



User's Guide

for

LibHuAirProp

**Library for the Calculation of Psychrometric,
Thermodynamic, and Transport Properties
for *Real* Humid Air, Steam, Water, and Ice**

I-P & SI Units

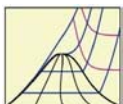
Version 10.0

FluidLAB for MATLAB

**Add-On for the comfortable use
of LibHuAirProp in MATLAB and Simulink**

*Based on ASHRAE Research Projects
RP-1485 and RP-1767*

Prepared by



**THERMO
FLUID
PROPERTIES**

www.thermofluidprop.com

Sebastian Herrmann
Hans-Joachim Kretzschmar
Matthias Kunick
Martin Suender

DISCLAIMER

ASHRAE has compiled this publication with care, but ASHRAE has not investigated, and ASHRAE expressly disclaims any duty to investigate, any product, service, process, procedure, design, or the like that may be described herein. The appearance of any technical data or editorial material in this publication does not constitute endorsement, warranty, or guaranty by ASHRAE of any product, service, process, procedure, design, or the like. ASHRAE does not warrant that the information in the publication is free of errors, and ASHRAE does not necessarily agree with any statement or opinion in this publication. The entire risk of the use of any information in this publication is assumed by the user.

The program developer, ASHRAE, and the distributors of this product do not warrant that the information in this software is free of errors. This information is provided “as is” without warranty of any kind, either expressed or implied. The entire risk as to the quality and performance of the program and data is with you. In no event will ASHRAE or the program developer be liable to you for any damages, including without limitation any lost profits, lost savings, or other incidental or consequential damages arising out of the use of or inability to use this information. ASHRAE reserves the right to discontinue support for this product at any time in the future, including without limitation discontinuing support of servers facilitating access to this product. In the event of discontinuation of support, ASHRAE will make a reasonable effort to notify purchasers before support is discontinued.

LICENSING AGREEMENT

Using this product indicates your acceptance of the terms and conditions of this agreement. The title and all copyrights and ownership rights of the product are retained by ASHRAE. You assume responsibility for the selection of the product to achieve your results and for the installation, use, and results obtained from the product.

This product may not be used on a LAN or WAN. You may use the product on a single machine. You may also copy the product into any machine-readable form for backup purposes in support of your use of the product on a single machine. You may not copy or transfer the product except as expressly provided for in this license. Specifically, you may not copy or transfer the product onto a machine other than your own unless the person to whom you are copying or transferring the product also has a license to use it. Doing so will result in the termination of your license, and ASHRAE will consider options to recover damages from unauthorized use of its intellectual property. Distribution to third parties in print or in electronic form is expressly prohibited unless authorized in writing by ASHRAE.

The program and data contained in the product are for your personal use only. “Personal use” includes showing the information at a meeting or group setting and allowing other individuals to view the content. “Personal use” does not include making copies, in whole or in part, of the content for the purposes of distribution, reusing the information contained in the product in your own presentation, or posting any of the files on a server for access by others. You shall not merge, adapt, translate, modify, rent, lease, sell, sublicense, assign, or otherwise transfer any of the content. To obtain permission to copy and paste this publication’s content for other than only personal use, go to www.ashrae.org/permissions.

LibHuAirProp Product Information

Do you need property values for moist air in I-P or SI units in your daily work?

► Use the property library LibHuAirProp ◀

Do you need these properties in Excel®, MATLAB®, Mathcad®, Mathcad Prime®, Engineering Equation Solver®, LabVIEW™, DYMOLA®, or SimulationX®?

► Use the add-ins FluidEXL, FluidLAB, FluidMAT, FluidPRIME, FluidEES, FluidVIEW, or FluidDYM ◀

What properties can be calculated using this software?

- thermodynamic properties psychrometric functions ◀
- transport properties backward functions ◀

What range of state is covered by this property library?

- unsaturated and saturated moist air ◀
- supersaturated moist air (liquid fog and ice fog) ◀
- temperatures from -143.15°C (-225.67°F) to 350°C (662°F) ◀
- pressures from 0.01 kPa (0.00145 psi) to 10,000 kPa (1450.4 psi) ◀

What are the references of LibHuAirProp?

Tables for moist air properties in the 2009, 2013, 2017, 2021, and 2025 ASHRAE Handbook of Fundamentals were calculated using LibHuAirProp

Psychrometrics

1.3

Table 2 Thermodynamic Properties of Moist Air at Standard Atmospheric Pressure, 101.325 kPa

Temp., °C <i>t</i>	Humidity Ratio <i>W_g</i> , kg _w /kg _{da}	Specific Volume, m ³ /kg _{da}			Specific Enthalpy, kJ/kg _{da}			Specific Entropy, kJ/(kg _{da} ·K)		Temp., °C <i>t</i>
		<i>v_{da}</i>	<i>v_{as}</i>	<i>v_g</i>	<i>h_{da}</i>	<i>h_{as}</i>	<i>h_g</i>	<i>s_{da}</i>	<i>s_g</i>	
-60	0.000067	0.6027	0.0000	0.6027	-60.341	0.016	-60.325	-0.2494	-0.2494	-60
-59	0.000076	0.6055	0.0000	0.6055	-59.335	0.018	-59.317	-0.2447	-0.2446	-59
-58	0.000087	0.6084	0.0000	0.6084	-58.329	0.021	-58.308	-0.2400	-0.2399	-58
-57	0.000100	0.6112	0.0000	0.6112	-57.323	0.024	-57.299	-0.2354	-0.2353	-57
-56	0.000114	0.6141	0.0000	0.6141	-56.317	0.027	-56.289	-0.2307	-0.2306	-56
-55	0.000129	0.6169	0.0000	0.6169	-55.311	0.031	-55.280	-0.2261	-0.2260	-55
-54	0.000147	0.6198	0.0000	0.6198	-54.305	0.035	-54.269	-0.2215	-0.2213	-54
-53	0.000167	0.6226	0.0000	0.6226	-53.299	0.040	-53.258	-0.2169	-0.2167	-53
-52	0.000190	0.6255	0.0000	0.6255	-52.293	0.046	-52.247	-0.2124	-0.2121	-52
-51	0.000215	0.6283	0.0000	0.6283	-51.287	0.052	-51.235	-0.2078	-0.2076	-51

Thermodynamic and psychrometric property algorithms from ASHRAE Research Project 1485

VOLUME 15, NUMBER 5

HVAC&R RESEARCH

SEPTEMBER 2009

FINAL REPORT

ASHRAE RP-1485

Thermodynamic Properties of Real Moist Air,
Dry Air, Steam, Water, and Ice

By S. Herrmann^a, H.-J. Kretzschmar^a, and D.P. Gatlley^b

^a Department of Technical Thermodynamics, Zittau/Goerlitz University of Applied Sciences, 02763 Zittau, Germany

^b Gatlley & Associates, Inc., Atlanta GA 30305, USA

November 17, 2008 (Submitted to TC for review)

March 12, 2009 (Final with corrections)

January 18, 2017 (Last update)

(For the documentation of corrections and modifications see the Appendix)

Thermodynamic Properties of Real Moist Air, Dry Air, Steam, Water, and Ice (RP-1485)

Sebastian Herrmann
Student Member ASHRAE

Hans-Joachim Kretzschmar, PhD
Member ASHRAE

Donald P. Gatlley, PE
Fellow/Life Member ASHRAE

Received February 14, 2009; accepted May 6, 2009

This paper is based on findings resulting from ASHRAE Research Project RP-1485.

This research updates the modeling of moist air as a real gas mixture using the virial equation of state. It includes the Hyland and Wexler model (1983a, 1983b) and considers the Nelson-Sauer model (2002). All new National Institute of Standards and Technology reference equations and the latest International Association for the Properties of Water and Steam (IAPWS) standards, as well as the current values for the molar masses and gas constants, have been incorporated. The deviations of the proposed model to the Hyland-Wexler and Nelson-Sauer models are very low at ambient pressures but increase with increasing pressures and temperatures. The range of validity of the new model is in pressure from 0.01 kPa up to 10 MPa, in temperature from -143.15°C up to 350°C, and in humidity ratio from 0 kg_w/kg_{da} up to 10 kg_w/kg_{da}. This model was used to produce moist air and H₂O saturation property tables for the psychrometric chapter in the 2009 ASHRAE Handbook—Fundamentals (ASHRAE 2009). The paper summarizes ASHRAE Research Project 1485 (RP-1485).

Transport property algorithms of moist air from ASHRAE Research Project 1767

FINAL REPORT

ASHRAE RP-1767

Transport Properties of Real Moist Air, Dry Air, Steam, and Water

By S. Herrmann^a, H.-J. Kretzschmar^a, V.C. Aute^b, D.P. Gatlery^c, and E. Vogel^d

^a Department of Technical Thermodynamics, Zittau/Goerlitz University of Applied Sciences,
02763 Zittau, Germany

^b Department of Mechanical Engineering, University of Maryland, College Park,
MD 20742, USA

^c Atlanta, GA, USA

^d Institute of Chemistry, University of Rostock, 18055 Rostock, Germany

December 31, 2018

Properties of dry air from the NIST Reference Equation of *Lemmon et al.* and properties of steam, water, and ice from the Industrial Formulation IAPWS-IF97, the Scientific Formulation IAPWS-95, and other current IAPWS formulations

Thermodynamic Properties of Air and Mixtures of Nitrogen, Argon, and Oxygen From 60 to 2000 K at Pressures to 2000 MPa

Eric W. Lemmon[¶]

Physical and Chemical Properties Division, National Institute of Standards and Technology, 325 Broadway, Boulder, Colorado 80303

Richard T Jacobsen and Steven G. Penoncello

Center for Applied Thermodynamic Studies, College of Engineering, University of Idaho, Moscow, Idaho 83844

Daniel G. Friend

Physical and Chemical Properties Division, National Institute of Standards and Technology, 325 Broadway, Boulder, Colorado 80303

Received June 25, 1999; revised manuscript received December 2, 1999

A thermodynamic property formulation for standard dry air based upon available experimental p - ρ - T , heat capacity, speed of sound, and vapor-liquid equilibrium data is presented. This formulation is valid for liquid, vapor, and supercritical air at temperatures from the solidification point on the bubble-point curve (59.75 K) to 2000 K at pressures up to 2000 MPa. In the absence of reliable experimental data for air above 873 K and 70 MPa, air properties were predicted from nitrogen data in this region. These values were included in the determination of the formulation to extend the range of validity. Experimental shock tube measurements on air give an indication of the extrapolation behavior of the equation of state up to temperatures and pressures of 5000 K and 98 GPa. The

0047-2689/2009/29(3)/331/55/\$37.00

331

J. Phys. Chem. Ref. Data, Vol. 29, No. 3, 2000

The International Association for the Properties of Water and Steam

Lucerne, Switzerland
August 2007

Revised Release on the IAPWS Industrial Formulation 1997
for the Thermodynamic Properties of Water and Steam
(The revision only relates to the extension of region 5 to 50 MPa)

©2007 International Association for the Properties of Water and Steam
Publication in whole or in part is allowed in all countries provided that attribution is given to the International Association for the Properties of Water and Steam

The International Association for the Properties of Water and Steam

Doorwerth, The Netherlands
September 2009

Revised Release on the IAPWS Formulation 1995 for the Thermodynamic
Properties of Ordinary Water Substance for General and Scientific Use

©2009 International Association for the Properties of Water and Steam
Publication in whole or in part is allowed in all countries provided that attribution is given to the International Association for the Properties of Water and Steam

Who are the authors of LibHuAirProp?

Dr. Sebastian Herrmann

Dr. Hans-Joachim Kretzschmar
Professor for Technical Thermodynamics

Dr. Matthias Kunick
Professor for Energy Systems Technology
Martin Suender

Kretzschmar Consulting Engineers, Amberg, Germany
Zittau/Goerlitz University of Applied Sciences, Zittau, Germany

Property Library for *Real Humid Air*, *Steam, Water, and Ice*

ASHRAE LibHuAirProp

Contents

0 Package Contents	0/1
Part I-P Units	I-P – 1/1
1 Property Library ASHRAE-LibHuAirProp-IP	I-P – 1/2
1.1 Function Overview	I-P – 1/2
1.1.1 Function Overview for Real Moist Air	I-P – 1/2
1.1.2 Function Overview for Steam and Water for Temperatures $t \geq 32^\circ\text{F}$	I-P – 1/6
1.1.3 Function Overview for Steam and Ice for Temperatures $t \leq 32^\circ\text{F}$	I-P – 1/8
1.2 Conversion of SI and I-P Units	I-P – 1/10
1.3 Calculation Algorithms	I-P – 1/13
1.3.1 Algorithms for Real Moist Air	I-P – 1/13
1.3.2 Algorithms for Steam and Water for Temperatures $t \geq 32^\circ\text{F}$	I-P – 1/14
1.3.3 Algorithms for Steam and Ice for Temperatures $t \leq 32^\circ\text{F}$	I-P – 1/14
1.3.4 Overview of the Applied Formulations for Steam, Water, and Ice	I-P – 1/14
2 Add-In FluidLAB for MATLAB [®] for ASHRAE-LibHuAirProp-IP	I-P – 2/1
2.1 Installing FluidLAB	I-P – 2/1
2.2 Licensing the LibHuAirProp Property Library	I-P – 2/5
2.3 Example Calculations of $h = f(p, t, W)$	I-P – 2/6
2.3.1 Preparation for the Example: Calculation of $h = f(p, t, W)$	I-P – 2/6
2.3.2 Example: Calculation of $h = f(p, t, W)$ in a M-File	I-P – 2/7
2.3.3 Example: Calculation of $h = f(p, t, W)$ in the Command Window	I-P – 2/11
2.4 Removing FluidLAB including LibHuAirProp	I-P – 2/12
3 Property Functions of ASHRAE-LibHuAirProp-IP	I-P – 3/1
3.1 Functions for Real Moist Air.....	I-P – 3/1
3.2 Functions for Steam and Water for Temperatures $t \geq 32^\circ\text{F}$	I-P – 3/42
3.3 Functions for Steam and Ice for Temperatures $t \leq 32^\circ\text{F}$	I-P – 3/54
4 Property Libraries for Calculating Heat Cycles, Boilers, Turbines, and Refrigerators. I-P – 4/1	
5 References	I-P – 5/1
6 Satisfied Customers	I-P – 6/1

Part SI Units	SI – 1/1
1 Property Library ASHRAE-LibHuAirProp-SI.....	SI – 1/2
1.1 Function Overview.....	SI – 1/2
1.1.1 Function Overview for Real Moist Air	SI – 1/2
1.1.2 Function Overview for Steam and Water for Temperatures $t \geq 0^\circ\text{C}$	SI – 1/6
1.1.3 Function Overview for Steam and Ice for Temperatures $t \leq 0^\circ\text{C}$	SI – 1/8
1.2 Conversion of SI and I-P Units	SI – 1/10
1.3 Calculation Algorithms.....	SI – 1/13
1.3.1 Algorithms for Real Moist Air	SI – 1/13
1.3.2 Algorithms for Steam and Water for Temperatures $t \geq 0^\circ\text{C}$	SI – 1/14
1.3.3 Algorithms for Steam and Ice for Temperatures $t \leq 0^\circ\text{C}$	SI – 1/14
1.3.4 Overview of the Applied Formulations for Steam, Water, and Ice.....	SI – 1/14
2 Add-On FluidLAB for MATLAB [®] for ASHRAE-LibHuAirProp-SI	SI – 2/1
2.1 Installing FluidLAB	SI – 2/1
2.2 Example Calculations of $h = f(p, t, W)$	SI – 2/1
2.2.1 Preparation for the Example: Calculation of $h = f(p, t, W)$	SI – 2/1
2.2.2 Example: Calculation of $h = f(p, t, W)$ in a M-File	SI – 2/3
2.2.3 Example: Calculation of $h = f(p, t, W)$ in the Command Window.....	SI – 2/6
2.3 Removing FluidLAB including LibHuAirProp.....	SI – 2/7
3 Property Functions of ASHRAE-LibHuAirProp-SI	SI – 3/1
3.1 Functions for Real Moist Air	SI – 3/1
3.2 Functions for Steam and Water for Temperatures $t \geq 0^\circ\text{C}$	SI – 3/42
3.3 Functions for Steam and Ice for Temperatures $t \leq 0^\circ\text{C}$	SI – 3/54
4 Property Libraries for Calculating Heat Cycles, Boilers, Turbines, and Refrigerators ..	SI – 4/1
5 References.....	SI – 5/1
6 Satisfied Customers	SI – 6/1

©  **KCE-**
ThermoFluidProperties UG
 Haager Weg 6
 92224 Amberg, Germany
 Phone: +49-172-7914607
 Fax: +49-3222-1095810
 E-mail: info@thermofluidprop.com
 Internet: www.thermofluidprop.com

Responsible person:

Dr. Sebastian Herrmann
 Phone: +49-172-5619222
 E-mail: Herrms@web.de

0 Package Contents

Add-On for 64-bit version of MATLAB®

The following ZIP file is delivered for your computer running a 64-bit version of MATLAB®.

ZIP file "CD_FluidLAB_ASHRAE_LibHuAirProp_x64.zip" for MATLAB®

The ZIP file contains the following files and folders:

FluidLAB_ASHRAE_LibHuAirProp_Users_Guide.pdf	User's Guide
FluidLAB_LibHuAirProp_64_Setup.msi	Self-extracting and self-installing program
setup.exe	Installation program for the FluidLAB Add-On for use in MATLAB®
vc redistrib_x64	Folder containing the "Microsoft Visual C++ 2010 x64 Redistributable Pack"
WindowsInstaller3_1	Folder containing the "Microsoft Windows Installer"

Part I-P Units

1 Property Library ASHRAE-LibHuAirProp-IP

1.1 Function Overview

1.1.1 Function Overview for Real Moist Air

Functional Dependence	Function Name	Property or Function	Unit of the Result	Page
$a = f(p, t, W)$	a_ptW_HAP_IP	Thermal diffusivity	ft ² /s	3/2
$\alpha_p = f(p, t, W)$	alphap_ptW_HAP_IP	Relative pressure coefficient	1/°R	3/3
$\beta_p = f(p, t, W)$	betap_ptW_HAP_IP	Isothermal stress coefficient	lb/ft ³	3/4
$c = f(p, t, W)$	c_ptW_HAP_IP	Speed of sound	ft/s	3/5
$c_p = f(p, t, W)$	cp_ptW_HAP_IP	Specific isobaric heat capacity	Btu/(lb·°R)	3/6
$c_v = f(p, t, W)$	cv_ptW_HAP_IP	Specific isochoric heat capacity	Btu/(lb·°R)	3/7
$f = f(p, t)$	f_pt_HAP_IP	Enhancement factor (decimal ratio)	-	3/8
$h = f(p, t, W)$	h_ptW_HAP_IP	Air-specific enthalpy	Btu/lb _a	3/9
$\eta = f(p, t, W)$	Eta_ptW_HAP_IP	Dynamic viscosity	lb·s/ft ²	3/10
$\kappa = f(p, t, W)$	Kappa_ptW_HAP_IP	Isentropic exponent	-	3/11
$\lambda = f(p, t, W)$	Lambda_ptW_HAP_IP	Thermal conductivity	Btu/(h·ft·°R)	3/12
$\nu = f(p, t, W)$	Ny_ptW_HAP_IP	Kinematic viscosity	ft ² /s	3/13
$p = f(t, s, W)$	p_tsW_HAP_IP	Pressure of humid air	psi	3/14
$p = f(z_{\text{ele}})$	p_zele_HAP_IP	Pressure of humid air from elevation	psi	3/15
$p_{\text{Air}} = f(p, t, W)$	pAIR_ptW_HAP_IP	Partial pressure of dry air in moist air	psi	3/16
$p_{\text{H}_2\text{O}} = f(p, t, W)$	pH2O_ptW_HAP_IP	Partial pressure of water vapor in moist air	psi	3/17
$p_{\text{H}_2\text{O}_s} = f(p, t)$	pH2Os_pt_HAP_IP	Partial saturation pressure of water vapour in moist air	psi	3/18

Functional Dependence	Function Name	Property or Function	Unit of the Result	Page
$\phi = f(p, t, W)$	phi_ptW_HAP_IP	Relative humidity (decimal ratio)	-	3/19
$Pr = f(p, t, W)$	Pr_ptW_HAP_IP	PRANDTL number	-	3/20
$\psi_{\text{Air}} = f(W)$	PsiAir_W_HAP_IP	Mole fraction of dry air in moist air	mol _a /mol	3/21
$\psi_{\text{H}_2\text{O}} = f(W)$	PsiH2O_W_HAP_IP	Mole fraction of water vapor in moist air	mol _w /mol	3/22
$\rho = f(p, t, W)$	Rho_ptW_HAP_IP	Density	lb/ft ³	3/23
$s = f(p, t, W)$	s_ptW_HAP_IP	Air-specific entropy	Btu/(lb _a ·°R)	3/24
$t = f(p, h, \phi)$	t_phphi_HAP_IP	Backward function: temperature from total pressure, air-specific enthalpy and relative humidity	°F	3/25
$t = f(p, h, W)$	t_phW_HAP_IP	Backward function: temperature from total pressure, enthalpy and humidity ratio	°F	3/26
$t = f(p, s, W)$	t_psW_HAP_IP	Backward function: temperature from total pressure, entropy and humidity ratio	°F	3/27
$t = f(p, t_{\text{wb}}, W)$	t_ptwbW_HAP_IP	Backward function: temperature from total pressure, wet-bulb temperature and humidity ratio	°F	3/28
$t_d = f(p, W)$	td_pW_HAP_IP	Dew-point/frost-point temperature	°F	3/29
$t_s = f(p, p_{\text{H}_2\text{O}})$	ts_ppH2O_HAP_IP	Backward function: saturation temperature of water from total pressure and partial pressure of water vapor	°F	3/30
$t_{\text{wb}} = f(p, t, W)$	twb_ptW_HAP_IP	Wet-bulb/ice-bulb temperature	°F	3/31
$u = f(p, t, W)$	u_ptW_HAP_IP	Air-specific internal energy	Btu/lb _a	3/32
$v = f(p, t, W)$	v_ptW_HAP_IP	Air-specific volume	ft ³ /lb _a	3/33
$W = f(p, t, p_{\text{H}_2\text{O}})$	W_ptpH2O_HAP_IP	Humidity ratio from total pressure, temperature, and partial pressure of water vapor	lb _w /lb _a	3/34
$W = f(p, t, \phi)$	W_ptphi_HAP_IP	Humidity ratio from total pressure, temperature, and relative humidity	lb _w /lb _a	3/35
$W = f(p, t_d)$	W_ptd_HAP_IP	Humidity ratio from total pressure and dew-point temperature	lb _w /lb _a	3/36

Functional Dependence	Function Name	Property or Function	Unit of the Result	Page
$W = f(p, t, t_{wb})$	W_pttwb_HAP_IP	Humidity ratio from total pressure, (dry bulb) temperature, and wet-bulb temperature	lb _w /lb _a	3/37
$W_s = f(p, t)$	Ws_pt_HAP_IP	Saturation humidity ratio	lb _w /lb _a	3/38
$\xi_{Air} = f(W)$	XiAir_W_HAP_IP	Mass fraction of dry air in moist air	lb _a /lb	3/39
$\xi_{H2O} = f(W)$	XiH2O_W_HAP_IP	Mass fraction of water vapor in moist air	lb _w /lb	3/40
$Z = f(p, t, W)$	Z_ptW_HAP_IP	Compression factor (decimal ratio)	-	3/41

Range of Validity of Thermodynamic Properties

Property	Range of Validity					
Pressure:	0.00145	≤	p	≤	1450.4	psi
Temperature:	-225.67	≤	t	≤	662	°F
Humidity ratio:	0	≤	W	≤	10	lb _w /lb _a
Relative humidity:	0	≤	ϕ	≤	1	(decimal ratio)
Dew-point temperature:	-225.67	≤	t_d	≤	662	°F
Wet-bulb temperature:	-225.67	≤	t_{wb}	≤	662	°F

Units

Symbol	Quantity	Unit
p	Pressure	psi
t	Temperature	°F
W	Humidity ratio	lb _w /lb _a (lb water / lb dry air)
ϕ	Relative humidity	(decimal ratio)
t_d	Dew point temperature	°F
t_{wb}	Wet bulb temperature	°F

Range of Validity of Transport Properties

Property	Range of Validity					
Pressure:	0.00145	≤	p	≤	1450.4	psi
Temperature:	-99.67	≤	t	≤	662	°F
Humidity ratio:	0	≤	W	≤	10	lb _w /lb _a
Relative humidity:	0	≤	ϕ	≤	1	(decimal ratio)

Molar Masses

Component	Molar Mass	Reference
Dry Air	63.859 lb/kmol	[17]
Water	39.7168998 lb/kmol	[5], [6]

Reference States

Property	Dry Air	Steam, Water, and Ice
Pressure	14.6959 psi	$p_s(32.018^\circ\text{F}) = 0.088714$ psi
Temperature	32°F	32.018°F
Enthalpy	0 Btu/lb	0.00026301926 Btu/lb
Entropy	0 Btu/(lb·°R)	0 Btu/(lb·°R)

1.1.2 Function Overview for Steam and Water for Temperatures $t \geq 32^\circ\text{F}$

Functional Dependence	Function Name	Property or Function	Unit of the Result	Page
$h_{\text{liq}} = f(p, t)$	hliq_pt_97_IP	Specific enthalpy of liquid water	Btu/lb	3/43
$h_{\text{liq,s}} = f(t)$	hliqs_t_97_IP	Specific enthalpy of saturated liquid water	Btu/lb	3/44
$h_{\text{vap,s}} = f(t)$	hvaps_t_97_IP	Specific enthalpy of saturated water vapor	Btu/lb	3/45
$p_s = f(t)$	ps_t_97_IP	Saturation pressure of water	psi	3/46
$s_{\text{liq}} = f(p, t)$	sliq_pt_97_IP	Specific entropy of liquid water	Btu/(lb·°R)	3/47
$s_{\text{liq,s}} = f(t)$	sliqs_t_97_IP	Specific entropy of saturated liquid water	Btu/(lb·°R)	3/48
$s_{\text{vap,s}} = f(t)$	svaps_t_97_IP	Specific entropy of saturated water vapor	Btu/(lb·°R)	3/49
$t_s = f(p)$	ts_p_97_IP	Saturation temperature of water	°F	3/50
$v_{\text{liq}} = f(p, t)$	vliq_pt_97_IP	Specific volume of liquid water	ft ³ /lb	3/51
$v_{\text{liq,s}} = f(t)$	vliqs_t_97_IP	Specific volume of saturated liquid water	ft ³ /lb	3/52
$v_{\text{vap,s}} = f(t)$	vvaps_t_97_IP	Specific volume of saturated water vapor	ft ³ /lb	3/53

Range of Validity

Property	Range of Validity				
Pressure:	0.00145	\leq	p	\leq	1450.4 psi
Temperature:	32	\leq	t	\leq	662 °F

Reference State

Property	Water Vapor and Liquid Water
Pressure	$p_s(32.018^\circ\text{F}) = 0.088714$ psi
Temperature	32.018°F
Enthalpy	0.00026301926 Btu/lb
Entropy	0 Btu/(lb·°R)

Units

Symbol	Quantity	Unit
p	Pressure	psi
t	Temperature	°F

1.1.3 Function Overview for Steam and Ice for Temperatures $t \leq 32^\circ\text{F}$

Functional Dependence	Function Name	Property	Unit of the Result	Page
$h_{\text{ice,sub}} = f(t)$	hicesub_t_06_IP	Specific enthalpy of saturated ice	Btu/lb	3/55
$h_{\text{vap,sub}} = f(t)$	hvapsub_t_95_IP	Specific enthalpy of saturated water vapor	Btu/lb	3/56
$p_{\text{mel}} = f(t)$	pmel_t_08_IP	Melting pressure of ice	psi	3/57
$p_{\text{sub}} = f(t)$	psub_t_08_IP	Sublimation pressure of ice	psi	3/58
$s_{\text{ice,sub}} = f(t)$	sicesub_t_06_IP	Specific entropy of saturated ice	Btu/(lb·°R)	3/59
$s_{\text{vap,sub}} = f(t)$	svapsub_t_95_IP	Specific entropy of saturated water vapor	Btu/(lb·°R)	3/60
$t_{\text{mel}} = f(p)$	tmel_p_08_IP	Melting temperature of ice	°F	3/61
$t_{\text{sub}} = f(p)$	tsub_p_08_IP	Sublimation temperature of ice	°F	3/62
$v_{\text{ice,sub}} = f(t)$	vicesub_t_06_IP	Specific volume of saturated ice	ft ³ /lb	3/63
$v_{\text{vap,sub}} = f(t)$	vvapsub_t_95_IP	Specific volume of saturated water vapor	ft ³ /lb	3/64

Range of Validity

Property	Range of Validity				
Pressure:	$p_{\text{sub}}(-225.67^{\circ}\text{F}) = 1.7407\text{E-}12$	\leq	p	\leq	1450.4 psi
Temperature:	-225.67	\leq	t	\leq	32 °F

Units

Symbol	Quantity	Unit
p	Pressure	psi
t	Temperature	°F

Reference State

Property	Water Vapor and Ice
Pressure	$p_s(32.018^{\circ}\text{F}) = 0.088714$ psi
Temperature	32.018°F
Enthalpy	0.00026301926 Btu/lb
Entropy	0 Btu/(lb·°R)

1.2 Conversion of SI and I-P Units

Property	Conversion: SI Units → I-P Units	Conversion: I-P Units → SI Units	Units SI	Units I-P
Thermal diffusivity a	$\frac{a_{IP}}{\frac{ft^2}{s}} = \frac{a_{SI}}{\frac{m^2}{s}} \times 10.76391042$	$\frac{a_{SI}}{\frac{m^2}{s}} = \frac{a_{IP}}{\frac{ft^2}{s}} \times 0.0929304$	m ² /s	ft ² /s
Relative pressure coefficient α_p	$\frac{\alpha_{p,IP}}{1} = \frac{\alpha_{p,SI}}{1} \times \frac{9}{5}$ °R K	$\frac{\alpha_{p,SI}}{1} = \frac{\alpha_{p,IP}}{1} \times \frac{5}{9}$ K °R	1/K	1/°R
Isothermal stress coefficient β_p	$\frac{\beta_{p,IP}}{\frac{lb}{ft^3}} = \frac{\beta_{p,SI}}{\frac{kg}{m^3}} \times 0.062428$	$\frac{\beta_{p,SI}}{\frac{kg}{m^3}} = \frac{\beta_{p,IP}}{\frac{lb}{ft^3}} \times 16.018463$	kg/m ³	lb/ft ³
Speed of sound c	$\frac{c_{IP}}{\frac{ft}{s}} = \frac{c_{SI}}{\frac{m}{s}} \times 3.2808399$	$\frac{c_{SI}}{\frac{m}{s}} = \frac{c_{IP}}{\frac{ft}{s}} \times 0.3048$	m/s	ft/s
Specific isobaric heat capacity c_p	$\frac{c_{p,IP}}{\frac{Btu}{lb \cdot ^\circ R}} = \frac{c_{p,SI}}{\frac{kJ}{kg \cdot K}} \times 0.2388459$	$\frac{c_{p,SI}}{\frac{kJ}{kg \cdot K}} = \frac{c_{p,IP}}{\frac{Btu}{lb \cdot ^\circ R}} \times 4.1868$	kJ/(kg·K)	Btu/(lb·°R)
Specific isochoric heat capacity c_v	$\frac{c_{v,IP}}{\frac{Btu}{lb \cdot ^\circ R}} = \frac{c_{v,SI}}{\frac{kJ}{kg \cdot K}} \times 0.2388459$	$\frac{c_{v,SI}}{\frac{kJ}{kg \cdot K}} = \frac{c_{v,IP}}{\frac{Btu}{lb \cdot ^\circ R}} \times 4.1868$	kJ/(kg·K)	Btu/(lb·°R)
Dynamic viscosity η	$\frac{\eta_{IP}}{\frac{lb \cdot s}{ft^2}} = \frac{\eta_{SI}}{\frac{Pa \cdot s}}{s} \times 0.02088543$	$\frac{\eta_{SI}}{\frac{Pa \cdot s}}{s} = \frac{\eta_{IP}}{\frac{lb \cdot s}{ft^2}} \times 47.880259$	Pa·s	lb·s/ft ²
Enhancement factor f	$f_{IP} = f_{SI}$	$f_{SI} = f_{IP}$	-	-

Property	Conversion: SI Units → I-P Units	Conversion: I-P Units → SI Units	Units SI	Units I-P
Air-specific enthalpy (moist air) h	$\frac{h_{IP}}{\text{Btu/lb}_a} = \frac{h_{SI}}{\text{kJ/kg}_a} \times 0.4299226 + 7.68565365666$	$\frac{h_{SI}}{\text{kJ/kg}_a} = \left(\frac{h_{IP}}{\text{Btu/lb}_a} - 7.68565365666 \right) \times 2.326$	kJ/kg _a	Btu/lb _a
Specific enthalpy (water, water vapor, ice) h_w	$\frac{h_{IP}}{\text{Btu/lb}} = \frac{h_{SI}}{\text{kJ/kg}} \times 0.4299226$	$\frac{h_{SI}}{\text{kJ/kg}} = \frac{h_{IP}}{\text{Btu/lb}} \times 2.326$	kJ/kg	Btu/lb
Isentropic exponent κ	$\kappa_{IP} = \kappa_{SI}$	$\kappa_{SI} = \kappa_{IP}$	-	-
Thermal conductivity λ	$\frac{\lambda_{IP}}{\text{Btu/h ft } ^\circ\text{R}} = \frac{\lambda_{SI}}{\text{W/m K}} \times 0.57778932$	$\frac{\lambda_{SI}}{\text{W/m K}} = \frac{\lambda_{IP}}{\text{Btu/h ft } ^\circ\text{R}} \times 1.73073467$	W/(m·K)	Btu/(h·ft·°R)
Kinematic viscosity ν	$\frac{\nu_{IP}}{\text{ft}^2/\text{s}} = \frac{\nu_{SI}}{\text{m}^2/\text{s}} \times 10.763910417$	$\frac{\nu_{SI}}{\text{m}^2/\text{s}} = \frac{\nu_{IP}}{\text{ft}^2/\text{s}} \times 0.092903040$	m ² /s	ft ² /s
Pressure p	$\frac{p_{IP}}{\text{psi}} = \frac{p_{SI}}{\text{kPa}} \times 0.14503774$	$\frac{p_{SI}}{\text{kPa}} = \frac{p_{IP}}{\text{psi}} \times 6.894757$	kPa	psi
Relative humidity ϕ	$\phi_{IP} = \phi_{SI}$	$\phi_{SI} = \phi_{IP}$	-	-
Prandtl number Pr	$Pr_{IP} = Pr_{SI}$	$Pr_{SI} = Pr_{IP}$	-	-
Mole fraction ψ	$\psi_{IP} = \psi_{SI}$	$\psi_{SI} = \psi_{IP}$	mol/mol	mol/mol
Density ρ	$\frac{\rho_{IP}}{\text{lb/ft}^3} = \frac{\rho_{SI}}{\text{kg/m}^3} \times 0.062428$	$\frac{\rho_{SI}}{\text{kg/m}^3} = \frac{\rho_{IP}}{\text{lb/ft}^3} \times 16.018463$	kg/m ³	lb/ft ³
Air-specific entropy (moist air) s	$\frac{s_{IP}}{\text{Btu/lb}_a \text{ } ^\circ\text{R}} = \frac{s_{SI}}{\text{kJ/kg}_a \text{ K}} \times 0.2388459 + 0.01616365106$	$\frac{s_{SI}}{\text{kJ/kg}_a \text{ K}} = \left(\frac{s_{IP}}{\text{Btu/lb}_a \text{ } ^\circ\text{R}} - 0.01616365106 \right) \times 4.1868$	kJ/(kg _a ·K)	Btu/(lb _a ·°R)

Property	Conversion: SI Units → I-P Units	Conversion: I-P Units → SI Units	Units SI	Units I-P
Specific entropy (water, water vapor, ice) s_w	$\frac{s_{IP}}{\frac{\text{Btu}}{\text{lb}_a \cdot ^\circ\text{R}}} = \frac{s_{SI}}{\frac{\text{kJ}}{\text{kg}_a \cdot \text{K}}} \times 0.23884589$	$\frac{s_{SI}}{\frac{\text{kJ}}{\text{kg}_a \cdot \text{K}}} = \frac{s_{IP}}{\frac{\text{Btu}}{\text{lb}_a \cdot ^\circ\text{R}}} \times 4.1868$	kJ/(kg _a ·K)	Btu/(lb _a ·°R)
Temperature t	$\frac{t_{IP}}{^\circ\text{F}} = \frac{t_{SI}}{^\circ\text{C}} \times \frac{9}{5} + 32$	$\frac{t_{SI}}{^\circ\text{C}} = \left(\frac{t_{IP}}{^\circ\text{F}} - 32 \right) \times \frac{5}{9}$	°C	°F
Air-specific internal energy (moist air) u	$(u = h - pv)$ $\frac{u_{IP}}{\frac{\text{Btu}}{\text{lb}_a}} = \frac{h_{SI}}{\frac{\text{kJ}}{\text{kg}_a}} \times 0.4299226 + 7.68565365666$ $- \frac{p_{SI}}{\text{kPa}} \times 0.145037738 \cdot \frac{v_{SI}}{\frac{\text{m}^3}{\text{kg}_a}} \times 16.018453$	$(u = h - pv)$ $\frac{u_{SI}}{\frac{\text{kJ}}{\text{kg}_a}} = \left(\frac{h_{IP}}{\frac{\text{Btu}}{\text{lb}_a}} - 7.68565365666 \right) \times 2.236$ $- \frac{p_{IP}}{\text{psi}} \times 6.894757293 \cdot \frac{v_{SI}}{\frac{\text{ft}^3}{\text{lb}_a}} \times 0.062428$	kJ/kg _a	Btu/lb _a
Air-specific volume (moist air) v	$\frac{v_{IP}}{\frac{\text{ft}^3}{\text{lb}_a}} = \frac{v_{SI}}{\frac{\text{m}^3}{\text{kg}_a}} \times 16.018453$	$\frac{v_{SI}}{\frac{\text{m}^3}{\text{kg}_a}} = \frac{v_{IP}}{\frac{\text{ft}^3}{\text{lb}_a}} \times 0.062428$	m ³ /kg _a	ft ³ /lb _a
Specific volume (water, water vapor, ice) v_w	$\frac{v_{IP}}{\frac{\text{ft}^3}{\text{lb}}} = \frac{v_{SI}}{\frac{\text{m}^3}{\text{kg}}} \times 16.018453$	$\frac{v_{SI}}{\frac{\text{m}^3}{\text{kg}}} = \frac{v_{IP}}{\frac{\text{ft}^3}{\text{lb}}} \times 0.062428$	m ³ /kg	ft ³ /lb
Humidity ratio W	$W_{IP} = W_{SI}$	$W_{SI} = W_{IP}$	kg _w /kg _a	lb _w /lb _a
Mass fraction ζ	$\zeta_{IP} = \zeta_{SI}$	$\zeta_{SI} = \zeta_{IP}$	kg _w /kg	lb _w /lb
Compression factor Z	$Z_{IP} = Z_{SI}$	$Z_{SI} = Z_{IP}$	-	-

1.3 Calculation Algorithms

1.3.1 Algorithms for Real Moist Air

The properties of moist air are calculated from the modified Hyland-Wexler model given in Herrmann, Kretzschmar, and Gatley (HKG) [1], [2]. The modifications incorporate:

- the value for the universal molar gas constant from the CODATA standard by Mohr and Taylor [22]
- the value for the molar mass of dry air from Gatley et al. [17] and that of water from IAPWS-95 [5], [6]
- the calculation of the ideal-gas parts of the heat capacity, enthalpy, and entropy for dry air from the fundamental equation of Lemmon et al. [14]
- the calculation of the ideal-gas parts of the heat capacity, enthalpy, and entropy for water vapor from IAPWS-IF97 [7], [8], [9] for $t \geq 32^\circ\text{F}$ and from IAPWS-95 [5], [6] for $t \leq 32^\circ\text{F}$
- the calculation of the vapor-pressure enhancement factor from the equation given by the models of Hyland and Wexler [21]
- the calculation of the second and third molar virial coefficients B_{aa} and C_{aaa} for dry air from the fundamental equation of Lemmon et al. [14] according to Feistel et al. [24]
- the calculation of the second and third molar virial coefficients B_{ww} and C_{www} for water and steam from IAPWS-95 [5], [6] according to Feistel et al. [24]
- the calculation of the air-water second molar cross-virial coefficient B_{aw} from Harvey and Huang [15]
- the calculation of the air-water third molar cross-virial coefficients C_{aaw} and C_{aww} from Nelson and Sauer [12], [13]
- the calculation of the saturation pressure of water from IAPWS-IF97 [7], [8], [9] for $t \geq 32^\circ\text{F}$ and of the sublimation pressure of water from IAPWS-08 [11] for $t \leq 32^\circ\text{F}$
- the calculation of the isothermal compressibility of saturated liquid water from IAPWS-IF97 [7], [8], [9] for $t \geq 32^\circ\text{F}$ and that of ice from IAPWS-06 [10] for $t \leq 32^\circ\text{F}$ in the determination of the vapor-pressure enhancement factor
- the calculation of Henry's constant from the IAPWS Guideline 2004 [16] in the determination of the enhancement factor. The mole fractions for the three main components of dry air were taken from Lemmon et al. [14]. Argon was not considered in the calculation of Henry's constant in the former research projects, but it is now the third component of dry air.

The transport properties of moist air are calculated from the model given in Herrmann et al. [3], [4].

1.3.2 Algorithms for Steam and Water for Temperatures $t \geq 32^\circ\text{F}$

The p - T diagram in Fig. 1 shows the formulations used for water and water vapor. The temperature range above 32°F is covered by IAPWS-IF97 [7], [8], [9]:

- The saturation line is calculated from the IAPWS-IF97 saturation pressure equation $p_s^{97}(t)$ and saturation temperature equation $t_s^{97}(p)$.
- The properties in the liquid region including saturated-liquid line are calculated from the fundamental equation of the IAPWS-IF97 region 1.
- The properties in the vapor region including saturated-vapor line are calculated from the fundamental equation of the IAPWS-IF97 region 2.

1.3.3 Algorithms for Steam and Ice for Temperatures $t \leq 32^\circ\text{F}$

- The sublimation curve is covered by the IAPWS-08 sublimation pressure equation $p_{\text{subl}}^{08}(t)$ [11] (see Fig. 1).
- The properties of ice including saturated ice are determined by the fundamental equation of the IAPWS-06 [10].
- The properties of vapor including saturated vapor are calculated from the fundamental equation of IAPWS-95 [5], [6].

1.3.4 Overview of the Applied Formulations for Steam, Water, and Ice

The following p - T diagram shows the used IAPWS Formulations and the ranges where they are applied.

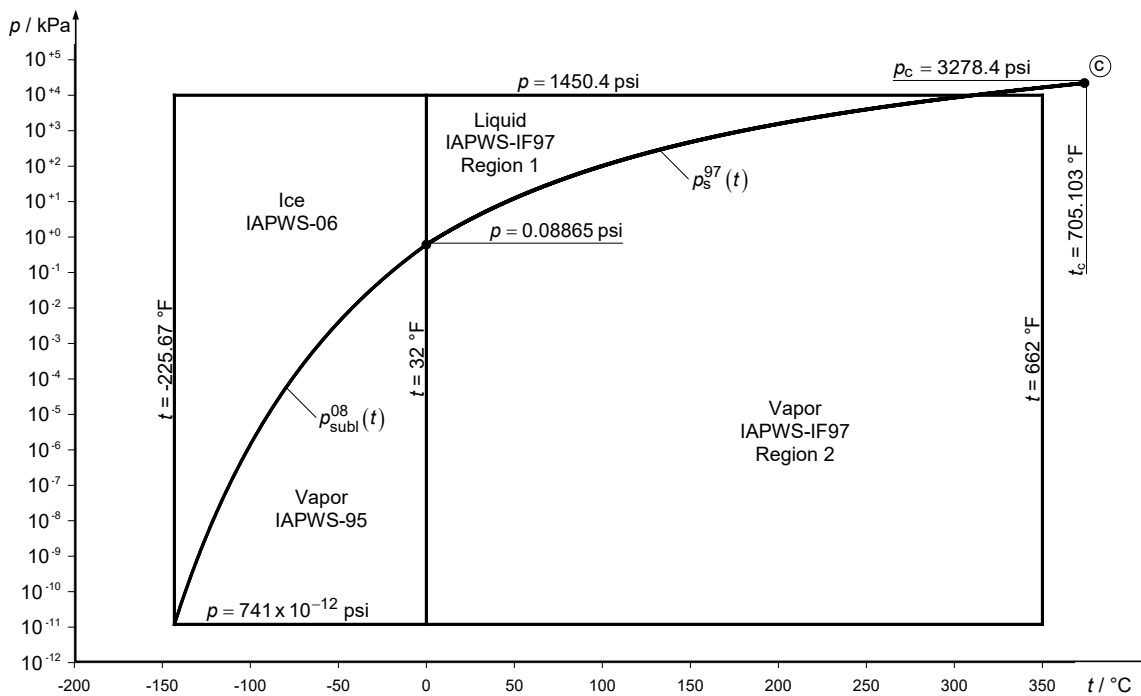


Figure 1: p - T diagram with used IAPWS formulations for steam, water, and ice.

2 Add-On FluidLAB for MATLAB® for ASHRAE-LibHuAirProp-IP

2.1 Installing FluidLAB

The FluidLAB Add-In has been developed to calculate thermodynamic properties in MATLAB® more conveniently. Within MATLAB® it enables the direct call of functions relating to real moist air, steam, water, and ice from the ASHRAE-LibHuAirProp-IP property library.

This section describes the installation of both FluidLAB LibHuAirProp_IP and LibHuAirProp_SI.

Before you begin, it is best to close any Windows® applications, since Windows® may need to be rebooted during the installation process.

After you have downloaded and extracted the zip-file "CD_FluidLAB_ASHRAE_LibHuAirProp.zip", you will see the folder

CD_FluidLAB_ASHRAE_LibHuAirProp

in your Windows Explorer®, Norton Commander® or other similar program you are using.

Open this folder by double-clicking on it.

In this folder you will see the following three files:

FluidLAB_ASHRAE_LibHuAirProp_Users_Guide.pdf

FluidLAB_ASHRAE_LibHuAirProp_64_Setup.msi

Setup.exe.

In order to run the installation of FluidLAB including the ASHRAE-LibHuAirProp property library, double-click on the file

setup.exe.

If the "Microsoft Visual C++ 2010 x64 Redistributable Pack" is not running on your computer yet, installation will start with a window noting that the "Visual C++ 2010 runtime library (x64)" will be installed on your machine (see Figure 2.1.1).

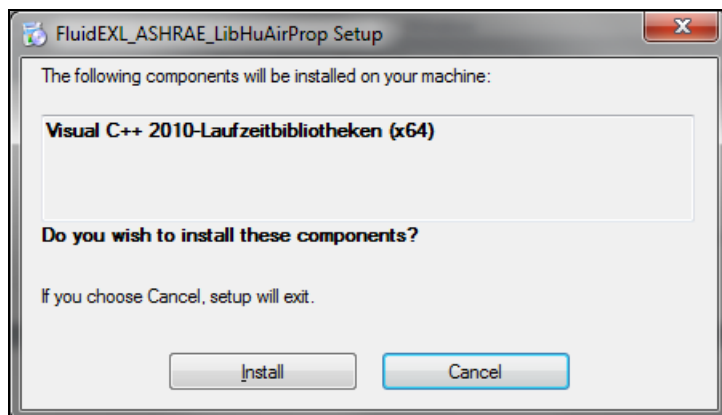


Figure 2.1.1: Installing the "Visual C++ 2010 runtime library (x64)"

Click on "Install" to continue.

Note.

If there is a newer version of "Microsoft Visual C++ 2010 x64 Redistributable" package installed on your computer than the one provided in this installation setup.exe will stop and is followed by an error message. In this case please start the installation again by double-clicking the file "FluidLAB_ASHRAE_LibHuAirProp_64_Setup.msi" and skip the next three steps described in the User's Guide.

In the following window you are required to accept the Microsoft® license terms to install the "Microsoft Visual C++ 2010 x64 Redistributable Pack" by ticking the box next to "I have read and accept the license terms" (see Figure 2.1.2).

Now click on "Install" to continue installation.

After the "Microsoft Visual C++ 2010 x64 Redistributable Pack" has been installed, you will see the sentence "Microsoft Visual C++ 2010 x64 Redistributable has been installed." Confirm this by clicking "Finish."

Installation of FluidLAB_LibHuAirProp starts with a window noting that the installer will guide you through the installation process. Click the "Next >" button to continue.

In the following dialog box, "Destination Location", the default path offered automatically for the installation of FluidLAB is

C:\Users\...\FluidLAB\LibHuAirProp

By clicking the "Browse..." button, you can change the installation directory before installation (see Figure 2.1.3).

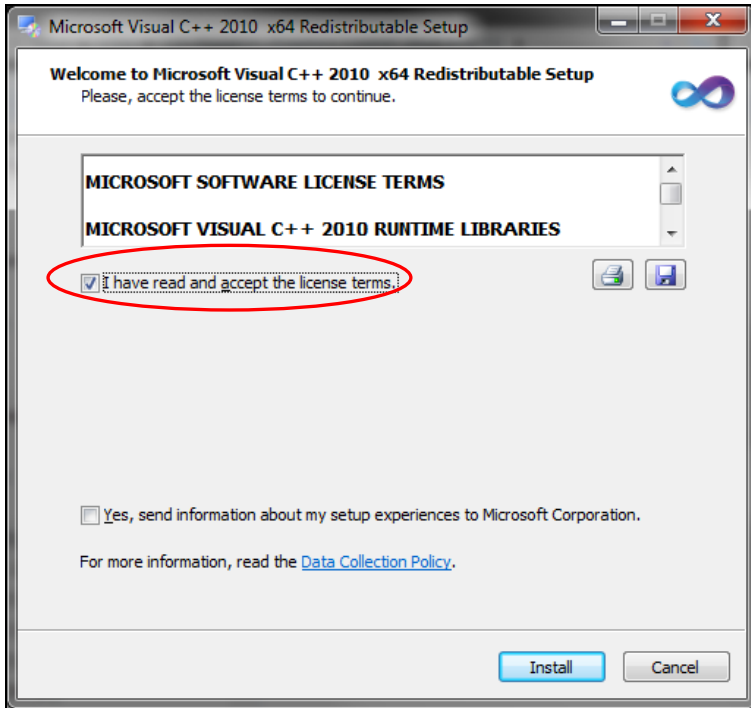


Figure 2.1.2: Accepting the license terms

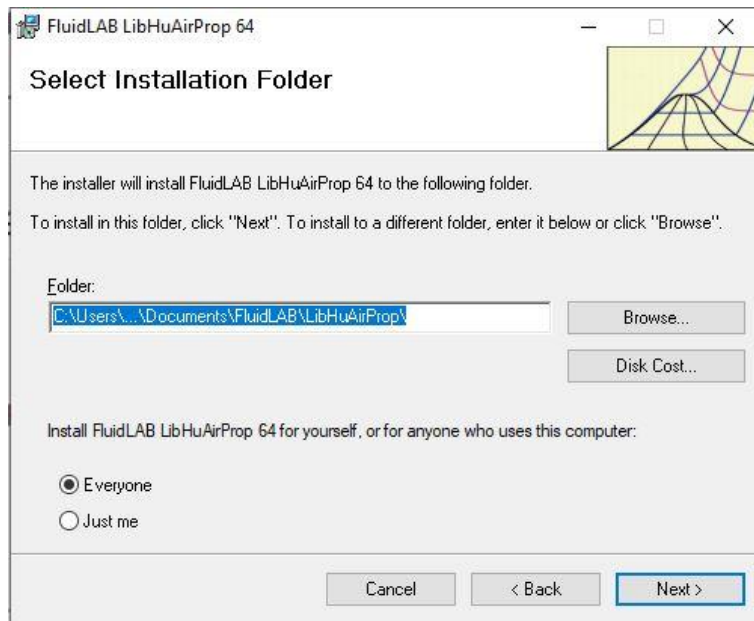


Figure 2.1.3: "Select Installation Folder"

Finally, click on "Next >" to continue installation; click "Next >" again in the "Confirm Installation" window which follows in order to start the installation of FluidLAB.

After FluidLAB has been installed, you will see the sentence "FluidLAB LibHuAirProp 64 has been successfully installed." Confirm this by clicking the "Close" button.

The installation program has copied the following files for LibHuAirProp_IP and LibHuAirProp_SI into the installation directory:

capt_ico_big.ico
 LC.dll
 LibHuAirProp_IP.dll
 LibHuAirProp_SI.dll

MATLAB®-Interface Program for calculable functions for LibHuAirProp

The installation program for the 64-bit Windows® version have copied the following function files for LibHuAirProp_IP and LibHuAirProp_SI into the installation directory:

I-P Functions

a_ptW_HAP_IP	v_ptW_HAP_IP
alphap_ptW_HAP_IP	W_ptpH2O_HAP_IP
betap_ptW_HAP_IP	W_ptphi_HAP_IP
c_ptW_HAP_IP	W_ptd_HAP_IP
cp_ptW_HAP_IP	W_pttwb_HAP_IP
cv_ptW_HAP_IP	Ws_pt_HAP_IP
Eta_ptW_HAP_IP	XiAir_W_HAP_IP
f_pt_HAP_IP	XiH2O_W_HAP_IP
h_ptW_HAP_IP	Z_ptW_HAP_IP
Kappa_ptW_HAP_IP	hliq_pt_97_IP

I-P – 2/4

Lambda_ptW_HAP_IP
Ny_ptW_HAP_IP
p_tsW_HAP_IP
p_zele_HAP_IP
pAIR_ptW_HAP_IP
pH2O_ptW_HAP_IP
pH2Os_pt_HAP_IP
phi_ptW_HAP_IP
PR_ptW_HAP_IP
PsiAir_W_HAP_IP
PsiH2O_W_HAP_IP
Rho_ptW_HAP_IP
s_ptW_HAP_IP
t_phphi_HAP_IP
t_phW_HAP_IP
t_psW_HAP_IP
t_ptwbW_HAP_IP
td_pW_HAP_IP
ts_ppH2O_HAP_IP
twb_ptW_HAP_IP
u_ptW_HAP_IP

SI Functions

a_ptW_HAP_SI
alphap_ptW_HAP_SI
betap_ptW_HAP_SI
c_ptW_HAP_SI
cp_ptW_HAP_SI
cv_ptW_HAP_SI
Eta_ptW_HAP_SI
f_pt_HAP_SI
h_ptW_HAP_SI
Kappa_ptW_HAP_SI
Lambda_ptW_HAP_SI
Ny_ptW_HAP_SI
p_tsW_HAP_SI
p_zele_HAP_SI
pAIR_ptW_HAP_SI
pH2O_ptW_HAP_SI
pH2Os_pt_HAP_SI
phi_ptW_HAP_SI
PR_ptW_HAP_SI
PsiAir_W_HAP_SI
PsiH2O_W_HAP_SI
Rho_ptW_HAP_SI

hliqs_t_97_IP
hvaps_t_97_IP
ps_t_97_IP
sliq_pt_97_IP
sliqs_t_97_IP
svaps_t_97_IP
ts_p_97_IP
vliq_pt_97_IP
vliqs_t_97_IP
vvaps_t_97_IP
hicesub_t_06_IP
hvapsub_t_95_IP
pmel_t_08_IP
psub_t_08_IP
sicesub_t_06_IP
svapsub_t_95_IP
tmel_p_08_IP
tsub_p_08_IP
vicesub_t_06_IP
vvapsub_t_95_IP

v_ptW_HAP_SI
W_ptpH2O_HAP_SI
W_ptphi_HAP_SI
W_ptd_HAP_SI
W_pttwb_HAP_SI
Ws_pt_HAP_SI
XiAir_W_HAP_SI
XiH2O_W_HAP_SI
Z_ptW_HAP_SI
hliq_pt_97_SI
hliqs_t_97_SI
hvaps_t_97_SI
ps_t_97_SI
sliq_pt_97_SI
sliqs_t_97_SI
svaps_t_97_SI
ts_p_97_SI
vliq_pt_97_SI
vliqs_t_97_SI
vvaps_t_97_SI
hicesub_t_06_SI
hvapsub_t_95_SI

s_ptW_HAP_SI	pmel_t_08_SI
t_phphi_HAP_SI	psub_t_08_SI
t_phW_HAP_SI	sicesub_t_06_SI
t_psW_HAP_SI	svapsub_t_95_SI
t_ptwbW_HAP_SI	tmel_p_08_SI
td_pW_HAP_SI	tsub_p_08_SI
ts_ppH2O_HAP_SI	vicesub_t_06_SI
twb_ptW_HAP_SI	vvapsub_t_95_SI
u_ptW_HAP_SI	

Please note that the 64-bit installation program has copied function files with the file extension
 .mexw64
 into your LibHuAirProp directory (the standard being
 " C:\Users\...\Documents\FluidLAB\LibHuAirProp").

2.2 Licensing the LibHuAirProp Property Library

The licensing procedure must be carried out when the prompt message appears. In this case, you will see the "License Information" window for LibHuAirProp (see Figure 2.2.1 below).

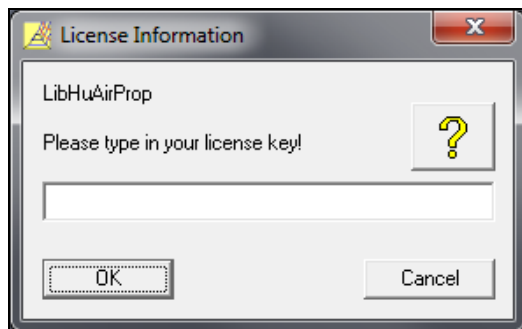


Figure 2.2.1: "License Information" window

Here you are asked to type in the license key which you have obtained from Kretzschmar Consulting Engineers. If you do not have this, or have any questions, you will find contact information on the "Content" page of this User's Guide or by clicking the yellow question mark in the "License Information" window.

If you do not enter a valid license it is still possible to use MATLAB® by clicking "Cancel". In this case, the LibHuAirProp property library will display the result "-11111111" for every calculation.

The "License Information" window will appear every time you use FluidLAB LibHuAirProp until you enter a license code to complete registration. If you decide not to use FluidLAB LibHuAirProp, you can uninstall the program following the instructions given in Section 2.4 of this User's Guide.

With this procedure both the LibHuAirProp-SI and LibHuAirProp-IP property libraries have been licensed.

2.3 Example Calculations of $h = f(p, t, W)$

2.3.1 Preparation for the Example: Calculation of $h = f(p, t, W)$

Now we will calculate, step by step, the air-specific enthalpy h of real moist air as a function of pressure p , temperature t and humidity ratio W using FluidLAB.

Please carry out the following instructions:

- Start Windows Explorer®, Total Commander®, My Computer or another file manager program. The following description refers to Windows Explorer®.
- Your Windows Explorer® should be set to "Details" for easier viewing. Click the "Views" button and select "Details."
- Switch into the program directory of FluidLAB, in which you will find the folder "\LibHuAirProp"; it is generally saved under: "C:\Users\...\Documents\FluidLAB"
- Create the folder "\LibHuAirProp_IP_Example" by clicking on "File" in the Explorer® menu, then "New" in the menu which appears and afterwards selecting "Folder". Name the new folder "\LibHuAirProp_IP_Example."
- You will now see the following window:



Figure 2.3.1: Folders "LibHuAirProp" and "LibHuAirProp_IP_Example"

- Switch into the directory "\LibHuAirProp" within "\FluidLAB", the standard being "C:\Users\...\Documents\FluidLAB\LibHuAirProp".
- You will see the following window:

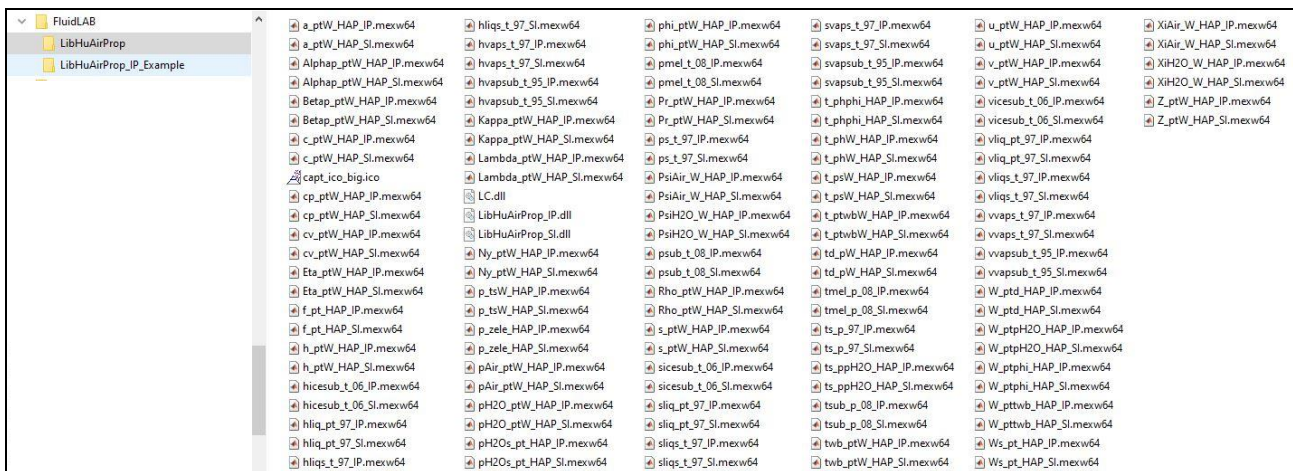


Figure 2.3.2: Contents of the folder "LibHuAirProp"

You will now have to copy the following files into the directory "C:\Users\...\Documents\FluidLAB\LibHuAirProp_IP_Example" in order to calculate the function $h = f(p, t, W)$.

- The following six files are needed:
 - h_ptW_HAP_IP.mexw64,
 - LC.dll,
 - LibHuAirProp_IP.dll,
- Click the file "h_ptW_HAP_IP.mexw64", then click "Edit" in the upper menu bar and select "Copy".
- Switch into the directory "C:\Users\...\Documents\FluidLAB\LibHuAirProp_IP_Example", click "Edit" and then "Paste".
- Repeat these steps in order to copy the other files listed above. You may also select all the above-named files and then copy them as a group (press the Control button to enable multiple markings).
- You will see the following window:



Figure 2.3.3: Contents of the folder "LibHuAirProp_IP_Example"

- Start MATLAB® (if you have not started it already).
- Click the button marked in the next figure in order to open the folder "\LibHuAirProp_IP_Example" in the "Current Directory" window.

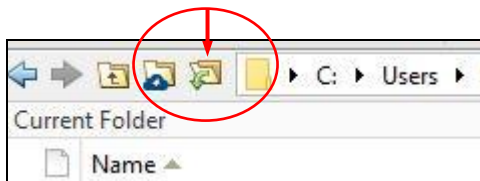


Figure 2.3.4: Selection of the working directory

- Find and select the directory "C:\Users\...\Documents\FluidLAB\LibHuAirProp_IP_Example" in the menu which appears.
- Confirm your selection by clicking the "OK" button.

For the example calculation in a M-File see Subsection 2.3.2 below and for the example calculation in the Command Window please see Subsection 2.3.3 on page I-P – 2/11.

2.3.2 Example: Calculation of $h = f(p, t, W)$ in a M-File

- First of all you need to create an M-File in MATLAB®. Within MATLAB® click "File", then select "New" and afterwards click "Script" in MATLAB® 2010 or later versions.
- If the "Editor" window appears as a separate window, you can embed it into MATLAB® by clicking the insertion arrow (see Figure 2.3.5) in order to obtain a better view.

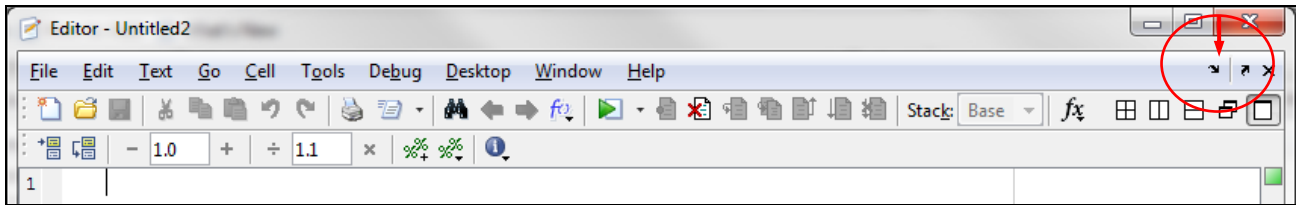


Figure 2.3.5: Embedding the "Editor - Untitled" window

- In Figure 2.3.6 you will see the "Editor - Untitled" window.

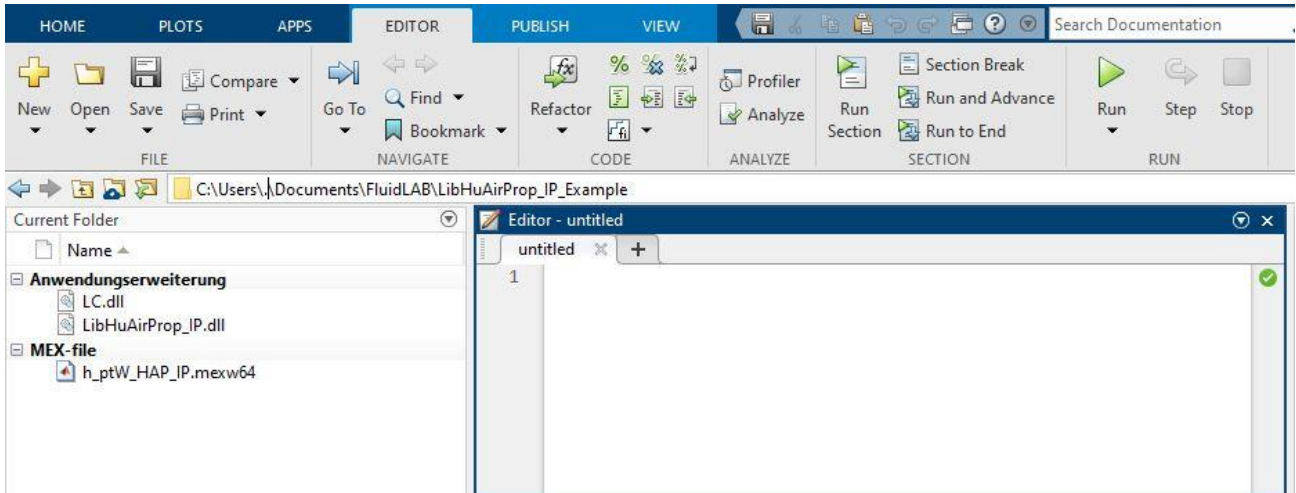


Figure 2.3.6: Embedded "Editor" window

- Now type the following lines in the "Editor - Untitled" window:

Text to be written:	Explanation:
<code>% h_ptW_HAP_IP.m</code>	file name as comment
<code>%%</code>	paragraph separation
<code>p=14.6959; % pressure in psi</code>	declaration of the variables pressure, temperature, and humidity ratio
<code>t=68; % temperature in °F</code>	
<code>W=0.01; % humidity ratio W in lb_w/lb_a</code>	
<code>%%</code>	paragraph separation
<code>h=h_ptW_HAP_IP(p, t, W)</code>	function call
<code>%%</code>	paragraph separation

Remarks:

- The program interprets the first line, starting with "%," to be a data description in "Current Directory".
- Paragraph separations which are mandatory are marked with "%%". This also serves to separate the declaration of variables and calculation instructions.
- The words which are printed in green, start with "%" and come after the variables are comments. They are not in fact absolutely necessary, but they are very helpful for

your overview and to make the process more easily understood.

- Omit the semicolons after the numerical values if you wish to see the result for h and the input parameters.

The values of the function parameters in their corresponding units stand for:

- First operand: Value for $p = 14.6959$ psi
(Range of validity: $p = 0.00145 \dots 1450.4$ psi)
- Second operand: Value for $t = 68$ °F
(Range of validity: $t = -226.67 \dots 662$ °F)
- Third operand: Value for $W = 0.01$ lb_w/lb_a (*lb water per lb air*)
(Range of validity: $W = 0 \dots 10$ lb_w/lb_a)

- Save the "M-File" by clicking the "File" button and then click "Save As...".
- The menu "Save file as:" appears; In this menu, the folder name "LibHuAirProp_IP_Example" must be displayed in the "Save in:" field.
- Next to "File name" you have to type "Example_h_ptW_HAP_IP.m" and afterwards click the "Save" button.

Note.

The name of the example file has to be different in comparison to the name of the used function. For example, the file could not be named "h_ptW_HAP_IP.m" in this case. Otherwise an error message will appear during the calculation.

- You will now see the following window (Figure 2.3.7):

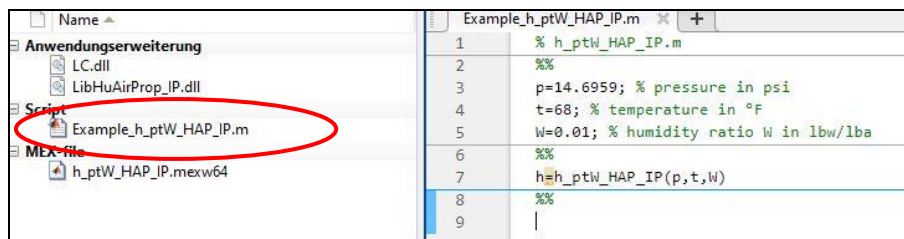


Figure 2.3.7: "Example_h_ptW_HAP_IP.m" M-file

- Within the "Current Directory" window, the file "Example_h_ptW_HAP_IP.m" appears.
- Right-click on this file and select "Run" in the menu which appears (see Figure 2.3.8).

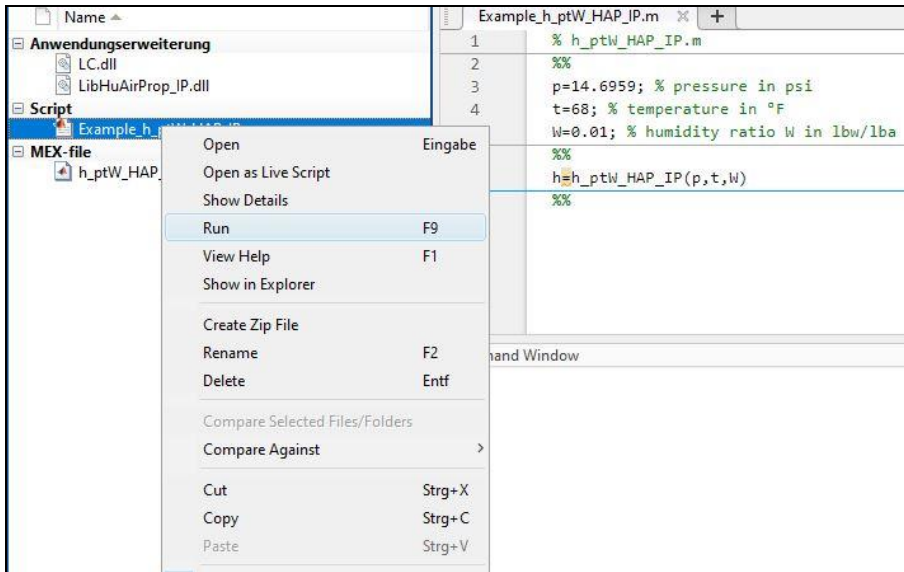


Figure 2.3.8: Running the "Example_h_ptW_HAP_IP.m" M-file

- You will see the following window:

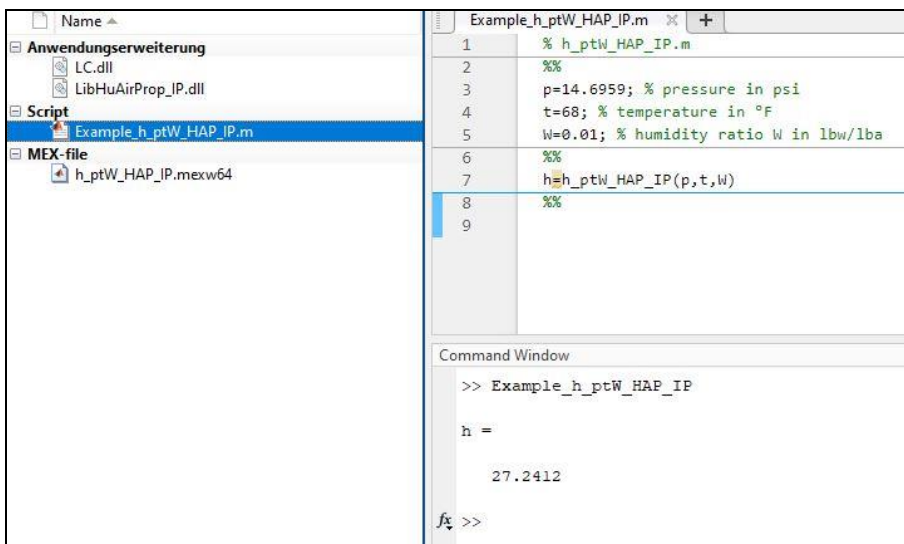


Figure 2.3.9: MATLAB® with calculated result

The result for h appears in the "Command Window".

⇒ The result in our sample calculation here is: " $h = 27.2412$ ". The corresponding unit is Btu/lb_a (see table of the property functions in Chapter 1).

To be able to calculate other values, you have to copy the associated mexw64 files as well because MATLAB® can only access functions that are located in the "Current Directory" window. The example calculated can be found in the directory C:\Users\...\Documents\FluidLAB\LibHuAirProp_IP_Example," and you may use it as a basis for further calculations using FluidLAB.

2.3.3 Example: Calculation of $h = f(p, t, W)$ in the Command Window

Please follow the instructions in Subsection 2.3.1 (pages I-P – 2/6 to I-P – 2/7).

- You will see the following window (Figure 2.3.10):

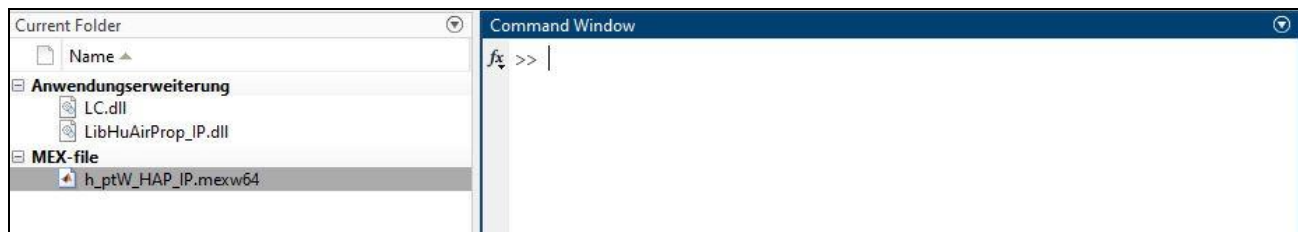


Figure 2.3.10: MATLAB® with necessary files

Corresponding to the table of the property functions in Chapter 1 you have to call up the function "h_ptW_HAP_IP" as follows for calculating $h = f(p, t, W)$.

Write "**h=h_ptW_HAP_IP(14.6959,68,0.01)**" within the "Command Window".

The values of the function parameters in their corresponding units stand for:

- **First operand: Value for $p = 14.6959$ psi**
(Range of validity: $p = 0.00145 \dots 1450.4$ psi)
 - **Second operand: Value for $t = 68$ °F**
(Range of validity: $t = -226.67 \dots 662$ °F)
 - **Third operand: Value for $W = 0.01$ lb_w/lb_a (lb water per lb air)**
(Range of validity: $W = 0 \dots 10$ lb_w/lb_a)
- Confirm your entry by pressing the "ENTER" button.
 - You will see the following window (Figure 2.3.11):

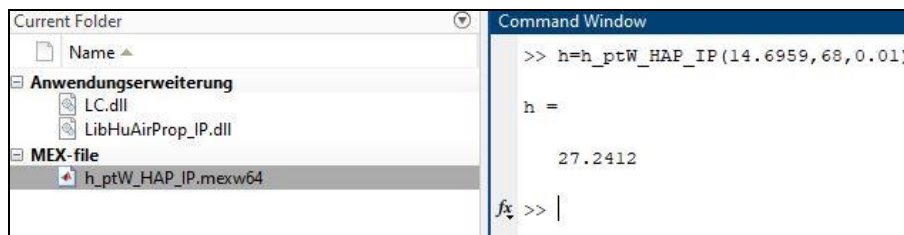


Figure 2.3.11: MATLAB® with calculated result

⇒ In the "Command Window" you will see the result "h = 27.2412". The corresponding unit is Btu/lb_a (see table of the property functions in Chapter 1).

To be able to calculate other values, you will have to copy the respective mexw64 files into the working directory as well, because MATLAB® can only access functions that are located in the "Current Directory" window.

2.4 Removing FluidLAB including LibHuAirProp

To remove the property library ASHRAE-LibHuAirProp from your hard drive in Windows®, click "Start" in the Windows® task bar, select "Settings" and click "Control Panel".

Now double-click on "Add or Remove Programs". In the list box of the "Add or Remove Programs" window that appears select "FluidLAB LibHuAirProp" by clicking on it and click the "Change/Remove" button.

In the following dialog box click "Automatic" and then click the "Next >" button.

Confirm the following menu "Perform Uninstall" by clicking the "Finish" button.

Finally, close the "Add or Remove Programs" and "Control Panel" windows.

Now, FluidLAB has been removed.

If there is no library other than ASHRAE-LibHuAirProp installed, the directory "FluidLAB" will be removed as well.

3 Property Functions of ASHRAE-LibHuAirProp-IP

3.1 Functions for Real Moist Air

Thermal Diffusivity $a = f(p, t, W)$
--

Function Name:

a_ptW_HAP_IP

Fortran Program:

REAL*8 FUNCTION A_PTW_HUAIRPROP(P,T,W), REAL*8 P,T,W

Input Values:

p - Total pressure p in psi
 t - Temperature t in °F
 W - Humidity ratio W in lb_w/lb_a

Result:a_ptW_HAP_IP - Thermal diffusivity of humid air in ft²/s**Range of Validity:**

Temperature t : from -99.67°F to 662°F
 Total pressure p : from 0.00145 psi to 1450.4 psi
 Humidity ratio W : $0 \leq W \leq W_s$

Comments:

- Thermal diffusivity $a = \frac{\lambda}{\rho \cdot c_p}$
- Calculation for supersaturated humid air ($W > W_s$) is not possible

Result for Wrong Input Values:

a_ptW_HAP_IP = -1000

References:

$\lambda(p, t, W)$ Herrmann et al. [3], [4]
 $\rho(p, t, W)$ Herrmann et al. [1], [2]
 $c_p(p, t, W)$ Herrmann et al. [1], [2]

Relative Pressure Coefficient $\alpha_p = f(p, t, W)$
Function Name:

alphap_ptW_HAP_IP

Fortran Program:

```
REAL*8 FUNCTION ALPHAP_PTW_HUAIRPROP(P,T,W), REAL*8 P,T,W
```

Input Values:

p - Total pressure p in psi
 t - Temperature t in °F
 W - Humidity ratio W in lb_w/lb_a

Result:

alphap_ptW_HAP_IP - Relative pressure coefficient of humid air in 1/°R

Range of Validity:

Temperature t : from -225.67°F to 662°F
 Total pressure p : from 0.00145 psi to 1450.4 psi
 Humidity ratio W : $0 \leq W \leq W_s$

Comments:

- Relative pressure coefficient $\alpha_p = \frac{1}{p} \left(\frac{\partial p}{\partial T} \right)_v$
- Calculation for supersaturated humid air ($W > W_s$) is not possible

Result for Wrong Input Values:

alphap_ptW_HAP_IP = -1000

References:

$\alpha_p(p, t, W)$ Herrmann et al. [1], [2]

Isothermal Stress Coefficient $\beta_p = f(p, t, W)$
Function Name:

betap_ptW_HAP_IP

Fortran Program:

REAL*8 FUNCTION BETAP_PTW_HUAIRPROP(P,T,W), REAL*8 P,T,W

Input Values:

p - Total pressure p in psi
 t - Temperature t in °F
 W - Humidity ratio W in lb_w/lb_a

Result:betap_ptW_HAP_IP - Isothermal stress coefficient of humid air in lb/ft³**Range of Validity:**

Temperature t : from -225.67°F to 662°F
 Total pressure p : from 0.00145 psi to 1450.4 psi
 Humidity ratio W : $0 \leq W \leq W_s$

Comments:

- Isothermal stress coefficient $\beta_p = -\frac{1}{p} \left(\frac{\partial p}{\partial v} \right)_T$
- Calculation for supersaturated humid air ($W > W_s$) is not possible

Result for Wrong Input Values:

betap_ptW_HAP_IP = -1000

References: $\beta_p(p, t, W)$ Herrmann et al. [1], [2]

Speed of Sound $c = f(p, t, W)$
Function Name:

c_ptW_HAP_IP

Fortran Program:

```
REAL*8 FUNCTION C_PTW_HUAIRPROP(P,T,W), REAL*8 P,T,W
```

Input Values:

p - Total pressure p in psi
 t - Temperature t in °F
 W - Humidity ratio W in lb_w/lb_a

Result:

c_ptW_HAP_IP - Speed of sound of humid air in ft/s

Range of Validity:

Temperature t : from -225.67°F to 662°F
 Total pressure p : from 0.00145 psi to 1450.4 psi
 Humidity ratio W : $0 \leq W \leq 10 \text{ lb}_w/\text{lb}_a$

Comments:

- Speed of sound $c = v \sqrt{-\left(\frac{\partial p}{\partial v}\right)_s}$

- When calculating supersaturated air an ideal mixing model (saturated humid air and water droplets (or ice crystals) homogeneously mixed) is applied

Result for Wrong Input Values:

c_ptW_HAP_IP = -1000

References:

$c(p, t, W)$ Herrmann et al. [1], [2]

Isobaric Heat Capacity $c_p = f(p, t, W)$
Function Name:

cp_ptW_HAP_IP

Fortran Program:

```
REAL*8 FUNCTION CP_PTW_HUAIRPROP(P,T,W), REAL*8 P,T,W
```

Input Values:

p - Total pressure p in psi
 t - Temperature t in °F
 W - Humidity ratio W in lb_w/lb_a

Result:

cp_ptW_HAP_IP - Isobaric heat capacity of humid air in Btu/(lb °R)

Range of Validity:

Temperature t : from -225.67°F to 662°F
 Total pressure p : from 0.00145 psi to 1450.4 psi
 Humidity ratio W : $0 \leq W \leq W_s$

Comments:

- Isobaric heat capacity $c_p = \left(\frac{\partial h}{\partial T} \right)_p$
- Calculation for supersaturated humid air ($W > W_s$) is not possible

Result for Wrong Input Values:

cp_ptW_HAP_IP = -1000

References:

$c_p(p, t, W)$ Herrmann et al. [1], [2]

Isochoric Heat Capacity $c_v = f(p, t, W)$
Function Name:

cv_ptW_HAP_IP

Fortran Program:

```
REAL*8 FUNCTION CV_PTW_HUAIRPROP(P,T,W), REAL*8 P,T,W
```

Input Values:

p - Total pressure p in psi
 t - Temperature t in °F
 W - Humidity ratio W in lb_w/lb_a

Result:

cv_ptW_HAP_IP - Isochoric heat capacity of humid air in Btu/(lb °R)

Range of Validity:

Temperature t : from -225.67°F to 662°F
 Total pressure p : from 0.00145 psi to 1450.4 psi
 Humidity ratio W : $0 \leq W \leq W_s$

Comments:

- Isochoric heat capacity $c_v = \left(\frac{\partial u}{\partial T} \right)_v$
- Calculation for supersaturated humid air ($W > W_s$) is not possible

Result for Wrong Input Values:

cv_ptW_HAP_IP = -1000

References:

$c_v(p, t, W)$ Herrmann et al. [3], [4]

Enhancement Factor $f = f(p,t)$ **Function Name:**

f_pt_HAP_IP

Fortran Program:

REAL*8 FUNCTION F_PT_HUAIRPROP(P,T), REAL*8 P,T

Input Values:

p - Total pressure p in psi
 t - Temperature t in °F

Result:

f_pt_HAP_IP - Enhancement factor of water (decimal ratio)

Range of Validity:

Temperature t : from -225.67°F to 662°F
 Total pressure p : from 0.00145 psi to 1450.4 psi

Comments:

- Enhancement factor $f = \frac{\rho_{H_2O,s}}{\rho_s(t)}$

with $\rho_s(t)$ for $t \geq 32^\circ\text{F}$ - Steam pressure of water

for $t < 32^\circ\text{F}$ - Sublimation pressure of water

- Describes the enhancement of the saturation pressure of water in the air atmosphere under elevated pressure

- Derived iteratively from the isothermal compressibility of liquid water, from Henry's constant [15], [16] and from the virial coefficients of air, water, and the air-water mixture

Result for Wrong Input Values:

f_pt_HAP_IP = -1000

References:

$f(p,t)$ Herrmann et al. [1], [2]

Air-Specific Enthalpy $h = f(p, t, W)$
Function Name:

h_ptW_HAP_IP

Fortran Program:

```
REAL*8 FUNCTION H_PTW_HUAIRPROP(P,T,W), REAL*8 P,T,W
```

Input Values:

p - Total pressure p in psi
 t - Temperature t in °F
 W - Humidity ratio W in lb_w/lb_a

Result:

h_ptW_HAP_IP - Air-specific enthalpy in Btu/lb_a

Range of Validity:

Temperature t : from -225.67°F to 662°F
 Total pressure p : from 0.00145 psi to 1450.4 psi
 Humidity ratio W : $0 \leq W \leq 10 \text{ lb}_w/\text{lb}_a$

Comments:

- When calculating supersaturated air an ideal mixing model (saturated humid air and water droplets (or ice crystals) homogeneously mixed) is applied

Result for Wrong Input Values:

h_ptW_HAP_IP = -1000

References:

$h(p, t, W)$ Herrmann et al. [1], [2]
 $h_w(p, t)$ IAPWS-IF97 [7], [8] and IAPWS-06 [11]
 $h_a(t)$ Lemmon et al. [14]

Dynamic Viscosity $\eta = f(p, t, W)$

Function Name:

Eta_ptW_HAP_IP

Fortran Program:

REAL*8 FUNCTION ETA_PTW_HUAIRPROP(P,T,W), REAL*8 P,T,W

Input Values:

p - Total pressure p in psi
 t - Temperature t in °F
 W - Humidity ratio W in lb_w/lb_a

Result:Eta_ptW_HAP_IP - Dynamic viscosity of humid air in (lb s/ft²)**Range of Validity:**

Temperature t : from -99.67°F to 662°F
 Total pressure p : from 0.00145 psi to 1450.4 psi
 Humidity ratio W : $0 \leq W \leq 10 \text{ lb}_w/\text{lb}_a$

Comments:

- A new very accurate algorithm is implemented between 32°F and 662°F
- When calculating supersaturated air an ideal mixing model (saturated humid air and water droplets (or ice crystals) homogeneously mixed) is applied

Result for Wrong Input Values:

Eta_ptW_HAP_IP = -1000

References:

$\eta(p, t, W)$ Herrmann et al. [3], [4]
 $\eta_w(p, t)$ IAPWS-IF97 [7], [8] and IAPWS-06 [19]
 $\eta_a(t)$ Lemmon et al. [18]

Isentropic Exponent $\kappa = f(p, t, W)$

Function Name:

Kappa_ptW_HAP_IP

Fortran Program:

```
REAL*8 FUNCTION KAPPA_PTW_HUAIRPROP(P,T, W), REAL*8 P,T,W
```

Input Values:

p - Total pressure p in psi
 t - Temperature t in °F
 W - Humidity ratio W in lb_w/lb_a

Result:

Kappa_ptW_HAP_IP - Isentropic exponent

Range of Validity:

Temperature t : from -225.67°F to 662°F
 Total pressure p : from 0.00145 psi to 1450.4 psi
 Humidity ratio W : $0 \leq W \leq 10 \text{ lb}_w/\text{lb}_a$

Comments:

- Isentropic exponent $\kappa = -\frac{v}{p} \left(\frac{\partial p}{\partial v} \right)_s$
- When calculating supersaturated air an ideal mixing model (saturated humid air and water droplets homogeneously mixed) is applied for $t \geq 32^\circ\text{F}$. For temperatures below (ice fog) the value of the saturated state is applied.

Result for Wrong Input Values:

Kappa_ptW_HAP_IP = -1000

References:

$v(p, t, W)$ Herrmann et al. [1], [2]

Thermal Conductivity $\lambda = f(p, t, W)$ **Function Name:**

Lambda_ptW_HAP_IP

Fortran Program:

REAL*8 FUNCTION LAMBDA_PTW_HUAIRPROP(P,T, W), REAL*8 P,T,W

Input Values:

p - Total pressure p in psi
 t - Temperature t in °F
 W - Humidity ratio W in lb_w/lb_a

Result:

Lambda_ptW_HAP_IP - Thermal conductivity in Btu/(h ft °R)

Range of Validity:

Temperature t : from -99.67°F to 662°F
 Total pressure p : from 0.00145 psi to 1450.4 psi
 Humidity ratio W : $0 \leq W \leq 10$ lb_w/lb_a

Comments:

- A new very accurate algorithm is implemented between 32°F and 662°F
- When calculating supersaturated air an ideal mixing model (saturated humid air and water droplets (or ice crystals) homogeneously mixed) is applied

Result for Wrong Input Values:

Lambda_ptW_HAP_IP = -1000

References:

$\lambda(p, t, W)$ Herrmann et al. [3], [4]
 $\lambda_w(p, t)$ IAPWS-IF97 [7], [8] and IAPWS-08 [20]
 $\lambda_a(t)$ Lemmon et al. [18]

Kinematic Viscosity $\nu = f(p, t, W)$
Function Name:

Ny_ptW_HAP_IP

Fortran Program:

```
REAL*8 FUNCTION NY_PTW_HUAIRPROP(P,T,W), REAL*8 P,T,W
```

Input Values:

p - Total pressure p in psi
 t - Temperature t in °F
 W - Humidity ratio W in lb_w/lb_a

Result:Ny_ptW_HAP_IP - Kinematic viscosity in ft²/s**Range of Validity:**

Temperature t : from -99.67°F to 662°F
 Total pressure p : from 0.00145 psi to 1450.4 psi
 Humidity ratio W : $0 \leq W \leq W_s$

Comments:

- Kinematic Viscosity $\nu = \frac{\eta}{\rho}$

Result for Wrong Input Values:

Ny_ptW_HAP_IP = -1000

References:

$\eta(p, t, W)$ Herrmann et al. [3], [4]
 $\rho(p, t, W)$ Herrmann et al. [1], [2]

Backward Function: Pressure $p = f(t,s,W)$ **Function Name:**

p_tsW_HAP_IP

Fortran Program:

REAL*8 FUNCTION P_TSW_HUAIRPROP(T,S,W), REAL*8 T,S,W

Input Values:

t - Temperature t in °F
 s - Air-specific entropy s in Btu/(lb_a °R)
 W - Humidity ratio W in lb_w/lb_a

Result:

p_tsW_HAP_IP - Total pressure of humid air in psi

Range of Validity:

Temperature t : from -225.67°F to 662°F
 Air-specific entropy s : from -6.32 Btu/(lb_a °R) to 9.32877 Btu/(lb_a °R)
 Humidity ratio W : $0 \leq W \leq 10$ lb_w/lb_a

Comments:- Iteration of total pressure p from $s = f(p,t,W)$ **Result for Wrong Input Values:**

p_tsW_HAP_IP = -1000

References: $s(p,t,W)$ Herrmann et al. [1], [2]

Pressure $p = f(z_{\text{ele}})$ **Function Name:**

p_zele_HAP_IP

Fortran Program:

REAL*8 FUNCTION P_ZELE_HUAIRPROP(ZELE), REAL*8 ZELE

Input Values:z_{ele} - Elevation z_{ele} in ft**Result:**

p_zele_HAP_IP - Pressure of humid air in psi

Range of Validity:Elevation z_{ele} from -16,404 ft to 36,089 ft**Comments:**

- Pressure of humid air from elevation

$$- p(z_{\text{ele}}) = 14.696 \text{ psi} \cdot \left(1 - 6.8754 \cdot 10^{-6} \cdot \frac{z_{\text{ele}}}{\text{ft}} \right)^{5.256}$$

Result for Wrong Input Values:

p_zele_HAP_IP = -1000

References:p(z_{ele}) ASHRAE [23]

Partial Pressure of Air $p_{\text{Air}} = f(p, t, W)$
Function Name:

pAir_ptW_HAP_IP

Fortran Program:

REAL*8 FUNCTION PAIR_PTW_HUAIRPROP(P,T,W), REAL*8 P,T,W

Input Values:

p - Total pressure p in psi
 t - Temperature t in °F
 W - Humidity ratio W in lb_w/lb_a

Result:

pAir_ptW_HAP_IP - Partial pressure of (dry) air in humid air in psi

Range of Validity:

Temperature t : from -225.67°F to 662°F
 Total pressure p : from 0.00145 psi to 1450.4 psi
 Humidity ratio W : $0 \leq W \leq 10 \text{ lb}_w/\text{lb}_a$

Comments:

- Partial pressure of (dry) air in humid air $p_{\text{Air}} = 1 - p_{\text{H}_2\text{O}}$
- Partial pressure of water vapor at saturation is calculated in case of supersaturated humid air ($W > W_s(p, t)$)
- The temperature value is used to calculate the saturation state

Result for Wrong Input Values:

pAir_ptW_HAP_IP = -1000

References: $p_{\text{H}_2\text{O}}(p, W)$ Herrmann et al. [1], [2]

Partial Pressure of Water Vapor $p_{\text{H}_2\text{O}} = f(p, t, W)$

Function Name:

pH2O_ptW_HAP_IP

Fortran Program:

```
REAL*8 FUNCTION PH2O_PTW_HUAIRPROP(P,T,W), REAL*8 P,T,W
```

Input Values:

p - Total pressure p in psi
 t - Temperature t in °F
 W - Humidity ratio W in lb_w/lb_a

Result:

pH2O_ptW_HAP_IP - Partial pressure of water vapor in humid air in psi

Range of Validity:

Temperature t : from -225.67°F to 662°F
 Total pressure p : from 0.00145 psi to 1450.4 psi
 Humidity ratio W : $0 \leq W \leq 10 \text{ lb}_w/\text{lb}_a$

Comments:

- Partial pressure of water vapor in humid air $p_{\text{H}_2\text{O}} = \frac{W \cdot p}{\left(\frac{R_a}{R_w} + W\right)}$
- Partial pressure of water vapor at saturation is calculated in case of supersaturated humid air ($W > W_s(p, t)$)
- The temperature value is used to calculate the saturation state

Result for Wrong Input Values:

pH2O_ptW_HAP_IP = -1000

References:

$p_{\text{H}_2\text{O}}(p, W)$ Herrmann et al. [1], [2]

Partial Sat. Pressure of Water Vapor in Humid Air $p_{\text{H}_2\text{O},s} = f(p, t)$
Function Name:

pH2Os_pt_HAP_IP

Fortran Program:

REAL*8 FUNCTION PH2OS_PT_HUAIRPROP(P,T), REAL*8 P,T

Input Values: p - Total pressure p in psi t - Temperature t in °F**Result:**

pH2Os_pt_HAP_IP - Partial saturation pressure of water vapor in humid air in psi

Range of Validity:Temperature t : from -225.67°F to 662°FTotal pressure p : from 0.00145 psi to 1450.4 psi**Comments:**- Partial pressure of water vapor at saturation $p_{\text{H}_2\text{O},s} = f \cdot p_s(t)$ with $p_s(t)$ for $t \geq 32^\circ\text{F}$ - Steam pressure of waterfor $t < 32^\circ\text{F}$ - Sublimation pressure of water**Result for Wrong Input Values:**

pH2Os_pt_HAP_IP = -1000

References: $f(p, t)$ Herrmann et al. [1], [2] $p_s(t)$ for $t \geq 32^\circ\text{F}$ IAPWS-IF97 [7], [8]for $t < 32^\circ\text{F}$ IAPWS-08 [11]

Relative Humidity $\phi = f(p, t, W)$

Function Name:

phi_ptW_HAP_IP

Fortran Program:

```
REAL*8 FUNCTION PHI_PTW_HUAIRPROP(P,T,W), REAL*8 P,T,W
```

Input Values:

p - Total pressure p in psi
 t - Temperature t in °F
 W - Humidity ratio W in lb_w/lb_a

Result:

phi_ptW_HAP_IP - Relative humidity (decimal ratio)

Range of Validity:

Temperature t : from -225.67°F to 662°F
 Total pressure p : from 0.00145 psi to 1450.4 psi
 Humidity ratio W : $0 \leq W \leq 10$ lb_w/lb_a

Comments:

- Relative humidity $\phi = \frac{p_{H2O}}{p_{H2O,s}}$

- This equation is valid for $p_{H2O} \leq p_{H2O,s}$ and for $0 \leq \phi \leq 1$

Result for Wrong Input Values:

phi_ptW_HAP_IP = -1000

References:

$\phi(p, t, W)$ Herrmann et al. [1], [2]

Prandtl Number $Pr = f(p, t, W)$ **Function Name:**

Pr_ptW_HAP_IP

Fortran Program:

REAL*8 FUNCTION PR_PTW_HUAIRPROP(P,T,W), REAL*8 P,T,W

Input Values:

p - Total pressure p in psi
 t - Temperature t in °F
 W - Humidity ratio W in lb_w/lb_a

Result:

Pr_ptW_HAP_IP - Prandtl number

Range of Validity:

Temperature t : from -99.67°F to 662°F
 Total pressure p : from 0.00145 psi to 1450.4 psi
 Humidity ratio W : $0 \leq W \leq 10 \text{ lb}_w/\text{lb}_a$

Comments:

- Prandtl number $Pr = \frac{\eta \cdot c_p}{\lambda}$
- Calculation for supersaturated humid air ($W > W_s$) is not possible

Result for Wrong Input Values:

Pr_ptW_HAP_IP = -1000

References:

$\eta(p, t, W)$ Herrmann et al. [3], [4]
 $c_p(p, t, W)$ Herrmann et al. [3], [4]
 $\lambda(p, t, W)$ Lemmon et al. [20]

Mole Fraction of Air $\psi_{\text{Air}} = f(W)$
Function Name:

PsiAir_W_HAP_IP

Fortran Program:

REAL*8 FUNCTION PSIAIR_W_HUAIRPROP(W), REAL*8 W

Input Values: W - Humidity ratio W in lb_w/lb_a **Result:**PsiAir_W_HAP_IP - Mole fraction of (dry) air in humid air in mol_a/mol **Range of Validity:**Humidity ratio W : $0 \leq W \leq 10 \text{lb}_w/\text{lb}_a$ **Comments:**

- Mole fraction of air $\psi_{\text{Air}} = 1 - \psi_{\text{H}_2\text{O}} = 1 - \left(\frac{W}{\frac{R_a}{R_{\text{H}_2\text{O}}} + W} \right)$

Result for Wrong Input Values:

PsiAir_W_HAP_IP = -1000

References: $\psi_{\text{Air}}(W)$ Herrmann et al. [1], [2]

Mole Fraction of Water $\psi_{H_2O} = f(W)$
Function Name:

PsiH2O_W_HAP_IP

Fortran Program:

REAL*8 FUNCTION PSIH2O_W_HUAIRPROP(W), REAL*8 W

Input Values: W - Humidity ratio W in lb_w/lb_a **Result:**

PsiH2O_W_HAP_IP - Mole fraction of water in humid air in molw/mol

Range of Validity:Humidity ratio W : $0 \leq W \leq 10 lb_w/lb_a$ **Comments:**

- Mole fraction of water $\psi_{H_2O} = \frac{W}{\frac{R_a}{R_{H_2O}} + W}$

Result for Wrong Input Values:

PsiH2O_W_HAP_IP = -1000

References: $\psi_{H_2O}(W)$ Herrmann et al. [1], [2]

Density $\rho = f(p, t, W)$ **Function Name:**

Rho_ptW_HAP_IP

Fortran Program:

REAL*8 FUNCTION RHO_PTW_HUAIRPROP(P,T,W), REAL*8 P,T,W

Input Values:

p - Total pressure p in psi
 t - Temperature t in °F
 W - Humidity ratio W in lb_w/lb_a

Result:Rho_ptW_HAP_IP - Density of humid air in lb/ft³**Range of Validity:**

Temperature t : from -225.67°F to 662°F
 Total pressure p : from 0.00145 psi to 1450.4 psi
 Humidity ratio W : $0 \leq W \leq 10 \text{ lb}_w/\text{lb}_a$

Comments:

- Density of humid air obtained from air-specific volume: $\rho = \frac{1+W}{v}$

Result for Wrong Input Values:

Rho_ptW_HAP_IP = -1000

References: $\rho(p, t, W)$ Herrmann et al. [1], [2]

Air-Specific Entropy $s = f(p, t, W)$ **Function Name:**

s_ptW_HAP_IP

Fortran Program:

REAL*8 FUNCTION S_PTW_HUAIRPROP(P,T,W), REAL*8 P,T,W

Input Values:

p - Total pressure p in psi
 t - Temperature t in °F
 W - Humidity ratio W in lb_w/lb_a

Result:s_ptW_HAP_IP - Air-specific entropy in Btu/(lb_a · °R)**Range of Validity:**

Temperature t : from -225.67°F to 662°F
 Total pressure p : from 0.00145 psi to 1450.4 psi
 Humidity ratio W : $0 \leq W \leq 10 \text{ lb}_w/\text{lb}_a$

Comments:

- When calculating supersaturated air an ideal mixing model (saturated humid air and water droplets (or ice crystals) homogeneously mixed) is applied

Result for Wrong Input Values:

s_ptW_HAP_IP = -1000

References: $s(p, t, W)$ Herrmann et al. [1], [2]

Backward Function: Temperature $t = f(p, h, \varphi)$
Function Name:

t_phphi_HAP_IP

Fortran Program:

```
REAL*8 FUNCTION T_PHPHI_HUAIRPROP(P,H,PHI), REAL*8 P,H,PHI
```

Input Values:

- p - Total pressure p in psi
- h - Air-specific enthalpy h in Btu/lb_a
- φ - Relative humidity φ (decimal ratio)

Result:

t_phphi_HAP_IP - Temperature from pressure, enthalpy, and relative humidity in °F

Range of Validity:

- Total pressure p : from 0.00145 psi to 1450.4 psi
- Air-specific enthalpy h : from -2469.22 Btu/lb_a to 12772.088 Btu/lb_a
- Relative humidity φ : $0 \leq \varphi \leq 1$

Comments:

- Iteration of temperature t from $h = f(p, t, W)$ using $W = f(p, t, \varphi)$

Result for Wrong Input Values:

t_phphi_HAP_IP = -1000

References:

$h(p, t, W)$ Herrmann et al. [1], [2]

Backward Function: Temperature $t = f(p, h, W)$ **Function Name:**

t_phW_HAP_IP

Fortran Program:

REAL*8 FUNCTION T_PHW_HUAIRPROP(P,H,W), REAL*8 P,H,W

Input Values:

p - Total pressure p in psi
 h - Air-specific enthalpy h in Btu/lb_a
 W - Humidity ratio W in lb_w/lb_a

Result:

t_phW_HAP_IP - Temperature from pressure, enthalpy, and humidity ratio in °F

Range of Validity:

Total pressure p : from 0.00145 psi to 1450.4 psi
 Air-specific enthalpy h : from -2469.22 Btu/lb_a to 12772.088 Btu/lb_a
 Humidity ratio W : $0 \leq W \leq 10$ lb_w/lb_a

Comments:- Iteration of temperature t from $h = f(p, t, W)$ **Result for Wrong Input Values:**

t_phW_HAP_IP = -1000

References: $h(p, t, W)$ Herrmann et al. [1], [2]

Backward Function: Temperature $t = f(p,s,W)$
Function Name:

t_psW_HAP_IP

Fortran Program:

```
REAL*8 FUNCTION T_PSW_HUAIRPROP(P,S,W), REAL*8 P,S,W
```

Input Values:

p - Total pressure p in psi
 s - Air-specific entropy in Btu/(lb_a · °R)
 W - Humidity ratio W in lb_w/lb_a

Result:

t_psW_HAP_IP - Temperature from pressure, entropy, and humidity ratio in °F

Range of Validity:

Total pressure p : from 0.00145 psi to 1450.4 psi
 Air-specific entropy s : from -6.32 Btu/(lb_a °R) to 9.32877 Btu/(lb_a °R)
 Humidity ratio W : $0 \leq W \leq 10 \text{ lb}_w/\text{lb}_a$

Comments:

- Iteration of temperature t from $s = f(p,t,W)$

Result for Wrong Input Values:

t_psW_HAP_IP = -1000

References:

$s(p,t,W)$ Herrmann et al. [1], [2]

Backward Function: Temperature $t = f(p, t_{wb}, W)$

Function Name:

t_ptwbW_HAP_IP

Fortran Program:

```
REAL*8 FUNCTION T_PTWBW_HUAIRPROP(P,TWB,W), REAL*8 P,TWB,W
```

Input Values:

p - Total pressure p in psi
 t_{wb} - Wet-bulb temperature in °F
 W - Humidity ratio W in lb_w/lb_a

Result:

t_ptwbW_HAP_IP - Temperature from pressure, wet bulb temperature and humidity ratio in °F

Range of Validity:

Total pressure p : from 0.00145 psi to 1450.4 psi
Wet bulb temperature t_{wb} : from -225.67°F to 662°F
Humidity ratio W : $0 \leq W \leq 10 \text{ lb}_w/\text{lb}_a$

Comments:

- Iteration of temperature t from $t_{wb} = f(p, t, W)$

Result for Wrong Input Values:

t_ptwbW_HAP_IP = -1000

References:

$t_{wb}(p, t, W)$ Herrmann et al. [1], [2]

Dew-Point/Frost-Point Temperature $t_d = f(p, W)$
Function Name:

td_pW_HAP_IP

Fortran Program:

REAL*8 FUNCTION TD_PW_HUAIRPROP(P,W), REAL*8 P,W

Input Values:

p - Total pressure p in psi
 W - Humidity ratio W in lb_w/lb_a

Result:

td_pW_HAP_IP - Dew-point/frost-point temperature in °F

Range of Validity:

Total pressure p : from 0.00145 psi to 1450.4 psi
Humidity ratio W : $0 \leq W \leq 10 \text{ lb}_w/\text{lb}_a$

Comments:

Dew-point temperature $t_d = t_s(\rho_{\text{H}_2\text{O}})$ for $t \geq 32^\circ\text{F}$ (saturation temperature of water in humid air)

$t_d = t_{\text{sub}}(\rho_{\text{H}_2\text{O}})$ for $t \leq 32^\circ\text{F}$ (sublimation temperature of water in humid air)

Result for Wrong Input Values:

td_pW_HAP_IP = -1000

References:

$t_s(\rho_{\text{H}_2\text{O}})$ for $t_d \geq 32^\circ\text{F}$ IAPWS-IF97 [7], [8]

$t_{\text{sub}}(\rho_{\text{H}_2\text{O}})$ for $t_d \leq 32^\circ\text{F}$ IAPWS-08 [11]

$\rho_{\text{H}_2\text{O}}$ Herrmann et. al. [1], [2]

Saturation Temperature $t_s = f(p, p_{H_2O})$

Function Name:

ts_ppH2O_HAP_IP

Fortran Program:

```
REAL*8 FUNCTION TS_PPH2O_HUAIRPROP(P,PH2O), REAL*8 P,PH2O
```

Input Values:

p - Total pressure p in psi
 p_{H_2O} - Partial saturation pressure of water p_{H_2O} in psi

Result:

ts_ppH2O_HAP_IP - Saturation temperature of water in humid air in °F

Range of Validity:

Total pressure p : from 0.00145 psi to 1450.4 psi
Partial pressure p_{H_2O} : from 0.00145 psi to 1450.4 psi

Comments:

- Iteration of saturation temperature t_s from $p_{H_2O,s} = f(p, t)$

Result for Wrong Input Values:

ts_ppH2O_HAP_IP = -1000

References:

$p_{H_2O,s}$ Herrmann et. al. [1], [2]

Wet-Bulb/Ice-Bulb Temperature $t_{wb} = f(p, t, W)$
Function Name:

twb_ptW_HAP_IP

Fortran Program:

```
REAL*8 FUNCTION TWB_PTW_HUAIRPROP(P,T,W), REAL*8 P,T,W
```

Input Values:

p - Total pressure p in psi
 t - Temperature t in °F
 W - Humidity ratio W in lb_w/lb_a

Result:

twb_ptW_HAP_IP - Wet-bulb/ice-bulb temperature in °F

Range of Validity:

Temperature t : from -225.67°F to 662°F
 Total pressure p : from 0.00145 psi to 1450.4 psi
 Humidity ratio W : $0 \leq W \leq 10 \text{lb}_w/\text{lb}_a$

Comments:

- Iteration of wet-bulb temperature t_{wb} from $h^{\text{unsaturated}}(p, t, W) = h^{\text{fog}}(p, t_{wb}, W)$
- Calculation for supersaturated humid air ($W > W_s$) is not possible

Result for Wrong Input Values:

twb_ptW_HAP_IP = -1000

References: $t_{wb}(p, t, W)$ Herrmann et al. [1], [2]

Air-Specific Internal Energy $u = f(p, t, W)$ **Function Name:**

u_ptW_HAP_IP

Fortran Program:

REAL*8 FUNCTION U_PTW_HUAIRPROP(P,T,W), REAL*8 P,T,W

Input Values:

p - Total pressure p in psi
 t - Temperature t in °F
 W - Humidity ratio W in lb_w/lb_a

Result:u_ptW_HAP_IP - Air-specific internal energy in Btu/lb_a**Range of Validity:**

Temperature t : from -225.67°F to 662°F
 Total pressure p : from 0.00145 psi to 1450.4 psi
 Humidity ratio W : $0 \leq W \leq 10$ lb_w/lb_a

Comments:- Internal energy $u = h - pv$ **Result for Wrong Input Values:**

u_ptW_HAP_IP = -1000

References: $u(p, t, W)$ Herrmann et al. [1], [2]

Air-Specific Volume $v = f(p, t, W)$
Function Name:

v_ptW_HAP_IP

Fortran Program:

```
REAL*8 FUNCTION V_PTW_HUAIRPROP(P,T,W), REAL*8 P,T,W
```

Input Values:

p - Total pressure p in psi
 t - Temperature t in °F
 W - Humidity ratio W in lb_w/lb_a

Result:

v_ptW_HAP_IP - Air-specific volume in ft³/lb_a

Range of Validity:

Temperature t : from -225.67°F to 662°F
 Total pressure p : from 0.00145 psi to 1450.4 psi
 Humidity ratio W : $0 \leq W \leq 10$ lb_w/lb_a

Comments:

- When calculating supersaturated air an ideal mixing model (saturated humid air and water droplets (or ice crystals) homogeneously mixed) is applied

Result for Wrong Input Values:

v_ptW_HAP_IP = -1000

References:

$v(p, t, W)$ Herrmann et al. [1], [2]

Humidity Ratio from Partial Pressure of Water Vapor $W = f(p, t, p_{H_2O})$

Function Name:

W_ptpH2O_HAP_IP

Fortran Program:

REAL*8 FUNCTION W_PTPH2O_HUAIRPROP(P,T,PH2O), REAL*8 P,T,PH2O

Input Values:

- p - Total pressure p in psi
- t - Temperature t in °F
- p_{H_2O} - Partial pressure of water p_{H_2O} in psi

Result:

W_ptpH2O_HAP_IP - Humidity ratio from pressure, temperature and partial pressure of water vapor in lb_w/lb_a

Range of Validity:

- Total pressure p : from 0.00145 psi to 1450.4 psi
- Temperature t : from -225.67°F to 662°F
- Partial pressure p_{H_2O} : from 0.00145 psi to 1450.4 psi

Comments:

- Iteration of humidity ratio W from $p_{H_2O} = f(p, t, W)$
- Result for supersaturated humid air is W_s

Result for Wrong Input Values:

W_ptpH2O_HAP_IP = -1000

References:

$p_{H_2O}(p, t, W)$ Herrmann et al. [1], [2]

Humidity Ratio from Relative Humidity $W = f(p, t, \varphi)$

Function Name:

W_ptphi_HAP_IP

Fortran Program:

```
REAL*8 FUNCTION W_PTPHI_HUAIRPROP(P,T,PHI), REAL*8 P,T,PHI
```

Input Values:

- p - Total pressure p in psi
- t - Temperature t in °F
- φ - Relative humidity (decimal ratio)

Result:

W_ptphi_HAP_IP - Humidity ratio from pressure, temperature and relative humidity in lb_w/lb_a

Range of Validity:

- Temperature t : from -225.67°F to 662°F
- Total pressure p : from 0.00145 psi to 1450.4 psi
- Relative humidity φ : $0 \leq \varphi \leq 1$

Comments:

- Iteration of humidity ratio W from $\varphi = f(p, t, W)$

Result for Wrong Input Values:

W_ptphi_HAP_IP = -1000

References:

- $\varphi(p, t, W)$ Herrmann et al. [1], [2]

Humidity Ratio from Dew-Point Temperature $W = f(p, t_d)$

Function Name:

$W_ptd_HAP_IP$

Fortran Program:

REAL*8 FUNCTION $W_PTD_HUAIRPROP(P,TD)$, REAL*8 P,TD

Input Values:

p - Total pressure p in psi
 t_d - Dew-point temperature t_d in °F

Result:

$W_ptd_HAP_IP$ - Humidity ratio from pressure and dew-point temperature
in lb_w/lb_a

Range of Validity:

Dew point temperature t_d : from -225.67°F to 662°F
Total pressure p : from 0.00145 psi to 1450.4 psi

Comments:

- Iteration of humidity ratio W from $t_d = f(p, W)$

Result for Wrong Input Values:

$W_ptd_HAP_IP = -1000$

References:

$t_d(p, W)$ Herrmann et al. [1], [2]

Humidity Ratio from Wet-Bulb Temperature $W = f(p, t, t_{wb})$

Function Name:

W_pttwb_HAP_IP

Fortran Program:

```
REAL*8 FUNCTION W_PTTWB_HUAIRPROP(P,T,TWB), REAL*8 P,T,TWB
```

Input Values:

p - Total pressure p in psi
 t - Temperature t in °F
 t_{wb} - Wet-bulb temperature in °F

Result:

W_pttwb_HAP_IP - Humidity ratio from pressure, temperature and wet-bulb temperature in lb_w/lb_a

Range of Validity:

Total pressure p : from 0.00145 psi to 1450.4 psi
 Temperature t : from -225.67°F to 662°F
 Wet-bulb temperature t_{wb} : from -225.67°F to 662°F

Comments:

- Iteration of humidity ratio W from $t_{wb} = f(p, t, W)$
- Calculation for supersaturated humid air ($W > W_s$) is not possible

Result for Wrong Input Values:

W_pttwb_HAP_IP = -1000

References:

$t_{wb}(p, t, W)$ Herrmann et al. [1], [2]

Saturation Humidity Ratio $W_s = f(p, t)$

Function Name:

Ws_pt_HAP_IP

Fortran Program:

```
REAL*8 FUNCTION WS_PT_HUAIRPROP(P,T), REAL*8 P,T
```

Input Values:

p - Total pressure p in psi
 t - Temperature t in °F

Result:

Ws_pt_HAP_IP - Saturation humidity ratio in lb_w/lb_a

Range of Validity:

Total pressure p : from 0.00145 psi to 1450.4 psi
 Temperature t : from -225.67°F to 662°F

Comments:

- Calculation of saturation humidity ratio W_s from $W_s = \frac{M_{H_2O}}{M_a} \frac{p_{H_2O,s}}{(p - p_{H_2O,s})}$

Result for Wrong Input Values:

Ws_pt_HAP_IP = -1000

References:

$p_{H_2O,s}$ Herrmann et al. [1], [2]

Mass Fraction of Air $\xi_{\text{Air}} = f(W)$
Function Name:

XiAir_W_HAP_IP

Fortran Program:

```
REAL*8 FUNCTION XIAIR_W_HUAIRPROP(W), REAL*8 W
```

Input Values:

W - Humidity ratio W in lb_w/lb_a

Result:

XiAir_W_HAP_IP - Mass fraction of (dry) air in humid air in lb_a/lb

Range of Validity:

Humidity ratio W : $0 \leq W \leq 10 \text{lb}_w/\text{lb}_a$

Comments:

- Mass fraction of (dry) air $\xi_{\text{Air}} = 1 - \xi_{\text{H}_2\text{O}} = 1 - \frac{W}{1+W}$

Result for Wrong Input Values:

XiAir_W_HAP_IP = -1000

References:

$\xi_{\text{Air}}(W)$ Herrmann et al. [1], [2]

Mass Fraction of Water Vapor in Humid Air $\zeta_{\text{H}_2\text{O}} = f(W)$
Function Name:

XiH2O_W_HAP_IP

Fortran Program:

REAL*8 FUNCTION XiH2O_W_HUAIRPROP(W), REAL*8 W

Input Values: W - Humidity ratio W in lb_w/lb_a **Result:**XiH2O_W_HAP_IP - Mass fraction of water vapor in humid air in lb_w/lb **Range of Validity:**Humidity ratio W : $0 \leq W \leq 10 \text{ lb}_w/\text{lb}_a$ **Comments:**- Mass fraction of water $\zeta_{\text{H}_2\text{O}} = \frac{W}{1+W}$ **Result for Wrong Input Values:**

XiH2O_W_HAP_IP = -1000

References: $\zeta_{\text{H}_2\text{O}}(W)$ Herrmann et al. [1], [2]

Compression Factor $Z = f(p, t, W)$

Function Name:

`Z_ptW_HAP_IP`

Fortran Program:

```
REAL*8 FUNCTION Z_PTW_HUAIRPROP(P,T,W), REAL*8 P,T,W
```

Input Values:

p - Total pressure p in psi
 t - Temperature t in °F
 W - Humidity ratio W in lb_w/lb_a

Result:

`Z_ptW_HAP_IP` - Compression factor (decimal ratio)

Range of Validity:

Total pressure p : from 0.00145 psi to 1450.4 psi
 Temperature t : from -225.67°F to 662°F
 Humidity ratio W : $0 \leq W \leq W_s$

Comments:

- Compression factor $Z = 1 + \frac{B_m}{\bar{v}} + \frac{C_m}{\bar{v}^2}$

$$\text{with } \bar{v} = \frac{M}{\rho} = \frac{M v}{1+W}$$

and M is the molar mass of humid air

- Calculation for supersaturated humid air ($W > W_s$) is not possible

Result for Wrong Input Values:

`Z_ptW_HAP_IP` = -1000

References:

$B_m(t, W), C_m(t, W)$ Herrmann et al. [1], [2]

$\rho(p, t, W), v(p, t, W)$ Herrmann et al. [1], [2]

3.2 Functions for Steam and Water for Temperatures $t \geq 32^\circ\text{F}$

Specific Enthalpy of Liquid Water $h_{\text{liq}} = f(p, t)$

Function Name:

hliq_pt_97_IP

Fortran Program:

```
REAL*8 FUNCTION HLIQ_PT_97(P,T), REAL*8 P,T
```

Input Values:

p - Pressure p in psi
 t - Temperature t in °F

Result:

hliq_pt_97_IP - Specific enthalpy of liquid water in Btu/lb

Range of Validity:

Pressure p : from $p_s(32^\circ\text{F}) = 0.08865$ psi to 1450.4 psi
 Temperature t : from 32°F to 662°F

Comments:

- Specific enthalpy of liquid water $h_{\text{liq}} = h^{97}(p, t)$ (Region 1)

Result for Wrong Input Values:

hliq_pt_97_IP = -1000

References:

$h^{97}(p, t)$ IAPWS-IF97 [7], [8]

Specific Enthalpy of Saturated Liquid Water $h_{\text{liq,s}} = f(t)$
Function Name:

hliqs_t_97_IP

Fortran Program:

REAL*8 FUNCTION HLIQS_T_97(T), REAL*8 T

Input Values: t - Temperature t in °F**Result:**

hliqs_t_97_IP - Specific enthalpy of saturated liquid water in Btu/lb

Range of Validity:Temperature t : from 32°F to 662°F**Comments:**- Specific enthalpy of liquid water $h_{\text{liq,s}} = h^{97}(p_s, t)$ (Region 1)with $p_s = p_s^{97}(t)$ **Result for Wrong Input Values:**

hliqs_t_97_IP = -1000

References: $h^{97}(p, t), p_s^{97}(t)$ IAPWS-IF97 [7], [8]

Specific Enthalpy of Saturated Water Vapor $h_{\text{vap},s} = f(t)$

Function Name:

hvaps_t_97_IP

Fortran Program:

```
REAL*8 FUNCTION HVAPS_T_97(T), REAL*8 T
```

Input Values:

t - Temperature t in °F

Result:

hvaps_t_97_IP - Specific enthalpy of saturated water vapor in Btu/lb

Range of Validity:

Temperature t : from 32°F to 662°F

Comments:

- Specific enthalpy of saturated water vapor $h_{\text{vap},s} = h^{97}(p_s, t)$ (Region 2)

with $p_s = p_s^{97}(t)$

Result for Wrong Input Values:

hvaps_t_97_IP = -1000

References:

$h^{97}(p, t), p_s^{97}(t)$ IAPWS-IF97 [7], [8]

Saturation Pressure of Water $p_s = f(t)$

Function Name:

ps_t_97_IP

Fortran Program:

```
REAL*8 FUNCTION PS_T_97(T), REAL*8 T
```

Input Values:

t - Temperature t in °F

Result:

ps_t_97_IP - Saturation pressure of water in psi

Range of Validity:

Temperature t : from 32°F to 662°F

Comments:

- Saturation pressure of water $p_s = p_s^{97}(t)$ (Region 4)

Result for Wrong Input Values:

ps_t_97_IP -1000

References:

$p_s^{97}(t)$ IAPWS-IF97 [7], [8]

Specific Entropy of Liquid Water $s_{\text{liq}} = f(p, t)$

Function Name:

sliq_pt_97_IP

Fortran Program:

```
REAL*8 FUNCTION SLIQ_PT_97(P,T), REAL*8 P,T
```

Input Values:

p - Pressure p in psi
 t - Temperature t in °F

Result:

sliq_pt_97_IP - Specific entropy of liquid water in Btu/(lb °R)

Range of Validity:

Pressure p : from $p_s(32^\circ\text{F}) = 0.08865$ psi to 1450.4 psi
 Temperature t : from 32°F to 662°F

Comments:

- Specific entropy of liquid water $s_{\text{liq}} = s^{97}(p, t)$ (Region 1)

Result for Wrong Input Values:

sliq_pt_97_IP = -1000

References:

$s^{97}(p, t)$ IAPWS-IF97 [7], [8]

Specific Entropy of Saturated Liquid Water $s_{\text{liq},s} = f(t)$
Function Name:

sliqs_t_97_IP

Fortran Program:

REAL*8 FUNCTION SLIQS_T_97(T), REAL*8 T

Input Values: t - Temperature t in °F**Result:**

sliqs_t_97_IP - Specific entropy of saturated liquid water in Btu/(lb °R)

Range of Validity:Temperature t : from 32°F to 662°F**Comments:**- Specific entropy of liquid water $s_{\text{liq},s} = s^{97}(p_s, t)$ (Region 1)with $p_s = p_s^{97}(t)$ **Result for Wrong Input Values:**

sliqs_t_97_IP = -1000

References: $s^{97}(p, t), p_s^{97}(t)$ IAPWS-IF97 [7], [8]

Specific Entropy of Saturated Water Vapor $s_{\text{vap},s} = f(t)$

Function Name:

svaps_t_97_IP

Fortran Program:

```
REAL*8 FUNCTION SVAPS_T_97(T), REAL*8 T
```

Input Values:

t - Temperature t in °F

Result:

svaps_t_97_IP - Specific entropy of saturated water vapor in Btu/(lb °R)

Range of Validity:

Temperature t : from 32°F to 662°F

Comments:

- Specific entropy of saturated water vapor $s_{\text{vap},s} = s^{97}(p_s, t)$ (Region 2)

with $p_s = p_s^{97}(t)$

Result for Wrong Input Values:

svaps_t_97_IP = -1000

References:

$s^{97}(p, t), p_s^{97}(t)$ IAPWS-IF97 [7], [8]

Saturation Temperature of Water $t_s = f(p)$

Function Name:

ts_p_97_IP

Fortran Program:

```
REAL*8 FUNCTION TS_P_97(P), REAL*8 P
```

Input Values:

p - Pressure p in psi

Result:

ts_p_97_IP - Saturation temperature of water in °F

Range of Validity:

Pressure p : from 0.08865 psi to 1450.4 psi

Comments:

- Saturation temperature of water $t_s = t_s^{97}(p)$ (Region 4)

Result for Wrong Input Values:

ts_p_97_IP = -1000

References:

$t_s^{97}(p)$ IAPWS-IF97 [7], [8]

Specific Volume of Liquid Water $v_{\text{liq}} = f(p, t)$

Function Name:

vliq_pt_97_IP

Fortran Program:

```
REAL*8 FUNCTION VLIQ_PT_97(P,T), REAL*8 P,T
```

Input Values:

p - Pressure p in psi
 t - Temperature t in °F

Result:

vliq_pt_97_IP - Specific volume of liquid water in ft³/lb

Range of Validity:

Pressure p : from $p_s(32^\circ\text{F}) = 0.08865$ psi to 1450.4 psi
 Temperature t : from 32°F to 662°F

Comments:

- Specific volume of liquid water $v_{\text{liq}} = v^{97}(p, t)$ (Region 1)

Result for Wrong Input Values:

vliq_pt_97_IP = -1000

References:

$v^{97}(p, t)$ IAPWS-IF97 [7], [8]

Specific Volume of Saturated Liquid Water $v_{\text{liq,s}} = f(t)$
Function Name:

vliqs_t_97_IP

Fortran Program:

REAL*8 FUNCTION VLIQS_T_97(T), REAL*8 T

Input Values: t - Temperature t in °F**Result:**vliqs_t_97_IP - Specific volume of saturated liquid water in ft³/lb**Range of Validity:**Temperature t : from 32°F to 662°F**Comments:**- Specific volume of liquid water $v_{\text{liq,s}} = v^{97}(\rho_s, t)$ (Region 1)with $\rho_s = \rho_s^{97}(t)$ **Result for Wrong Input Values:**

vliqs_t_97_IP = -1000

References: $v^{97}(\rho, t), \rho_s^{97}(t)$ IAPWS-IF97 [7], [8]

Specific Volume of Saturated Water Vapor $v_{\text{vap},s} = f(t)$
Function Name:

vvaps_t_97_IP

Fortran Program:

```
REAL*8 FUNCTION  VVAPS_T_97(T), REAL*8 T
```

Input Values:

t - Temperature t in °F

Result:

vvaps_t_97_IP - Specific volume of saturated water vapor in ft³/lb

Range of Validity:

Temperature t : from 32°F to 662°F

Comments:

- Specific volume of saturated water vapor $v_{\text{vap},s} = v^{97}(p_s, t)$ (Region 2)

with $p_s = p_s^{97}(t)$

Result for Wrong Input Values:

vvaps_t_97_IP = -1000

References:

$v^{97}(p, t), p_s^{97}(t)$ IAPWS-IF97 [7], [8]

3.3 Functions for Steam and Ice for Temperatures $t \leq 32^\circ\text{F}$

Specific Enthalpy of Saturated Ice $h_{\text{ice,sub}} = f(t)$
Function Name:

hicesub_t_06_IP

Fortran Program:

```
REAL*8 FUNCTION HICESUB_T_06(T), REAL*8 T
```

Input Values:

t - Temperature t in °F

Result:

hicesub_t_06_IP - Specific enthalpy of saturated ice in Btu/lb

Range of Validity:

Temperature t : from -225.67°F to 32°F

Comments:

- Specific enthalpy of saturated ice $h_{\text{ice,sub}} = h^{06}(\rho_{\text{sub}}, t)$

with $\rho_{\text{sub}} = \rho_{\text{sub}}^{08}(t)$

Result for Wrong Input Values:

hicesub_t_06_IP = -1000

References:

$h^{06}(\rho, t)$ IAPWS-06 [10]

$\rho_{\text{sub}}^{08}(t)$ IAPWS-08 [11]

Specific Enthalpy of Saturated Water Vapor $h_{\text{vap,sub}} = f(t)$
Function Name:

hvapsub_t_95_IP

Fortran Program:

```
REAL*8 FUNCTION HVAPSUB_T_95(T), REAL*8 T
```

Input Values:

t - Temperature t in °F

Result:

hvapsub_t_95_IP - Specific enthalpy of saturated water vapor in Btu/lb

Range of Validity:

Temperature t : from -225.67°F to 32°F

Comments:

- Specific enthalpy of saturated water vapor $h_{\text{vap,sub}} = h^{95}(p_{\text{sub}}, t)$

with $p_{\text{sub}} = p_{\text{sub}}^{08}(t)$

Result for Wrong Input Values:

hvapsub_t_95_IP = -1000

References:

$h^{95}(p, t)$ IAPWS-95 [5], [6]

$p_{\text{sub}}^{08}(t)$ IAPWS-08 [11]

Melting Pressure of Ice $p_{\text{mel}} = f(t)$ **Function Name:**

pmel_t_08_IP

Fortran Program:

REAL*8 FUNCTION PMEL_T_08 (T), REAL*8 T

Input Values: t - Temperature t in °F**Result:**

pmel_t_08_IP - Melting pressure of ice in psi

Range of Validity:Temperature t : from -7.573°F to 32°F**Result for Wrong Input Values:**

pmel_t_08_IP = -1000

References: $p_{\text{mel}}^{08}(t)$ IAPWS-08 [11]

Sublimation Pressure of Ice $p_{\text{sub}} = f(t)$

Function Name:

psub_t_08_IP

Fortran Program:

```
REAL*8 FUNCTION PSUB_T_08 (T), REAL*8 T
```

Input Values:

t - Temperature t in °F

Result:

psub_t_08_IP - Sublimation pressure of ice in psi

Range of Validity:

Temperature t : from -225.67°F to 32°F

Result for Wrong Input Values:

psub_t_08_IP = -1000

References:

$p_{\text{sub}}^{08}(t)$ IAPWS-08 [11]

Specific Entropy of Saturated Ice $s_{\text{ice,sub}} = f(t)$

Function Name:

sicesub_t_06_IP

Fortran Program:

```
REAL*8 FUNCTION SICESUB_T_06(T), REAL*8 T
```

Input Values:

t - Temperature t in °F

Result:

sicesub_t_06_IP - Specific entropy of saturated ice in Btu/(lb °R)

Range of Validity:

Temperature t : from -225.67°F to 32°F

Comments:

- Specific entropy of saturated ice $s_{\text{ice,sub}} = s^{06}(p_{\text{sub}}, t)$
 with $p_{\text{sub}} = p_{\text{sub}}^{08}(t)$

Result for Wrong Input Values:

sicesub_t_06_IP = -1000

References:

$s^{06}(p, t)$ IAPWS-06 [10]

$p_{\text{sub}}^{08}(t)$ IAPWS-08 [11]

Specific Entropy of Saturated Water Vapor $s_{\text{vap,sub}} = f(t)$
Function Name:

svapsub_t_95_IP

Fortran Program:

REAL*8 FUNCTION SVAPSUB_T_95(T), REAL*8 T

Input Values: t - Temperature t in °F**Result:**

svapsub_t_95_IP - Specific entropy of saturated water vapor in Btu/(lb °R)

Range of Validity:Temperature t : from -225.67°F to 32°F**Comments:**- Specific entropy of saturated water vapor $s_{\text{vap,sub}} = s^{95}(p_{\text{sub}}, t)$ with $p_{\text{sub}} = p_{\text{sub}}^{08}(t)$ **Result for Wrong Input Values:**

svapsub_t_95_IP = -1000

References: $s^{95}(p, t)$ IAPWS-95 [7], [8] $p_{\text{sub}}^{08}(t)$ IAPWS-08 [11]

Melting Temperature of Ice $t_{\text{mel}} = f(p)$

Function Name:

tmel_p_08_IP

Fortran Program:

```
REAL*8 FUNCTION TMEL_P_08(P), REAL*8 P
```

Input Values:

p - Pressure p in psi

Result:

tmel_p_08_IP - Melting temperature of ice in °F

Range of Validity:

Pressure p : from p_s (32°F) = 0.08865 psi to 1450.4 psi

Result for Wrong Input Values:

tmel_p_08_IP = -1000

References:

$t_{\text{mel}}^{08}(p)$ IAPWS-08 [11]

Sublimation Temperature of Ice $t_{\text{sub}} = f(p)$

Function Name:

tsub_p_08_IP

Fortran Program:

```
REAL*8 FUNCTION TSUB_P_08(P), REAL*8 P
```

Input Values:

p - Pressure p in psi

Result:

tsub_p_08_IP - Sublimation temperature of ice in °F

Range of Validity:

Pressure p : from $p_{\text{subl}}(-225.67^\circ\text{F}) = 1.7407 \times 10^{-12}$ psi to $p_{\text{subl}}(32^\circ\text{F}) = 0.08865$ psi

Result for Wrong Input Values:

tsub_p_08_IP = -1000

References:

$t_{\text{sub}}^{08}(p)$ IAPWS-08 [11]

Specific Volume of Saturated Ice $v_{\text{ice,sub}} = f(t)$
Function Name:

vicesub_t_06_IP

Fortran Program:

REAL*8 FUNCTION VICESUB_T_06(T), REAL*8 T

Input Values: t - Temperature t in °F**Result:**vicesub_t_06_IP - Specific volume of saturated ice in ft³/lb**Range of Validity:**Temperature t : from -225.67°F to 32°F**Comments:**- Specific volume of saturated ice $v_{\text{ice,sub}} = v^{06}(\rho_{\text{sub}}, t)$ with $\rho_{\text{sub}} = \rho_{\text{sub}}^{08}(t)$ **Result for Wrong Input Values:**

vicesub_t_06_IP = -1000

References: $v^{06}(\rho, t)$ IAPWS-06 [10] $\rho_{\text{sub}}^{08}(t)$ IAPWS-08 [11]

Specific Volume of Saturated Water Vapor $v_{\text{vap,sub}} = f(t)$
Function Name:

vvapsub_t_95_IP

Fortran Program:

```
REAL*8 FUNCTION  VVAPSUB_T_95(T), REAL*8 T
```

Input Values:

t - Temperature t in °F

Result:

vvapsub_t_95_IP - Specific volume of saturated water vapor in ft³/lb

Range of Validity:

Temperature t : from -225.67°F to 32°F

Comments:

- Specific volume of saturated water vapor $v_{\text{vap,sub}} = v^{95}(\rho_{\text{sub}}, t)$

with $\rho_{\text{sub}} = \rho_{\text{sub}}^{08}(t)$

Result for Wrong Input Values:

vvapsub_t_95_IP = -1000

References:

$v^{95}(\rho, t)$ IAPWS-95 [7], [8]

$\rho_{\text{sub}}^{08}(t)$ IAPWS-08 [11]

4. Property Libraries for Calculating Heat Cycles, Boilers, Turbines and Refrigerators

Water and Steam

Library LibIF97

- Industrial Formulation IAPWS-IF97 (Revision 2007)
- Supplementary Standards IAPWS-IF97-S01, -S03rev, -S04, and -S05
- IAPWS Revised Advisory Note No. 3 on Thermodynamic Derivatives (2008)

Library LibIF97_META

- Industrial Formulation IAPWS-IF97 (Revision 2007) for metastable steam

Humid Combustion Gas Mixtures

Library LibHuGas

- Model: Ideal mixture of the real fluids:
 CO₂ - Span, Wagner H₂O - IAPWS-95
 O₂ - Schmidt, Wagner N₂ - Span et al.
 Ar - Tegeler et al.
 and of the ideal gases:
 SO₂, CO, Ne
 (Scientific Formulation of Bücker et al.)
- Consideration of:
- Dissociation from VDI 4670
 - Poynting effect

Humid Air

Library LibHuAir

- Model: Ideal mixture of the real fluids:
- Dry air from Lemmon et al.
 - Steam, water and ice from IAPWS-IF97 and IAPWS-06
- Consideration of:
- Condensation and freezing of steam
 - Dissociation from VDI 4670
 - Poynting effect from ASHRAE RP-1485

Extremely Fast Property Calculations

- Spline-Based Table
 Look-up Method (SBTL)
- Library LibSBTL_IF97**
Library LibSBTL_95
Library LibSBTL_HuAir
- For steam, water, humid air, carbon dioxide and other fluids and mixtures according IAPWS Guideline 2015 for Computational Fluid Dynamics (CFD), real-time and non-stationary simulations

Carbon Dioxide Including Dry Ice

Library LibCO2

Formulation of Span and Wagner (1996)

Seawater

Library LibSeaWa

IAPWS Industrial Formulation 2013

Ice

Library LibICE

Ice from IAPWS-06, Melting and sublimation pressures from IAPWS-08, Water from IAPWS-IF97, Steam from IAPWS-95 and -IF97

Ideal Gas Mixtures

Library LibIdGasMix

Model: Ideal mixture of the ideal gases:

Ar	NO	He	Propylene
Ne	H ₂ O	F ₂	Propane
N ₂	SO ₂	NH ₃	Iso-Butane
O ₂	H ₂	Methane	n-Butane
CO	H ₂ S	Ethane	Benzene
CO ₂	OH	Ethylene	Methanol
Air			

Consideration of:

- Dissociation from the VDI Guideline 4670

Library LibIDGAS

Model: Ideal gas mixture from VDI Guideline 4670

Consideration of:

- Dissociation from the VDI Guideline 4670

Humid Air

Library ASHRAE LibHuAirProp

Model: Virial equation from ASHRAE Report RP-1485 for real mixture of the real fluids:
 - Dry air
 - Steam

Consideration of:

- Enhancement of the partial saturation pressure of water vapor at elevated total pressures

www.ashrae.org/bookstore

Dry Air Including Liquid Air

Library LibRealAir

Formulation of Lemmon et al. (2000)

Refrigerants

Ammonia

Library LibNH3

Formulation of Tillner-Roth et al. (1993)

R134a

Library LibR134a

Formulation of Tillner-Roth and Baehr (1994)

Iso-Butane

Library LibButane_Iso

Formulation of Bücker and Wagner (2006)

n-Butane

Library LibButane_n

Formulation of Bücker and Wagner (2006)

Mixtures for Absorption Processes

Ammonia/Water Mixtures

Library LibAmWa

IAPWS Guideline 2001 of Tillner-Roth and Friend (1998)

Helmholtz energy equation for the mixing term (also useable for calculating the Kalina Cycle)

Water/Lithium Bromide Mixtures

Library LibWaLi

Formulation of Kim and Infante Ferreira (2004)

Gibbs energy equation for the mixing term

Liquid Coolants

Liquid Secondary Refrigerants

Library LibSecRef

Liquid solutions of water with

C ₂ H ₆ O ₂	Ethylene glycol
C ₃ H ₈ O ₂	Propylene glycol
C ₂ H ₅ OH	Ethanol
CH ₃ OH	Methanol
C ₃ H ₈ O ₃	Glycerol
K ₂ CO ₃	Potassium carbonate
CaCl ₂	Calcium chloride
MgCl ₂	Magnesium chloride
NaCl	Sodium chloride
C ₂ H ₃ KO ₂	Potassium acetate
CHKO ₂	Potassium formate
LiCl	Lithium chloride
NH ₃	Ammonia

Formulation of the International Institute of Refrigeration (IIR 2010)

Ethanol**Library LibC2H5OH**

Formulation of
Schroeder et al. (2014)

Methanol**Library LibCH3OH**

Formulation of
de Reuck and Craven (1993)

Propane**Library LibPropane**

Formulation of
Lemmon et al. (2009)

Siloxanes as ORC Working Fluids

Octamethylcyclotetrasiloxane $C_8H_{24}O_4Si_4$ **Library LibD4**

Decamethylcyclopentasiloxane $C_{10}H_{30}O_5Si_5$ **Library LibD5**

Tetradecamethylhexasiloxane $C_{14}H_{42}O_6Si_6$ **Library LibMD4M**

Hexamethyldisiloxane $C_6H_{18}OSi_2$ **Library LibMM**

Formulation of Colonna et al. (2006)

Dodecamethylcyclohexasiloxane $C_{12}H_{36}O_6Si_6$ **Library LibD6**

Decamethyltetrasiloxane $C_{10}H_{30}O_3Si_4$ **Library LibMD2M**

Dodecamethylpentasiloxane $C_{12}H_{36}O_4Si_5$ **Library LibMD3M**

Octamethyltrisiloxane $C_8H_{24}O_2Si_3$ **Library LibMDM**

Formulation of Colonna et al. (2008)

Nitrogen and Oxygen**Libraries
LibN2 and LibO2**

Formulations of Span et al. (2000)
and Schmidt and Wagner (1985)

Hydrogen**Library LibH2**

Formulation of
Leachman et al. (2009)

Helium**Library LibHe**

Formulation of
Arp et al. (1998)

Hydrocarbons

Decane $C_{10}H_{22}$ **Library LibC10H22**

Isopentane C_5H_{12} **Library LibC5H12_Iso**

Neopentane C_5H_{12} **Library LibC5H12_Neo**

Isohexane C_6H_{14} **Library LibC6H14**

Toluene C_7H_8 **Library LibC7H8**

Formulation of Lemmon and Span (2006)

Further Fluids

Carbon monoxide **CO** **Library LibCO**

Carbonyl sulfide **COS** **Library LibCOS**

Hydrogen sulfide **H₂S** **Library LibH2S**

Nitrous oxide **N₂O** **Library LibN2O**

Sulfur dioxide **SO₂** **Library LibSO2**

Acetone C_3H_6O **Library LibC3H6O**

Formulation of Lemmon and Span (2006)

**For more information please contact:**

KCE-ThermoFluidProperties UG
Prof. Dr. Hans-Joachim Kretzschmar
Haager Weg 6
92224 Amberg, Germany

Internet: www.thermofluidprop.com
Email: info@thermofluidprop.com
Phone: +49-9621-1762047
Mobile: +49-172-7914607
Fax: +49-3222-1095810

The following thermodynamic and transport properties can be calculated^a:**Thermodynamic Properties**

- Vapor pressure p_s
- Saturation temperature T_s
- Density ρ
- Specific volume v
- Enthalpy h
- Internal energy u
- Entropy s
- Exergy e
- Isobaric heat capacity c_p
- Isochoric heat capacity c_v
- Isentropic exponent κ
- Speed of sound w
- Surface tension σ

Transport Properties

- Dynamic viscosity η
- Kinematic viscosity ν
- Thermal conductivity λ
- Prandtl number Pr
- Thermal diffusivity a

Backward Functions

- $T, v, s(p, h)$
- $T, v, h(p, s)$
- $p, T, v(h, s)$
- $p, T(v, h)$
- $p, T(v, u)$

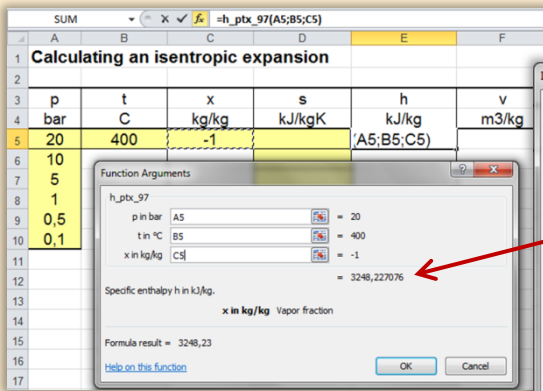
Thermodynamic Derivatives

- Partial derivatives used in process modeling can be calculated.

^a Not all of these property functions are available in all property libraries.

Property Software for Calculating Heat Cycles, Boilers, Turbines and Refrigerators

Add-In FluidEXL^{Graphics} for Excel[®]



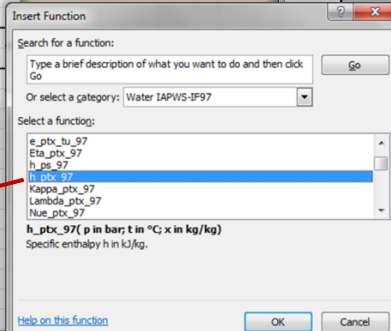
Calculating an isentropic expansion

bar	C	kg/kg	kJ/kgK	h	v
20	400	-1		A5;B5;C5	
10					
5					
1					
0,5					
0,1					

Function Arguments dialog for h_ptx_97:

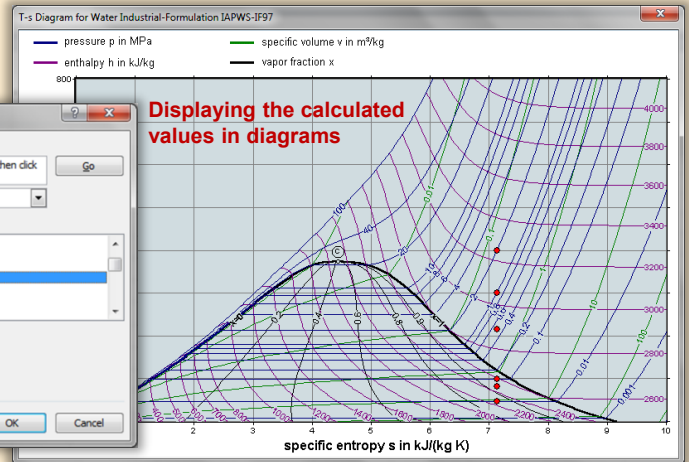
- p in bar: A5 = 20
- t in °C: B5 = 400
- x in kg/kg: C5 = -1
- Specific enthalpy h in kJ/kg: = 3248,227076
- x in kg/kg Vapor fraction: = 3248,23

Choosing a property library and a function



Insert Function dialog box showing the selection of the h_ptx_97 function from the Water IAPWS-IF97 category.

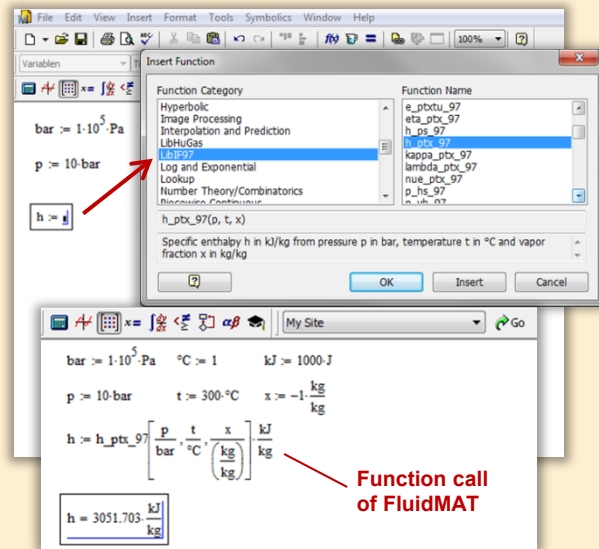
Displaying the calculated values in diagrams



Menu for the input of given property values

Add-On FluidMAT for Mathcad[®]
Add-On FluidPRIME for Mathcad Prime[®]

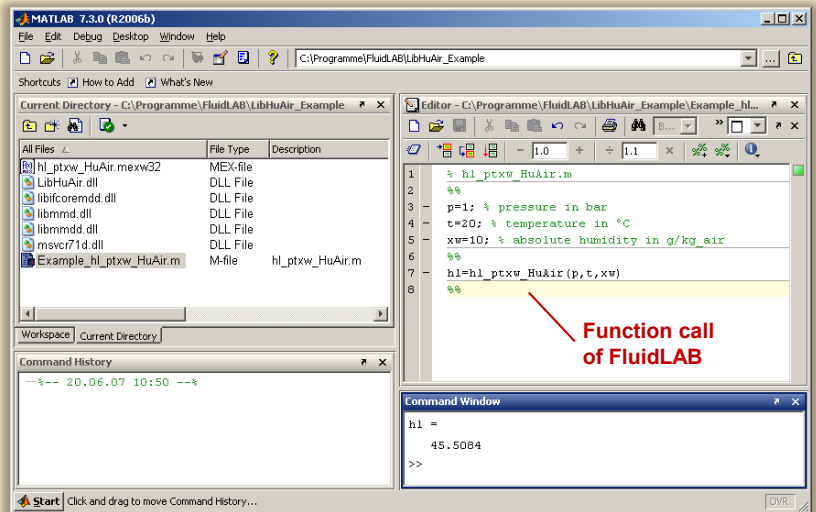
The property libraries can be used in Mathcad[®] and Mathcad Prime[®].



Mathcad interface showing the FluidMAT function call: $h = h_{ptx_97} \left[\frac{p}{\text{bar}}, \frac{t}{^\circ\text{C}}, \frac{x}{\left(\frac{\text{kg}}{\text{kg}}\right)} \right] \frac{\text{kJ}}{\text{kg}}$. The result is $h = 3051.703 \frac{\text{kJ}}{\text{kg}}$.

Add-On FluidLAB for MATLAB[®] and SIMULINK[®]

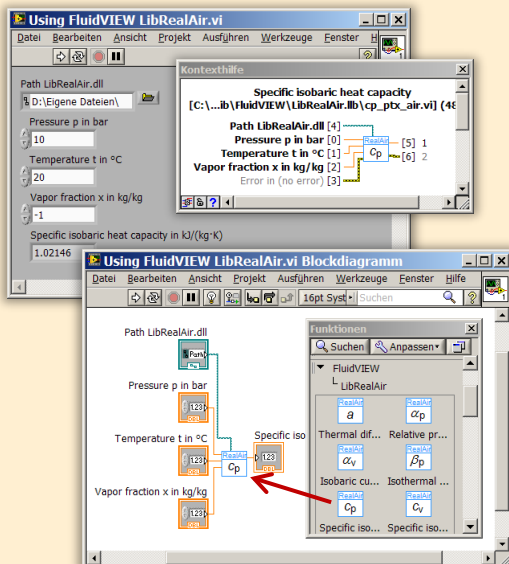
Using the Add-In FluidLAB the property functions can be called in MATLAB[®] and SIMULINK[®].



MATLAB 7.3.0 (R2006b) interface showing the FluidLAB function call in a script: `h1 = hl_ptxw_HuAir(m, p, t, xw)`. The Command Window shows the result: `h1 = 45.5084`.

Add-On FluidVIEW for LabVIEW[™]

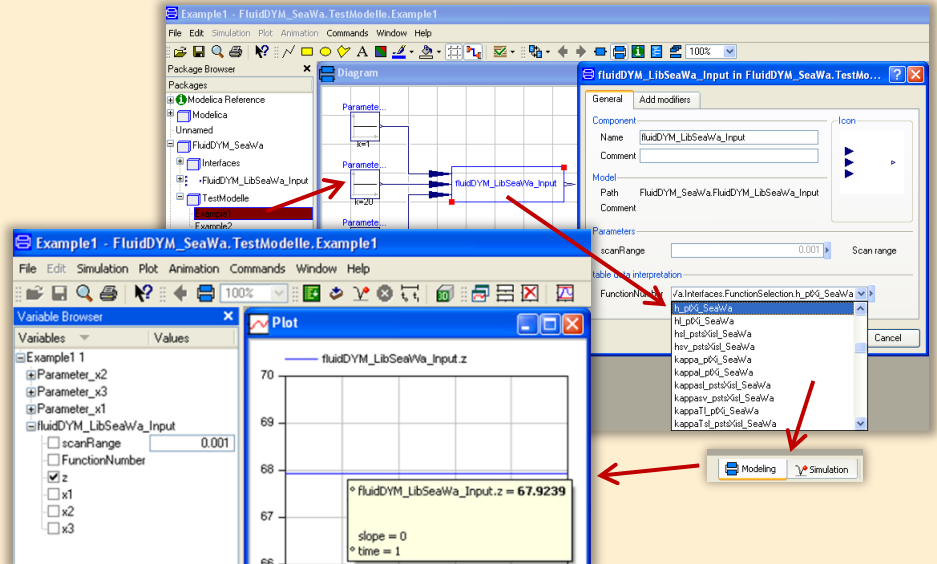
The property functions can be calculated in LabVIEW[™].



LabVIEW interface showing the FluidVIEW function block. Inputs include Pressure p in bar (10), Temperature t in °C (20), and Vapor fraction x in kg/kg (-1). The output is Specific isobaric heat capacity Cp in kJ/(kg K) (1.02146).

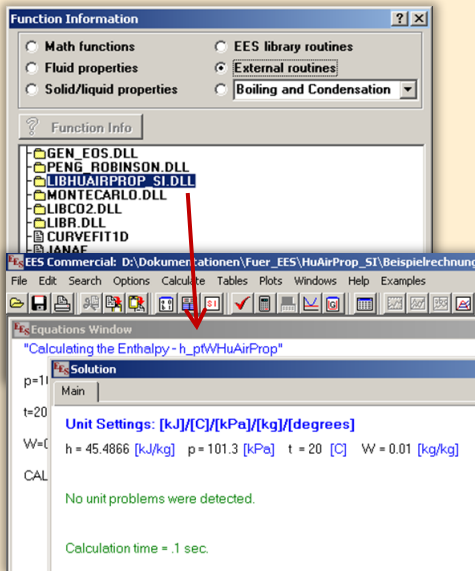
Add-On FluidDYM for DYMOLA[®] (Modelica) and SimulationX[®]

The property functions can be called in DYMOLA[®] and SimulationX[®].

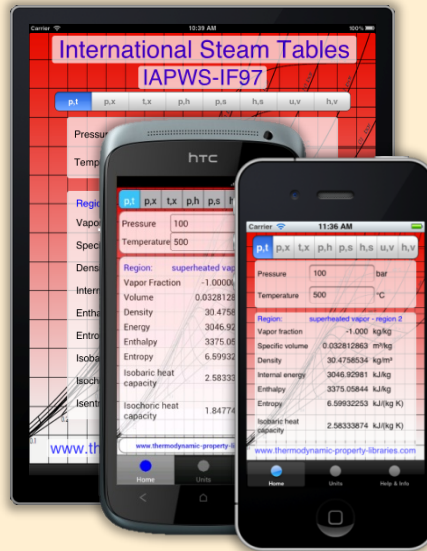


SimulationX interface showing the FluidDYM function block. The output plot shows the value of fluidDYM_LibSeaWa_Input.z as 67.9239.

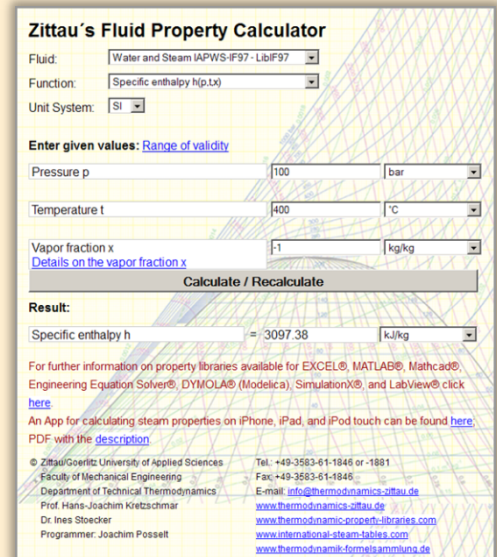
Add-On FluidEES for Engineering Equation Solver®



App International Steam Tables for iPhone, iPad, iPod touch, Android Smartphones and Tablets



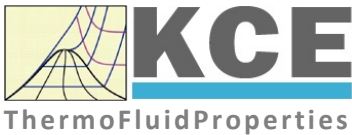
Online Property Calculator at www.thermofluidprop.com



Property Software for Pocket Calculators

<p>FluidCasio</p> <p>fx 9750 G II CFX 9850 fx-GG20 CFX 9860 G Graph 85 ALGEBRA FX 2.0</p>				<p>FluidHP</p> <p>HP 48 HP 49</p>		<p>FluidTI</p> <p>TI Nspire CX CAS TI 83 TI Nspire CAS TI 84 TI 89 TI Voyage 200</p>	
---	--	--	--	---	--	--	--

For more information please contact:



KCE-ThermoFluidProperties UG
 Prof. Dr. Hans-Joachim Kretzschmar
 Haager Weg 6
 92224 Amberg, Germany

Internet: www.thermofluidprop.com
 Email: info@thermofluidprop.com
 Phone: +49-9621-1762047
 Mobile: +49-172-7914607
 Fax: +49-3222-1095810

The following thermodynamic and transport properties^a can be calculated in Excel®, MATLAB®, Mathcad®, Engineering Equation Solver® (EES), DYMOLA® (Modelica), SimulationX® and LabVIEW™:

Thermodynamic Properties

- Vapor pressure p_s
- Saturation temperature T_s
- Density ρ
- Specific volume v
- Enthalpy h
- Internal energy u
- Entropy s
- Exergy e
- Isobaric heat capacity c_p
- Isochoric heat capacity c_v
- Isentropic exponent κ
- Speed of sound w
- Surface tension σ

Transport Properties

- Dynamic viscosity η
- Kinematic viscosity ν
- Thermal conductivity λ
- Prandtl number Pr
- Thermal diffusivity a

Backward Functions

- $T, v, s(p, h)$
- $T, v, h(p, s)$
- $p, T, v(h, s)$
- $p, T(v, h)$
- $p, T(v, u)$

Thermodynamic Derivatives

- Partial derivatives used in process modeling can be calculated.

^a Not all of these property functions are available in all property libraries.

5 References

- [1] Herrmann, S.; Kretzschmar, H.-J.; Gatley, D.P.: Thermodynamic Properties of Real Moist Air, Dry Air, Steam, Water, and Ice. HVAC&R Research 5, 961-986 (2009).
- [2] Herrmann, S.; Kretzschmar, H.-J.; Gatley, D.P.: Thermodynamic Properties of Real Moist Air, Dry Air, Steam, Water, and Ice. ASHRAE RP-1485, American Society of Heating, Refrigerating, and Air-Conditioning Engineers, Inc., Atlanta, GA (2009).
- [3] Herrmann, S.; Kretzschmar, H.-J.; Aute, V.C.; Gatley, D.P.; Vogel, E.: Transport Properties of Real Moist Air, Dry Air, Steam, and Water. ASHRAE RP-1767, American Society of Heating, Refrigerating, and Air-Conditioning Engineers, Inc., Atlanta, GA (2018).
- [4] Herrmann, S.; Kretzschmar, H.-J.; Aute, V.C.; Gatley, D.P.; Vogel, E.: Transport Properties of Real Moist Air, Dry Air, Steam, and Water. Science and Technology for the Built Environment 27, 393-401 (2021).
- [5] IAPWS. R6-95, Revised Release on the IAPWS Formulation 1995 for the Thermodynamic Properties of Ordinary Water Substance for General and Scientific Use. (2009), available from www.iapws.org.
- [6] Wagner, W.; Pruß, A.: The IAPWS Formulation 1995 for the Thermodynamic Properties of Ordinary Water Substance for General and Scientific Use. J. Phys. Chem. Ref. Data 31, 387-535 (2002).
- [7] IAPWS. R7-97, Revised Release on the IAPWS Industrial Formulation 1997 for the Thermodynamic Properties of Water and Steam IAPWS-IF97. (2007), available from www.iapws.org.
- [8] Wagner, W.; Kretzschmar, H.-J.: International Steam Tables. Springer, Berlin (2008).
- [9] Parry, W.T.; Bellows, J.C.; Gallagher, J.S.; Harvey, A.H.: ASME International Steam Tables for Industrial Use. ASME Press, New York (2000).
- [10] IAPWS. R10-06, Revised Release on the Equation of State 2006 for H₂O Ice Ih. (2009); available from www.iapws.org.
- [11] IAPWS. R14-08, Revised Release on the Pressure along the Melting and Sublimation Curves of Ordinary Water Substance. (2008); available from www.iapws.org.
- [12] Nelson, H.F.; Sauer, H.J.: Formulation of High-Temperature Properties for Moist Air. HVAC&R Research 8, 311-334 (2002).
- [13] Gatley, D.P.: Understanding Psychrometrics, 2nd ed., ASHRAE, Atlanta (2005).
- [14] Lemmon, E.W.; Jacobsen, R.T.; Penoncello, S.G.; Friend, D.G.: Thermodynamic Properties of Air and Mixture of Nitrogen, Argon, and Oxygen from 60 to 2000 K at Pressures to 2000 MPa. J. Phys. Chem. Ref. Data 29, 331-385 (2000).
- [15] Harvey, A.H.; Huang, P.H.: First-Principles Calculation of the Air-Water Second Virial Coefficient. Int. J. Thermophys. 28, 556-565 (2007).
- [16] IAPWS. G7-04, Guideline on the Henry's Constant and Vapor-Liquid Distribution Constant for Gases in H₂O and D₂O at High Temperatures. (2004), available from www.iapws.org.

- [17] Gatley, D.P.; Herrmann, S.; Kretzschmar, H.-J.: A Twenty-First Century Molar Mass for Dry Air. HVAC&R Research 14, 655-662 (2008).
- [18] Lemmon, E.W.; Jacobsen, R.T.: Viscosity and Thermal Conductivity Equations for Nitrogen, Oxygen, Argon, and Air. Int. J. Thermophys. 25, 21-69 (2004).
- [19] IAPWS. R12-08, Release on the IAPWS Formulation 2008 for the Viscosity of Ordinary Water Substance. (2008), available from www.iapws.org.
- [20] IAPWS. R15-11, Revised Release on the IAPWS Formulation 1985 for the Thermal Conductivity of Ordinary Water Substance. (2008), available from www.iapws.org.
- [21] Hyland, R.W.; Wexler, A.: Formulations for the Thermodynamic Properties of Dry Air from 173.15 K to 473.15 K, and of Saturated Moist Air from 173.15 K to 372.15 K, at Pressures to 5 MPa. ASHRAE Trans. 89, 520-535 (1983).
- [22] Mohr, P.J.; Taylor, P.N.: CODATA Recommended Values of the Fundamental Physical Constants: 2002. Rev. Mod. Phys. 77, 1-107 (2005).
- [23] ASHRAE. 2009 Handbook of Fundamentals. Chapter 1 - Psychrometrics. American Society of Heating, Refrigerating, and Air-Conditioning Engineers, Inc., Atlanta, GA (2009).
- [24] Feistel, R.; Lovell-Smith, J.W.; Hellmuth, O.: Virial Approximation of the TEOS-10 Equation for the Fugacity of Water in Humid Air. Int. J. Thermophys. 36, 44-68 (2015).

6 Satisfied Customers

Period from 2018 to 2022

The following companies and institutions use the property libraries:

- FluidEXL *Graphics* for Excel® incl. VBA
- FluidLAB for MATLAB® and Simulink
- FluidMAT for Mathcad®
- FluidPRIME for Mathcad Prime®
- FluidEES for Engineering Equation Solver® EES
- FluidDYM for Dymola® (Modelica) and SimulationX®
- FluidVIEW for LabVIEW™
- FluidPYT for Python
- FluidJAVA for Java
- DLLs for Windows Applications
- Shared Objects for Linux
- Shared Objects for macOS.

2022

ASTG, Graz, Austria	12/2022
Wandschneider + Gutjahr, Hamburg	
RWE Supply & Trading, Essen	11/2022
Stadtwerke Rosenheim	
CEA, Saclay, France	10/2022
RWE Supply & Trading, Essen	
SEEC Saudi Energy Efficiency Center, Riyadh, Saudi Arabia	
MAN, Copenhagen, Denmark	
Hermeler & Partner Consulting Engineers, Sassenberg	09/2022
Envi Con, Nürnberg	
Drill Cool Systems, Bakersfield CA, USA	
RWE Supply & Trading, Essen	
Maerz Ofenbau, Zürich, Switzerland	
Saale Energie, Schkopau	
ERGO, Dresden	
Mainova, Frankfurt/Main	
Bundeswehr, Koblenz	08/2022
RWE Supply & Trading, Essen	
Grenzbach Corporation, Newnan GE, USA	
AGRANA, Gmuend, Austria	07/2022
MIBRAG, Zeitz	
Hochschule Niederrhein, Krefeld	
ULT, Löbau	06/2022
LEAG, Cottbus	

VPC Group, Vetschau	
Wärme, Hamburg	
ILK, Dresden	
Stricker IB, Küsnacht a. Rigi, Switzerland	
LEAG, Cottbus	05/2022
RWE Supply & Trading, Essen	
IGT Tomalla, Kreuztal	
B+T Engineering, Dübendorf, Switzerland	
Stricker IB, Küsnacht a. Rigi, Switzerland	
Vogelsang & Benning, Bochum	04/2022
Frischli, Rehburg-Loccum	
BPS Consulting, Sprengel	03/2022
HS Hannover, Maschinenbau & BioVT	
M+M Turbinentechnik, Bad Salzungen	
Uni. Strathclyde, Glasgow, UK	02/2022
Delta Energy Group, Jiaozhou City, Qingdao, China	
Wetzel IB, Guben	
Wijbenga, PC Geldermalsen, The Netherlands	
Voith Paper, Heidenheim	
HS Zittau/Görlitz, Maschinenwesen	01/2022
Thermische Abfallbehandlung, Lauterbach	
Webb Institute, Glen Cove NY, USA	
TU Berlin, Umweltverfahrenstechnik	
SachsenEnergie, Dresden	
Doosan, Chang-won-si, Gyeongsangnam-do, South Korea	
KW3, LH Veenendaal, The Netherlands	
Université du Luxembourg, Esch-sur-Alzette	
Enseleit IB, Mansfeld	
Caliqua/Equans, Zürich, Switzerland	
Rudnick & Enners, Alpenrod	

2021

Wenisch IB, Vetschau	12/2021
PPCHEM, Hinwil, Switzerland	
KW3, The Netherlands	
BASF Ludwigshafen	
Air-Consult, Jena	
Sjerp & Jongeneel, RB Zoetermeer, The Netherlands	11/2021
Maerz Ofenbau, Zürich, Switzerland	
RWE Supply & Trading, Essen	
Hahn IB, Dresden	10/2021
Therm, South Africa	
RWE Supply & Trading, Essen	
TH Nürnberg, Verfahrenstechnik	09/2021
RWE Supply & Trading, Essen	
Enseleit IB, Mansfeld	

SachsenEnergie, Dresden	
BSH Hausgeräte, Berlin	
Norsk Energi, Oslo, Norway	08/2021
AKM Industrieanlagen, Haltern	
Drill Cool Systems, Bakersfield CA, USA	
Siemens Energy Global, Erlangen	07/2021
Wulff & Umag, Husum	
Planungsbüro Waidhas, Chemnitz	
Burkhardt Energie Technik, Mühlhausen	
Lücke IB, Paderborn	06/2021
TU Dresden, Energieverfahrenstechnik	
Wärme, Hamburg	
AL-KO Therm, Kötz	
PCK Raffinerie, Schwedt	
Vogelsang & Benning, Bochum	05/2021
MTU, München	
VPC Group, Vetschau	
AVG, Köln	04/2021
TH Ulm, Institut für Fahrzeugtechnik	
Marty IB, Oberwil, Switzerland	
HypTec, Lebring, Austria	
Lopez IB, Getxo, Bizkaia, Spain	03/2021
GM Remediation Systems, Leoben, Austria	
Jager Kältetechnik, Osnabrück	
T&M Automation, GR Leidschendam, The Netherlands	
RWE Supply & Trading, Essen	
Stadtwerke Leipzig	
Beuth Hochschule für Technik, Berlin	
Beleth IB, Woeth	02/2021
ZTL, Thal, Austria	
ETABO Bochum	
RWE Supply & Trading, Essen	
Onyx Germany, Berlin	
TU Dresden, Kältetechnik	
GOHL-KTK, Durmersheim	
Therm Development, South Africa	
thermofin, Heinsdorfergrund	
RWE Supply & Trading, Essen	01/2021
STEAG, Essen	
ETA Energieberatung, Pfaffenhofen	
Enex Power, Kirchseeon	

2020

Drill Cool, Bakersfield CA, USA	12/2020
Manders, The Netherlands	
RWE Supply & Trading, Essen	

NEOWAT Lodz, Poland	
University of Duisburg-Essen, Duisburg	11/2020
Stellenbosch University, South Africa	
University De France-COMTe, France	
RWE, Essen	
STEAG, Herne	
Isenmann Ingenieurbüro	
University of Stuttgart, ITLR, Stuttgart	
Norsk Energi, Oslo, Norway	
TGM Kanis, Nürnberg	
Stadtwerke Neuburg	10/2020
Smurfit Kappa, Roermond, The Netherlands	
RWE, Essen	
Hochschule Zittau/Görlitz, Wirtschaftsingenieurwesen	
Stadtwerke, Neuburg	
ILK, Dresden	
ATESTEO, Alsdorf	
Hochschule Zittau/Görlitz, Maschinenwesen	
TH Nürnberg, Verfahrenstechnik	
Drill Cool, Bakersfield CA,USA	09/2020
RWE, Essen	
2Meyers Ingenieurbüro, Nürnberg	
FELUWA, Mürlenbach	
Stadtwerke Neuburg	
Caverion, Wien, Austria	
GMVA Niederrhein, Oberhausen	
INWAT Lodz, Poland	
Troche Ingenieurbüro, Hayingen	08/2020
CEA Saclay, France	
VPC, Vetschau	07/2020
FSK System-Kälte-Klima, Dortmund	
Exergie Etudes, Sarl, Switzerland	
AWG Wuppertal	
STEAG Energy Services, Zwingenberg	
Hochschule Braunschweig	06/2020
DBI, Leipzig	
GOHL-KTK, Dumersheim	
TU Dresden, Energieverfahrenstechnik	
BASF SE, ESI/EE, Ludwigshafen	
Wärme Hamburg	
Ruchti Ingenieurbüro, Uster, Switzerland	
IWB, Basel, Switzerland	
Midiplan, Bietingen-Bissingen	05/2020
Knieschke, Ingenieurbüro	
RWE, Essen	
Leser, Hamburg	

AGRANA, Gmünd, Austria	
EWT Wassertechnik, Celle	
Hochschule Darmstadt	04/2020
MTU München CCP	
HAW Hamburg	03/2020
Hanon, Novi Jicin, Czech Republic	
TU Dresden, Kältetechnik	
MAN, Copenhagen, Denmark	
EnerTech, Radebeul	02/2020
LEAG, Cottbus	
B+B Engineering Magdeburg	
Hochschule Offenburg	
WIB, Dennheritz	01/2020
Universität Duisburg-Essen, Strömungsmaschinen	
Kältetechnik Dresden-Bremen	
TH Ingolstadt	
Vattenfall AB, Jokkmokk, Sweden	
Fraunhofer UMSICHT	

2019

PEU Leipzig, Rötha	12/2019
MB-Holding, Vestenbergsgreuth	
RWE, Essen	
Georg-Büchner-Hochschule, Darmstadt	11/2019
EEB ENERKO, Aldenhoven	
Robert Benoufa Energietechnik, Wiesloch	
Kehrein & Kubanek Klimatechnik, Moers	10/2019
Hanon Systems Autopal Services, Hluk, Czech Republic	
CEA Saclay, Gif Sur Yvette cedex, France	
Saudi Energy Efficiency Center SEEC, Riyadh, Saudi Arabia	
VPC, Vetschau	09/2019
jGanser PM + Engineering, Forchheim	
Endress+Hauser Flowtec AG, Reinach, Switzerland	
Ruchti IB, Uster, Switzerland	
ZWILAG Zwischenlager Würenlingen, Switzerland	08/2019
Hochschule Zittau/Görlitz, Faculty Maschinenwesen	
Stadtwerke Neubrandenburg	
Physikalisch Technische Bundesanstalt PTB, Braunschweig	
GMVA Oberhausen	07/2019
Endress+Hauser Flowtec AG, Reinach, Switzerland	
WARNICA, Waterloo, Canada	
MIBRAG, Zeitz	06/2019
Pöyry, Zürich, Switzerland	
RWTH Aachen, Institut für Strahlantriebe und Turbomaschinen	
Midiplan, Bietigheim-Bissingen	
GKS Schweinfurt	

Comparex Leipzig for LEAG, Berlin	06/2018
Münstermann, Telgte	05/2018
TH Nürnberg, Verfahrenstechnik	
Universität Madrid, Madrid, Spanien	
HS Zittau/Görlitz, Wirtschaftsingenieurwesen	
HS Niederrhein, Krefeld	
Wilhelm-Büchner HS, Pfungstadt	03/2018
GRS, Köln	
WIB, Dennheritz	
RONAL AG, Härklingen, Schweiz	02/2018
Ingenieurbüro Leipert, Riegelsberg	
AIXPROCESS, Aachen	
KRONES, Neutraubling	
Doosan Lentjes, Ratingen	01/2018

Part SI Units

1 Property Library ASHRAE-LibHuAirProp-SI

1.1 Function Overview

1.1.1 Function Overview for Real Moist Air

Functional Dependence	Function Name	Property or Function	Unit of the Result	Page
$a = f(p, t, W)$	a_ptW_HAP_SI	Thermal diffusivity	m ² /s	3/2
$\alpha_p = f(p, t, W)$	alphap_ptW_HAP_SI	Relative pressure coefficient	1/K	3/3
$\beta_p = f(p, t, W)$	betap_ptW_HAP_SI	Isothermal stress coefficient	kg/m ³	3/4
$c = f(p, t, W)$	c_ptW_HAP_SI	Speed of sound	m/s	3/5
$c_p = f(p, t, W)$	cp_ptW_HAP_SI	Specific isobaric heat capacity	kJ/(kg·K)	3/6
$c_v = f(p, t, W)$	cv_ptW_HAP_SI	Specific isochoric heat capacity	kJ/(kg·K)	3/7
$f = f(p, t)$	f_pt_HAP_SI	Enhancement factor (decimal ratio)	-	3/8
$h = f(p, t, W)$	h_ptW_HAP_SI	Air-specific enthalpy	kJ/kg _a	3/9
$\eta = f(p, t, W)$	Eta_ptW_HAP_SI	Dynamic viscosity	Pa·s	3/10
$\kappa = f(p, t, W)$	Kappa_ptW_HAP_SI	Isentropic exponent	-	3/11
$\lambda = f(p, t, W)$	Lambda_ptW_HAP_SI	Thermal conductivity	W/(m·K)	3/12
$\nu = f(p, t, W)$	Ny_ptW_HAP_SI	Kinematic viscosity	m ² /s	3/13
$p = f(t, s, W)$	p_tsW_HAP_SI	Pressure of humid air	kPa	3/14
$p = f(z_{\text{ele}})$	p_zele_HAP_SI	Pressure of humid air from elevation	kPa	3/15
$p_{\text{Air}} = f(p, t, W)$	pAIR_ptW_HAP_SI	Partial pressure of dry air in moist air	kPa	3/16
$p_{\text{H}_2\text{O}} = f(p, t, W)$	pH2O_ptW_HAP_SI	Partial pressure of water vapor in moist air	kPa	3/17
$p_{\text{H}_2\text{O}_s} = f(p, t)$	pH2Os_pt_HAP_SI	Partial saturation pressure of water vapor	kPa	3/18

Functional Dependence	Function Name	Property or Function	Unit of the Result	Page
$\phi = f(p, t, W)$	phi_ptW_HAP_SI	Relative humidity (decimal ratio)	-	3/19
$Pr = f(p, t, W)$	Pr_ptW_HAP_SI	PRANDTL number	-	3/20
$\psi_{Air} = f(W)$	PsiAir_W_HAP_SI	Mole fraction of dry air in moist air	mol _a /mol	3/21
$\psi_{H_2O} = f(W)$	PsiH2O_W_HAP_SI	Mole fraction of water vapor in moist air	mol _w /mol	3/22
$\rho = f(p, t, W)$	Rho_ptW_HAP_SI	Density	kg/m ³	3/23
$s = f(p, t, W)$	s_ptW_HAP_SI	Air-specific entropy	kJ/(kg _a ·K)	3/24
$t = f(p, h, \phi)$	t_phphi_HAP_SI	Backward function: temperature from total pressure, air-specific enthalpy and relative humidity	°C	3/25
$t = f(p, h, W)$	t_phW_HAP_SI	Backward function: temperature from total pressure, air-specific enthalpy and humidity ratio	°C	3/26
$t = f(p, s, W)$	t_psW_HAP_SI	Backward function: temperature from total pressure, air-specific entropy and humidity ratio	°C	3/27
$t = f(p, t_{wb}, W)$	t_ptwbW_HAP_SI	Backward function: temperature from total pressure, wet-bulb temperature and humidity ratio	°C	3/28
$t_d = f(p, W)$	td_pW_HAP_SI	Dew-point/frost-point temperature	°C	3/29
$t_s = f(p, p_{H_2O})$	ts_ppH2O_HAP_SI	Backward function: saturation temperature of water from total pressure and partial pressure of water vapor	°C	3/30
$t_{wb} = f(p, t, W)$	twb_ptW_HAP_SI	Wet-bulb/ice-bulb temperature	°C	3/31
$u = f(p, t, W)$	u_ptW_HAP_SI	Air-specific internal energy	kJ/kg _a	3/32
$v = f(p, t, W)$	v_ptW_HAP_SI	Air-specific volume	m ³ /kg _a	3/33
$W = f(p, t, p_{H_2O})$	W_ptpH2O_HAP_SI	Humidity ratio from total pressure, temperature, and partial pressure of water vapor	kg _w /kg _a	3/34
$W = f(p, t, \phi)$	W_ptphi_HAP_SI	Humidity ratio from total pressure, temperature, and relative humidity	kg _w /kg _a	3/35
$W = f(p, t_d)$	W_ptd_HAP_SI	Humidity ratio from total pressure and dew-point temperature	kg _w /kg _a	3/36

Functional Dependence	Function Name	Property or Function	Unit of the Result	Page
$W = f(p, t, t_{wb})$	W_pttwb_HAP_SI	Humidity ratio from total pressure, (dry bulb) temperature, and wet-bulb temperature	kg _w /kg _a	3/37
$W_s = f(p, t)$	Ws_pt_HAP_SI	Saturation humidity ratio	kg _w /kg _a	3/38
$\xi_{Air} = f(W)$	XiAir_W_HAP_SI	Mass fraction of dry air in moist air	kg _a /kg	3/39
$\xi_{H_2O} = f(W)$	XiH2O_W_HAP_SI	Mass fraction of water vapor in moist air	kg _w /kg	3/40
$Z = f(p, t, W)$	Z_ptW_HAP_SI	Compression factor (decimal ratio)	-	3/41

Range of Validity of Thermodynamic Properties

Property	Range of Validity
Pressure:	$0.01 \leq p \leq 10\,000$ kPa
Temperature:	$-143.15 \leq t \leq 350$ °C
Humidity ratio:	$0 \leq W \leq 10$ kg _w /kg _a
Relative humidity:	$0 \leq \phi \leq 1$ (decimal ratio)
Dew-point temperature:	$-143.15 \leq t_d \leq 350$ °C
Wet-bulb temperature:	$-143.15 \leq t_{wb} \leq 350$ °C

Units

Symbol	Quantity	Unit
p	Pressure	kPa
t	Temperature	°C
W	Humidity ratio	kg _w /kg _a (kg water / kg dry air)
ϕ	Relative humidity	(decimal ratio)
t_d	Dew point temperature	°C
t_{wb}	Wet bulb temperature	°C

Range of Validity of Transport Properties

Property	Range of Validity
Pressure:	$0.01 \leq p \leq 10\,000$ kPa
Temperature:	$-73.15 \leq t \leq 350$ °C
Humidity ratio:	$0 \leq W \leq 10$ kg _w /kg _a
Relative humidity:	$0 \leq \phi \leq 1$ (decimal ratio)

Molar Masses

Component	Molar Mass	Reference
Dry Air	28.966 kg/kmol	[17]
Water	18.015268 kg/kmol	[5], [6]

Reference States

Property	Dry Air	Steam, Water, and Ice
Pressure	101.325 kPa	$p_s(0.01^\circ\text{C}) = 0.611657$ kPa
Temperature	0°C	0.01°C
Enthalpy	0 kJ/kg	0.000611782 kJ/kg
Entropy	0 kJ/(kg K)	0 kJ/(kg K)

1.1.2 Function Overview for Steam and Water for Temperatures $t \geq 0^\circ\text{C}$

Functional Dependence	Function Name	Property	Unit of the Result	Page
$h_{\text{liq}} = f(p, t)$	hliq_pt_97_SI	Specific enthalpy of liquid water	kJ/kg	3/43
$h_{\text{liq,s}} = f(t)$	hliqs_t_97_SI	Specific enthalpy of saturated liquid water	kJ/kg	3/44
$h_{\text{vap,s}} = f(t)$	hvaps_t_97_SI	Specific enthalpy of saturated water vapor	kJ/kg	3/45
$p_s = f(t)$	ps_t_97_SI	Saturation pressure of water	kPa	3/46
$s_{\text{liq}} = f(p, t)$	sliq_pt_97_SI	Specific entropy of liquid water	kJ/(kg·K)	3/47
$s_{\text{liq,s}} = f(t)$	sliqs_t_97_SI	Specific entropy of saturated liquid water	kJ/(kg·K)	3/48
$s_{\text{vap,s}} = f(t)$	svaps_t_97_SI	Specific entropy of saturated water vapor	kJ/(kg·K)	3/49
$t_s = f(p)$	ts_p_97_SI	Saturation temperature of water	$^\circ\text{C}$	3/50
$v_{\text{liq}} = f(p, t)$	vliq_pt_97_SI	Specific volume of liquid water	m^3/kg	3/51
$v_{\text{liq,s}} = f(t)$	vliqs_t_97_SI	Specific volume of saturated liquid water	m^3/kg	3/52
$v_{\text{vap,s}} = f(t)$	vvaps_t_97_SI	Specific volume of saturated water vapor	m^3/kg	3/53

Range of Validity

Property	Range of Validity
Pressure:	$0.01 \leq p \leq 10\,000$ kPa
Temperature:	$0 \leq t \leq 350$ °C

Reference State

Property	Water Vapor and Liquid Water
Pressure	$p_s(0.01^\circ\text{C}) = 0.611657$ kPa
Temperature	0.01°C
Enthalpy	0.000611782 kJ/kg
Entropy	0 kJ/(kg K)

Units

Symbol	Quantity	Unit
p	Pressure	kPa
t	Temperature	°C

1.1.3 Function Overview for Steam and Ice for Temperatures $t \leq 0^\circ\text{C}$

Functional Dependence	Function Name	Property	Unit of the Result	Page
$h_{\text{ice,sub}} = f(t)$	hicesub_t_06_SI	Specific enthalpy of saturated ice	kJ/kg	3/55
$h_{\text{vap,sub}} = f(t)$	hvapsub_t_95_SI	Specific enthalpy of saturated water vapor	kJ/kg	3/56
$p_{\text{mel}} = f(t)$	pmel_t_08_SI	Melting pressure of ice	kPa	3/57
$p_{\text{sub}} = f(t)$	psub_t_08_SI	Sublimation pressure of ice	kPa	3/58
$s_{\text{ice,sub}} = f(t)$	sicesub_t_06_SI	Specific entropy of saturated ice	kJ/(kg·K)	3/59
$s_{\text{vap,sub}} = f(t)$	svapsub_t_95_SI	Specific entropy of saturated water vapor	kJ/(kg·K)	3/60
$t_{\text{mel}} = f(p)$	tmel_p_08_SI	Melting temperature of ice	$^\circ\text{C}$	3/61
$t_{\text{sub}} = f(p)$	tsub_p_08_SI	Sublimation temperature of ice	$^\circ\text{C}$	3/62
$v_{\text{ice,sub}} = f(t)$	vicesub_t_06_SI	Specific volume of saturated ice	m^3/kg	3/63
$v_{\text{vap,sub}} = f(t)$	vvapsub_t_95_SI	Specific volume of saturated water vapor	m^3/kg	3/64

Range of Validity

Property	Range of Validity
Pressure:	$p_{\text{sub}}(-143.15^\circ\text{C}) = 1.2002 \times 10^{-11} \leq p \leq 10\,000 \text{ kPa}$
Temperature:	$-143.15 \leq t \leq 0 \quad ^\circ\text{C}$

Units

Symbol	Quantity	Unit
p	Pressure	kPa
t	Temperature	$^\circ\text{C}$

Reference State

Property	Water Vapor and Ice
Pressure	$p_s(0.01^\circ\text{C}) = 0.611657 \text{ kPa}$
Temperature	0.01°C
Enthalpy	$0.000611782 \text{ kJ/kg}$
Entropy	$0 \text{ kJ}/(\text{kg K})$

1.2 Conversion of SI and I-P Units

Property	Conversion: SI Units → I-P Units	Conversion: I-P Units → SI Units	Units SI	Units I-P
Thermal diffusivity a	$\frac{a_{IP}}{\frac{ft^2}{s}} = \frac{a_{SI}}{\frac{m^2}{s}} \times 10.76391042$	$\frac{a_{SI}}{\frac{m^2}{s}} = \frac{a_{IP}}{\frac{ft^2}{s}} \times 0.0929304$	m ² /s	ft ² /s
Relative pressure coefficient α_p	$\frac{\alpha_{p,IP}}{\frac{1}{^\circ R}} = \frac{\alpha_{p,SI}}{\frac{1}{K}} \times \frac{9}{5}$	$\frac{\alpha_{p,SI}}{\frac{1}{K}} = \frac{\alpha_{p,IP}}{\frac{1}{^\circ R}} \times \frac{5}{9}$	1/K	1/°R
Isothermal stress coefficient β_p	$\frac{\beta_{p,IP}}{\frac{lb}{ft^3}} = \frac{\beta_{p,SI}}{\frac{kg}{m^3}} \times 0.062428$	$\frac{\beta_{p,SI}}{\frac{kg}{m^3}} = \frac{\beta_{p,IP}}{\frac{lb}{ft^3}} \times 16.018463$	kg/m ³	lb/ft ³
Speed of sound c	$\frac{c_{IP}}{\frac{ft}{s}} = \frac{c_{SI}}{\frac{m}{s}} \times 3.2808399$	$\frac{c_{SI}}{\frac{m}{s}} = \frac{c_{IP}}{\frac{ft}{s}} \times 0.3048$	m/s	ft/s
Specific isobaric heat capacity c_p	$\frac{c_{p,IP}}{\frac{Btu}{lb \cdot ^\circ R}} = \frac{c_{p,SI}}{\frac{kJ}{kg \cdot K}} \times 0.2388459$	$\frac{c_{p,SI}}{\frac{kJ}{kg \cdot K}} = \frac{c_{p,IP}}{\frac{Btu}{lb \cdot ^\circ R}} \times 4.1868$	kJ/(kg·K)	Btu/(lb·°R)
Specific isochoric heat capacity c_v	$\frac{c_{v,IP}}{\frac{Btu}{lb \cdot ^\circ R}} = \frac{c_{v,SI}}{\frac{kJ}{kg \cdot K}} \times 0.2388459$	$\frac{c_{v,SI}}{\frac{kJ}{kg \cdot K}} = \frac{c_{v,IP}}{\frac{Btu}{lb \cdot ^\circ R}} \times 4.1868$	kJ/(kg·K)	Btu/(lb·°R)
Dynamic viscosity η	$\frac{\eta_{IP}}{\frac{lb \cdot s}{ft^2}} = \frac{\eta_{SI}}{\frac{Pa \cdot s}}{\frac{Pa}{s}} \times 0.02088543$	$\frac{\eta_{SI}}{\frac{Pa \cdot s}}{\frac{Pa}{s}} = \frac{\eta_{IP}}{\frac{lb \cdot s}{ft^2}} \times 47.880259$	Pa·s	lb·s/ft ²
Enhancement factor f	$f_{IP} = f_{SI}$	$f_{SI} = f_{IP}$	-	-

Property	Conversion: SI Units → I-P Units	Conversion: I-P Units → SI Units	Units SI	Units I-P
Air-specific enthalpy (moist air) h	$\frac{h_{IP}}{\frac{\text{Btu}}{\text{lb}_a}} = \frac{h_{SI}}{\frac{\text{kJ}}{\text{kg}_a}} \times 0.4299226 + 7.68565365666$	$\frac{h_{SI}}{\frac{\text{kJ}}{\text{kg}_a}} = \left(\frac{h_{IP}}{\frac{\text{Btu}}{\text{lb}_a}} - 7.68565365666 \right) \times 2.326$	kJ/kg _a	Btu/lb _a
Specific enthalpy (water, water vapor, ice) h_w	$\frac{h_{IP}}{\frac{\text{Btu}}{\text{lb}}} = \frac{h_{SI}}{\frac{\text{kJ}}{\text{kg}}} \times 0.4299226$	$\frac{h_{SI}}{\frac{\text{kJ}}{\text{kg}}} = \frac{h_{IP}}{\frac{\text{Btu}}{\text{lb}}} \times 2.326$	kJ/kg	Btu/lb
Isentropic exponent κ	$\kappa_{IP} = \kappa_{SI}$	$\kappa_{SI} = \kappa_{IP}$	-	-
Thermal conductivity λ	$\frac{\lambda_{IP}}{\frac{\text{Btu}}{\text{h ft } ^\circ\text{R}}} = \frac{\lambda_{SI}}{\frac{\text{W}}{\text{m K}}} \times 0.57778932$	$\frac{\lambda_{SI}}{\frac{\text{W}}{\text{m K}}} = \frac{\lambda_{IP}}{\frac{\text{Btu}}{\text{h ft } ^\circ\text{R}}} \times 1.73073467$	W/(m·K)	Btu/(h·ft·°R)
Kinematic viscosity ν	$\frac{\nu_{IP}}{\frac{\text{ft}^2}{\text{s}}} = \frac{\nu_{SI}}{\frac{\text{m}^2}{\text{s}}} \times 10.763910417$	$\frac{\nu_{SI}}{\frac{\text{m}^2}{\text{s}}} = \frac{\nu_{IP}}{\frac{\text{ft}^2}{\text{s}}} \times 0.092903040$	m ² /s	ft ² /s
Pressure p	$\frac{p_{IP}}{\text{psi}} = \frac{p_{SI}}{\text{kPa}} \times 0.14503774$	$\frac{p_{SI}}{\text{kPa}} = \frac{p_{IP}}{\text{psi}} \times 6.894757$	kPa	psi
Relative humidity ϕ	$\phi_{IP} = \phi_{SI}$	$\phi_{SI} = \phi_{IP}$	-	-
Prandtl number Pr	$Pr_{IP} = Pr_{SI}$	$Pr_{SI} = Pr_{IP}$	-	-
Mole fraction ψ	$\psi_{IP} = \psi_{SI}$	$\psi_{SI} = \psi_{IP}$	mol/mol	mol/mol
Density ρ	$\frac{\rho_{IP}}{\frac{\text{lb}}{\text{ft}^3}} = \frac{\rho_{SI}}{\frac{\text{kg}}{\text{m}^3}} \times 0.062428$	$\frac{\rho_{SI}}{\frac{\text{kg}}{\text{m}^3}} = \frac{\rho_{IP}}{\frac{\text{lb}}{\text{ft}^3}} \times 16.018463$	kg/m ³	lb/ft ³
Air-specific entropy (moist air) s	$\frac{s_{IP}}{\frac{\text{Btu}}{\text{lb}_a \text{ } ^\circ\text{R}}} = \frac{s_{SI}}{\frac{\text{kJ}}{\text{kg}_a \text{ K}}} \times 0.2388459 + 0.01616365106$	$\frac{s_{SI}}{\frac{\text{kJ}}{\text{kg}_a \text{ K}}} = \left(\frac{s_{IP}}{\frac{\text{Btu}}{\text{lb}_a \text{ } ^\circ\text{R}}} - 0.01616365106 \right) \times 4.1868$	kJ/(kg _a ·K)	Btu/(lb _a ·°R)

Property	Conversion: SI Units → I-P Units	Conversion: I-P Units → SI Units	Units SI	Units I-P
Specific entropy (water, water vapor, ice) s_w	$\frac{s_{IP}}{\frac{\text{Btu}}{\text{lb}_a \cdot ^\circ\text{R}}} = \frac{s_{SI}}{\frac{\text{kJ}}{\text{kg}_a \cdot \text{K}}} \times 0.23884589$	$\frac{s_{SI}}{\frac{\text{kJ}}{\text{kg}_a \cdot \text{K}}} = \frac{s_{IP}}{\frac{\text{Btu}}{\text{lb}_a \cdot ^\circ\text{R}}} \times 4.1868$	kJ/(kg _a ·K)	Btu/(lb _a ·°R)
Temperature t	$\frac{t_{IP}}{^\circ\text{F}} = \frac{t_{SI}}{^\circ\text{C}} \times \frac{9}{5} + 32$	$\frac{t_{SI}}{^\circ\text{C}} = \left(\frac{t_{IP}}{^\circ\text{F}} - 32 \right) \times \frac{5}{9}$	°C	°F
Air-specific internal energy (moist air) u	$(u = h - pv)$ $\frac{u_{IP}}{\frac{\text{Btu}}{\text{lb}_a}} = \frac{h_{SI}}{\frac{\text{kJ}}{\text{kg}_a}} \times 0.4299226 + 7.68565365666$ $- \frac{p_{SI}}{\text{kPa}} \times 0.145037738 \cdot \frac{v_{SI}}{\frac{\text{m}^3}{\text{kg}_a}} \times 16.018453$	$(u = h - pv)$ $\frac{u_{SI}}{\frac{\text{kJ}}{\text{kg}_a}} = \left(\frac{h_{IP}}{\frac{\text{Btu}}{\text{lb}_a}} - 7.68565365666 \right) \times 2.236$ $- \frac{p_{IP}}{\text{psi}} \times 6.894757293 \cdot \frac{v_{IIP}}{\frac{\text{ft}^3}{\text{lb}_a}} \times 0.062428$	kJ/kg _a	Btu/lb _a
Air-specific volume (moist air) v	$\frac{v_{IP}}{\frac{\text{ft}^3}{\text{lb}_a}} = \frac{v_{SI}}{\frac{\text{m}^3}{\text{kg}_a}} \times 16.018453$	$\frac{v_{SI}}{\frac{\text{m}^3}{\text{kg}_a}} = \frac{v_{IP}}{\frac{\text{ft}^3}{\text{lb}_a}} \times 0.062428$	m ³ /kg _a	ft ³ /lb _a
Specific volume (water, water vapor, ice) v_w	$\frac{v_{IP}}{\frac{\text{ft}^3}{\text{lb}}} = \frac{v_{SI}}{\frac{\text{m}^3}{\text{kg}}} \times 16.018453$	$\frac{v_{SI}}{\frac{\text{m}^3}{\text{kg}}} = \frac{v_{IP}}{\frac{\text{ft}^3}{\text{lb}}} \times 0.062428$	m ³ /kg	ft ³ /lb
Humidity ratio W	$W_{IP} = W_{SI}$	$W_{SI} = W_{IP}$	kg _w /kg _a	lb _w /lb _a
Mass fraction ζ	$\zeta_{IP} = \zeta_{SI}$	$\zeta_{SI} = \zeta_{IP}$	kg _w /kg	lb _w /lb
Compression factor Z	$Z_{IP} = Z_{SI}$	$Z_{SI} = Z_{IP}$	-	-

1.3 Calculation Algorithms

1.3.1 Algorithms for Real Moist Air

The properties of moist air are calculated from the modified Hyland-Wexler model given in Herrmann, Kretzschmar, and Gatley (HKG) [1], [2]. The modifications incorporate:

- the value for the universal molar gas constant from the CODATA standard by Mohr and Taylor [22]
- the value for the molar mass of dry air from Gatley et al. [17] and that of water from IAPWS-95 [5], [6]
- the calculation of the ideal-gas parts of the heat capacity, enthalpy, and entropy for dry air from the fundamental equation of Lemmon et al. [14]
- the calculation of the ideal-gas parts of the heat capacity, enthalpy, and entropy for water vapor from IAPWS-IF97 [7], [8], [9] for $t \geq 0^\circ\text{C}$ and from IAPWS-95 [5], [6] for $t \leq 0^\circ\text{C}$
- the calculation of the vapor-pressure enhancement factor from the equation given by the models of Hyland and Wexler [21]
- the calculation of the second and third molar virial coefficients B_{aa} and C_{aaa} for dry air from the fundamental equation of Lemmon et al. [14] according to Feistel et al. [24]
- the calculation of the second and third molar virial coefficients B_{ww} and C_{www} for water and steam from IAPWS-95 [5], [6] according to Feistel et al. [24]
- the calculation of the air-water second molar cross-virial coefficient B_{aw} from Harvey and Huang [15]
- the calculation of the air-water third molar cross-virial coefficients C_{aaw} and C_{aww} from Nelson and Sauer [12], [13]
- the calculation of the saturation pressure of water from IAPWS-IF97 [7], [8], [9] for $t \geq 0^\circ\text{C}$ and of the sublimation pressure of water from IAPWS-08 [11] for $t \leq 0^\circ\text{C}$
- the calculation of the isothermal compressibility of saturated liquid water from IAPWS-IF97 [7], [8], [9] for $t \geq 0^\circ\text{C}$ and that of ice from IAPWS-06 [10] for $t \leq 0^\circ\text{C}$ in the determination of the vapor-pressure enhancement factor
- the calculation of Henry's constant from the IAPWS Guideline 2004 [16] in the determination of the enhancement factor. The mole fractions for the three main components of dry air were taken from Lemmon et al. [14]. Argon was not considered in the calculation of Henry's constant in the former research projects, but it is now the third component of dry air.

The transport properties of moist air are calculated from the model given in Herrmann et al. [3], [4].

1.3.2 Algorithms for Steam and Water for Temperatures $t \geq 0^\circ\text{C}$

The p - T diagram in Fig. 1 shows the formulations used for water and water vapor. The temperature range above 0°C is covered by IAPWS-IF97 [7], [8], [9]:

- The saturation line is calculated from the IAPWS-IF97 saturation pressure equation $p_s^{97}(t)$ and saturation temperature equation $t_s^{97}(p)$.
- The properties in the liquid region including saturated-liquid line are calculated from the fundamental equation of the IAPWS-IF97 region 1.
- The properties in the vapor region including saturated-vapor line are calculated from the fundamental equation of the IAPWS-IF97 region 2.

1.3.3 Algorithms for Steam and Ice for Temperatures $t \leq 0^\circ\text{C}$

- The sublimation curve is covered by the IAPWS-08 sublimation pressure equation $p_{\text{subl}}^{08}(t)$ [11] (see Fig. 1).
- The properties of ice including saturated ice are determined by the fundamental equation of the IAPWS-06 [10].
- The properties of vapor including saturated vapor are calculated from the fundamental equation of IAPWS-95 [5], [6].

1.3.4 Overview of the Applied Formulations for Steam, Water, and Ice

The following p - T diagram shows the used IAPWS Formulations and the ranges where they are applied.

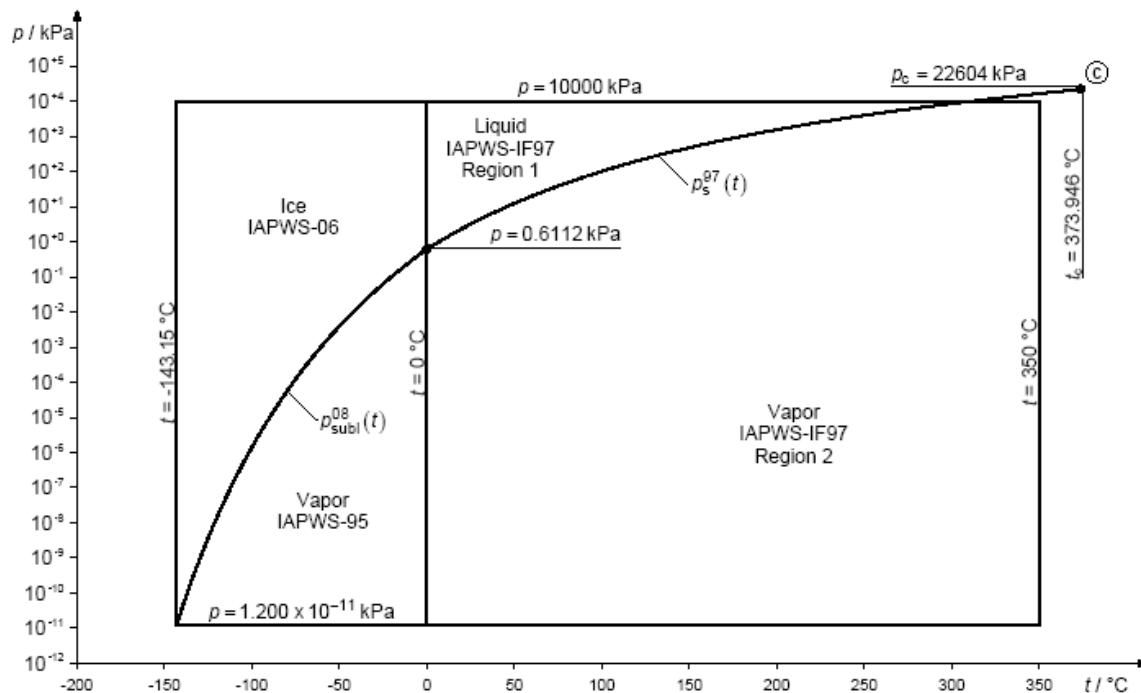


Figure 1: p - T diagram with used IAPWS formulations for steam, water, and ice.

2 Add-On FluidLAB for MATLAB® for ASHRAE-LibHuAirProp-SI

2.1 Installing FluidLAB

The FluidLAB Add-On has been developed to calculate thermodynamic properties in MATLAB® more conveniently. Within MATLAB® it enables the direct call of functions relating to real moist air, steam, water, and ice from the ASHRAE-LibHuAirProp-SI property library.

The installation of FluidLAB and ASHRAE-LibHuAirProp-SI is described in Section 2.1 in "Part I-P Units" of this User's Guide.

2.2 Example Calculations of $h = f(p, t, W)$

2.2.1 Preparation for the Example: Calculation of $h = f(p, t, W)$

Now we will calculate, step by step, the air-specific enthalpy h of moist air as a function of mixture pressure p , given temperature t and given humidity ratio W , using FluidLAB.

Please carry out the following instructions:

- Start Windows Explorer®, Total Commander®, My Computer or another file manager program. The following description refers to Windows Explorer®.
- Your Windows Explorer® should be set to "Details" for easier viewing. Click the "Views" button and select "Details".
- Switch into the program directory of FluidLAB in which you will find the folder "\LibHuAirProp"; it is generally saved under: "C:\Users\...\Documents\FuildLAB".
- Create the folder "\LibHuAirProp_SI_Example" by clicking on "File" in the Explorer® menu, then "New" in the menu which appears and afterwards selecting "Folder". Name the new folder "\LibHuAirProp_SI_Example".
- You will now see the following window:

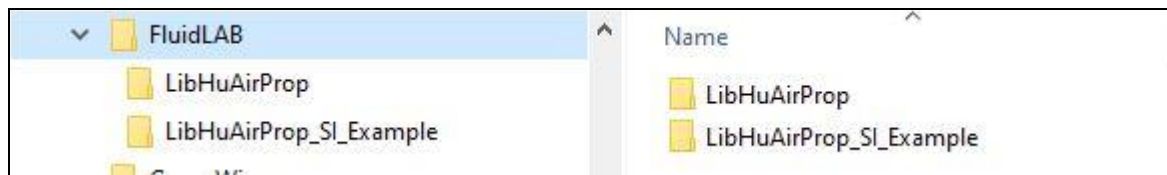


Figure 2.2.1: Folders "LibHuAirProp" and "LibHuAirProp_SI_Example"

- Switch into the directory "\LibHuAirProp" within "\FluidLAB", the standard being: "C:\Users\...\Documents\FuildLAB\LibHuAirProp".
- You will see the following window:



Figure 2.2.2: Contents of the folder "LibHuAirProp"

You will now have to copy the following files into the directory "C:\Users\...\Documents\FluidLAB\LibHuAirProp_SI_Example" in order to calculate the function $h = f(p, t, W)$.

- The following files are needed:
 - h_ptW_HAP_SI.mexw64
 - LC.dll
 - LibHuAirProp_SI.dll
- Click the file "h_ptW_HAP_SI.mexw64", then click "Edit" in the upper menu bar and select "Copy".
- Switch into the directory "C:\Users\...\Documents\FluidLAB\LibHuAirProp_SI_Example", click "Edit" and then "Paste".
- Repeat these steps in order to copy the other files listed above. You may also select all the above-named files and then copy them as a group (press the Control button to enable multiple markings).
- You will see the following contents of the folder "LibHuAirProp_SI_Example":

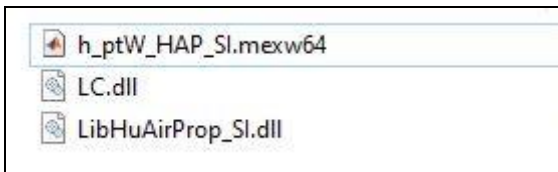


Figure 2.2.3: Contents of the folder "LibHuAirProp_SI_Example"

- Start MATLAB® (if you have not started it already).
- Click the button marked in the following image in order to open the folder "LibHuAirProp_SI_Example" in the "Current Directory" window.

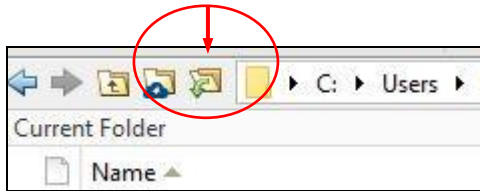


Figure 2.2.4: Selection of the working directory

- Find and select the directory "C:\Users\...\Documents\FluidLAB\LibHuAirProp_SI_Example" in the menu that appears and confirm your selection by clicking the "OK" button.

For the example calculation in a M-File see Subsection 2.2.2 below and for the example calculation in the Command Window please see Subsection 2.2.3 on page SI – 2/6.

2.2.2 Example: Calculation of $h = f(p, t, W)$ in a M-File

- First of all you need to create an M-File in MATLAB®. Within MATLAB® click "File", then select "New" and afterwards click "Script" in MATLAB 2010 or later versions.
- If the "Editor" window appears as a separate window, you can embed it into MATLAB® by clicking the insertion arrow (see Figure 2.2.5) in order to obtain a better view.

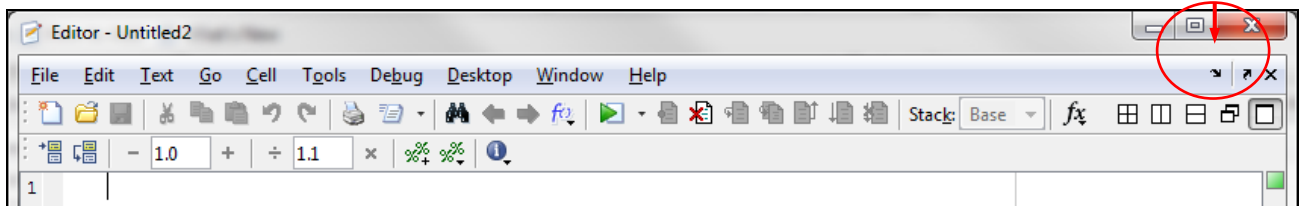


Figure 2.2.5: Embedding the "Editor - Untitled" window

- In Figure 2.2.6 you will see the "Editor - Untitled" window.



Figure 2.2.6: Embedded "Editor" window

- Now type the following lines in the "Editor - Untitled" window:

Text to be written:	Explanation:
<code>% h_ptW_HAP_SI.m</code>	file name as comment
<code>%%</code>	paragraph separation
<code>p=101.325; % pressure in kPa</code>	declaration of the variables pressure, temperature, and humidity ratio
<code>t=20; % temperature in °C</code>	
<code>W=0.01; % humidity ratio W in kg_w/kg_a</code>	
<code>%%</code>	paragraph separation
<code>h=h_ptW_HAP_SI(p, t, W)</code>	function call
<code>%%</code>	paragraph separation

Remarks:

- The program interprets the first line, starting with "%," to be a data description in "Current Directory".
- Paragraph separations which are mandatory are marked with "%%". This also serves to separate the declaration of variables and calculation instructions.
- The words which are printed in green, start with "%" and come after the variables are comments. They are not in fact absolutely necessary, but they are very helpful for your overview and to make the process more easily understood.
- Omit the semicolons after the numerical values if you wish to see the result for h and the input parameters.

The values of the function parameters in their corresponding units stand for:

- **First operand: Value for $p = 101.325$ kPa**
(Range of validity: $p = 0.01 \dots 10000$ kPa)
- **Second operand: Value for $t = 20^\circ\text{C}$**
(Range of validity: $t = -143.15 \dots 350^\circ\text{C}$)
- **Third operand: Value for $W = 0.01$ kg_w/kg_a (kg water per kg air)**
(Range of validity: $W = 0 \dots 10$ kg_w/kg_a)
- Save the "M-File" by clicking the "File" button and then click "Save As...".
- The menu "Save file as:" appears; In this menu, the folder name "LibHuAirProp_SI_Example" must be displayed in the "Save in:" field.
- Next to "File name" you have to type "Example_h_ptW_HAP_SI.m" and afterwards click the "Save" button.

Note.

The name of the example file has to be different in comparison to the name of the used function. For example, the file could not be named "h_ptW_HAP_SI.m" in this case. Otherwise an error message will appear during the calculation.

- You will now see the following window (Figure 2.2.7):

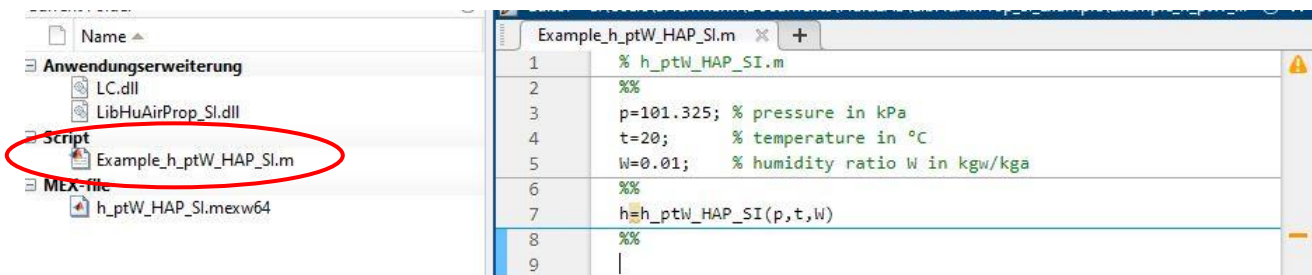


Figure 2.2.7: "Example_h_ptW_HAP_SI.m" M-file

- Within the "Current Directory" window the file "Example_h_ptW_HuAirProp_SI.m" appears.
- Right-click on this file and select "Run" in the menu which appears (see Figure 2.2.8).

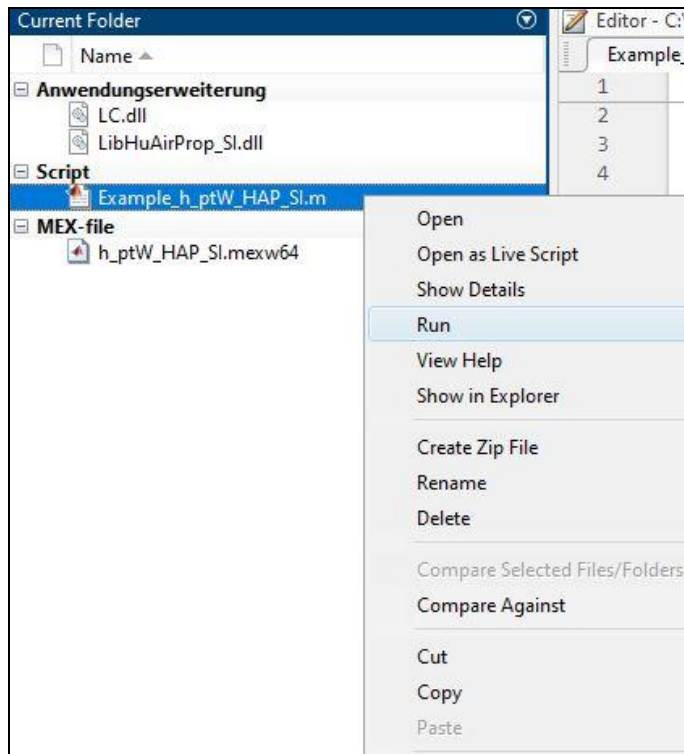


Figure 2.2.8: Running the "Example_h_ptW_HAP_SI.m" M-file

- You will see the following window:

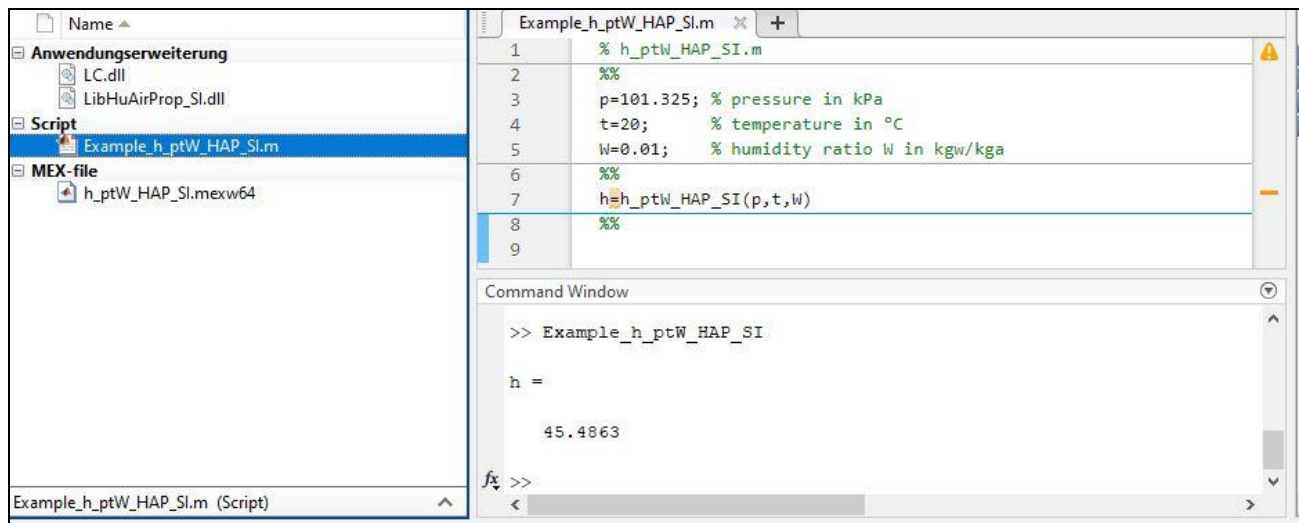


Figure 2.2.9: MATLAB® with calculated result

The result for h appears in the "Command Window".

⇒ The result in our sample calculation here is: "h = 45.4863". The corresponding unit is kJ/kg (see table of the property functions in Chapter 1).

To be able to calculate other values, you have to copy the associated mexw64 files as well because MATLAB® can only access functions that are located in the "Current Directory" window. The calculated example can be found in the directory "C:\Users\...\Documents\FluidLAB\LibHuAirProp_SI_Example", and you may use it as a basis for further calculations with FluidLAB.

2.2.3 Example: Calculation of $h = f(p, t, W)$ in the Command Window

Please follow the instructions in Subsection 2.2.1 (pages SI – 2/1 to SI – 2/3).

- You will see the following window (Figure 2.2.10):

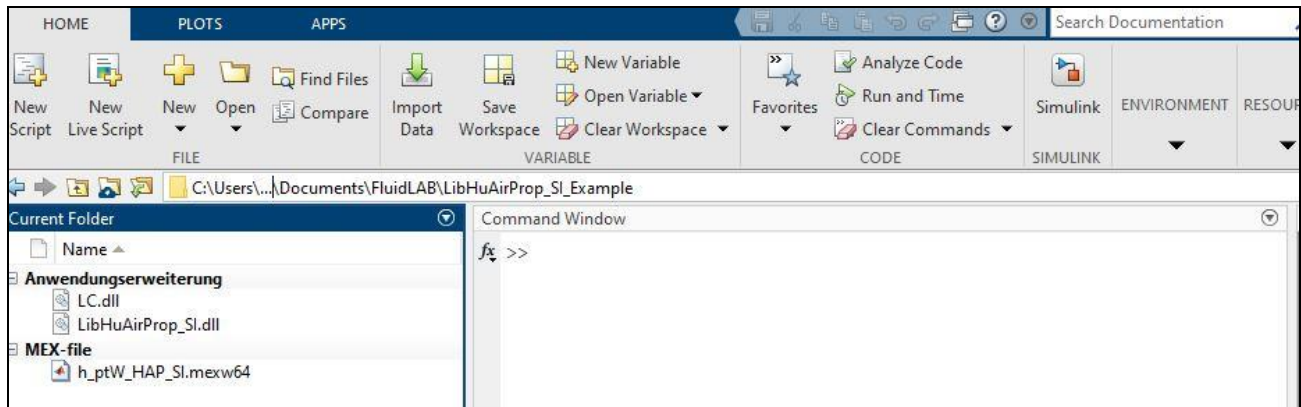


Figure 2.2.10: MATLAB® with necessary files

Corresponding to the table of the property functions in Chapter 1 you have to call up the function "h_ptW_HAP_SI" as follows for calculating $h = f(p, t, W)$

Write "`h=h_ptW_HAP_SI(101.325,20,0.01)`" within the "Command Window".

The values of the function parameters in their corresponding units stand for:

- First operand: Value for $p = 101.325$ kPa
(Range of validity: $p = 0.01 \dots 10\,000$ kPa)
- Second operand: Value for $t = 20^\circ\text{C}$
(Range of validity: $t = -143.15 \dots 350^\circ\text{C}$)
- Third operand: Value for $W = 0.01$ kg_w/kg_a (kg water per kg air)
(Range of validity: $W = 0 \dots 10$ kg_w/kg_a)
- Confirm your entry by pressing the "ENTER" button.
- You will see the following window (Figure 2.2.11):

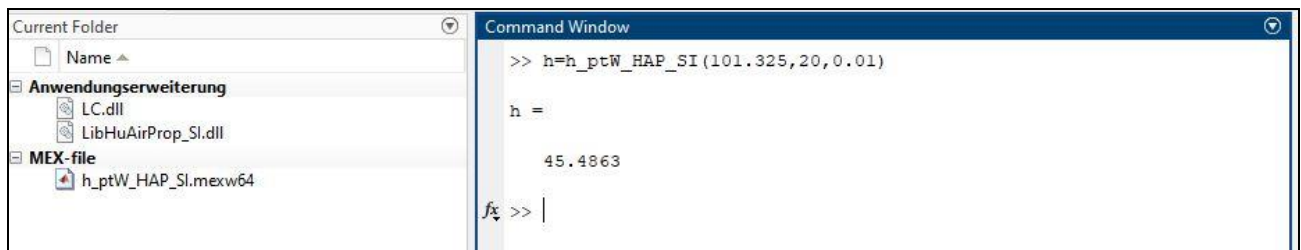


Figure 2.2.11: MATLAB® with calculated result

The result for h appears in the "Command Window".

⇒ In the "Command Window" you will see the result "`h = 45.4863`". The corresponding unit is kJ/kg (see table of the property functions in Chapter 1).

To be able to calculate other values, you will have to copy the respective mexw64 files into the working directory as well, because MATLAB® can only access functions that are located in the "Current Directory" window.

2.3 Removing FluidLAB including LibHuAirProp

The de-installation of FluidLAB and ASHRAE-LibHuAirProp-SI is described in Section 2.4 in "Part I-P Units" of this User's Guide.

3 Property Functions of ASHRAE-LibHuAirProp-SI

3.1 Functions for Real Moist Air

Thermal Diffusivity $a = f(p, t, W)$
--

Function Name:

a_ptW_HAP_SI

Fortran Program:

REAL*8 FUNCTION A_PTW_HUAIRPROP(P,T,W), REAL*8 P,T,W

Input Values:

p - Total pressure p in kPa
 t - Temperature t in °C
 W - Humidity ratio W in kg_w/kg_a

Result:a_ptW_HAP_SI - Thermal diffusivity of humid air in m^2/s **Range of Validity:**

Temperature t : from -73.15°C to 350°C
 Total pressure p : from 0.01 kPa to 10 000 kPa
 Humidity ratio W : $0 \leq W \leq W_s$

Comments:

- Thermal diffusivity $a = \frac{\lambda}{\rho \cdot c_p}$
- Calculation for supersaturated humid air ($W > W_s$) is not possible

Result for Wrong Input Values:

a_ptW_HAP_SI = -1000

References:

$\lambda(p, t, W)$ Herrmann et al. [3], [4]
 $\rho(p, t, W)$ Herrmann et al. [1], [2]
 $c_p(p, t, W)$ Herrmann et al. [1], [2]

Relative Pressure Coefficient $\alpha_p = f(p, t, W)$
Function Name:

alphap_ptW_HAP_SI

Fortran Program:

```
REAL*8 FUNCTION ALPHAP_PTW_HUAIRPROP(P,T,W), REAL*8 P,T,W
```

Input Values:

p - Total pressure p in kPa
 t - Temperature t in °C
 W - Humidity ratio W in kg_w/kg_a

Result:

alphap_ptW_HAP_SI - Relative pressure coefficient of humid air in 1/K

Range of Validity:

Temperature t : from -143.15°C to 350°C
 Total pressure p : from 0.01 kPa to 10 000 kPa
 Humidity ratio W : $0 \leq W \leq W_s$

Comments:

- Relative pressure coefficient $\alpha_p = \frac{1}{p} \left(\frac{\partial p}{\partial T} \right)_v$
- Calculation for supersaturated humid air ($W > W_s$) is not possible

Result for Wrong Input Values:

alphap_ptW_HAP_SI = -1000

References:

$\rho(p, t, W)$ Herrmann et al. [1], [2]

Isothermal Stress Coefficient $\beta_p = f(p, t, W)$
Function Name:

betap_ptW_HAP_SI

Fortran Program:

REAL*8 FUNCTION BETAP_PTW_HUAIRPROP(P,T,W), REAL*8 P,T,W

Input Values:

p - Total pressure p in kPa
 t - Temperature t in °C
 W - Humidity ratio W in kg_w/kg_a

Result:betap_ptW_HAP_SI - Isothermal stress coefficient of humid air in kg/m³**Range of Validity:**

Temperature t : from -143.15°C to 350°C
 Total pressure p : from 0.01 kPa to 10 000 kPa
 Humidity ratio W : $0 \leq W \leq W_s$

Comments:

- Isothermal stress coefficient $\beta_p = -\frac{1}{p} \left(\frac{\partial p}{\partial v} \right)_T$
- Calculation for supersaturated humid air ($W > W_s$) is not possible

Result for Wrong Input Values:

betap_ptW_HAP_SI = -1000

References: $v(p, t, W)$ Herrmann et al. [1], [2]

Speed of Sound $c = f(p, t, W)$

Function Name:

c_ptW_HAP_SI

Fortran Program:

REAL*8 FUNCTION C_PTW_HUAIRPROP(P,T,W), REAL*8 P,T,W

Input Values:

p - Total pressure p in kPa
 t - Temperature t in °C
 W - Humidity ratio W in kg_w/kg_a

Result:

c_ptW_HAP_SI - Speed of sound of humid air in m/s

Range of Validity:

Temperature t : from -143.15°C to 350°C
 Total pressure p : from 0.01 kPa to 10 000 kPa
 Humidity ratio W : $0 \leq W \leq 10 \text{kg}_w/\text{kg}_a$

Comments:

- Speed of sound $c = v \sqrt{-\left(\frac{\partial p}{\partial v}\right)_s}$

- When calculating supersaturated air an ideal mixing model (saturated humid air and water droplets (or ice crystals) homogeneously mixed) is applied

Result for Wrong Input Values:

c_ptW_HAP_SI = -1000

References:

$v(p, t, W)$ Herrmann et al. [1], [2]

Specific Isobaric Heat Capacity $c_p = f(p, t, W)$
Function Name:

cp_ptW_HAP_SI

Fortran Program:

```
REAL*8 FUNCTION CP_PTW_HUAIRPROP(P,T,W), REAL*8 P,T,W
```

Input Values:

p - Total pressure p in kPa
 t - Temperature t in °C
 W - Humidity ratio W in kg_w/kg_a

Result:

cp_ptW_HAP_SI - Specific isobaric heat capacity of humid air in kJ/(kg K)

Range of Validity:

Temperature t : from -143.15°C to 350°C
 Total pressure p : from 0.01 kPa to 10 000 kPa
 Humidity ratio W : $0 \leq W \leq W_s$

Comments:

- Specific isobaric heat capacity $c_p = \left(\frac{\partial h}{\partial T} \right)_p$
- Calculation for supersaturated humid air ($W > W_s$) is not possible

Result for Wrong Input Values:

cp_ptW_HAP_SI = -1000

References: $h(p, t, W)$ Herrmann et al. [1], [2]

Specific Isochoric Heat Capacity $c_v = f(p, t, W)$
Function Name:

cv_ptW_HAP_SI

Fortran Program:

```
REAL*8 FUNCTION CV_PTW_HUAIRPROP(P,T,W), REAL*8 P,T,W
```

Input Values:

p - Total pressure p in kPa
 t - Temperature t in °C
 W - Humidity ratio W in kg_w/kg_a

Result:

cv_ptW_HAP_SI - Specific isochoric heat capacity of humid air in kJ/(kg K)

Range of Validity:

Temperature t : from -143.15°C to 350°C
 Total pressure p : from 0.01 kPa to 10 000 kPa
 Humidity ratio W : $0 \leq W \leq W_s$

Comments:

- Specific isochoric heat capacity $c_v = \left(\frac{\partial u}{\partial T} \right)_v$
- Calculation for supersaturated humid air ($W > W_s$) is not possible

Result for Wrong Input Values:

cv_ptW_HAP_SI = -1000

References:

$c_v(p, t, W)$ Herrmann et al. [3], [4]

Enhancement Factor $f = f(p,t)$ **Function Name:**

f_pt_HAP_SI

Fortran Program:

REAL*8 FUNCTION F_PT_HUAIRPROP(P,T), REAL*8 P,T

Input Values:

p - Total pressure p in kPa
 t - Temperature t in °C

Result:

f_pt_HAP_SI - Enhancement factor of water (decimal ratio)

Range of Validity:

Temperature t : from -143.15°C to 350°C
 Total pressure p : from 0.01 kPa to 10 000 kPa

Comments:

- Enhancement factor $f = \frac{\rho_{\text{H}_2\text{O},s}}{\rho_s(t)}$

with $\rho_s(t)$ for $t \geq 0.01^\circ\text{C}$ - Steam pressure of water

for $t < 0.01^\circ\text{C}$ - Sublimation pressure of water

- Describes the enhancement of the saturation pressure of water in the air atmosphere under elevated pressure

- Derived iteratively from the isothermal compressibility of liquid water, from Henry's constant [15], [16] and from the virial coefficients of air, water, and the air-water mixture

Result for Wrong Input Values:

f_pt_HAP_SI = -1000

References:

$f(p,t)$ Herrmann et al. [1], [2]

Air-Specific Enthalpy $h = f(p, t, W)$
--

Function Name:

h_ptW_HAP_SI

Fortran Program:

```
REAL*8 FUNCTION H_PTW_HUAIRPROP(P,T,W), REAL*8 P,T,W
```

Input Values:

p - Total pressure p in kPa
 t - Temperature t in °C
 W - Humidity ratio W in kg_w/kg_a

Result:h_ptW_HAP_SI - Air-specific enthalpy in kJ/kg_a**Range of Validity:**

Temperature t : from -143.5°C to 350°C
 Total pressure p : from 0.01 kPa to 10 000 kPa
 Humidity ratio W : $0 \leq W \leq 10 \text{ kg}_w/\text{kg}_a$

Comments:

- When calculating supersaturated air an ideal mixing model (saturated humid air and water droplets (or ice crystals) homogeneously mixed) is applied

Result for Wrong Input Values:

h_ptW_HAP_SI = -1000

References:

$h(p, t, W)$ Herrmann et al. [1], [2]
 $h_w(p, t)$ IAPWS-IF97 [7], [8] and IAPWS-06 [11]
 $h_a(t)$ Lemmon et al. [14]

Dynamic Viscosity $\eta = f(p, t, W)$ **Function Name:**

Eta_ptW_HAP_SI

Fortran Program:

REAL*8 FUNCTION ETA_PTW_HUAIRPROP(P,T,W), REAL*8 P,T,W

Input Values:

p - Total pressure p in kPa
 t - Temperature t in °C
 W - Humidity ratio W in kg_w/kg_a

Result:

Eta_ptW_HAP_SI - Dynamic viscosity of humid air in Pa s

Range of Validity:

Temperature t : from -73.15°C to 350°C
 Total pressure p : from 0.01 kPa to 10 000 kPa
 Humidity ratio W : $0 \leq W \leq 10 \text{ kg}_w/\text{kg}_a$

Comments:

- A new very accurate algorithm is implemented between 0°C and 350°C
- When calculating supersaturated air an ideal mixing model (saturated humid air and water droplets (or ice crystals) homogeneously mixed) is applied

Result for Wrong Input Values:

Eta_ptW_HAP_SI = -1000

References:

$\eta(p, t, W)$ Herrmann et al. [3], [4]
 $\eta_a(t)$ Lemmon et al. [18]
 $\eta_w(p, t)$ IAPWS-IF97 [7], [8] and IAPWS-08 [19]

Isentropic Exponent $\kappa = f(p, t, W)$

Function Name:

Kappa_ptW_HAP_SI

Fortran Program:

```
REAL*8 FUNCTION KAPPA_PTW_HUAIRPROP(P,T, W), REAL*8 P,T,W
```

Input Values:

p - Total pressure p in kPa
 t - Temperature t in °C
 W - Humidity ratio W in kg_w/kg_a

Result:

Kappa_ptW_HAP_SI - Isentropic exponent

Range of Validity:

Temperature t : from -143.5°C to 350°C
 Total pressure p : from 0.01 kPa to 10 000 kPa
 Humidity ratio W : $0 \leq W \leq 10 \text{ kg}_w/\text{kg}_a$

Comments:

- Isentropic exponent $\kappa = -\frac{v}{p} \left(\frac{\partial p}{\partial v} \right)_s$
- When calculating supersaturated air an ideal mixing model (saturated humid air and water droplets homogeneously mixed) is applied for $t \geq 0.01^\circ\text{C}$. For temperatures below (ice fog) the value of the saturated state is applied.

Result for Wrong Input Values:

Kappa_ptW_HAP_SI = -1000

References:

$v(p, t, W)$ Herrmann et al. [1], [2]

Thermal Conductivity $\lambda = f(p, t, W)$
Function Name:

Lambda_ptW_HAP_SI

Fortran Program:

```
REAL*8 FUNCTION LAMBDA_PTW_HUAIRPROP(P,T, W), REAL*8 P,T,W
```

Input Values:

p - Total pressure p in kPa
 t - Temperature t in °C
 W - Humidity ratio W in kg_w/kg_a

Result:

Lambda_ptW_HAP_SI - Thermal conductivity in $\text{W}/(\text{m K})$

Range of Validity:

Temperature t : from -73.5°C to 350°C
 Total pressure p : from 0.01 kPa to 10 000 kPa
 Humidity ratio W : $0 \leq W \leq 10 \text{ kg}_w/\text{kg}_a$

Comments:

- A new very accurate algorithm is implemented between 0°C and 350°C
- When calculating supersaturated air an ideal mixing model (saturated humid air and water droplets (or ice crystals) homogeneously mixed) is applied

Result for Wrong Input Values:

Lambda_ptW_HAP_SI = -1000

References:

$\lambda(p, t, W)$ Herrmann et al. [3], [4]
 $\lambda_a(t)$ Lemmon et al. [18]
 $\lambda_w(p, t)$ IAPWS-IF97 [7], [8] and IAPWS-08 [20]

Kinematic Viscosity $\nu = f(p, t, W)$

Function Name:

Ny_ptW_HAP_SI

Fortran Program:

```
REAL*8 FUNCTION NY_PTW_HUAIRPROP(P,T,W), REAL*8 P,T,W
```

Input Values:

p - Total pressure p in kPa
 t - Temperature t in °C
 W - Humidity ratio W in kg_w/kg_a

Result:

Ny_ptW_HAP_SI - Kinematic viscosity in m^2/s

Range of Validity:

Temperature t : from -73.5°C to 350°C
 Total pressure p : from 0.01 kPa to 10 000 kPa
 Humidity ratio W : $0 \leq W \leq W_s$

Comments:

- Kinematic Viscosity $\nu = \frac{\eta}{\rho}$

Result for Wrong Input Values:

Ny_ptW_HAP_SI = -1000

References:

$\eta(p, t, W)$ Herrmann et al. [3], [4]
 $\rho(p, t, W)$ Herrmann et al. [1], [2]

Backward Function: Total Pressure $p = f(t,s,W)$ **Function Name:**

p_tsW_HAP_SI

Fortran Program:

REAL*8 FUNCTION P_TSW_HUAIRPROP(T,S,W), REAL*8 T,S,W

Input Values:

t - Temperature t in °C
 s - Air-specific entropy s in kJ/(kg_a K)
 W - Humidity ratio W in kg_w/kg_a

Result:

p_tsW_HAP_SI - Total pressure in kPa

Range of Validity:

Temperature t : from -143.5°C to 350°C
 Air-specific entropy s : from -26.53 kJ/(kg_a K) to 38.990 kJ/(kg_a K)
 Humidity ratio W : $0 \leq W \leq 10$ kg_w/kg_a

Comments:- Iteration of total pressure p from $s = f(p,t,W)$ **Result for Wrong Input Values:**

p_tsW_HAP_SI = -1000

References: $s(p,t,W)$ Herrmann et al. [1], [2]

Pressure $p = f(z_{\text{ele}})$
Function Name:

p_zele_HAP_SI

Fortran Program:

```
REAL*8 FUNCTION P_ZELE_HUAIRPROP(ZELE), REAL*8 ZELE
```

Input Values:

z_{ele} - Elevation z_{ele} in m

Result:

p_zele_HAP_SI - Pressure of humid air in kPa

Range of Validity:

Elevation z_{ele} from -5,000 m to 11,000 m

Comments:

- Pressure of humid air from elevation

$$- p(z_{\text{ele}}) = 101.325 \text{ kPa} \cdot \left(1 - 2.25577 \cdot 10^{-5} \cdot \frac{z_{\text{ele}}}{\text{m}} \right)^{5.256}$$

Result for Wrong Input Values:

p_zele_HAP_SI = -1000

References:

$p(z_{\text{ele}})$ ASHRAE [23]

Partial Pressure of Dry Air $p_{\text{Air}} = f(p, t, W)$
Function Name:

pAir_ptW_HAP_SI

Fortran Program:

```
REAL*8 FUNCTION PAIR_PTW_HUAIRPROP(P,T,W), REAL*8 P,T,W
```

Input Values:

p - Total pressure p in kPa
 t - Temperature t in °C
 W - Humidity ratio W in kg_w/kg_a

Result:

pAir_ptW_HAP_SI - Partial pressure of (dry) air in humid air in kPa

Range of Validity:

Temperature t : from -143.15°C to 350°C
 Total pressure p : from 0.01 kPa to 10 000 kPa
 Humidity ratio W : $0 \leq W \leq 10 \text{ kg}_w/\text{kg}_a$

Comments:

- Partial pressure of (dry) air in humid air $p_{\text{Air}} = 1 - p_{\text{H}_2\text{O}}$
- Partial pressure of water vapor at saturation is calculated in case of supersaturated humid air ($W > W_s(p, t)$)
- The temperature value is used to calculate the saturation state

Result for Wrong Input Values:

pAir_ptW_HAP_SI = -1000

References:

$p_{\text{H}_2\text{O}}(p, W)$ Herrmann et al. [1], [2]

Partial Pressure of Water Vapor $p_{\text{H}_2\text{O}} = f(p, t, W)$

Function Name:

pH2O_ptW_HAP_SI

Fortran Program:

```
REAL*8 FUNCTION PH2O_PTW_HUAIRPROP(P,T,W), REAL*8 P,T,W
```

Input Values:

p - Total pressure p in kPa
 t - Temperature t in °C
 W - Humidity ratio W in kg_w/kg_a

Result:

pH2O_ptW_HAP_SI - Partial pressure of water vapor in humid air in kPa

Range of Validity:

Temperature t : from -143.15°C to 350°C
 Total pressure p : from 0.01 kPa to 10 000 kPa
 Humidity ratio W : $0 \leq W \leq 10 \text{ kg}_w/\text{kg}_a$

Comments:

- Partial pressure of water vapor in humid air $p_{\text{H}_2\text{O}} = \frac{W \cdot p}{\left(\frac{R_a}{R_w} + W\right)}$
- Partial pressure of water vapor at saturation is calculated in case of supersaturated humid air ($W > W_s(p, t)$)
- The temperature value is used to calculate the saturation state

Result for Wrong Input Values:

pH2O_ptW_HAP_SI = -1000

References:

$p_{\text{H}_2\text{O}}(p, W)$ Herrmann et al. [1], [2]

Partial Saturation Pressure of Water Vapor $p_{\text{H}_2\text{O},s} = f(p, t)$
Function Name:

pH2Os_pt_HAP_SI

Fortran Program:

REAL*8 FUNCTION PH2OS_PT_HUAIRPROP(P,T), REAL*8 P,T

Input Values:

p - Total pressure p in kPa
 t - Temperature t in °C

Result:

pH2Os_pt_HAP_SI - Partial saturation pressure of water vapor in humid air in kPa

Range of Validity:

Temperature t : from -143.15°C to 350°C
 Total pressure p : from 0.01 kPa to 10 000 kPa

Comments:

- Partial pressure of steam at saturation $p_{\text{H}_2\text{O},s} = f \cdot p_s(t)$
 with $p_s(t)$ for $t \geq 0.01^\circ\text{C}$ - Steam pressure of water
 for $t < 0.01^\circ\text{C}$ - Sublimation pressure of water

Result for Wrong Input Values:

pH2Os_pt_HAP_SI = -1000

References:

$f(p, t)$		Herrmann et al. [1], [2]
$p_s(t)$	for $t \geq 0.01^\circ\text{C}$	IAPWS-IF97 [7], [8]
	for $t < 0.01^\circ\text{C}$	IAPWS-08 [11]

Relative Humidity $\varphi = f(p, t, W)$
Function Name:

phi_ptW_HAP_SI

Fortran Program:

```
REAL*8 FUNCTION PHI_PTW_HUAIRPROP(P,T,W), REAL*8 P,T,W
```

Input Values:

p - Total pressure p in kPa
 t - Temperature t in °C
 W - Humidity ratio W in kg_w/kg_a

Result:

phi_ptW_HAP_SI - Relative humidity (decimal ratio)

Range of Validity:

Temperature t : from -143.15°C to 350°C
 Total pressure p : from 0.01 kPa to 10 000 kPa
 Humidity ratio W : $0 \leq W \leq 10 \text{ kg}_w/\text{kg}_a$

Comments:

- Relative humidity $\varphi = \frac{p_{\text{H}_2\text{O}}}{p_{\text{H}_2\text{O},s}}$
- This equation is valid for $p_{\text{H}_2\text{O}} \leq p_{\text{H}_2\text{O},s}$ and for $0 \leq \varphi \leq 1$

Result for Wrong Input Values:

phi_ptW_HAP_SI = -1000

References:

$\varphi(p, t, W)$ Herrmann et al. [1], [2]

Prandtl Number $Pr = f(p, t, W)$
Function Name:

Pr_ptW_HAP_SI

Fortran Program:

REAL*8 FUNCTION PR_PTW_HUAIRPROP(P,T,W), REAL*8 P,T,W

Input Values:

p - Total pressure p in kPa
 t - Temperature t in °C
 W - Humidity ratio W in kg_w/kg_a

Result:

Pr_ptW_HAP_SI - Prandtl number

Range of Validity:

Temperature t : from -73.5°C to 350°C
 Total pressure p : from 0.01 kPa to 10 000 kPa
 Humidity ratio W : $0 \leq W \leq 10 \text{ kg}_w/\text{kg}_a$

Comments:

- Prandtl number $Pr = \frac{\eta \cdot c_p}{\lambda}$
- Calculation for supersaturated humid air ($W > W_s$) is not possible

Result for Wrong Input Values:

Pr_ptW_HAP_SI = -1000

References:

$\eta(p, t, W)$ Herrmann et al. [3], [4]
 $c_p(p, t, W)$ Herrmann et al. [3], [4]
 $\lambda(p, t, W)$ Lemmon et al. [20]

Mole Fraction of Dry Air $\psi_{\text{Air}} = f(W)$
Function Name:

PsiAir_W_HAP_SI

Fortran Program:

REAL*8 FUNCTION PSIAIR_W_HUAIRPROP(W), REAL*8 W

Input Values: W - Humidity ratio W in kg_w/kg_a **Result:**PsiAir_W_HAP_SI - Mole fraction of (dry) air in humid air in mol_a/mol **Range of Validity:**Humidity ratio W : $0 \leq W \leq 10 \text{ kg}_w/\text{kg}_a$ **Comments:**

- Mole fraction of air $\psi_{\text{Air}} = 1 - \psi_{\text{H}_2\text{O}} = 1 - \left(\frac{W}{\frac{R_a}{R_{\text{H}_2\text{O}}} + W} \right)$

Result for Wrong Input Values:

PsiAir_W_HAP_SI = -1000

References: $\psi_{\text{Air}}(W)$ Herrmann et al. [1], [2]

Mole Fraction of Water $\psi_{H_2O} = f(W)$
Function Name:

PsiH2O_W_HAP_SI

Fortran Program:

REAL*8 FUNCTION PSIH2O_W_HUAIRPROP(W), REAL*8 W

Input Values: W - Humidity ratio W in kg_w/kg_a **Result:**

PsiH2O_W_HAP_SI - Mole fraction of water in humid air in molw/mol

Range of Validity:Humidity ratio W : $0 \leq W \leq 10 kg_w/kg_a$ **Comments:**

- Mole fraction of water $\psi_{H_2O} = \frac{W}{\frac{R_a}{R_{H_2O}} + W}$

Result for Wrong Input Values:

PsiH2O_W_HAP_SI = -1000

References: $\psi_{H_2O}(W)$ Herrmann et al. [1], [2]

Density $\rho = f(p, t, W)$
Function Name:

Rho_ptW_HAP_SI

Fortran Program:

```
REAL*8 FUNCTION RHO_PTW_HUAIRPROP(P,T,W), REAL*8 P,T,W
```

Input Values:

p - Total pressure p in kPa
 t - Temperature t in °C
 W - Humidity ratio W in kg_w/kg_a

Result:Rho_ptW_HAP_SI - Density of humid air in kg/m^3 **Range of Validity:**

Temperature t : from -143.15°C to 350°C
 Total pressure p : from 0.01 kPa to 10 000 kPa
 Humidity ratio W : $0 \leq W \leq 10 \text{kg}_w/\text{kg}_a$

Comments:

- Density of humid air obtained from air-specific volume: $\rho = \frac{1+W}{v}$

Result for Wrong Input Values:

Rho_ptW_HAP_SI = -1000

References:

$\rho(p, t, W)$ Herrmann et al. [1], [2]

Air-Specific Entropy $s = f(p, t, W)$ **Function Name:**

s_ptW_HAP_SI

Fortran Program:

REAL*8 FUNCTION S_PTW_HUAIRPROP(P,T,W), REAL*8 P,T,W

Input Values:

p - Total pressure p in kPa
 t - Temperature t in °C
 W - Humidity ratio W in kg_w/kg_a

Result:s_ptW_HAP_SI - Air-specific entropy in $\text{kJ}/(\text{kg}_a \text{K})$ **Range of Validity:**

Temperature t : from -143.15°C to 350°C
 Total pressure p : from 0.01 kPa to 10 000 kPa
 Humidity ratio W : $0 \leq W \leq 10 \text{kg}_w/\text{kg}_a$

Comments:

- When calculating supersaturated air an ideal mixing model (saturated humid air and water droplets (or ice crystals) homogeneously mixed) is applied

Result for Wrong Input Values:

s_ptW_HAP_SI = -1000

References: $s(p, t, W)$ Herrmann et al. [1], [2]

Backward Function: Temperature $t = f(p, h, \varphi)$
Function Name:

t_phphi_HAP_SI

Fortran Program:

```
REAL*8 FUNCTION T_PHPHI_HUAIRPROP(P,H,PHI), REAL*8 P,H,PHI
```

Input Values:

- p - Total pressure p in kPa
- h - Air-specific enthalpy h in kJ/kg_a
- φ - Relative humidity φ (decimal ratio)

Result:

t_phphi_HAP_SI - Temperature from pressure, enthalpy, and relative humidity in °C

Range of Validity:

- Total pressure p : from 0.01 kPa to 10 000 kPa
- Air-specific enthalpy h : from -5745 kJ/kg_a to 29690 kJ/kg_a
- Relative humidity φ : $0 \leq \varphi \leq 1$

Comments:

- Iteration of temperature t from $h = f(p, t, W)$ using $W = f(p, t, \varphi)$

Result for Wrong Input Values:

t_phphi_HAP_SI = -1000

References: $h(p, t, W)$ Herrmann et al. [1], [2]

Backward Function: Temperature $t = f(p, h, W)$ **Function Name:**

t_phW_HAP_SI

Fortran Program:

REAL*8 FUNCTION T_PHW_HUAIRPROP(P,H,W), REAL*8 P,H,W

Input Values:

p - Total pressure p in kPa
 h - Air-specific enthalpy h in kJ/kg_a
 W - Humidity ratio W in kg_w/kg_a

Result:

t_phW_HAP_SI - Temperature from pressure, enthalpy, and humidity ratio in °C

Range of Validity:

Total pressure p : from 0.01 kPa to 10 000 kPa
 Air-specific enthalpy h : from -5745 kJ/kg_a to 29690 kJ/kg_a
 Humidity ratio W : $0 \leq W \leq 10$ kg_w/kg_a

Comments:- Iteration of temperature t from $h = f(p, t, W)$ **Result for Wrong Input Values:**

t_phW_HAP_SI = -1000

References: $h(p, t, W)$ Herrmann et al. [1], [2]

Backward Function: Temperature $t = f(p, s, W)$
Function Name:

t_psW_HAP_SI

Fortran Program:

```
REAL*8 FUNCTION T_PSW_HUAIRPROP(P,S,W), REAL*8 P,S,W
```

Input Values:

p - Total pressure p in kPa
 s - Air-specific entropy s in kJ/(kg_a K)
 W - Humidity ratio W in kg_w/kg_a

Result:

t_psW_HAP_SI - Temperature from pressure, entropy, and humidity ratio in °C

Range of Validity:

Total pressure p : from 0.01 kPa to 10 000 kPa
 Air-specific entropy s : from -26.53 kJ/(kg_a K) to 38.990 kJ/(kg_a K)
 Humidity ratio W : $0 \leq W \leq 10 \text{ kg}_w/\text{kg}_a$

Comments:

- Iteration of temperature t from $s = f(p, t, W)$

Result for Wrong Input Values:

t_psW_HAP_SI = -1000

References:

$s(p, t, W)$ Herrmann et al. [1], [2]

Backward Function: Temperature $t = f(p, t_{wb}, W)$
Function Name:

t_ptwbW_HAP_SI

Fortran Program:

REAL*8 FUNCTION T_PTWBW_HUAIRPROP(P,TWB,W), REAL*8 P,TWB,W

Input Values:

p - Total pressure p in kPa
 t_{wb} - Wet-bulb temperature in °C
 W - Humidity ratio W in kg_w/kg_a

Result:

t_ptwbW_HAP_SI - Temperature from pressure, wet bulb temperature and humidity ratio in °C

Range of Validity:

Total pressure p : from 0.01 kPa to 10 000 kPa
Wet bulb temperature t_{wb} : from -143.15°C to 350°C
Humidity ratio W : $0 \leq W \leq 10 \text{ kg}_w/\text{kg}_a$

Comments:

- Iteration of temperature t from $t_{wb} = f(p, t, W)$

Result for Wrong Input Values:

t_ptwbW_HAP_SI = -1000

References:

$t_{wb}(p, t, W)$ Herrmann et al. [1], [2]

Dew-Point/Frost-Point Temperature $t_d = f(p, W)$
Function Name:

td_pW_HAP_SI

Fortran Program:

REAL*8 FUNCTION TD_PW_HUAIRPROP(P,W), REAL*8 P,W

Input Values:

p - Total pressure p in kPa
 W - Humidity ratio W in kg_w/kg_a

Result:

td_pW_HAP_SI - Dew-point/frost-point temperature in °C

Range of Validity:

Total pressure p : from 0.01 kPa to 10 000 kPa
 Humidity ratio W : $0 \leq W \leq 10 \text{ kg}_w/\text{kg}_a$

Comments:

Dew-point temperature $t_d = t_s(\rho_{\text{H}_2\text{O}})$ for $t \geq 0.01^\circ\text{C}$ (saturation temperature of water in humid air)

$t_d = t_{\text{sub}}(\rho_{\text{H}_2\text{O}})$ for $t \leq 0.01^\circ\text{C}$ (sublimation temperature of water in humid air)

Result for Wrong Input Values:

td_pW_HAP_SI = -1000

References:

$t_s(\rho_{\text{H}_2\text{O}})$ for $t_d \geq 0.01^\circ\text{C}$ IAPWS-IF97 [7], [8]

$t_{\text{sub}}(\rho_{\text{H}_2\text{O}})$ for $t_d \leq 0.01^\circ\text{C}$ IAPWS-08 [11]

$\rho_{\text{H}_2\text{O}}$ Herrmann et. al. [1], [2]

Saturation Temperature $t_s = f(p, p_{H_2O})$

Function Name:

ts_ppH2O_HAP_SI

Fortran Program:

```
REAL*8 FUNCTION TS_PPH2O_HUAIRPROP(P,PH2O), REAL*8 P,PH2O
```

Input Values:

p - Total pressure p in kPa
 p_{H_2O} - Partial pressure of water vapor p_{H_2O} in kPa

Result:

ts_ppH2O_HAP_SI - Saturation temperature in °C

Range of Validity:

Total pressure p : from 0.01 kPa to 10 000 kPa
 Partial Pressure p_{H_2O} : from 0.01 kPa to 10 000 kPa

Comments:

- Iteration of saturation temperature t_s from $p_{H_2O,s} = f(p, t)$

Result for Wrong Input Values:

ts_ppH2O_HAP_SI = -1000

References:

$p_{H_2O,s}$ Herrmann et. al. [1], [2]

Wet-Bulb/Ice-Bulb Temperature $t_{wb} = f(p, t, W)$
Function Name:

twb_ptW_HAP_SI

Fortran Program:

```
REAL*8 FUNCTION TWB_PTW_HUAIRPROP(P,T,W), REAL*8 P,T,W
```

Input Values:

p - Total pressure p in kPa
 t - Temperature t in °C
 W - Humidity ratio W in kg_w/kg_a

Result:

twb_ptW_HAP_SI - Wet-bulb/ice-bulb temperature in °C

Range of Validity:

Temperature t : from -143.15°C to 350°C
 Total pressure p : from 0.01 kPa to 10 000 kPa
 Humidity ratio W : $0 \leq W \leq 10 \text{ kg}_w/\text{kg}_a$

Comments:

- Iteration of wet-bulb/ice-bulb temperature t_{wb}
 from $h^{\text{unsaturated}}(p, t, W) = h^{\text{fog}}(p, t_{wb}, W)$
- Calculation for supersaturated humid air ($W > W_s$) is not possible

Result for Wrong Input Values:

twb_ptW_HAP_SI = -1000

References: $t_{wb}(p, t, W)$ Herrmann et al. [1], [2]

Air-Specific Internal Energy $u = f(p, t, W)$ **Function Name:**

u_ptW_HAP_SI

Fortran Program:

REAL*8 FUNCTION U_PTW_HUAIRPROP(P,T,W), REAL*8 P,T,W

Input Values:

p - Total pressure p in kPa
 t - Temperature t in °C
 W - Humidity ratio W in kg_w/kg_a

Result:u_ptW_HAP_SI - Air-specific internal energy in kJ/kg_a**Range of Validity:**

Temperature t : from -143.15°C to 350°C
 Total pressure p : from 0.01 kPa to 10 000 kPa
 Humidity ratio W : $0 \leq W \leq 10$ kg_w/kg_a

Comments:- Internal energy $u = h - pv$ **Result for Wrong Input Values:**

u_ptW_HAP_SI = -1000

References: $u(p, t, W)$ Herrmann et al. [1], [2]

Air-Specific Volume $v = f(p, t, W)$
Function Name:

v_ptW_HAP_SI

Fortran Program:

```
REAL*8 FUNCTION V_PTW_HUAIRPROP(P,T,W), REAL*8 P,T,W
```

Input Values:

p - Total pressure p in kPa
 t - Temperature t in °C
 W - Humidity ratio W in kg_w/kg_a

Result:

v_ptW_HAP_SI - Air-specific volume in m^3/kg_a

Range of Validity:

Temperature t : from -143.15°C to 350°C
 Total pressure p : from 0.01 kPa to 10 000 kPa
 Humidity ratio W : $0 \leq W \leq 10 \text{kg}_w/\text{kg}_a$

Comments:

- When calculating supersaturated air an ideal mixing model (saturated humid air and water droplets (or ice crystals) homogeneously mixed) is applied

Result for Wrong Input Values:

v_ptW_HAP_SI = -1000

References:

$v(p, t, W)$ Herrmann et al. [1], [2]

Humidity Ratio from Partial Pressure of Steam $W = f(p, t, p_{H_2O})$
Function Name:

W_ptpH2O_HAP_SI

Fortran Program:

```
REAL*8 FUNCTION W_PTPH2O_HUAIRPROP(P,T,PH2O), REAL*8 P,T,PH2O
```

Input Values:

- p - Total pressure p in kPa
- t - Temperature t in °C
- p_{H_2O} - Partial pressure of water p_{H_2O} in kPa

Result:

W_ptpH2O_HAP_SI - Humidity ratio from temperature and partial pressure of water vapor in kg_w/kg_a

Range of Validity:

- Total pressure p : from 0.01 kPa to 10 000 kPa
- Temperature t : from -143.15°C to 350°C
- Partial pressure p_{H_2O} : from 0.01 kPa to 10 000 kPa

Comments:

- Iteration of humidity ratio W from $p_{H_2O} = f(p, t, W)$
- Result for supersaturated humid air is W_s

Result for Wrong Input Values:

W_ptpH2O_HAP_SI = -1000

References:

$p_{H_2O}(p, t, W)$ Herrmann et al. [1], [2]

Humidity Ratio from Relative Humidity $W = f(p, t, \varphi)$

Function Name:

`W_ptphi_HAP_SI`

Fortran Program:

```
REAL*8 FUNCTION W_PTPHI_HUAIRPROP(P,T,PHI), REAL*8 P,T,PHI
```

Input Values:

p - Total pressure p in kPa
 t - Temperature t in °C
 φ - Relative humidity (decimal ratio)

Result:

`W_ptphi_HAP_SI` - Humidity ratio from temperature and relative humidity
in kg_w/kg_a

Range of Validity:

Temperature t : from -143.15°C to 350°C
Total pressure p : from 0.01 kPa to 10 000 kPa
Relative humidity φ : $0 \leq \varphi \leq 1$

Comments:

- Iteration of humidity ratio W from $\varphi = f(p, t, W)$

Result for Wrong Input Values:

`W_ptphi_HAP_SI` = -1000

References:

$\varphi(p, t, W)$ Herrmann et al. [1], [2]

Humidity Ratio from Dew-Point Temperature $W = f(p, t_d)$

Function Name:

W_ptd_HAP_SI

Fortran Program:

REAL*8 FUNCTION W_PTD_HUAIRPROP(P,TD), REAL*8 P,TD

Input Values:

p - Total pressure p in kPa
 t_d - Dew-point temperature t_d in °C

Result:

W_ptd_HAP_SI - Humidity ratio from temperature and dew-point temperature
in kg_w/kg_a

Range of Validity:

Dew point temperature t_d : from -143.15°C to 350°C
Total pressure p : from 0.01 kPa to 10 000 kPa

Comments:

- Iteration of humidity ratio W from $t_d = f(p, W)$

Result for Wrong Input Values:

W_ptd_HAP_SI = -1000

References:

$t_d(p, W)$ Herrmann et al. [1], [2]

Humidity Ratio from Wet-Bulb Temperature $W = f(p, t, t_{wb})$

Function Name:

W_pttwb_HAP_SI

Fortran Program:

```
REAL*8 FUNCTION W_PTTWB_HUAIRPROP(P,T,TWB), REAL*8 P,T,TWB
```

Input Values:

p - Total pressure p in kPa
 t - Temperature t in °C
 t_{wb} - Wet-bulb temperature in °C

Result:

W_pttwb_HAP_SI - Humidity ratio from temperature and wet-bulb temperature
in kg_w/kg_a

Range of Validity:

Total pressure p : from 0.01 kPa to 10 000 kPa
Temperature t : from -143.15°C to 350°C
Wet-bulb temperature t_{wb} : from -143.15°C to 350°C

Comments:

- Iteration of humidity ratio W from $t_{wb} = f(p, t, W)$
- Calculation for supersaturated humid air ($W > W_s$) is not possible

Result for Wrong Input Values:

W_pttwb_HAP_SI = -1000

References:

$t_{wb}(p, t, W)$ Herrmann et al. [1], [2]

Saturation Humidity Ratio $W_s = f(p, t)$

Function Name:

Ws_pt_HAP_SI

Fortran Program:

REAL*8 FUNCTION WS_PT_HUAIRPROP(P,T), REAL*8 P,T

Input Values: p - Total pressure p in kPa t - Temperature t in °C**Result:**Ws_pt_HAP_SI - Saturation humidity ratio (mass fraction) in kg_w/kg_a **Range of Validity:**Total pressure p : from 0.01 kPa to 10 000 kPaTemperature t : from -143.15°C to 350°C**Comments:**

- Calculation of saturation humidity ratio W_s from $W_s = \frac{M_{\text{H}_2\text{O}}}{M_a} \frac{p_{\text{H}_2\text{O},s}}{(p - p_{\text{H}_2\text{O},s})}$

Result for Wrong Input Values:

Ws_pt_HAP_SI = -1000

References: $p_{\text{H}_2\text{O},s}$ Herrmann et al. [1], [2]

Mass Fraction of Dry Air $\xi_{\text{Air}} = f(W)$
Function Name:

XiAir_W_HAP_SI

Fortran Program:

```
REAL*8 FUNCTION XIAIR_W_HUAIRPROP(W), REAL*8 W
```

Input Values:

W - Humidity ratio W in kg_w/kg_a

Result:

XiAir_W_HAP_SI - Mass fraction of (dry) air in humid air in kg_a/kg

Range of Validity:

Humidity ratio W : $0 \leq W \leq 10 \text{kg}_w/\text{kg}_a$

Comments:

- Mass fraction of (dry) air $\xi_{\text{Air}} = 1 - \xi_{\text{H}_2\text{O}} = 1 - \frac{W}{1 + W}$

Result for Wrong Input Values:

XiAir_W_HAP_SI = -1000

References:

$\xi_{\text{Air}}(W)$ Herrmann et al. [1], [2]

Mass Fraction of Water Vapor $\xi_{\text{H}_2\text{O}} = f(W)$
Function Name:

XiH2O_W_HAP_SI

Fortran Program:

REAL*8 FUNCTION XIH2O_W_HUAIRPROP(W), REAL*8 W

Input Values: W - Humidity ratio W in kg_w/kg_a **Result:**XiH2O_W_HAP_SI - Mass fraction of water vapor in humid air in kg_w/kg **Range of Validity:**Humidity ratio W : $0 \leq W \leq 10 \text{ kg}_w/\text{kg}_a$ **Comments:**- Mass fraction of water vapor $\xi_{\text{H}_2\text{O}} = \frac{W}{1+W}$ **Result for Wrong Input Values:**

XiH2O_W_HAP_SI = -1000

References: $\xi_{\text{H}_2\text{O}}(W)$ Herrmann et al. [1], [2]

Compression Factor $Z = f(p, t, W)$

Function Name:

Z_ptW_HAP_SI

Fortran Program:

```
REAL*8 FUNCTION Z_PTW_HUAIRPROP(P,T,W), REAL*8 P,T,W
```

Input Values:

p - Total pressure p in kPa
 t - Temperature t in °C
 W - Humidity ratio W in kg_w/kg_a

Result:

Z_ptW_HAP_SI - Compression factor (decimal ratio)

Range of Validity:

Total pressure p : from 0.01 kPa to 10 000 kPa
 Temperature t : from -143.15°C to 350°C
 Humidity ratio W : $0 \leq W \leq W_s$

Comments:

- Compression factor $Z = 1 + \frac{B_m}{\bar{v}} + \frac{C_m}{\bar{v}^2}$

$$\text{with } \bar{v} = \frac{M}{\rho} = \frac{Mv}{1+W}$$

and M is the molar mass of humid air

- Calculation for supersaturated humid air ($W > W_s$) is not possible

Result for Wrong Input Values:

Z_ptW_HAP_SI = -1000

References:

$B_m(t, W), C_m(t, W)$ Herrmann et al. [1], [2]

$\rho(p, t, W), v(p, t, W)$ Herrmann et al. [1], [2]

3.2 Functions for Steam and Water for Temperatures $t \geq 0^\circ\text{C}$

Specific Enthalpy of Liquid Water $h_{\text{liq}} = f(p, t)$
Function Name:

hliq_pt_97_SI

Fortran Program:

REAL*8 FUNCTION HLIQ_PT_97(P,T), REAL*8 P,T

Input Values:

p - Pressure p in kPa
 t - Temperature t in °C

Result:

hliq_pt_97_SI - Specific enthalpy of liquid water in kJ/kg

Range of Validity:

Pressure p : from $p_s(0^\circ\text{C}) = 0.6112$ kPa to 10000 kPa
 Temperature t : from 0°C to 350°C

Comments:

- Specific enthalpy of liquid water $h_{\text{liq}} = h^{97}(p, t)$ (Region 1)

Result for Wrong Input Values:

hliq_pt_97_SI = -1000

References:

$h^{97}(p, t)$ IAPWS-IF97 [7], [8]

Specific Enthalpy of Saturated Liquid Water $h_{\text{liq,s}} = f(t)$
Function Name:

hliqs_t_97_SI

Fortran Program:

REAL*8 FUNCTION HLIQS_T_97(T), REAL*8 T

Input Values: t - Temperature t in °C**Result:**

hliqs_t_97_SI - Specific enthalpy of saturated liquid water in kJ/kg

Range of Validity:Temperature t : from 0°C to 350°C**Comments:**- Specific enthalpy of liquid water $h_{\text{liq,s}} = h^{97}(p_s, t)$ (Region 1)with $p_s = p_s^{97}(t)$ **Result for Wrong Input Values:**

hliqs_t_97_SI = -1000

References: $h^{97}(p, t), p_s^{97}(t)$ IAPWS-IF97 [7], [8]

Specific Enthalpy of Saturated Water Vapor $h_{\text{vap},s} = f(t)$
Function Name:

hvaps_t_97_SI

Fortran Program:

```
REAL*8 FUNCTION HVAPS_T_97(T), REAL*8 T
```

Input Values:

t - Temperature t in °C

Result:

hvaps_t_97_SI - Specific enthalpy of saturated water vapor in kJ/kg

Range of Validity:

Temperature t : from 0°C to 350°C

Comments:

- Specific enthalpy of saturated water vapor $h_{\text{vap},s} = h^{97}(p_s, t)$ (Region 2)

with $p_s = p_s^{97}(t)$

Result for Wrong Input Values:

hvaps_t_97_SI = -1000

References:

$h^{97}(p, t), p_s^{97}(t)$ IAPWS-IF97 [7], [8]

Saturation Pressure of Water $p_s = f(t)$ **Function Name:**

ps_t_97_SI

Fortran Program:

REAL*8 FUNCTION PS_T_97(T), REAL*8 T

Input Values: t - Temperature t in °C**Result:**

ps_t_97_SI - Saturation pressure of water in kPa

Range of Validity:Temperature t : from 0°C to 350°C**Comments:**- Saturation pressure of water $p_s = p_s^{97}(t)$ (Region 4)**Result for Wrong Input Values:**

ps_t_97_SI -1000

References: $p_s^{97}(t)$ IAPWS-IF97 [7], [8]

Specific Entropy of Liquid Water $s_{\text{liq}} = f(p, t)$

Function Name:

sliq_pt_97_SI

Fortran Program:

```
REAL*8 FUNCTION SLIQ_PT_97(P,T), REAL*8 P,T
```

Input Values:

p - Pressure p in kPa
 t - Temperature t in °C

Result:

sliq_pt_97_SI - Specific entropy of liquid water in kJ/(kg K)

Range of Validity:

Pressure p : from $p_s(0^\circ\text{C}) = 0.6112$ kPa to 10000 kPa
 Temperature t : from 0°C to 350°C

Comments:

- Specific entropy of liquid water $s_{\text{liq}} = s^{97}(p, t)$ (Region 1)

Result for Wrong Input Values:

sliq_pt_97_SI = -1000

References:

$s^{97}(p, t)$ IAPWS-IF97 [7], [8]

Specific Entropy of Saturated Liquid Water $s_{\text{liq,s}} = f(t)$
Function Name:

sliqs_t_97_SI

Fortran Program:

REAL*8 FUNCTION SLIQS_T_97(T), REAL*8 T

Input Values: t - Temperature t in °C**Result:**

sliqs_t_97_SI - Specific entropy of saturated liquid water in kJ/(kg K)

Range of Validity:Temperature t : from 0°C to 350°C**Comments:**- Specific entropy of liquid water $s_{\text{liq,s}} = s^{97}(p_s, t)$ (Region 1)with $p_s = p_s^{97}(t)$ **Result for Wrong Input Values:**

sliqs_t_97_SI = -1000

References: $s^{97}(p, t), p_s^{97}(t)$ IAPWS-IF97 [7], [8]

Specific Entropy of Saturated Water Vapor $s_{\text{vap},s} = f(t)$

Function Name:

svaps_t_97_SI

Fortran Program:

REAL*8 FUNCTION SVAPS_T_97(T), REAL*8 T

Input Values: t - Temperature t in °C**Result:**

svaps_t_97_SI - Specific entropy of saturated water vapor in kJ/(kg K)

Range of Validity:Temperature t : from 0°C to 350°C**Comments:**

- Specific entropy of saturated water vapor $s_{\text{vap},s} = s^{97}(p_s, t)$ (Region 2)
 with $p_s = p_s^{97}(t)$

Result for Wrong Input Values:

svaps_t_97_SI = -1000

References: $s^{97}(p, t), p_s^{97}(t)$ IAPWS-IF97 [7], [8]

Saturation Temperature of Water $t_s = f(p)$

Function Name:

ts_p_97_SI

Fortran Program:

REAL*8 FUNCTION TS_P_97(P), REAL*8 P

Input Values: p - Pressure p in kPa**Result:**

ts_p_97_SI - Saturation temperature of water in °C

Range of Validity:Pressure p : from 0.6112 kPa to 10 000 kPa**Comments:**- Saturation temperature of water $t_s = t_s^{97}(p)$ (Region 4)**Result for Wrong Input Values:**

ts_p_97_SI = -1000

References: $t_s^{97}(p)$ IAPWS-IF97 [7], [8]

Specific Volume of Liquid Water $v_{\text{liq}} = f(p, t)$

Function Name:

vliq_pt_97_SI

Fortran Program:

```
REAL*8 FUNCTION VLIQ_PT_97(P,T), REAL*8 P,T
```

Input Values:

p - Pressure p in kPa
 t - Temperature t in °C

Result:

vliq_pt_97_SI - Specific volume of liquid water in m³/kg

Range of Validity:

Pressure p : from $p_s(0^\circ\text{C}) = 0.6112$ kPa to 10 000 kPa
 Temperature t : from 0°C to 350°C

Comments:

- Specific volume of liquid water $v_{\text{liq}} = v^{97}(p, t)$ (Region 1)

Result for Wrong Input Values:

vliq_pt_97_SI = -1000

References:

$v^{97}(p, t)$ IAPWS-IF97 [7], [8]

Specific Volume of Saturated Liquid Water $v_{\text{liq,s}} = f(t)$
Function Name:

vliqs_t_97_SI

Fortran Program:

REAL*8 FUNCTION VLIQS_T_97(T), REAL*8 T

Input Values: t - Temperature t in °C**Result:**vliqs_t_97_SI - Specific volume of saturated liquid water in m^3/kg **Range of Validity:**Temperature t : from 0°C to 350°C**Comments:**- Specific volume of liquid water $v_{\text{liq,s}} = v^{97}(\rho_s, t)$ (Region 1)with $\rho_s = \rho_s^{97}(t)$ **Result for Wrong Input Values:**

vliqs_t_97_SI = -1000

References: $v^{97}(\rho, t), \rho_s^{97}(t)$ IAPWS-IF97 [7], [8]

Specific Volume of Saturated Water Vapor $v_{\text{vap},s} = f(t)$

Function Name:

vvaps_t_97_SI

Fortran Program:

```
REAL*8 FUNCTION  VVAPS_T_97(T), REAL*8 T
```

Input Values:

t - Temperature t in °C

Result:

vvaps_t_97_SI - Specific volume of saturated water vapor in m^3/kg

Range of Validity:

Temperature t : from 0°C to 350°C

Comments:

- Specific volume of saturated water vapor $v_{\text{vap},s} = v^{97}(p_s, t)$ (Region 2)

with $p_s = p_s^{97}(t)$

Result for Wrong Input Values:

vvaps_t_97_SI = -1000

References:

$v^{97}(p, t), p_s^{97}(t)$ IAPWS-IF97 [7], [8]

3.3 Functions for Steam and Ice for Temperatures $t \leq 0^\circ\text{C}$

Specific Enthalpy of Saturated Ice $h_{\text{ice,sub}} = f(t)$
Function Name:

hicesub_t_06_SI

Fortran Program:

```
REAL*8 FUNCTION HICESUB_T_06(T), REAL*8 T
```

Input Values:

t - Temperature t in °C

Result:

hicesub_t_06_SI - Specific enthalpy of saturated ice in kJ/kg

Range of Validity:

Temperature t : from -143.15°C to 0°C

Comments:

- Specific enthalpy of saturated ice $h_{\text{ice,sub}} = h^{06}(\rho_{\text{sub}}, t)$
 with $\rho_{\text{sub}} = \rho_{\text{sub}}^{08}(t)$

Result for Wrong Input Values:

hicesub_t_06_SI = -1000

References:

$h^{06}(\rho, t)$ IAPWS-06 [10]

$\rho_{\text{sub}}^{08}(t)$ IAPWS-08 [11]

Specific Enthalpy of Saturated Water Vapor $h_{\text{vap,sub}} = f(t)$
Function Name:

hvapsub_t_95_SI

Fortran Program:

```
REAL*8 FUNCTION HVAPSUB_T_95(T), REAL*8 T
```

Input Values:

t - Temperature t in °C

Result:

hvapsub_t_95_SI - Specific enthalpy of saturated water vapor in kJ/kg

Range of Validity:

Temperature t : from -143.15°C to 0°C

Comments:

- Specific enthalpy of saturated water vapor $h_{\text{vap,sub}} = h^{95}(p_{\text{sub}}, t)$

with $p_{\text{sub}} = p_{\text{sub}}^{08}(t)$

Result for Wrong Input Values:

hvapsub_t_95_SI = -1000

References:

$h^{95}(p, t)$ IAPWS-95 [5], [6]

$p_{\text{sub}}^{08}(t)$ IAPWS-08 [11]

Melting Pressure $p_{\text{mel}} = f(t)$
Function Name:

pmel_t_08_SI

Fortran Program:

```
REAL*8 FUNCTION PMEL_T_08 (T), REAL*8 T
```

Input Values:

t - Temperature t in °C

Result:

pmel_t_08_SI - Melting pressure of ice in kPa

Range of Validity:

Temperature t : from -21.985°C to 0°C

Result for Wrong Input Values:

pmel_t_08_SI = -1000

References:

$p_{\text{mel}}^{08}(t)$ IAPWS-08 [11]

Sublimation Pressure $p_{\text{sub}} = f(t)$

Function Name:

psub_t_08_SI

Fortran Program:

```
REAL*8 FUNCTION PSUB_T_08 (T), REAL*8 T
```

Input Values:

t - Temperature t in °C

Result:

psub_t_08_SI - Sublimation pressure of ice in kPa

Range of Validity:

Temperature t : from -143.15°C to 0°C

Result for Wrong Input Values:

psub_t_08_SI = -1000

References:

$p_{\text{sub}}^{08}(t)$ IAPWS-08 [11]

Specific Entropy of Saturated Ice $s_{\text{ice,sub}} = f(t)$

Function Name:

sicesub_t_06_SI

Fortran Program:

```
REAL*8 FUNCTION SICESUB_T_06(T), REAL*8 T
```

Input Values:

t - Temperature t in °C

Result:

sicesub_t_06_SI - Specific entropy of saturated ice in kJ/(kg K)

Range of Validity:

Temperature t : from -143.15°C to 0°C

Comments:

- Specific entropy of saturated ice $s_{\text{ice,sub}} = s^{06}(p_{\text{sub}}, t)$
 with $p_{\text{sub}} = p_{\text{sub}}^{08}(t)$

Result for Wrong Input Values:

sicesub_t_06_SI = -1000

References:

$s^{06}(p, t)$ IAPWS-06 [10]

$p_{\text{sub}}^{08}(t)$ IAPWS-08 [11]

Specific Entropy of Saturated Water Vapor $s_{\text{vap,sub}} = f(t)$
Function Name:

svapsub_t_95_SI

Fortran Program:

REAL*8 FUNCTION SVAPSUB_T_95(T), REAL*8 T

Input Values: t - Temperature t in °C**Result:**

svapsub_t_95_SI - Specific entropy of saturated water vapor in kJ/(kg K)

Range of Validity:Temperature t : from -143.15°C to 0°C**Comments:**- Specific entropy of saturated water vapor $s_{\text{vap,sub}} = s^{95}(p_{\text{sub}}, t)$ with $p_{\text{sub}} = p_{\text{sub}}^{08}(t)$ **Result for Wrong Input Values:**

svapsub_t_95_SI = -1000

References: $s^{95}(p, t)$ IAPWS-95 [7], [8] $p_{\text{sub}}^{08}(t)$ IAPWS-08 [11]

Melting Temperature $t_{\text{mel}} = f(p)$
Function Name:

tmel_p_08_SI

Fortran Program:

```
REAL*8 FUNCTION TMEL_P_08(P), REAL*8 P
```

Input Values:

p - Pressure p in kPa

Result:

tmel_p_08_SI - Melting temperature of ice in °C

Range of Validity:

Pressure p : from $p_s(0^\circ\text{C}) = 0.6112$ kPa to 10 000 kPa

Result for Wrong Input Values:

tmel_p_08_SI = -1000

References:

$t_{\text{mel}}^{08}(p)$ IAPWS-08 [11]

Sublimation Temperature $t_{\text{sub}} = f(p)$

Function Name:

tsub_p_08_SI

Fortran Program:

```
REAL*8 FUNCTION TSUB_P_08(P), REAL*8 P
```

Input Values:

p - Pressure p in kPa

Result:

tsub_p_08_SI - Sublimation temperature of ice in °C

Range of Validity:

Pressure p : from $p_{\text{subl}}(-143.15^\circ\text{C}) = 1.2002 \times 10^{-11}$ kPa to $p_{\text{subl}}(0^\circ\text{C}) = 0.6112$ kPa

Result for Wrong Input Values:

tsub_p_08_SI = -1000

References:

$t_{\text{sub}}^{08}(p)$ IAPWS-08 [11]

Specific Volume of Saturated Ice $v_{\text{ice,sub}} = f(t)$
Function Name:

vicesub_t_06_SI

Fortran Program:

REAL*8 FUNCTION VICESUB_T_06(T), REAL*8 T

Input Values: t - Temperature t in °C**Result:**vicesub_t_06_SI - Specific volume of saturated ice in m^3/kg **Range of Validity:**Temperature t : from -143.15°C to 0°C **Comments:**- Specific volume of saturated ice $v_{\text{ice,sub}} = v^{06}(\rho_{\text{sub}}, t)$ with $\rho_{\text{sub}} = \rho_{\text{sub}}^{08}(t)$ **Result for Wrong Input Values:**

vicesub_t_06_SI = -1000

References: $v^{06}(\rho, t)$ IAPWS-06 [10] $\rho_{\text{sub}}^{08}(t)$ IAPWS-08 [11]

Specific Volume of Saturated Water Vapor $v_{\text{vap,sub}} = f(t)$
Function Name:

vvapsub_t_95_SI

Fortran Program:

```
REAL*8 FUNCTION  VVAPSUB_T_95(T), REAL*8 T
```

Input Values:

t - Temperature t in °C

Result:

vvapsub_t_95_SI - Specific volume of saturated water vapor in m³/kg

Range of Validity:

Temperature t : from -143.15°C to 0°C

Comments:

- Specific volume of saturated water vapor $v_{\text{vap,sub}} = v^{95}(\rho_{\text{sub}}, t)$

with $\rho_{\text{sub}} = \rho_{\text{sub}}^{08}(t)$

Result for Wrong Input Values:

vvapsub_t_95_SI = -1000

References:

$v^{95}(\rho, t)$ IAPWS-95 [7], [8]

$\rho_{\text{sub}}^{08}(t)$ IAPWS-08 [11]

4. Property Libraries for Calculating Heat Cycles, Boilers, Turbines and Refrigerators

Water and Steam

Library LibIF97

- Industrial Formulation IAPWS-IF97 (Revision 2007)
- Supplementary Standards IAPWS-IF97-S01, -S03rev, -S04, and -S05
- IAPWS Revised Advisory Note No. 3 on Thermodynamic Derivatives (2008)

Library LibIF97_META

- Industrial Formulation IAPWS-IF97 (Revision 2007) for metastable steam

Humid Combustion Gas Mixtures

Library LibHuGas

- Model: Ideal mixture of the real fluids:
 CO₂ - Span, Wagner H₂O - IAPWS-95
 O₂ - Schmidt, Wagner N₂ - Span et al.
 Ar - Tegeler et al.
 and of the ideal gases:
 SO₂, CO, Ne
 (Scientific Formulation of Bückler et al.)
 Consideration of:
- Dissociation from VDI 4670
 - Poynting effect

Humid Air

Library LibHuAir

- Model: Ideal mixture of the real fluids:
- Dry air from Lemmon et al.
 - Steam, water and ice from IAPWS-IF97 and IAPWS-06
- Consideration of:
- Condensation and freezing of steam
 - Dissociation from VDI 4670
 - Poynting effect from ASHRAE RP-1485

Extremely Fast Property Calculations

- Spline-Based Table
 Look-up Method (SBTL)
Library LibSBTL_IF97
Library LibSBTL_95
Library LibSBTL_HuAir
 For steam, water, humid air, carbon dioxide and other fluids and mixtures according IAPWS Guideline 2015 for Computational Fluid Dynamics (CFD), real-time and non-stationary simulations

Carbon Dioxide Including Dry Ice

Library LibCO2

Formulation of Span and Wagner (1996)

Seawater

Library LibSeaWa

IAPWS Industrial Formulation 2013

Ice

Library LibICE

Ice from IAPWS-06, Melting and sublimation pressures from IAPWS-08, Water from IAPWS-IF97, Steam from IAPWS-95 and -IF97

Ideal Gas Mixtures

Library LibIdGasMix

Model: Ideal mixture of the ideal gases:

Ar	NO	He	Propylene
Ne	H ₂ O	F ₂	Propane
N ₂	SO ₂	NH ₃	Iso-Butane
O ₂	H ₂	Methane	n-Butane
CO	H ₂ S	Ethane	Benzene
CO ₂	OH	Ethylene	Methanol
Air			

Consideration of:

- Dissociation from the VDI Guideline 4670

Library LibIDGAS

Model: Ideal gas mixture from VDI Guideline 4670

Consideration of:

- Dissociation from the VDI Guideline 4670

Humid Air

Library ASHRAE LibHuAirProp

Model: Virial equation from ASHRAE Report RP-1485 for real mixture of the real fluids:
 - Dry air
 - Steam

Consideration of:

- Enhancement of the partial saturation pressure of water vapor at elevated total pressures

www.ashrae.org/bookstore

Dry Air Including Liquid Air

Library LibRealAir

Formulation of Lemmon et al. (2000)

Refrigerants

Ammonia

Library LibNH3

Formulation of Tillner-Roth et al. (1993)

R134a

Library LibR134a

Formulation of Tillner-Roth and Baehr (1994)

Iso-Butane

Library LibButane_Iso

Formulation of Bückler and Wagner (2006)

n-Butane

Library LibButane_n

Formulation of Bückler and Wagner (2006)

Mixtures for Absorption Processes

Ammonia/Water Mixtures

Library LibAmWa

IAPWS Guideline 2001 of Tillner-Roth and Friend (1998)

Helmholtz energy equation for the mixing term (also useable for calculating the Kalina Cycle)

Water/Lithium Bromide Mixtures

Library LibWaLi

Formulation of Kim and Infante Ferreira (2004)

Gibbs energy equation for the mixing term

Liquid Coolants

Liquid Secondary Refrigerants

Library LibSecRef

Liquid solutions of water with

C ₂ H ₆ O ₂	Ethylene glycol
C ₃ H ₈ O ₂	Propylene glycol
C ₂ H ₅ OH	Ethanol
CH ₃ OH	Methanol
C ₃ H ₈ O ₃	Glycerol
K ₂ CO ₃	Potassium carbonate
CaCl ₂	Calcium chloride
MgCl ₂	Magnesium chloride
NaCl	Sodium chloride
C ₂ H ₃ KO ₂	Potassium acetate
CHKO ₂	Potassium formate
LiCl	Lithium chloride
NH ₃	Ammonia

Formulation of the International Institute of Refrigeration (IIR 2010)

Ethanol**Library LibC2H5OH**

Formulation of
Schroeder et al. (2014)

Methanol**Library LibCH3OH**

Formulation of
de Reuck and Craven (1993)

Propane**Library LibPropane**

Formulation of
Lemmon et al. (2009)

Siloxanes as ORC Working Fluids

Octamethylcyclotetrasiloxane $C_8H_{24}O_4Si_4$ **Library LibD4**

Decamethylcyclopentasiloxane $C_{10}H_{30}O_5Si_5$ **Library LibD5**

Tetradecamethylhexasiloxane $C_{14}H_{42}O_6Si_6$ **Library LibMD4M**

Hexamethyldisiloxane $C_6H_{18}OSi_2$ **Library LibMM**

Formulation of Colonna et al. (2006)

Dodecamethylcyclohexasiloxane $C_{12}H_{36}O_6Si_6$ **Library LibD6**

Decamethyltetrasiloxane $C_{10}H_{30}O_3Si_4$ **Library LibMD2M**

Dodecamethylpentasiloxane $C_{12}H_{36}O_4Si_5$ **Library LibMD3M**

Octamethyltrisiloxane $C_8H_{24}O_2Si_3$ **Library LibMDM**

Formulation of Colonna et al. (2008)

Nitrogen and Oxygen**Libraries
LibN2 and LibO2**

Formulations of Span et al. (2000)
and Schmidt and Wagner (1985)

Hydrogen**Library LibH2**

Formulation of
Leachman et al. (2009)

Helium**Library LibHe**

Formulation of
Arp et al. (1998)

Hydrocarbons

Decane $C_{10}H_{22}$ **Library LibC10H22**

Isopentane C_5H_{12} **Library LibC5H12_Iso**

Neopentane C_5H_{12} **Library LibC5H12_Neo**

Isohexane C_6H_{14} **Library LibC6H14**

Toluene C_7H_8 **Library LibC7H8**

Formulation of Lemmon and Span (2006)

Further Fluids

Carbon monoxide **CO** **Library LibCO**

Carbonyl sulfide **COS** **Library LibCOS**

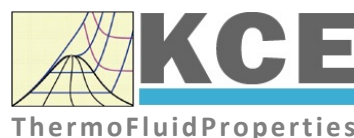
Hydrogen sulfide **H₂S** **Library LibH2S**

Nitrous oxide **N₂O** **Library LibN2O**

Sulfur dioxide **SO₂** **Library LibSO2**

Acetone C_3H_6O **Library LibC3H6O**

Formulation of Lemmon and Span (2006)

**For more information please contact:**

KCE-ThermoFluidProperties UG
Prof. Dr. Hans-Joachim Kretschmar
Haager Weg 6
92224 Amberg, Germany

Internet: www.thermofluidprop.com
Email: info@thermofluidprop.com
Phone: +49-9621-1762047
Mobile: +49-172-7914607
Fax: +49-3222-1095810

The following thermodynamic and transport properties can be calculated^a:**Thermodynamic Properties**

- Vapor pressure p_s
- Saturation temperature T_s
- Density ρ
- Specific volume v
- Enthalpy h
- Internal energy u
- Entropy s
- Exergy e
- Isobaric heat capacity c_p
- Isochoric heat capacity c_v
- Isentropic exponent κ
- Speed of sound w
- Surface tension σ

Transport Properties

- Dynamic viscosity η
- Kinematic viscosity ν
- Thermal conductivity λ
- Prandtl number Pr
- Thermal diffusivity a

Backward Functions

- $T, v, s(p, h)$
- $T, v, h(p, s)$
- $p, T, v(h, s)$
- $p, T(v, h)$
- $p, T(v, u)$

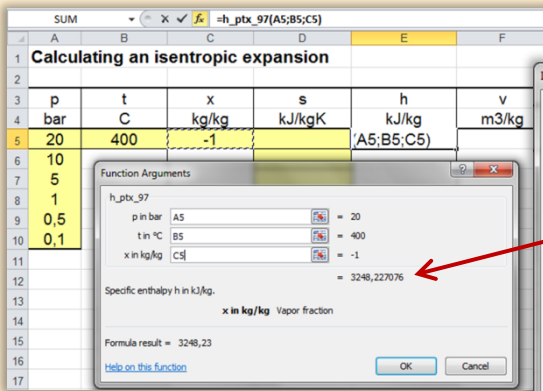
Thermodynamic Derivatives

- Partial derivatives used in process modeling can be calculated.

^a Not all of these property functions are available in all property libraries.

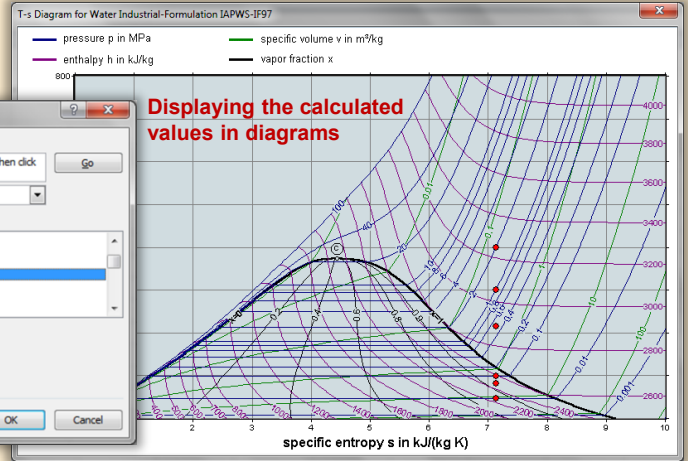
Property Software for Calculating Heat Cycles, Boilers, Turbines and Refrigerators

Add-In FluidEXL^{Graphics} for Excel[®]



Excel spreadsheet with columns: bar, C, kg/kg, kJ/kgK, h, m³/kg. Values: 20, 400, -1, 3248, 2270.6. Dialog box for h_ptx_97 with arguments: p in bar (A5) = 20, t in °C (B5) = 400, x in kg/kg (C5) = -1. Formula result: 3248,23.

Choosing a property library and a function

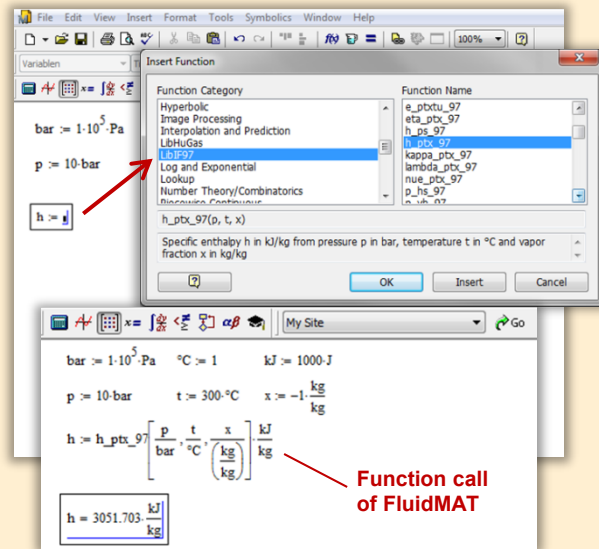


Displaying the calculated values in diagrams

Menu for the input of given property values

Add-On FluidMAT for Mathcad[®]
Add-On FluidPRIME for Mathcad Prime[®]

The property libraries can be used in Mathcad[®] and Mathcad Prime[®].

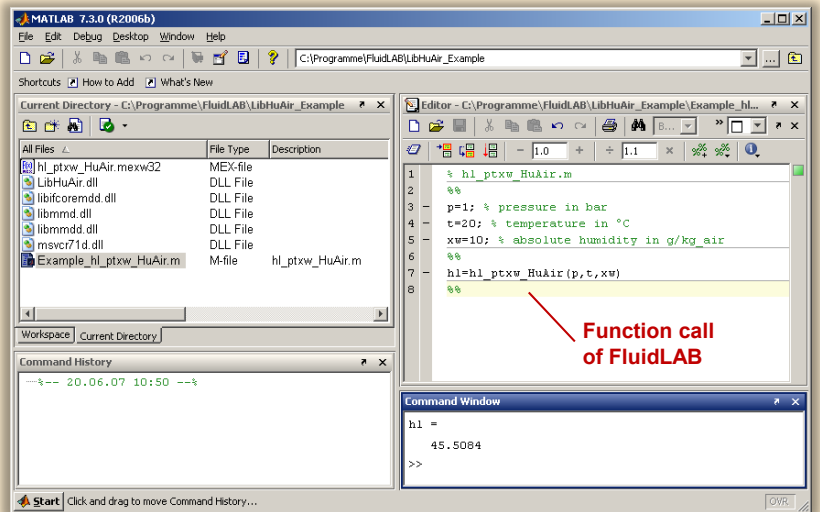


Mathcad interface showing 'Insert Function' dialog with 'LibHuAir' selected. Below, a function call: $h = h_{ptx_97} \left[\frac{p}{\text{bar}}, \frac{t}{^\circ\text{C}}, \frac{x}{\left(\frac{\text{kg}}{\text{kg}}\right)} \right] \frac{\text{kJ}}{\text{kg}}$. Result: $h = 3051.703 \frac{\text{kJ}}{\text{kg}}$.

Function call of FluidMAT

Add-On FluidLAB for MATLAB[®] and SIMULINK[®]

Using the Add-In FluidLAB the property functions can be called in MATLAB[®] and SIMULINK[®].

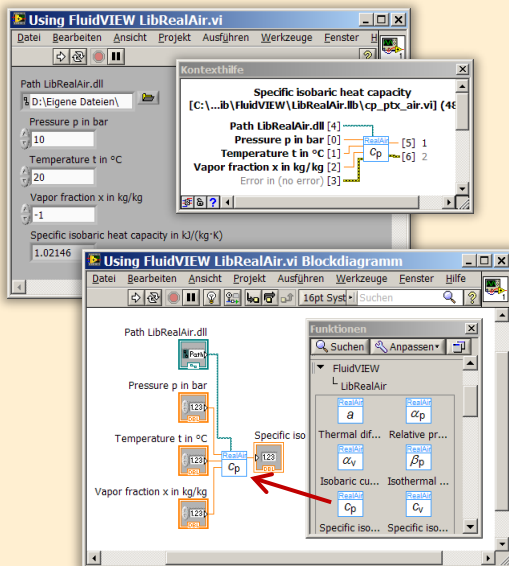


MATLAB 7.3.0 (R2006b) interface. Script: `h1 = hl_ptxw_HuAir(m, p, t, x)`. Command Window output: `h1 = 45.5084`. Editor shows function call of FluidLAB.

Function call of FluidLAB

Add-On FluidVIEW for LabVIEW[™]

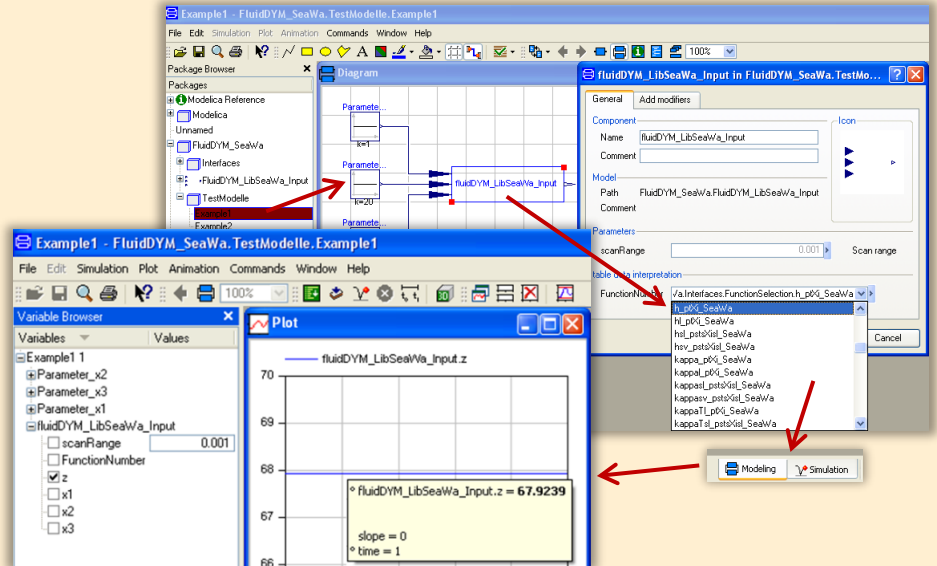
The property functions can be calculated in LabVIEW[™].



LabVIEW interface showing 'Using FluidVIEW LibRealAir.vi' block diagram. Inputs: Pressure p in bar (10), Temperature t in °C (20), Vapor fraction x in kg/kg (-1). Output: Specific isobaric heat capacity in kJ/(kg K) (1.02146).

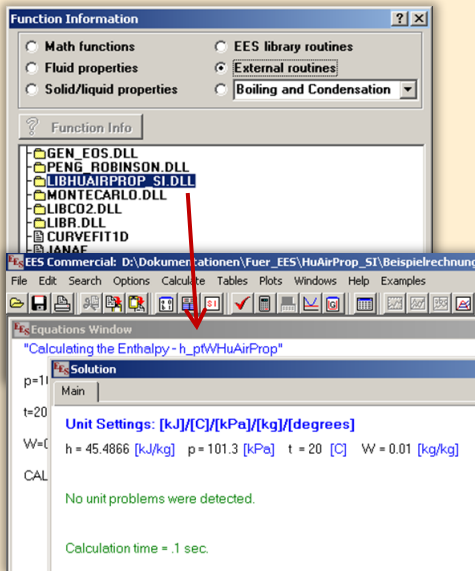
Add-On FluidDYM for DYMOLA[®] (Modelica) and SimulationX[®]

The property functions can be called in DYMOLA[®] and SimulationX[®].

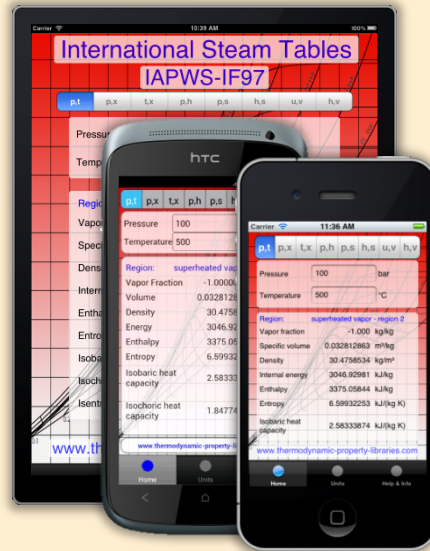


SimulationX interface showing 'fluidDYM_LibSeaWa_Input' block. Variable Browser shows 'z' value: 67.9239. Plot shows 'fluidDYM_LibSeaWa_Input.z' vs 'time' with a value of 67.9239 at time 1.

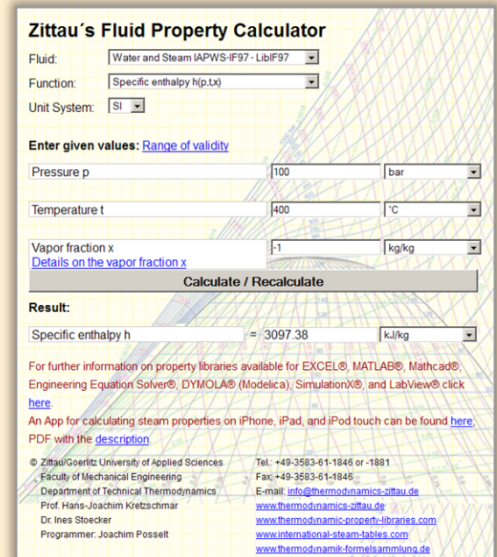
Add-On FluidEES for Engineering Equation Solver®



App International Steam Tables for iPhone, iPad, iPod touch, Android Smartphones and Tablets



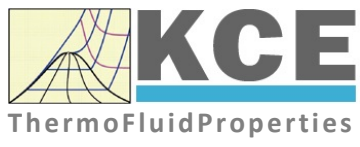
Online Property Calculator at www.thermofluidprop.com



Property Software for Pocket Calculators

<p>FluidCasio</p> <p>fx 9750 G II CFX 9850 fx-GG20 CFX 9860 G Graph 85 ALGEBRA FX 2.0</p>				<p>FluidHP</p> <p>HP 48 HP 49</p>		<p>FluidTI</p> <p>TI Nspire CX CAS TI 83 TI Nspire CAS TI 84 TI 89 TI Voyage 200</p>	
---	--	--	--	---	--	--	--

For more information please contact:



KCE-ThermoFluidProperties UG
 Prof. Dr. Hans-Joachim Kretzschmar
 Haager Weg 6
 92224 Amberg, Germany

Internet: www.thermofluidprop.com
 Email: info@thermofluidprop.com
 Phone: +49-9621-1762047
 Mobile: +49-172-7914607
 Fax: +49-3222-1095810

The following thermodynamic and transport properties^a can be calculated in Excel®, MATLAB®, Mathcad®, Engineering Equation Solver® (EES), DYMOLA® (Modelica), SimulationX® and LabVIEW™:

- | | | | |
|--|---|---|---|
| <p>Thermodynamic Properties</p> <ul style="list-style-type: none"> • Vapor pressure p_s • Saturation temperature T_s • Density ρ • Specific volume v • Enthalpy h • Internal energy u • Entropy s • Exergy e • Isobaric heat capacity c_p • Isochoric heat capacity c_v • Isentropic exponent κ • Speed of sound w • Surface tension σ | <p>Transport Properties</p> <ul style="list-style-type: none"> • Dynamic viscosity η • Kinematic viscosity ν • Thermal conductivity λ • Prandtl number Pr • Thermal diffusivity a | <p>Backward Functions</p> <ul style="list-style-type: none"> • $T, v, s(p, h)$ • $T, v, h(p, s)$ • $p, T, v(h, s)$ • $p, T(v, h)$ • $p, T(v, u)$ | <p>Thermodynamic Derivatives</p> <ul style="list-style-type: none"> • Partial derivatives used in process modeling can be calculated. |
|--|---|---|---|

^a Not all of these property functions are available in all property libraries.

5 References

- [1] Herrmann, S.; Kretzschmar, H.-J.; Gatley, D.P.: Thermodynamic Properties of Real Moist Air, Dry Air, Steam, Water, and Ice. HVAC&R Research 5, 961-986 (2009).
- [2] Herrmann, S.; Kretzschmar, H.-J.; Gatley, D.P.: Thermodynamic Properties of Real Moist Air, Dry Air, Steam, Water, and Ice. ASHRAE RP-1485, American Society of Heating, Refrigerating, and Air-Conditioning Engineers, Inc., Atlanta, GA (2009).
- [3] Herrmann, S.; Kretzschmar, H.-J.; Aute, V.C.; Gatley, D.P.; Vogel, E.: Transport Properties of Real Moist Air, Dry Air, Steam, and Water. ASHRAE RP-1767, American Society of Heating, Refrigerating, and Air-Conditioning Engineers, Inc., Atlanta, GA (2018).
- [4] Herrmann, S.; Kretzschmar, H.-J.; Aute, V.C.; Gatley, D.P.; Vogel, E.: Transport Properties of Real Moist Air, Dry Air, Steam, and Water. Science and Technology for the Built Environment 27, 393-401 (2021).
- [5] IAPWS. R6-95, Revised Release on the IAPWS Formulation 1995 for the Thermodynamic Properties of Ordinary Water Substance for General and Scientific Use. (2009), available from www.iapws.org.
- [6] Wagner, W.; Pruß, A.: The IAPWS Formulation 1995 for the Thermodynamic Properties of Ordinary Water Substance for General and Scientific Use. J. Phys. Chem. Ref. Data 31, 387-535 (2002).
- [7] IAPWS. R7-97, Revised Release on the IAPWS Industrial Formulation 1997 for the Thermodynamic Properties of Water and Steam IAPWS-IF97. (2007), available from www.iapws.org.
- [8] Wagner, W.; Kretzschmar, H.-J.: International Steam Tables. Springer, Berlin (2008).
- [9] Parry, W.T.; Bellows, J.C.; Gallagher, J.S.; Harvey, A.H.: ASME International Steam Tables for Industrial Use. ASME Press, New York (2000).
- [10] IAPWS. R10-06, Revised Release on the Equation of State 2006 for H₂O Ice Ih. (2009); available from www.iapws.org.
- [11] IAPWS. R14-08, Revised Release on the Pressure along the Melting and Sublimation Curves of Ordinary Water Substance. (2008); available from www.iapws.org.
- [12] Nelson, H.F.; Sauer, H.J.: Formulation of High-Temperature Properties for Moist Air. HVAC&R Research 8, 311-334 (2002).
- [13] Gatley, D.P.: Understanding Psychrometrics, 2nd ed., ASHRAE, Atlanta (2005).
- [14] Lemmon, E.W.; Jacobsen, R.T.; Penoncello, S.G.; Friend, D.G.: Thermodynamic Properties of Air and Mixture of Nitrogen, Argon, and Oxygen from 60 to 2000 K at Pressures to 2000 MPa. J. Phys. Chem. Ref. Data 29, 331-385 (2000).
- [15] Harvey, A.H.; Huang, P.H.: First-Principles Calculation of the Air-Water Second Virial Coefficient. Int. J. Thermophys. 28, 556-565 (2007).
- [16] IAPWS. G7-04, Guideline on the Henry's Constant and Vapor-Liquid Distribution Constant for Gases in H₂O and D₂O at High Temperatures. (2004), available from www.iapws.org.

- [17] Gatley, D.P.; Herrmann, S.; Kretzschmar, H.-J.: A Twenty-First Century Molar Mass for Dry Air. HVAC&R Research 14, 655-662 (2008).
- [18] Lemmon, E.W.; Jacobsen, R.T.: Viscosity and Thermal Conductivity Equations for Nitrogen, Oxygen, Argon, and Air. Int. J. Thermophys. 25, 21-69 (2004).
- [19] IAPWS. R12-08, Release on the IAPWS Formulation 2008 for the Viscosity of Ordinary Water Substance. (2008), available from www.iapws.org.
- [20] IAPWS. R15-11, Revised Release on the IAPWS Formulation 1985 for the Thermal Conductivity of Ordinary Water Substance. (2008), available from www.iapws.org.
- [21] Hyland, R.W.; Wexler, A.: Formulations for the Thermodynamic Properties of Dry Air from 173.15 K to 473.15 K, and of Saturated Moist Air from 173.15 K to 372.15 K, at Pressures to 5 MPa. ASHRAE Trans. 89, 520-535 (1983).
- [22] Mohr, P.J.; Taylor, P.N.: CODATA Recommended Values of the Fundamental Physical Constants: 2002. Rev. Mod. Phys. 77, 1-107 (2005).
- [23] ASHRAE. 2009 Handbook of Fundamentals. Chapter 1 - Psychrometrics. American Society of Heating, Refrigerating, and Air-Conditioning Engineers, Inc., Atlanta, GA (2009).
- [24] Feistel, R.; Lovell-Smith, J.W.; Hellmuth, O.: Virial Approximation of the TEOS-10 Equation for the Fugacity of Water in Humid Air. Int. J. Thermophys. 36, 44-68 (2015).

6 Satisfied Customers

Period from 2018 to 2022

The following companies and institutions use the property libraries:

- FluidEXL *Graphics* for Excel® incl. VBA
- FluidLAB for MATLAB® and Simulink
- FluidMAT for Mathcad®
- FluidPRIME for Mathcad Prime®
- FluidEES for Engineering Equation Solver® EES
- FluidDYM for Dymola® (Modelica) and SimulationX®
- FluidVIEW for LabVIEW™
- FluidPYT for Python
- FluidJAVA for Java
- DLLs for Windows Applications
- Shared Objects for Linux
- Shared Objects for macOS.

2022

ASTG, Graz, Austria	12/2022
Wandschneider + Gutjahr, Hamburg	
RWE Supply & Trading, Essen	11/2022
Stadtwerke Rosenheim	
CEA, Saclay, France	10/2022
RWE Supply & Trading, Essen	
SEEC Saudi Energy Efficiency Center, Riyadh, Saudi Arabia	
MAN, Copenhagen, Denmark	
Hermeler & Partner Consulting Engineers, Sassenberg	09/2022
Envi Con, Nürnberg	
Drill Cool Systems, Bakersfield CA, USA	
RWE Supply & Trading, Essen	
Maerz Ofenbau, Zürich, Switzerland	
Saale Energie, Schkopau	
ERGO, Dresden	
Mainova, Frankfurt/Main	
Bundeswehr, Koblenz	08/2022
RWE Supply & Trading, Essen	
Grenzebach Corporation, Newnan GE, USA	
AGRANA, Gmuend, Austria	07/2022
MIBRAG, Zeitz	
Hochschule Niederrhein, Krefeld	
ULT, Löbau	06/2022
LEAG, Cottbus	

VPC Group, Vetschau	
Wärme, Hamburg	
ILK, Dresden	
Stricker IB, Küsnacht a. Rigi, Switzerland	
LEAG, Cottbus	05/2022
RWE Supply & Trading, Essen	
IGT Tomalla, Kreuztal	
B+T Engineering, Dübendorf, Switzerland	
Stricker IB, Küsnacht a. Rigi, Switzerland	
Vogelsang & Benning, Bochum	04/2022
Frischli, Rehburg-Loccum	
BPS Consulting, Sprengel	03/2022
HS Hannover, Maschinenbau & BioVT	
M+M Turbinentechnik, Bad Salzungen	
Uni. Strathclyde, Glasgow, UK	02/2022
Delta Energy Group, Jiaozhou City, Qingdao, China	
Wetzel IB, Guben	
Wijbenga, PC Geldermalsen, The Netherlands	
Voith Paper, Heidenheim	
HS Zittau/Görlitz, Maschinenwesen	01/2022
Thermische Abfallbehandlung, Lauterbach	
Webb Institute, Glen Cove NY, USA	
TU Berlin, Umweltverfahrenstechnik	
SachsenEnergie, Dresden	
Doosan, Chang-won-si, Gyeongsangnam-do, South Korea	
KW3, LH Veenendaal, The Netherlands	
Université du Luxembourg, Esch-sur-Alzette	
Enseleit IB, Mansfeld	
Caliqua/Equans, Zürich, Switzerland	
Rudnick & Enners, Alpenrod	

2021

Wenisch IB, Vetschau	12/2021
PPCHEM, Hinwil, Switzerland	
KW3, The Netherlands	
BASF Ludwigshafen	
Air-Consult, Jena	
Sjerp & Jongeneel, RB Zoetermeer, The Netherlands	11/2021
Maerz Ofenbau, Zürich, Switzerland	
RWE Supply & Trading, Essen	
Hahn IB, Dresden	10/2021
Therm, South Africa	
RWE Supply & Trading, Essen	
TH Nürnberg, Verfahrenstechnik	09/2021
RWE Supply & Trading, Essen	
Enseleit IB, Mansfeld	

SachsenEnergie, Dresden	
BSH Hausgeräte, Berlin	
Norsk Energi, Oslo, Norway	08/2021
AKM Industrieanlagen, Haltern	
Drill Cool Systems, Bakersfield CA, USA	
Siemens Energy Global, Erlangen	07/2021
Wulff & Umag, Husum	
Planungsbüro Waidhas, Chemnitz	
Burkhardt Energie Technik, Mühlhausen	
Lücke IB, Paderborn	06/2021
TU Dresden, Energieverfahrenstechnik	
Wärme, Hamburg	
AL-KO Therm, Kötz	
PCK Raffinerie, Schwedt	
Vogelsang & Benning, Bochum	05/2021
MTU, München	
VPC Group, Vetschau	
AVG, Köln	04/2021
TH Ulm, Institut für Fahrzeugtechnik	
Marty IB, Oberwil, Switzerland	
HypTec, Lebring, Austria	
Lopez IB, Getxo, Bizkaia, Spain	03/2021
GM Remediation Systems, Leoben, Austria	
Jager Kältetechnik, Osnabrück	
T&M Automation, GR Leidschendam, The Netherlands	
RWE Supply & Trading, Essen	
Stadtwerke Leipzig	
Beuth Hochschule für Technik, Berlin	
Beleth IB, Woeth	02/2021
ZTL, Thal, Austria	
ETABO Bochum	
RWE Supply & Trading, Essen	
Onyx Germany, Berlin	
TU Dresden, Kältetechnik	
GOHL-KTK, Durmersheim	
Therm Development, South Africa	
thermofin, Heinsdorfergrund	
RWE Supply & Trading, Essen	01/2021
STEAG, Essen	
ETA Energieberatung, Pfaffenhofen	
Enex Power, Kirchseeon	

2020

Drill Cool, Bakersfield CA, USA	12/2020
Manders, The Netherlands	
RWE Supply & Trading, Essen	

NEOWAT Lodz, Poland	
University of Duisburg-Essen, Duisburg	11/2020
Stellenbosch University, South Africa	
University De France-COMTe, France	
RWE, Essen	
STEAG, Herne	
Isenmann Ingenieurbüro	
University of Stuttgart, ITLR, Stuttgart	
Norsk Energi, Oslo, Norway	
TGM Kanis, Nürnberg	
Stadtwerke Neuburg	10/2020
Smurfit Kappa, Roermond, The Netherlands	
RWE, Essen	
Hochschule Zittau/Görlitz, Wirtschaftsingenieurwesen	
Stadtwerke, Neuburg	
ILK, Dresden	
ATESTEO, Alsdorf	
Hochschule Zittau/Görlitz, Maschinenwesen	
TH Nürnberg, Verfahrenstechnik	
Drill Cool, Bakersfield CA,USA	09/2020
RWE, Essen	
2Meyers Ingenieurbüro, Nürnberg	
FELUWA, Mürlenbach	
Stadtwerke Neuburg	
Caverion, Wien, Austria	
GMVA Niederrhein, Oberhausen	
INWAT Lodz, Poland	
Troche Ingenieurbüro, Hayingen	08/2020
CEA Saclay, France	
VPC, Vetschau	07/2020
FSK System-Kälte-Klima, Dortmund	
Exergie Etudes, Sarl, Switzerland	
AWG Wuppertal	
STEAG Energy Services, Zwingenberg	
Hochschule Braunschweig	06/2020
DBI, Leipzig	
GOHL-KTK, Dumersheim	
TU Dresden, Energieverfahrenstechnik	
BASF SE, ESI/EE, Ludwigshafen	
Wärme Hamburg	
Ruchti Ingenieurbüro, Uster, Switzerland	
IWB, Basel, Switzerland	
Midiplan, Bietingen-Bissingen	05/2020
Knieschke, Ingenieurbüro	
RWE, Essen	
Leser, Hamburg	

AGRANA, Gmünd, Austria	
EWT Wassertechnik, Celle	
Hochschule Darmstadt	04/2020
MTU München CCP	
HAW Hamburg	03/2020
Hanon, Novi Jicin, Czech Republic	
TU Dresden, Kältetechnik	
MAN, Copenhagen, Denmark	
EnerTech, Radebeul	02/2020
LEAG, Cottbus	
B+B Engineering Magdeburg	
Hochschule Offenburg	
WIB, Dennheritz	01/2020
Universität Duisburg-Essen, Strömungsmaschinen	
Kältetechnik Dresden-Bremen	
TH Ingolstadt	
Vattenfall AB, Jokkmokk, Sweden	
Fraunhofer UMSICHT	

2019

PEU Leipzig, Rötha	12/2019
MB-Holding, Vestenbergsgreuth	
RWE, Essen	
Georg-Büchner-Hochschule, Darmstadt	11/2019
EEB ENERKO, Aldenhoven	
Robert Benoufa Energietechnik, Wiesloch	
Kehrein & Kubanek Klimatechnik, Moers	10/2019
Hanon Systems Autopal Services, Hluk, Czech Republic	
CEA Saclay, Gif Sur Yvette cedex, France	
Saudi Energy Efficiency Center SEEC, Riyadh, Saudi Arabia	
VPC, Vetschau	09/2019
jGanser PM + Engineering, Forchheim	
Endress+Hauser Flowtec AG, Reinach, Switzerland	
Ruchti IB, Uster, Switzerland	
ZWILAG Zwischenlager Würenlingen, Switzerland	08/2019
Hochschule Zittau/Görlitz, Faculty Maschinenwesen	
Stadtwerke Neubrandenburg	
Physikalisch Technische Bundesanstalt PTB, Braunschweig	
GMVA Oberhausen	07/2019
Endress+Hauser Flowtec AG, Reinach, Switzerland	
WARNICA, Waterloo, Canada	
MIBRAG, Zeitz	06/2019
Pöyry, Zürich, Switzerland	
RWTH Aachen, Institut für Strahlantriebe und Turbomaschinen	
Midiplan, Bietigheim-Bissingen	
GKS Schweinfurt	

Comparex Leipzig for LEAG, Berlin	06/2018
Münstermann, Telgte	05/2018
TH Nürnberg, Verfahrenstechnik	
Universität Madrid, Madrid, Spanien	
HS Zittau/Görlitz, Wirtschaftsingenieurwesen	
HS Niederrhein, Krefeld	
Wilhelm-Büchner HS, Pfungstadt	03/2018
GRS, Köln	
WIB, Dennheritz	
RONAL AG, Härklingen, Schweiz	02/2018
Ingenieurbüro Leipert, Riegelsberg	
AIXPROCESS, Aachen	
KRONES, Neutraubling	
Doosan Lentjes, Ratingen	01/2018