



**Property Library for  
Seawater  
Calculated from the  
IAPWS Industrial Formulation 2013**

**FluidDYM  
with LibSeaWa  
for DYMOLA®**

Prof. Hans-Joachim Kretzschmar

Dr. Sebastian Herrmann

Dr. Matthias Kunick

# **Property Library for Seawater Calculated from the IAPWS Industrial Formulation 2013 FluidDYM LibSeaWa**

## **Content**

- 0. Package Contents
- 1. Property Functions
- 2. Application of FluidDYM in Dymola®
  - 2.1 Installing FluidDYM
  - 2.2 Example: Calculation of the Specific Enthalpy  $h = f(p, t, \xi)$  of Seawater
  - 2.3 Removing FluidDYM
- 3. Program Documentation
- 4. Property Libraries for Calculating Heat Cycles, Boilers, Turbines, and Refrigerators
- 5. References
- 6. Satisfied Customers

For steam tables and further property libraries for EES®, Excel®, MATLAB® and Mathcad® see the following link:

[www.international-steam-tables.com](http://www.international-steam-tables.com)

---

© KCE-ThermoFluidProperties UG (with limited liability) & Co. KG  
Professor Hans-Joachim Kretzschmar  
Wallotstr. 3, 01307 Dresden, Germany  
Phone: +49-351-27597860  
Mobile: +49-172-7914607  
Fax: +49-3222-4262250  
Email: [info@thermofluidprop.com](mailto:info@thermofluidprop.com)  
Internet: [www.thermofluidprop.com](http://www.thermofluidprop.com)

## 0 Package Contents

### 0.1 Zip file for 32-bit DYMOLA®

#### "CD\_FluidDYM\_LibCO2.zip"

Including the following files:

FluidDYM_LibCO2_Setup.exe	Installation Program for the FluidDYM Add-In for use in DYMOLA®
LibCO2.dll	Dynamic Link Library f
FluidDYM_LibCO2_Docu.pdf	User's Guide
Folder "Users_Guide"	Includes the complete User's Guide

### 0.2 Zip file for 64-bit MATLAB®

#### "CD\_FluidDYM\_LibCO2\_64.zip"

Including the following files and folders:

##### **Files:**

Setup.exe	- Self-extracting and self-installing program for FluidLAB
FluidDYM_LibCO2_64.msi	- Installation program for the FluidLAB Add-On for use in MATLAB®
LibCO2.dll	- Dynamic Link Library for carbon dioxide for use in MATLAB®
FluidLAB_LibCO2_Docu.pdf	- User's Guide

##### **Folders:**

vcredist_x64	- Folder containing the "Microsoft Visual C++ 2010 x64 Redistributable Pack"
WindowsInstaller3_1	- Folder containing the "Microsoft Windows Installer"

# 1. Property Functions

## 1.1 Functions

Functional Dependence	Function Name in Excel®	Call as Fortran Program	Property or Function	Unit of the Result	Reference	Page
$a = f(p, t, \xi)$	a_ptXI_SeaWa	= A_PT XI_SEAWA(P,T,XI)	Thermal diffusivity	m <sup>2</sup> /s	[1], [2]	3/2
$a = f(t, \xi)$	al_tXI_SeaWa	= AL_TXI_SEAWA(T,XI)	Thermal diffusivity of subcooled liquid	m <sup>2</sup> /s	[1], [2]	3/3
$a_{sl} = f(p_s, t_s, \xi_{sl})$	asl_pstsXisl_SeaWa	= ASL_PST SXISL_SEAWA(PS,TS,XISL)	Thermal diffusivity of saturated liquid	m <sup>2</sup> /s	[1], [2]	3/4
$a_{sv} = f(p_s, t_s, \xi_{sv})$	asv_pstsXisl_SeaWa	= ASV_PST SXISL_SEAWA(PS,TS,XISL)	Thermal diffusivity of saturated vapor	m <sup>2</sup> /s	[1], [2]	3/5
$\alpha_l = f(p, t, \xi)$	alphan_ptXi_SeaWa	= ALPHAL_PT XI_SEAWA(P,T,XI)	Thermal expansion coefficient of subcooled liquid	1/K	[1]	3/6
$\alpha_{sl} = f(p_s, t_s, \xi_{sl})$	alphasl_pstsXisl_SeaWa	= ALPHASL_PST SXISL_SEAWA(PS,TS,XISL)	Thermal expansion coefficient of saturated liquid	1/K	[1]	3/7
$\beta_l = f(p, t, \xi)$	betan_ptXi_SeaWa	= BETAL_PT XI_SEAWA(P,T,XI)	Haline contraction coefficient of subcooled liquid	kg/kg	[1]	3/8
$\beta_{sl} = f(p, t, \xi_{sl})$	betasl_pstsXisl_SeaWa	= BETASL_PST SXISL_SEAWA(PS,TS,XISL)	Haline contraction coefficient of saturated liquid	kg/kg	[1]	3/9
$\beta_{sl} = f(p, t, \xi)$	betasl_ptXi_SeaWa	= BETAISL_PT XI_SEAWA(P,T,XI)	Isentropic temperature-pressure coefficient of subcooled liquid	K/kPa	[1]	3/10
$\beta_{ssl} = f(p, t, \xi_{sl})$	betalssl_pstsXisl_SeaWa	= BETAISL_PST SXISL_SEAWA(PS,TS,XISL)	Isentropic temperature-pressure coefficient of saturated liquid	K/kPa	[1]	3/11
$c_p = f(p, t, \xi)$	cp_ptXI_SeaWa	= CP_PT XI_SEAWA(P,T,XI)	Specific isobaric heat capacity	kJ/(kg·K)	[1], [2]	3/12
$c_{p,l} = f(p, t, \xi)$	cpl_ptXI_SeaWa	= CPL_PT XI_SEAWA(P,T,XI)	Specific isobaric heat capacity of subcooled liquid	kJ/(kg·K)	[1], [2]	3/13
$c_{p,sl} = f(p_s, t_s, \xi_{sl})$	cpsl_pstsXisl_SeaWa	= CPSL_PST SXISL_SEAWA(PS,TS,XISL)	Specific isobaric heat capacity of saturated liquid	kJ/(kg·K)	[1], [2]	3/14

Functional Dependence	Function Name in Excel®	Call as Fortran Program	Property or Function	Unit of the Result	Reference	Page
$c_{p_{sv}} = f(p_s, t_s, \xi_{sl})$	cpsv_pstsXisl_SeaWa	= CPSV_PSTSXISL_SEAWA(PS,TS,XISL)	Specific isobaric heat capacity of saturated vapor	kJ/(kg·K)	[1], [2]	3/15
$\eta = f(p, t, \xi)$	eta_ptXI_SeaWa	= ETA_PTXI_SEAWA(P,T,XI)	Dynamic viscosity	Pa·s	[1], [2]	3/16
$\eta_l = f(t, \xi)$	etal_tXI_SeaWa	= ETAL_TXI_SEAWA(T,XI)	Dynamic viscosity of subcooled liquid	Pa·s	[1], [2]	3/17
$\eta_{sl} = f(p_s, t_s, \xi_{sl})$	etasl_pstsXisl_SeaWa	= ETASL_PSTSXISL_SEAWA(PS,TS,XISL)	Dynamic viscosity of saturated liquid	Pa·s	[1], [2]	3/18
$\eta_{sv} = f(p_s, t_s, \xi_{sl})$	etasv_pstsXisl_SeaWa	= ETASV_PSTSXISL_SEAWA(PS,TS,XISL)	Dynamic viscosity of saturated vapor	Pa·s	[1], [2]	3/19
$f_l = f(p, t, \xi)$	fl_ptXI_SeaWa	= FL_PTXI_SEAWA(P,T,XI)	Specific Helmholtz energy of subcooled liquid	kJ/kg	[1]	3/20
$f_{sl} = f(p_s, t_s, \xi_{sl})$	fsl_pstsXisl_SeaWa	= FSL_PSTSXISL_SEAWA(PS,TS,XISL)	Specific Helmholtz energy of saturated liquid	kJ/kg	[1]	3/21
$\phi = f(p, t, \xi)$	phil_ptXI_SeaWa	= PHIL_PTXI_SEAWA(P,T,XI)	Osmotic coefficient of subcooled liquid	[-]	[1]	3/22
$\phi_{sl} = f(p_s, t_s, \xi_{sl})$	phisl_pstsXisl_SeaWa	= PHISL_PSTSXISL_SEAWA(PS,TS,XISL)	Osmotic coefficient of saturated liquid	[-]	[1]	3/23
$h = f(p, t, \xi)$	h_ptXI_SeaWa	= H_PTXI_SEAWA(P,T,XI)	Specific enthalpy	kJ/kg	[1], [2]	3/24
$h_l = f(p, t, \xi)$	hl_ptXI_SeaWa	= HL_PTXI_SEAWA(P,T,XI)	Specific enthalpy of subcooled liquid	kJ/kg	[1], [2]	3/25
$h_{sl} = f(p_s, t_s, \xi_{sl})$	hsl_pstsXisl_SeaWa	= HSL_PSTSXISL_SEAWA(PS,TS,XISL)	Specific enthalpy of saturated liquid	kJ/kg	[1], [2]	3/26
$h_{sv} = f(p_s, t_s, \xi_{sl})$	hsv_pstsXisl_SeaWa	= HSV_PSTSXISL_SEAWA(PS,TS,XISL)	Specific enthalpy of saturated steam	kJ/kg	[1], [2]	3/27
$\kappa = f(p, t, \xi)$	kappa_ptXI_SeaWa	= KAPPA_PTXI_SEAWA(P,T,XI)	Isentropic exponent	[-]	[1]	3/28
$\kappa_l = f(p, t, \xi)$	kappal_ptXI_SeaWa	= KAPPAL_PTXI_SEAWA(P,T,XI)	Isentropic exponent of subcooled liquid	[-]	[1]	3/29
$\kappa_{sl} = f(p_s, t_s, \xi_{sl})$	kappasl_pstsXisl_SeaWa	= KAPPASL_PSTSXISL_SEAWA(PS,TS,XISL)	Isentropic exponent of saturated liquid	[-]	[1]	3/30

Functional Dependence	Function Name in Excel®	Call as Fortran Program	Property or Function	Unit of the Result	Reference	Page
$\kappa_{sv} = f(p_s, t_s, \xi_{sl})$	kappasv_pstsXisl_SeaWa	= KAPPASV_PSTSXISL_SEAWA(P,TS,XISL)	Isentropic exponent of saturated vapor	[-]	[1]	3/31
$\kappa_{T_l} = f(p, t, \xi)$	kappaTI_ptXI_SeaWa	= KAPPATL_PTXI_SEAWA(P,T,XI)	Isothermal compressibility of subcooled liquid	1/kPa	[1]	3/32
$\kappa_{T_{sl}} = f(p_s, t_s, \xi_{sl})$	kappaTsl_pstsXisl_SeaWa	= KAPPATSL_PSTSXISL_SEAWA(P,TS,XISL)	Isothermal compressibility of saturated liquid	1/kPa	[1]	3/33
$\kappa_{ls_l} = f(p, t, \xi)$	kappalsl_ptXI_SeaWa	= KAPPAISL_PTXI_SEAWA(P,T,XI)	Isentropic compressibility of subcooled liquid	1/kPa	[1]	3/34
$\kappa_{ls_{sl}} = f(p_s, t_s, \xi_{sl})$	kappaIssl_pstsXisl_SeaWa	= KAPPAISL_PSTSXISL_SEAWA(P,TS,XISL)	Isentropic compressibility of saturated liquid	1/kPa	[1]	3/35
$\lambda = f(p, t, \xi)$	lambda_ptXI_SeaWa	= LAMBDA_PTXI_SEAWA(P,T,XI)	Thermal conductivity	W/(m*K)	[3], [4], [15]	3/36
$\lambda_l = f(t, \xi)$	lambdal_tXI_SeaWa	= LAMBDAL_TXI_SEAWA(T,XI)	Thermal conductivity of subcooled liquid	W/(m*K)	[3], [4], [15]	3/37
$\lambda_{sl} = f(p_s, t_s, \xi_{sl})$	lambdasl_pstsXisl_SeaWa	= LAMBDA_SL_PSTSXISL_SEAWA(P,TS,XISL)	Thermal conductivity of saturated liquid	W/(m*K)	[3], [4], [15]	3/38
$\lambda_{sv} = f(p_s, t_s, \xi_{sl})$	lambdasv_pstsXisl_SeaWa	= LAMBDA_SV_PSTSXISL_SEAWA(P,TS,XISL)	Thermal conductivity of saturated vapor	W/(m*K)	[3], [4], [15]	3/39
$\mu_l = f(p, t, \xi)$	myl_ptXI_SeaWa	= MYL_PTXI_SEAWA(P,T,XI)	Relative chem. potential of subcooled liquid	kJ/kg	[1]	3/40
$\mu_{sl} = f(p_s, t_s, \xi_{sl})$	mysl_pstsXisl_SeaWa	= MYSL_PSTSXISL_SEAWA(P,TS,XISL)	Relative chem. potential of saturated liquid	kJ/kg	[1]	3/41
$\mu_{W_l} = f(p, t, \xi)$	mywl_ptXI_SeaWa	= MYWL_PTXI_SEAWA(P,T,XI)	Relative chem. potential of H <sub>2</sub> O of subcooled liquid	kJ/kg	[1]	3/42
$\mu_{W_{sl}} = f(p_s, t_s, \xi_{sl})$	mywsl_pstsXisl_SeaWa	= MYWSL_PSTSXISL_SEAWA(P,TS,XISL)	Relative chem. potential of H <sub>2</sub> O of saturated liquid	kJ/kg	[1]	3/43
$\mu_{Salt_l} = f(p, t, \xi)$	mySaltl_ptXI_SeaWa	= MYSALT_PTXI_SEAWA(P,T,XI)	Relative chem. potential of sea salt of subcooled liquid	kJ/kg	[1]	3/44
$\mu_{Salt_{sl}} = f(p_s, t_s, \xi_{sl})$	mySaltsl_pstsXisl_SeaWa	= MYSALTSL_PSTSXISL_SEAWA(P,TS,XISL)	Relative chem. potential of sea salt of saturated liquid	kJ/kg	[1]	3/45

Functional Dependence	Function Name in Excel®	Call as Fortran Program	Property or Function	Unit of the Result	Reference	Page
$\nu = f(p, t, \xi)$	ny_ptXI_SeaWa	= NY_PT XI_SEAWA(P,T,XI)	Kinematic viscosity	m <sup>2</sup> /s	[1], [2]	3/46
$\nu_l = f(t, \xi)$	nyl_tXI_SeaWa	= NYL_TXI_SEAWA(T,XI)	Kinematic viscosity of subcooled liquid	m <sup>2</sup> /s	[1], [2]	3/47
$\nu_{sl} = f(p_s, t_s, \xi_{sl})$	nysl_pstsXisl_SeaWa	= NYSL_PST SXISL_SEAWA(PS,TS,XISL)	Kinematic viscosity of saturated liquid	m <sup>2</sup> /s	[1], [2]	3/48
$\nu_{sv} = f(p_s, t_s, \xi_{sl})$	nysv_pstsXisl_SeaWa	= NYSV_PST SXISL_SEAWA(PS,TS,XISL)	Kinematic viscosity of saturated vapor	m <sup>2</sup> /s	[1], [2]	3/49
$p_s = f(t_s, \xi_{sl})$	ps_tsXisl_SeaWa	= PS_TSXISL_SEAWA(TS,XISL)	Boiling pressure	bar	[1] ,[2] ,[4]	3/50
$p_{mel} = f(t, \xi)$	pmel_tXi_SeaWa	= PMEL_TXI_SEAWA(T,XI)	Freezing pressure	bar	[1] ,[4], [5]	3/51
$p_{tr} = f(\xi)$	ptr_Xi_SeaWa	= PTR_XI_SEAWA(T,XI)	Triple point pressure	bar	[1] ,[4], [5]	3/52
$Pr = f(p, t, \xi)$	Pr_ptXI_SeaWa	= PR_PT XI_SEAWA(P,T,XI)	Prandtl Number	[-]	[1], [2], [3]	3/53
$Pr_l = f(t, \xi)$	Prl_tXI_SeaWa	= PRL_TXI_SEAWA(T,XI)	Prandtl Number of subcooled liquid	[-]	[1], [2], [3]	3/54
$Pr_{sl} = f(p_s, t_s, \xi_{sl})$	Prsl_pstsXisl_SeaWa	= PRSL_PST SXISL_SEAWA(PS,TS,XISL)	Prandtl Number of saturated liquid	[-]	[1], [2], [3]	3/55
$Pr_{sv} = f(p_s, t_s, \xi_{sl})$	Prsv_pstsXisl_SeaWa	= PRSV_PST SXISL_SEAWA(PS,TS,XISL)	Prandtl Number of saturated vapor	[-]	[1], [2], [3]	3/56
$\rho = f(p, t, \xi)$	rho_ptXI_SeaWa	= RHO_PT XI_SEAWA(P,T,XI)	Density	kg/m <sup>3</sup>	[1], [2]	3/57
$\rho_l = f(p, t, \xi)$	rho_l_ptXI_SeaWa	= RHOL_PT XI_SEAWA(P,T,XI)	Density of subcooled liquid	kg/m <sup>3</sup>	[1], [2]	3/58
$\rho_{sl} = f(p_s, t_s, \xi_{sl})$	rhosl_pstsXisl_SeaWa	= RHOSL_PST SXISL_SEAWA(PS,TS,XISL)	Density of saturated liquid	kg/m <sup>3</sup>	[1], [2]	3/59
$\rho_{sv} = f(p_s, t_s, \xi_{sl})$	rhosv_pstsXisl_SeaWa	= RHOSV_PST SXISL_SEAWA(PS,TS,XISL)	Density of saturated vapor	kg/m <sup>3</sup>	[1], [2]	3/60
$s = f(p, t, \xi)$	s_ptXI_SeaWa	= S_PT XI_SEAWA(P,T,XI)	Specific entropy	kJ/(kg*K)	[1], [2]	3/61
$s_l = f(p, t, \xi)$	sl_ptXI_SeaWa	= SL_PT XI_SEAWA(P,T,XI)	Specific entropy of subcooled liquid	kJ/(kg*K)	[1], [2]	3/62
$s_{sl} = f(p_s, t_s, \xi_{sl})$	ssl_pstsXisl_SeaWa	= SSL_PST SXISL_SEAWA(PS,TS,XISL)	Specific entropy of saturated liquid	kJ/(kg*K)	[1], [2]	3/63

Functional Dependence	Function Name in Excel®	Call as Fortran Program	Property or Function	Unit of the Result	Reference	Page
$s_{sv} = f(p_s, t_s, \xi_{sl})$	ssv_pstsXisl_SeaWa	= SSV_PSTSXISL_SEAWA(PS,TS,XISL)	Specific entropy of saturated vapor	kJ/(kg*K)	[1], [2]	3/64
$t_s = f(p_s, \xi_{sl})$	ts_psXisl_SeaWa	= TS_PSXISL_SEAWA(PS,XISL)	Boiling temperature	°C	[1], [2], [4]	3/65
$t_{mel} = f(p, \xi)$	tmel_pXi_SeaWa	= TMEL_PXI_SEAWA(P,XI)	Freezing temperature	°C	[1], [4], [5]	3/66
$t_{tr} = f(\xi)$	ttr_Xi_SeaWa	= TTR_XI_SEAWA(T,XI)	Triple point temperature	°C	[1], [4], [5]	3/67
Region = $f(p, t, \xi)$	Region_ptXi_SeaWa	= REGION_PTXI_SEAWA(P,T,XI)	Region	[-]	[1], [2], [4]	3/68
Region = $f(p, h, \xi)$	Region_phXi_SeaWa	= REGION_PHXI_SEAWA(P,H,XI)	Region	[-]	[1], [2], [4]	3/69
Region = $f(p, s, \xi)$	Region_psXi_SeaWa	= REGION_P SXI_SEAWA(P,S,XI)	Region	[-]	[1], [2], [4]	3/70
$t = f(p, h, \xi)$	t_phXi_SeaWa	= T_PHXI_SEAWA(P,H,XI)	Backward function: Temperature from pressure and specific enthalpy	°C	[1], [2], [4]	3/71
$t = f(p, s, \xi)$	t_psXi_SeaWa	= T_P SXI_SEAWA(P,S,XI)	Backward function: Temperature from pressure and specific entropy	°C	[1], [2], [4]	3/72
$u = f(p, t, \xi)$	u_ptXi_SeaWa	= U_PTXI_SEAWA(P,T,XI)	Specific internal energy	kJ/kg	[1], [2]	3/73
$u_l = f(p, t, \xi)$	ul_ptXi_SeaWa	= UL_PTXI_SEAWA(P,T,XI)	Specific internal energy of subcooled liquid	kJ/kg	[1], [2]	3/74
$u_{sl} = f(p_s, t_s, \xi_{sl})$	usl_pstsXisl_SeaWa	= USL_PSTSXISL_SEAWA(PS,TS,XISL)	Specific internal energy of saturated liquid	kJ/kg	[1], [2]	3/75
$u_{sv} = f(p_s, t_s, \xi_{sl})$	usv_pstsXisl_SeaWa	= USV_PSTSXISL_SEAWA(PS,TS,XISL)	Specific internal energy of saturated vapor	kJ/kg	[1], [2]	3/76
$v = f(p, t, \xi)$	v_ptXi_SeaWa	= V_PTXI_SEAWA(P,T,XI)	Specific volume	m³/kg	[1], [2]	3/77
$v_l = f(p, t, \xi)$	vl_ptXi_SeaWa	= VL_PTXI_SEAWA(P,T,XI)	Specific internal energy of subcooled liquid	m³/kg	[1], [2]	3/78
$v_{sl} = f(p_s, t_s, \xi_{sl})$	vsl_pstsXisl_SeaWa	= VSL_PSTSXISL_SEAWA(PS,TS,XISL)	Specific volume of saturated liquid	m³/kg	[1], [2]	3/79
$v_{sv} = f(p_s, t_s, \xi_{sl})$	vsv_pstsXisl_SeaWa	= VSV_PSTSXISL_SEAWA(PS,TS,XISL)	Specific volume of saturated vapor	m³/kg	[1], [2]	3/80



Functional Dependence	Function Name in Excel®	Call as Fortran Program	Property or Function	Unit of the Result	Reference	Page
$w = f(p, t, \xi)$	w_ptXI_SeaWa	= W_PT XI_SEAWA(P,T,XI)	Speed of sound	m/s	[1]	3/81
$w_l = f(p, t, \xi)$	wl_ptXI_SeaWa	= WL_PT XI_SEAWA(P,T,XI)	Speed of sound of liquid	m/s	[1]	3/82
$w_{sl} = f(p_s, t_s, \xi_{sl})$	wsl_pstsXisl_SeaWa	= WSL_PST SXISL_SEAWA(PS,TS,XISL)	Speed of sound of saturated liquid	m/s	[1]	3/83
$w_{sv} = f(p_s, t_s, \xi_{sv})$	wsv_pstsXisl_SeaWa	= WSV_PST SXISL_SEAWA(PS,TS,XISL)	Speed of sound of saturated vapor	m/s	[1]	3/84
$x = f(p, t, \xi)$	x_ptXI_SeaWa	= X_PT XI_SEAWA(P,T,XI)	Vapor fraction	kg/kg	[1]	3/85
$\xi_{sl} = f(p_s, t_s)$	Xisl_psts_SeaWa	= XISL_PSTS_SEAWA(PS,TS)	Mass fraction of sea salt in saturated liquid	kg/kg	[1]	3/86
$\xi_{sv} = f(p_s, t_s)$	Xisv_psts_SeaWa	= XISV_PSTS_SEAWA(PS,TS)	Mass fraction of sea salt in saturated vapor	kg/kg	[1]	3/87

### Range of Validity

Pressure:	0.002 093	$\leq$	$p$	$\leq$	1000	bar
Temperature:	0	$\leq$	$t$	$\leq$	220	°C
Salinity:	0	$\leq$	$\xi$	$\leq$	0.2	kg <sub>salt</sub> /kg <sub>mixture</sub>

### Reference State

Property	SeaWater
Pressure	1.01325 bar
Temperature	0 °C
Salinity	0.003516504 kg/kg
Enthalpy	0 kJ/ kg
Entropy	0 kJ/( kg K)

### Variable Types for Function Call

All functions and variables:	REAL*8
------------------------------	--------

**Property functions with deviation to the range of validity (cp. Chapter. 3)**

## 2. Application of FluidDYM in Dymola®

The FluidDYM Add-In has been developed to calculate thermodynamic properties in Dymola® more conveniently. Within Dymola it enables the direct call of functions relating to seawater from the LibSeaWa property library. The 32-bit version of FluidDYM LibSeaWa runs on both the 32-bit and 64-bit version of DYMOLA®.

### 2.1 Installing FluidDYM

In this section, the installation of FluidDYM and LibSeaWa is described.

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\_FluidDYM\_LibSeaWa.zip," (32-bit version)

"CD\_FluidDYM\_LibSeaWa\_64.zip," (64-bit version)

you will see the folder

CD\_FluidDYM\_LibSeaWa (32-bit version)

CD\_FluidDYM\_LibSeaWa\_64 (64-bit version)

in your Windows Explorer®, Norton Commander® etc.

Now, open this folder by double-clicking on it.

Within the folder for the **32-bit version** you will see the following files

FluidDYM\_LibSeaWa\_Users\_Guide.pdf

FluidDYM\_LibSeaWa\_Setup.exe (32-bit version)

and the folder

"Users\_Guide."

Within the folder for the **64-bit version** you will see the following files

FluidDYM\_LibSeaWa\_Users\_Guide.pdf

FluidDYM\_LibSeaWa\_64\_Setup.msi

Setup.exe

and the folder

"Users\_Guide."

In order to run the installation of **32-bit** FluidDYM including the LibSeaWa property library double-click the file

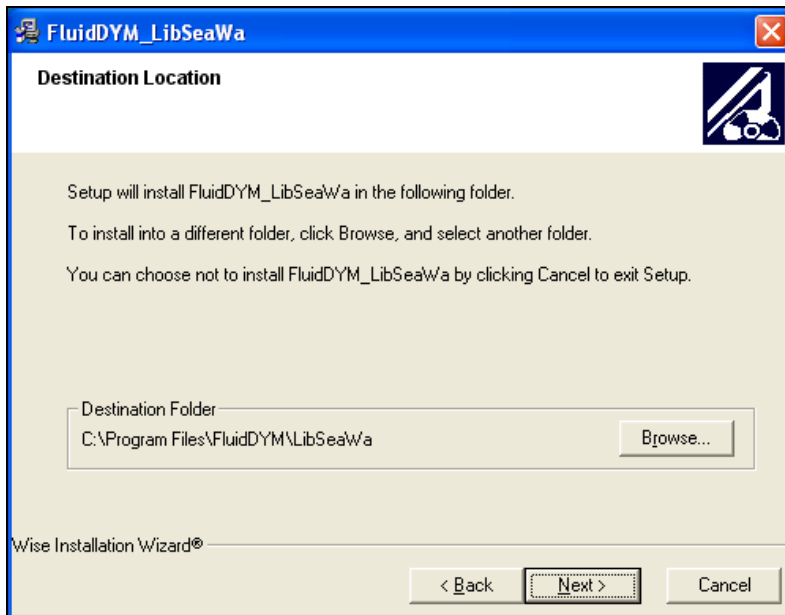
FluidDYM\_LibSeaWa\_Setup.exe.

Installation may start with a window noting that all Windows® programs should be closed. When this is the case, the installation can be continued. Click the "Continue" button.

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

C:\Program Files\FluidDYM\LibSeaWa.

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



**Figure 2.1:** Dialog window "Destination Location"

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

After FluidDYM has been installed, you will see the sentence "FluidDYM LibSeaWa has been successfully installed." Confirm this by clicking the "Finish" button.

The installation of FluidDYM 32-bit has been completed.

In order to run the installation of **64-bit** FluidDYM including the LibSeaWa property library double-click the file

Setup.exe.

Installation may start with a window noting that all Windows® programs should be closed. When this is the case, the installation can be continued. Click the "Continue" button.

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

C:\Users\...\Documents\FuildDYM\_64\LibSeaWa.

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

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

After FluidDYM has been installed, you will see the sentence "FluidDYM LibSeaWa has been successfully installed." Confirm this by clicking the "Finish" button.

The installation of FluidDYM 64-bit has been completed.

The installation program has copied the following files into the directory

C:\Program Files\FuildDYM\LibSeaWa	(for English version of Windows)
C:\Programme\FuildDYM\LibSeaWa	(for German version of Windows)):

- Dynamic link library "LibSeaWa.dll".
- Link up Dynamic link library "LibSeaWa\_Dym.dll" and other necessary system DLL files.
- Library File "LibSeaWa\_DYM.lib"
- Header File "LibSeaWa\_DYM.h" and other necessary system DLL files.
- Modelica File "FluidDYM\_LibSeaWa.mo", includes the following property functions:

a_ptXi_SeaWa	mywl_ptXi_SeaWa
al_tXi_SeaWa	mywsl_pstsXisl_SeaWa
asl_pstsXisl_SeaWa	mySaltl_ptXi_SeaWa
asv_pstsXisl_SeaWa	mySaltsl_pstsXisl_SeaWa
alpha_ptXi_SeaWa	ny_ptXi_SeaWa
alphasl_pstsXisl_SeaWa	nyl_tXi_SeaWa
betal_ptXi_SeaWa	nysl_pstsXisl_SeaWa
betasl_pstsXisl_SeaWa	nysv_pstsXisl_SeaWa
betalsl_ptXi_SeaWa	ps_tsXisl_SeaWa
betalsl_pstsXisl_SeaWa	pmel_tXi_SeaWa
cp_ptXi_SeaWa	ptr_Xi_SeaWa
cpl_ptXi_SeaWa	Pr_ptXi_SeaWa
cpsl_pstsXisl_SeaWa	Prl_tXi_SeaWa
cpsv_pstsXisl_SeaWa	Prsl_pstsXisl_SeaWa
eta_ptXi_SeaWa	Prsv_pstsXisl_SeaWa
etal_tXi_SeaWa	rho_ptXi_SeaWa
etasl_pstsXisl_SeaWa	rho_l_ptXi_SeaWa
etasv_pstsXisl_SeaWa	rhosl_pstsXisl_SeaWa
fl_ptXi_SeaWa	rhosv_pstsXisl_SeaWa
fsl_pstsXisl_SeaWa	s_ptXi_SeaWa
phil_ptXi_SeaWa	sl_ptXi_SeaWa
phisl_pstsXisl_SeaWa	ssl_pstsXisl_SeaWa
h_ptXi_SeaWa	ssv_pstsXisl_SeaWa
hl_ptXi_SeaWa	ts_psXisl_SeaWa
hsl_pstsXisl_SeaWa	tmel_pXi_SeaWa
hsv_pstsXisl_SeaWa	ttr_Xi_SeaWa
kappa_ptXi_SeaWa	Region_ptXi_SeaWa
kappal_ptXi_SeaWa	Region_phXi_SeaWa
kappasl_pstsXisl_SeaWa	Region_psXi_SeaWa
kappasv_pstsXisl_SeaWa	t_phXi_SeaWa
kappaTl_ptXi_SeaWa	t_psXi_SeaWa
kappaTsl_pstsXisl_SeaWa	u_ptXi_SeaWa
kappalsl_ptXi_SeaWa	ul_ptXi_SeaWa
kappalsl_pstsXisl_SeaWa	usl_pstsXisl_SeaWa
lambda_ptXi_SeaWa	usv_pstsXisl_SeaWa
lambdal_tXi_SeaWa	v_ptXi_SeaWa
lambdasl_pstsXisl_SeaWa	vl_ptXi_SeaWa

lambdasv_pstsXisl_SeaWa	vsl_pstsXisl_SeaWa
myl_ptXi_SeaWa	vsv_pstsXisl_SeaWa
mysl_pstsXisl_SeaWa	w_ptXi_SeaWa
wl_ptXi_SeaWa	x_ptXi_SeaWa
wsl_pstsXisl_SeaWa	Xisl_psts_SeaWa
wsv_pstsXisl_SeaWa	Xisv_psts_SeaWa

Now, you have to overwrite the file "LibSeaWa.dll" and the folder "Users\_Guide" in your LibSeaWa directory with the files of the same names provided in your CD folder with FluidDYM.

To do this, open the CD folder "CD\_FluidDYM\_LibSeaWa" in "My Computer" and click on the file "LibSeaWa.dll" in order to highlight it. Hold Ctrl and click on the folder "Users\_Guide" to mark it as well.

Then click on the "Edit" menu in your Explorer and select "Copy".

Now, open your LibSeaWa directory (the standard being

C:\Program Files\FluidDYM\LibSeaWa	(for English version of Windows)
C:\Programme\FluidDYM\LibSeaWa	(for German version of Windows))

and insert the "LibSeaWa.dll" and the "Users\_Guide" folder by clicking the "Edit" menu in your Explorer and then select "Paste".

Answer the question whether you want to replace the files by clicking the "Yes" button. Now, you have overwritten the file "LibSeaWa.dll" and the folder "Users\_Guide" successfully.

Next, copy the folder "Users\_Guide" into your Dymola LibSeaWa directory with the file of the same name provided on your CD with FluidDYM.

To do this, open the CD in "My Computer" and click on the folder "Users\_Guide" to highlight it.

Then click on the "Edit" menu in your Explorer and select "Copy".

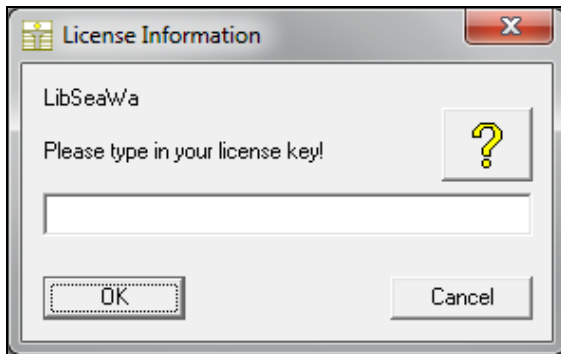
Now, open your Dymola LibSeaWa directory (standard:

C:\Program Files\FluidDYM\LibSeaWa	(for English version of Windows)
C:\Programme\FluidDYM\LibSeaWa	(for German version of Windows))

and insert the folder "Users\_Guide" by clicking the "Edit" menu in your Explorer and then selecting "Paste". Now, the folder "Users\_Guide" has been successfully placed in your installation directory.

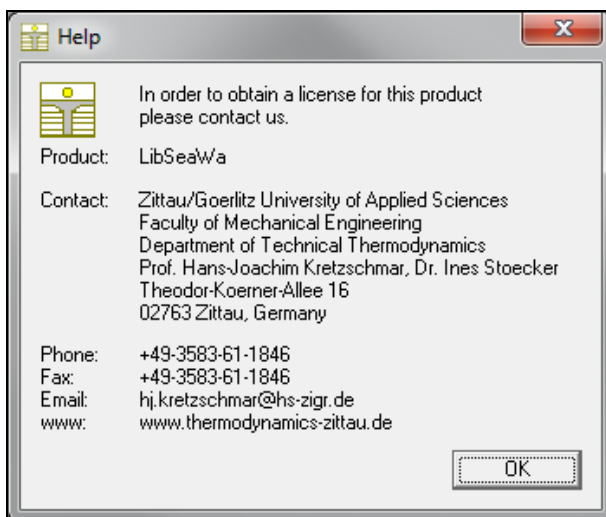
## Licensing the LibSeaWa Property Library

The licensing procedure has to be carried out when Dymola® is running and a model simulation starts. In this case, you will see the "License Information" window (see Figure 2.2).



**Figure 2.2:** "License Information" window

Here you will have to type in the license key which you have obtained from the Zittau/Goerlitz University of Applied Sciences. You can find contact information on the "Content" page of this User's Guide or by clicking the yellow question mark in the "License Information" window. Then the following window will appear:



**Figure 2.3:** "Help" window

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

The "License Information" window will appear every time you start Dymola unless you uninstall FluidDYM\_LibSeaWa according to the description in section 2.3 of this User's Guide.

Should you not wish to license the LibSeaWa property library, you have to delete the files

LibSeaWa.dll	LibSeaWa_DYM.h
LibSeaWa_DYM.dll	LibSeaWa_DYM.mo
LibSeaWa_DYM.lib	

in the installation folder of Dymola® (the standard being

C:\Program Files\FluidDYM\LibSeaWa	(for English version of Windows)
C:\Programme\FluidDYM\LibSeaWa	(for German version of Windows))

using an appropriate program such as Explorer® or Norton Commander.



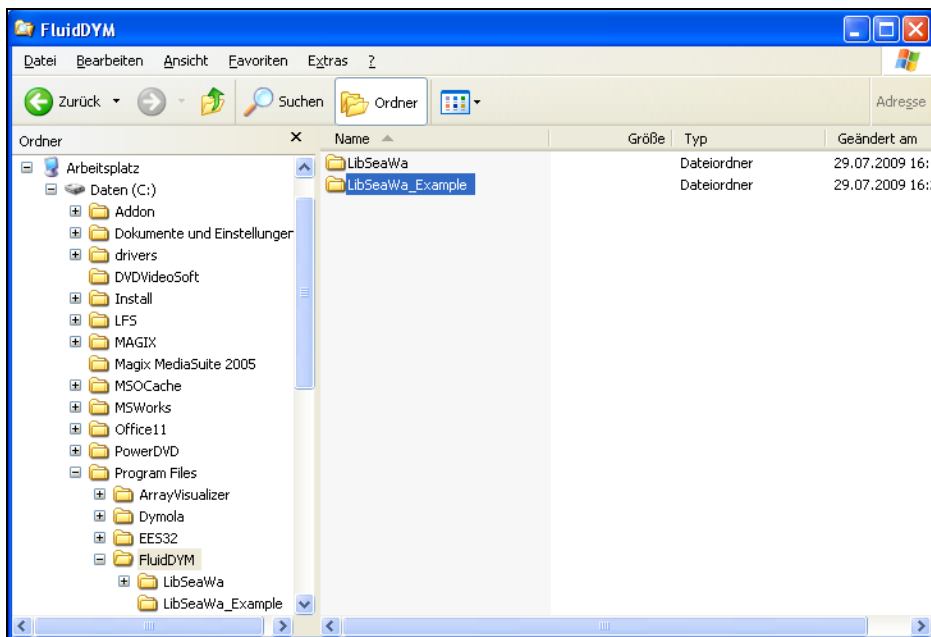
## 2.2 Example: Calculation of the Specific Enthalpy $h = f(p, t, \xi)$ of Seawater

Now we will calculate, step by step, the specific enthalpy  $h$  of seawater as a function of pressure  $p$ , temperature  $t$  and water mass fraction of sea salt  $\xi$ , using Dymola®.

Please carry out the following instructions:

- Start Windows Explorer®, Total Commander®, My Computer or another file manager program.  
The description here refers to Windows Explorer.
- Your Windows Explorer should be set to Details for a better view. Click the "View" (Ansicht) button and select "Details".
- Switch into the program directory of FluidDYM in which you will find the folder "LibSeaWa"; the standard location is:  

C:\Program Files\FluidDYM\LibSeaWa      (for English version of Windows)  
C:\Programme\FluidDYM\LibSeaWa      (for German version of Windows))
- Create the folder "LibSeaWa\_Example" by clicking on "File" in the Explorer menu, then "New" in the menu which appears, and then selecting "Folder". Name the new folder "LibSeaWa\_Example".
- You will see the following window:

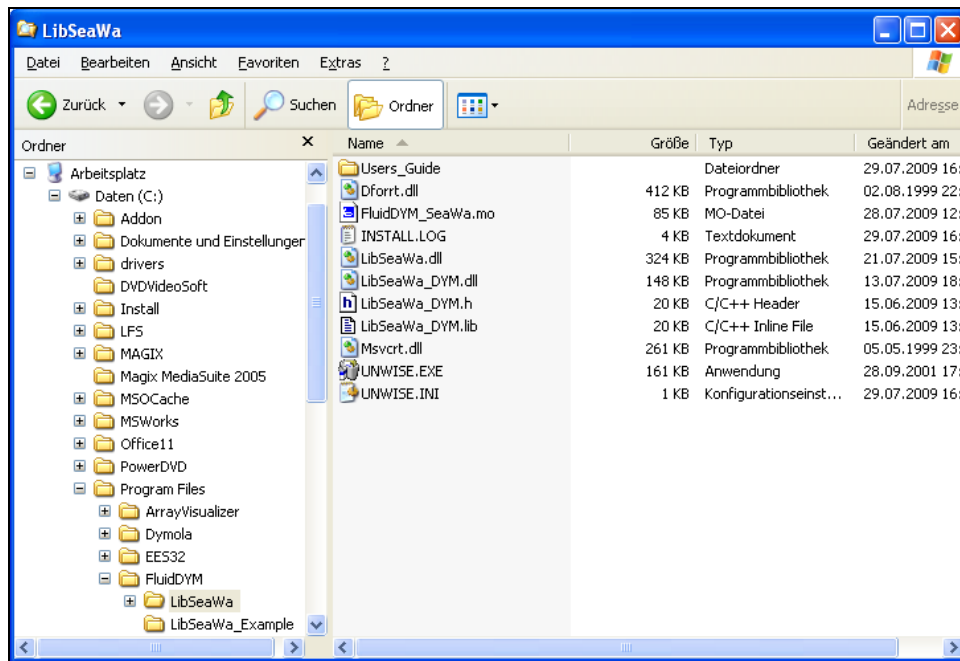


**Figure 2.4:** Highlighted *LibSeaWa\_Example* directory in Program Files

- Switch into the directory "LibSeaWa" within "FluidDYM", the standard being:  

C:\Program Files\FluidDYM\LibSeaWa      (for English version of Windows)  
C:\Programme\FluidDYM\LibSeaWa      (for German version of Windows)).

- You will see the following window:



**Figure 2.5:** *LibSeaWa* directory including installed files

In order to calculate the function  $h = f(p, t, \xi)$ , the following files are necessary. Copy them into the directory

C:\Program Files\FluidDYM\LibSeaWa (for English version of Windows)  
 C:\Programme\FluidDYM\LibSeaWa (for German version of Windows)):

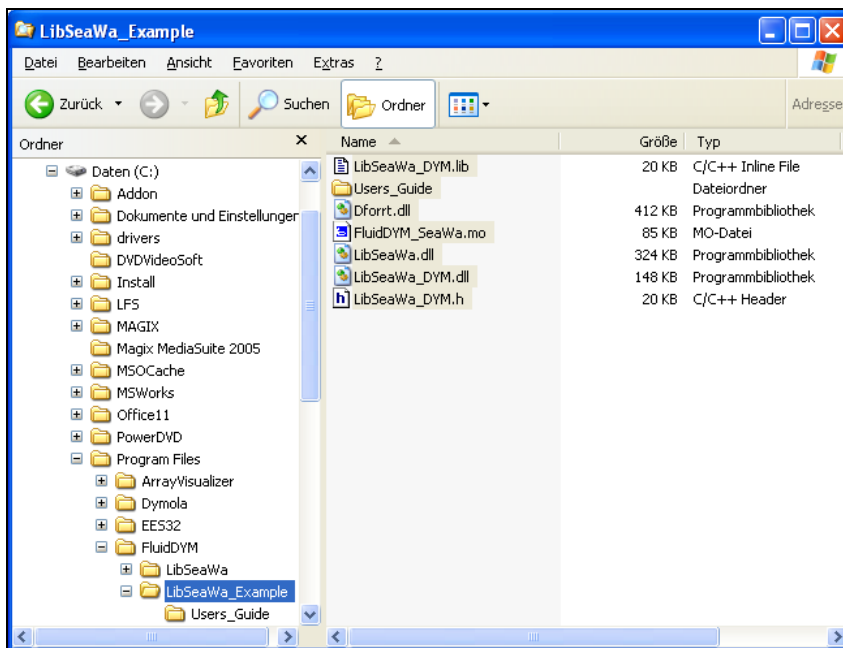
- "LibSeaWa.dll"
- "LibSeaWa\_DYM.dll"
- "LibSeaWa\_DYM.lib"
- "LibSeaWa\_DYM.h"
- "FluidDYM\_SeaWa.mo"
- "DFORRT.dll"
- "msvcr71d.dll"
- the folder "Users\_Guide"

- Mark up these files, then click "Edit" in the upper menu bar and select "Copy".
- Switch into the directory

C:\Program Files\FluidDYM\LibSeaWa\_Example (for English version of Windows)  
 C:\Programme\FluidDYM\LibSeaWa\_Example (for German version of Windows)),

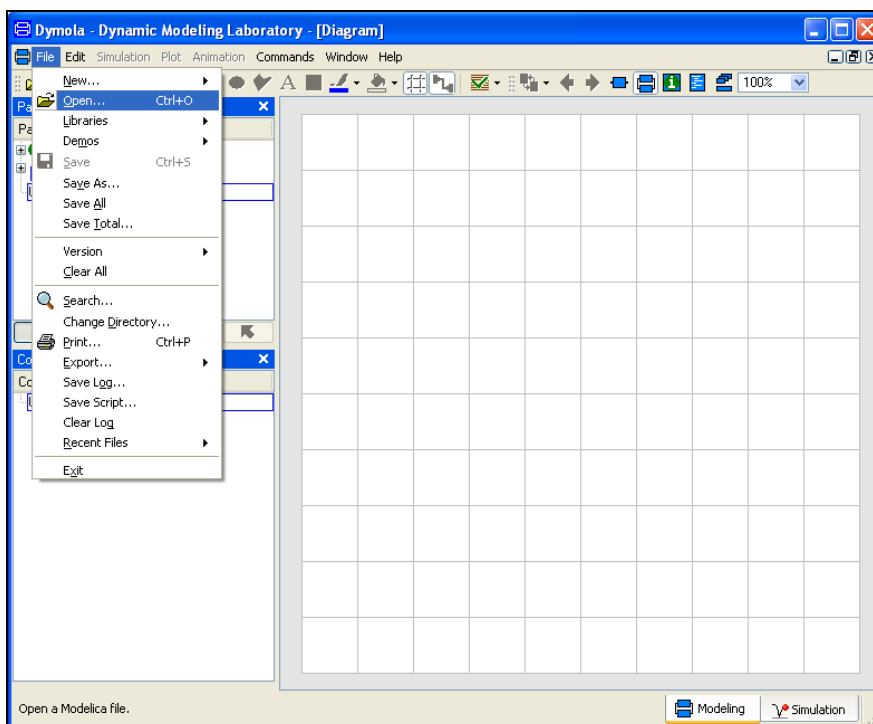
click "Edit" and then "Paste".

- You will see the following window:



**Figure 2.6:** *LibSeaWa\_Example* directory including the newly-copied files

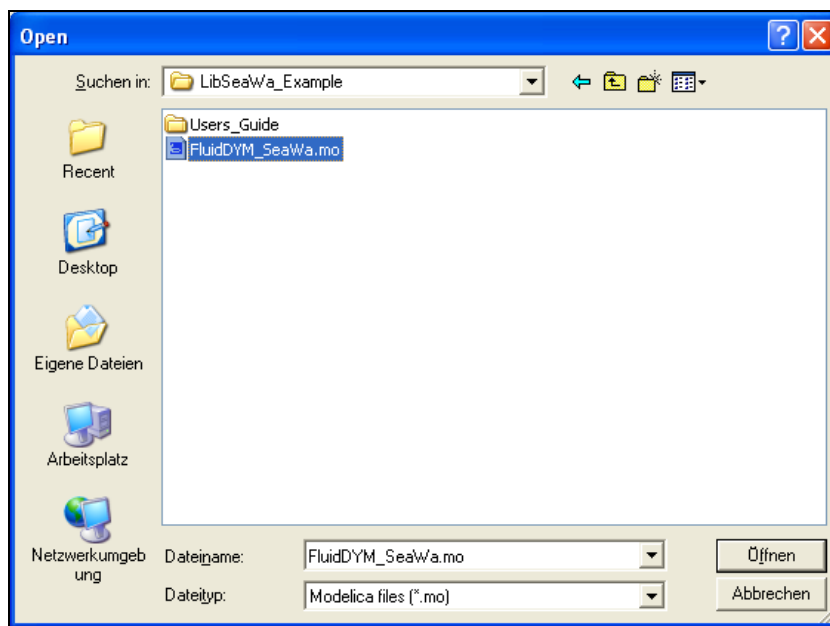
- Start Dymola®.
- Now click on "File" in the Dymola menu bar and select "Open" (see Figure 2.7).



**Figure 2.7:** Selecting the menu entry "Open"

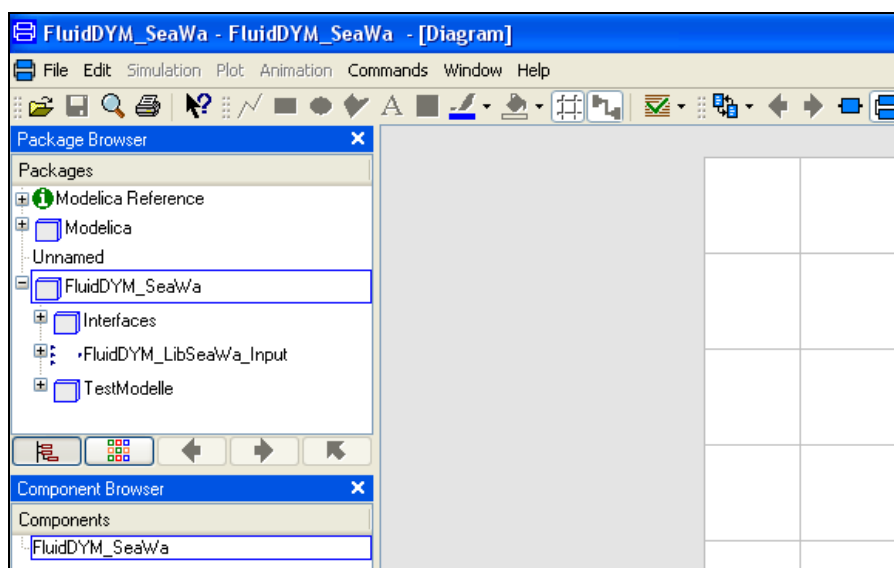
- Search and click on the directory  
     "C:\Program Files\FluidDYM\LibSeaWa\_Example" (for English version of Windows)  
     "C:\Programme\FluidDYM\LibSeaWa\_Example" (for German version of Windows)  
 in the appearing menu.

- Select the "FluidDYM\_SeaWa.mo" file and click on the "Open" button (see Figure 2.8)



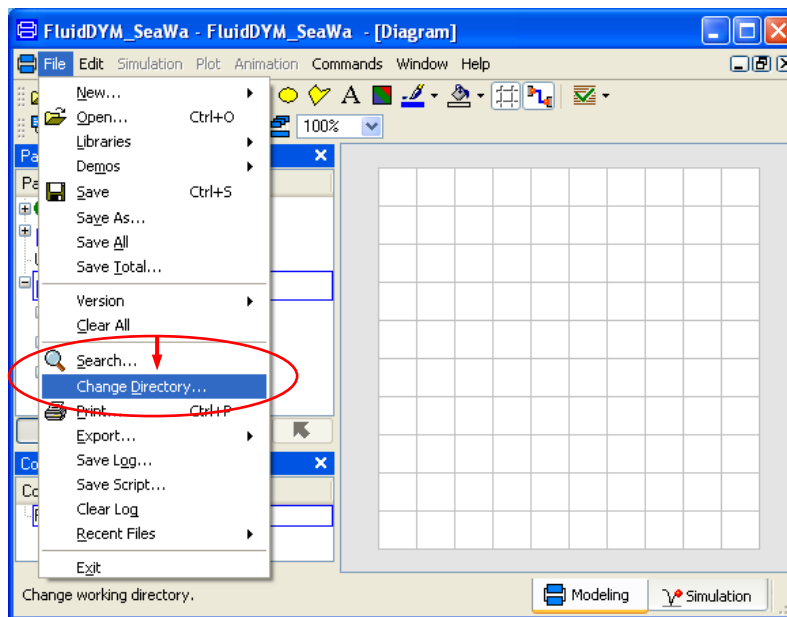
**Figure 2.8:** Selecting the *FluidDYM\_SeaWa.mo* file

- The library will be loaded by Dymola which may take a few seconds.
- After Dymola has finished loading the LibSeaWa library, you will see the window shown in Figure 2.9.



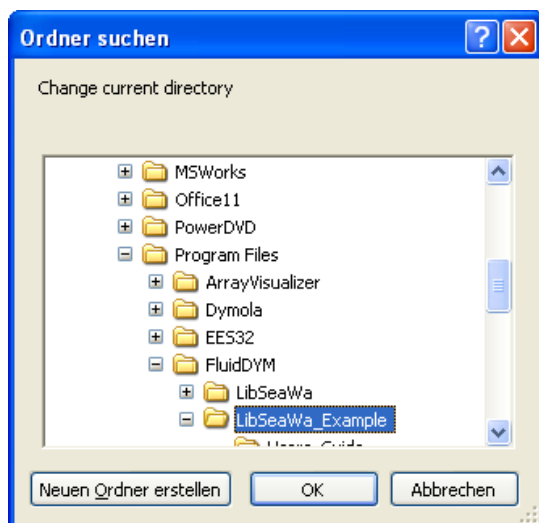
**Figure 2.9:** Dymola window after loading the *LibSeaWa* library

- Now, click on "File" in the Dymola menu bar and select "Change Directory..." in order to open the folder "\LibSeaWa\_Example" (see Figure 2.10).



**Figure 2.10:** Selecting the menu entry "Change Directory..."


- Search and click on the directory
  - "C:\Program Files\FluidDYM\LibSeaWa\_Example" (for English version of Windows)
  - "C:\Programme\FluidDYM\LibSeaWa\_Example" (for German version of Windows)
 in the menu that appears (see Figure 2.11).

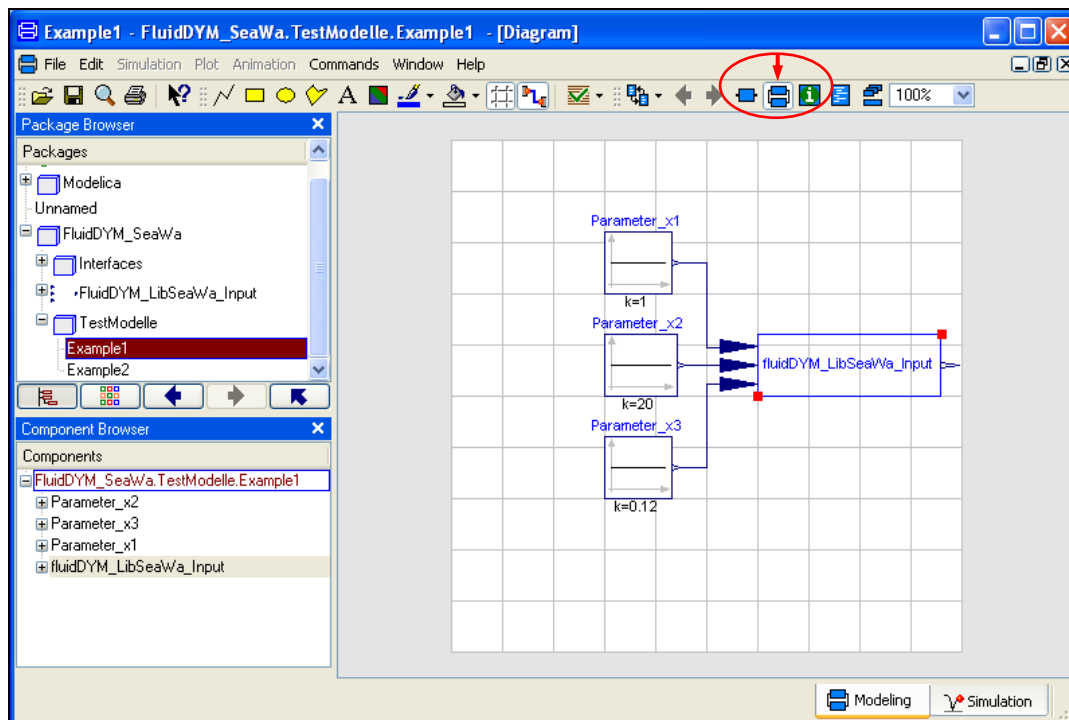


**Figure 2.11:** Selecting the *LibSeaWa\_Example* directory

- Confirm your selection by clicking the "OK" button.

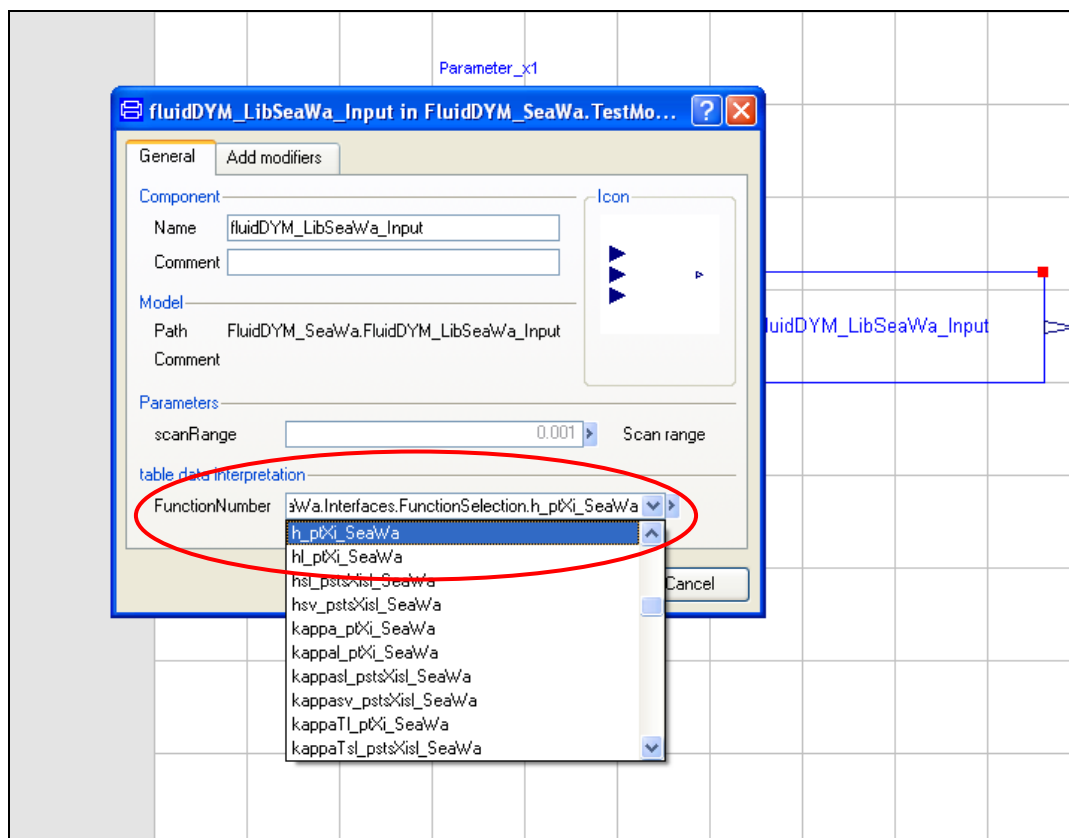
As indicated in the table of property functions in Chapter 1, you have to call up the function "h\_ptXi\_SeaWa" as follows for calculating  $h = f(p, t, \xi)$ .

- Click on the Dymola-Block "Testmodelle," which can be found in the FluidDYM\_SeaWa package in the "Package Browser" on the left hand side of the Dymola window. Here choose Example1 by double-clicking on it.
- Now click on the  button in the Dymola menu bar in order to switch to the Diagram Mode. You will see the following window:



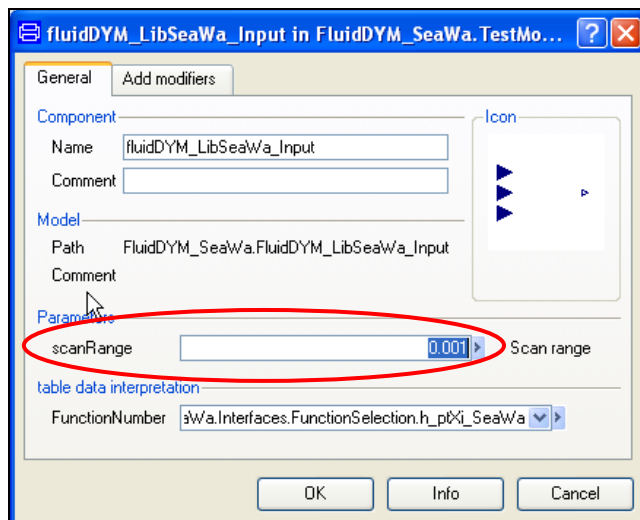
**Figure 2.12:** Dymola in Diagram Mode

- Now double-click on the "fluidDYM\_LibSeaWa\_Input" block on the right hand side of the Dymola window.
- Search and click the "h\_ptXi\_SeaWa" function next to "Function Number" in the menu that appears (see Figure 2.13).



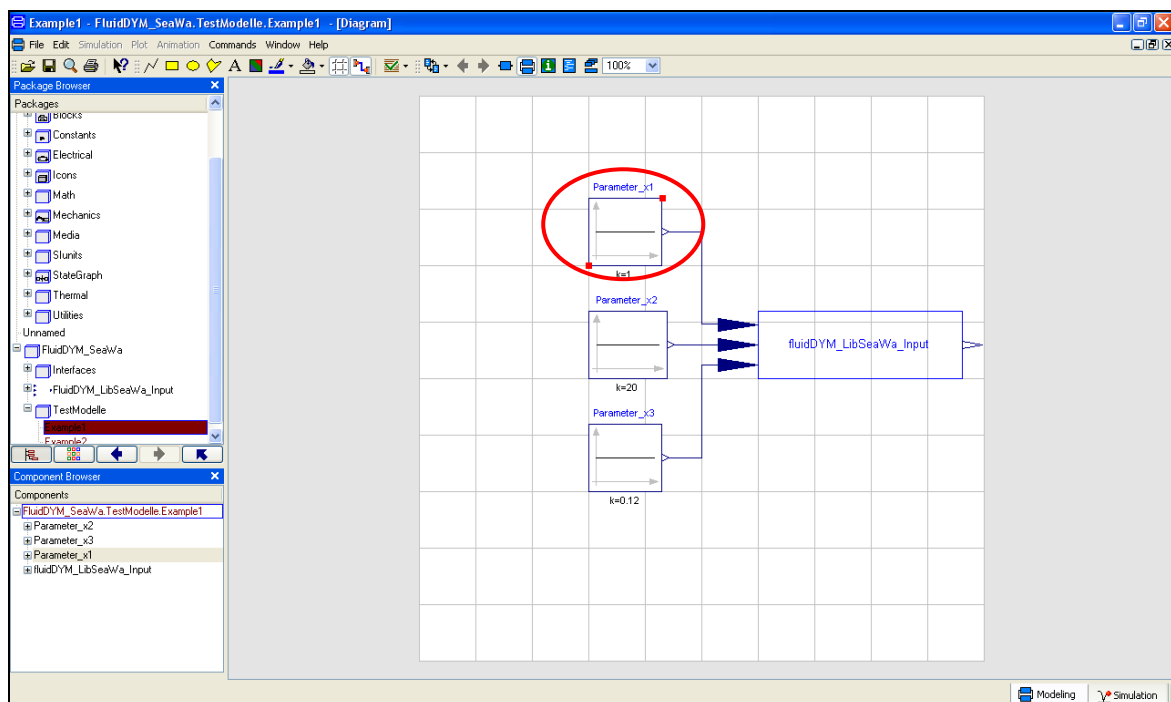
**Figure 2.13:** Choosing the function  $h\_ptXi\_SeaWa$

- You can set the scan range (how many times the property will be calculated per second) next to "scanRange". The preset value 0.001 means that the property will be calculated 1000 times per second. If you enter the value 1, for example, the property will be calculated once per second. Do not change the preset value of 0.001 for our example calculation.



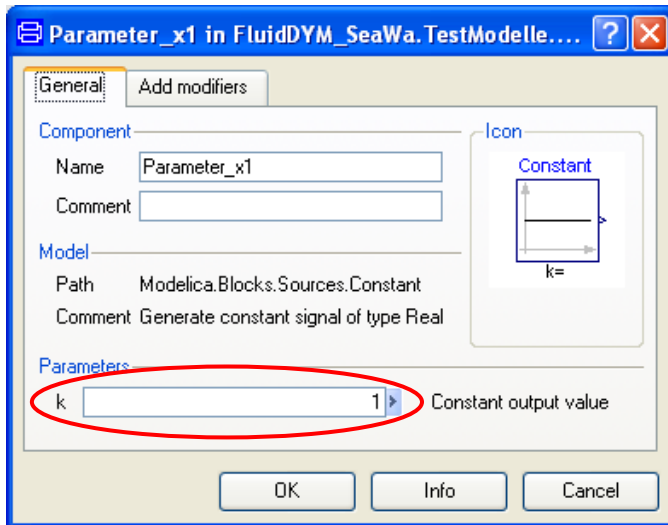
**Figure 2.14:** Setting the scan range

- Now we will configure the input parameters  $x_1$  to  $x_3$ , where  $x_1$  represents the pressure  $p$ ,  $x_2$  represents the temperature  $t$ , and  $x_3$  represents the salinity  $X_i$ . When calculating a function with only two input parameters, the third input parameter  $x_3$  will not be defined.
- First, double click on the "Parameter\_x1" block which represents the first input parameter, here the pressure  $p$  in bar.



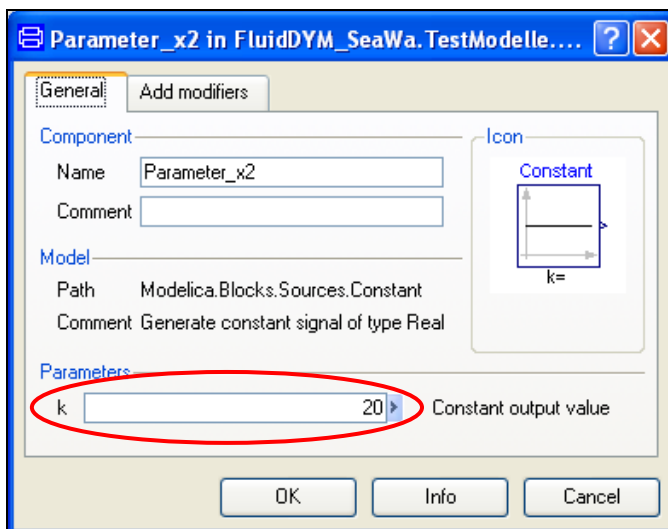
**Figure 2.15:** "Parameter\_x1" block in Dymola

- Enter the value 1 on the line next to "k" in the dialog window which appears and then click the "OK" button (see Figure 2.16).



**Figure 2.16:** Entering the value for the pressure  $p$

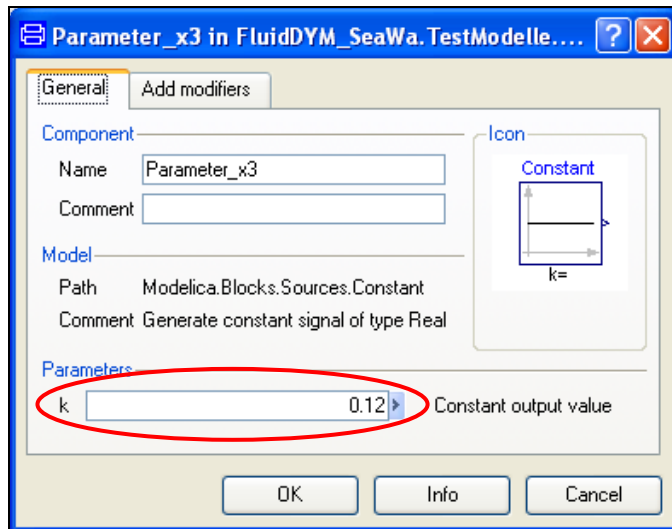
- Now, double click on the "Parameter\_x2" block which represents the second input parameter, here the pressure  $t$  in °C.
- Enter the value 20 on the line next to "k" in the dialog window which appears and then click the "OK" button (see Figure 2.17).



**Figure 2.17:** Entering the value for the temperature  $t$

- Now, double click on the "Parameter\_x3" block which represents the third input parameter, here the salinity  $X_i$  in  $\text{kg}_{\text{Salt}}/\text{kg}_{\text{Seawater}}$ .
- Enter the value 0.12 on the line next to "k" in the dialog window which appears and then click the "OK" button (see Figure 2.18).





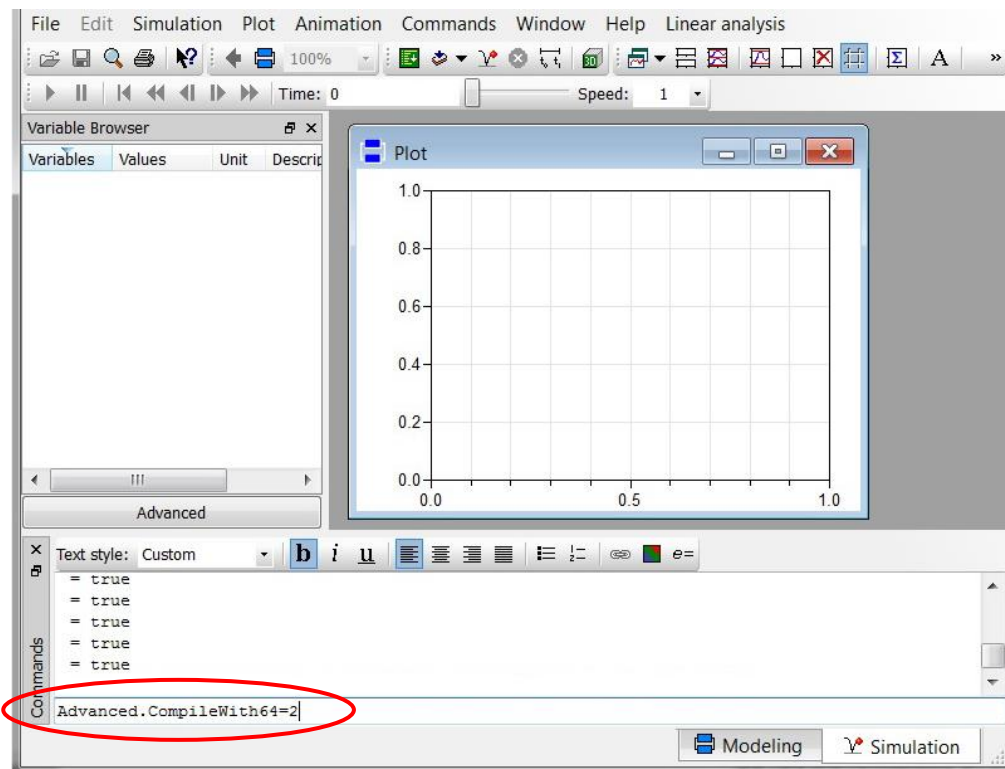
**Figure 2.18:** Entering the value for the salinity  $X_i$

All parameters have now been defined.

### **IMPORTANT NOTICE:**

Per default the 64-bit version of Dymola creates a 32-bit simulation process. If you want to create a 64-bit simulation process you must have installed the 64-bit version of FluidDYM and you now need to enter the following command into the command line of Dymola and confirm your entry by pressing the Enter key:

**"Advanced.CompileWith64=2"**

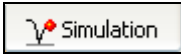


**Figure 2.19:** "Simulation Mode" window with 64-bit command

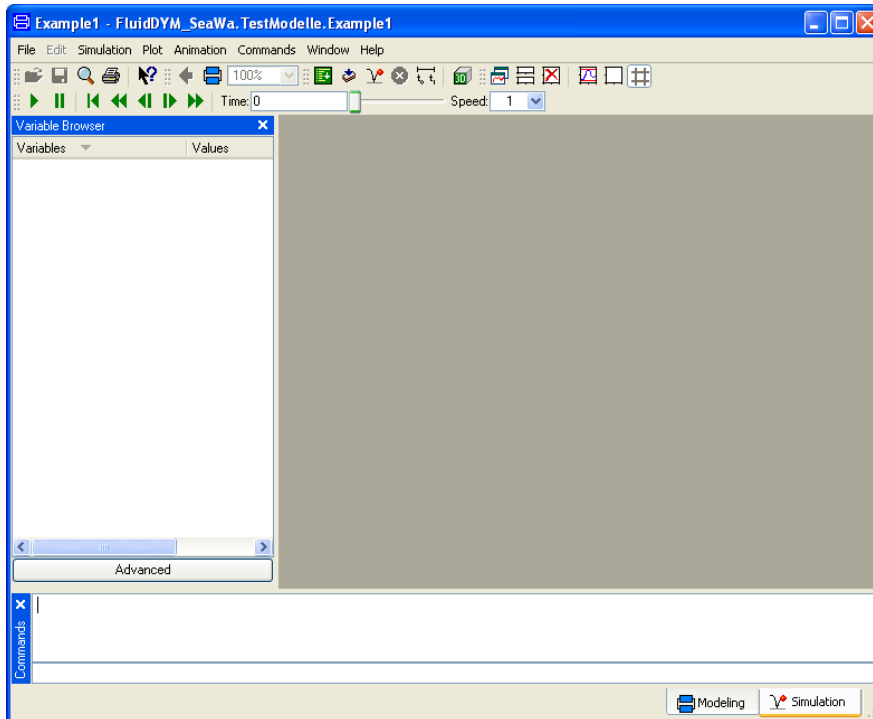
Now, your 64-bit Dymola creates 64-bit simulation processes with FluidDYM.

Please note that if you restart Dymola and want to create 64-bit simulation processes again, you will always have to enter this command anew.


For further information concerning this matter, please see the Dymola user's guide.

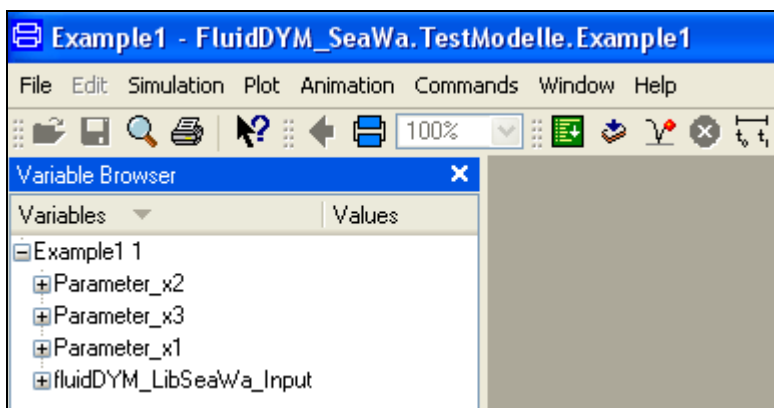
- Click on the  button in the lower right area of Dymola in order to switch into the "Simulation Mode".

In Figure 2.20 you can see what the Dymola "Simulation Mode" looks like.



**Figure 2.20:** "Simulation Mode" window

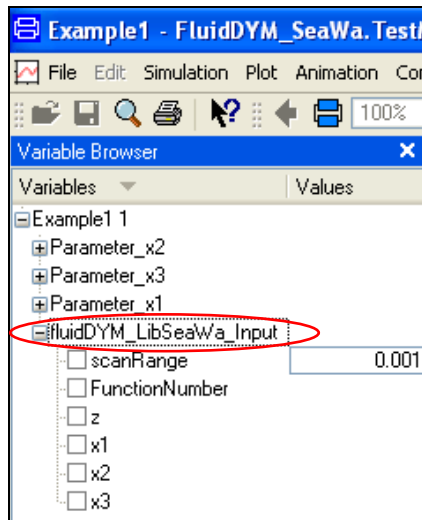
- Click on the "Simulate" Button  in the Dymola menu bar to start the calculation. Now the model will be compiled and the simulation started.
- Afterwards you will see the following entries within the "Variable Browser" window in Dymola (see Figure 2.21):



**Figure 2.21:** "Variable Browser" with new entries

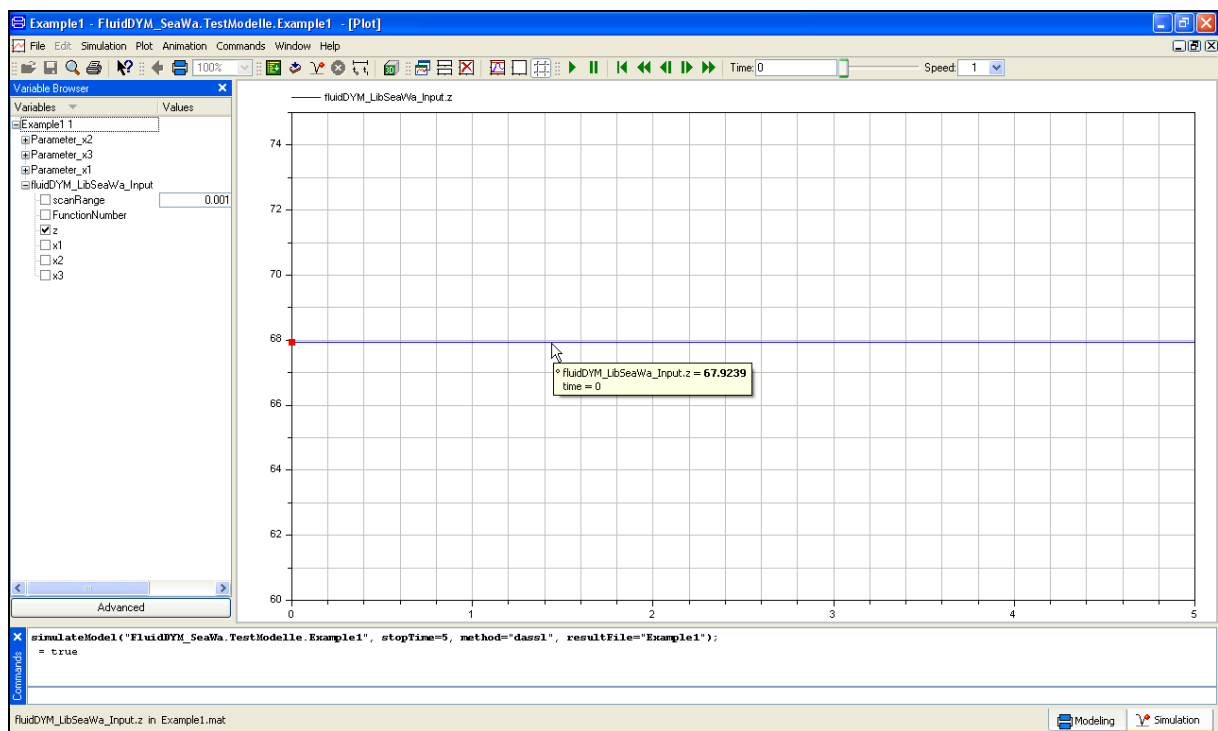
- By clicking on the "NewPlotWindow" Button , a new diagram window will be opened.

- Click on "fluidDYM\_LibSeaWa\_Input" within the "Variable Browser"; then you will see the input and output parameters "scanRange", "FunctionNumber", "z", "x1", "x2" and "x3" (see Figure 2.22).



**Figure 2.22:** Parameters of *fluidDYM\_LibSeaWa\_Input*


- After clicking on the output parameter "z", of the calculated property will be represented graphically in the "PlotWindow".
- Move the mouse over the curve to see the result of the simulation at a specific point in time (see Figure 2.23).



**Figure 2.23:** "DiagramWindow" showing the result


The result for  $h$  appears in the "DiagramWindow"

⇒ The result in our sample calculation here is: " $h = 67.9239$ ". The corresponding unit is kJ/kg (see table of the property functions in Chapter 1).


- Now click on the Modeling button  in the lower right area of Dymola in order to switch into the "Modeling Mode". Here you can arbitrarily change the values for  $p$ ,  $t$ , or  $X_i$  in the appropriate blocks.

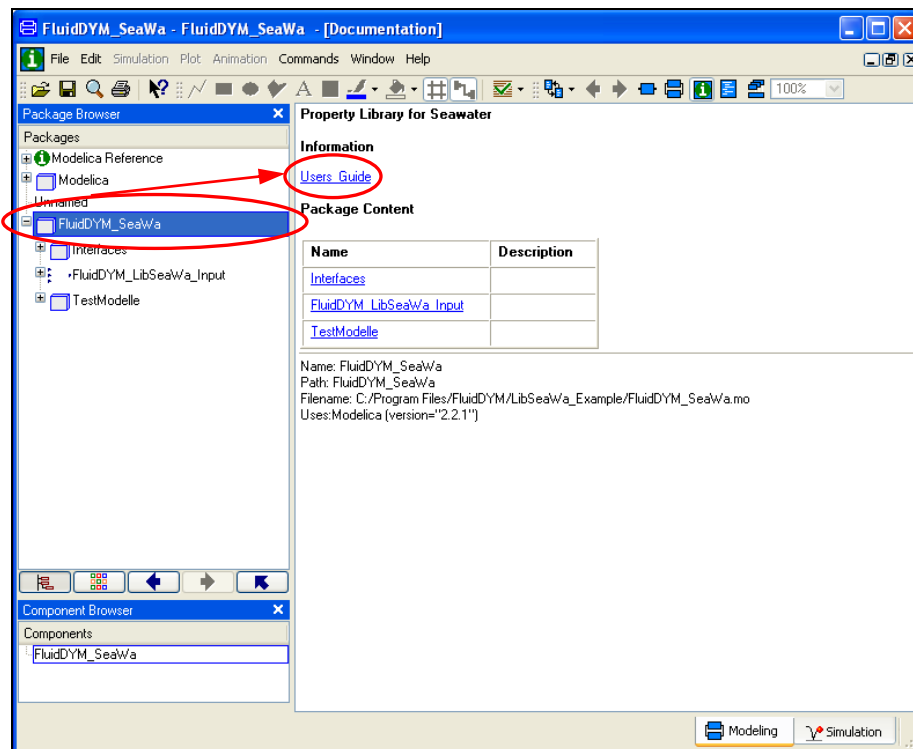
## Help Systems in Dymola®

Dymola® provides detailed help functions. You can choose to read the program documentation or the help page of a specific property function, as desired.

Within the "Modeling-Mode"  the help may be accessed via two different steps.

First we will show you how to access the program documentation of the property library.


- Make sure Dymola is set to the "Modeling-Mode".
- Now click the  button in the Dymola menu bar to choose the "Documentation Mode".
- Double-click on the "FluidDYM\_SeaWa" Block at the left and then click on "Users\_Guide" (see Figure 2.24).

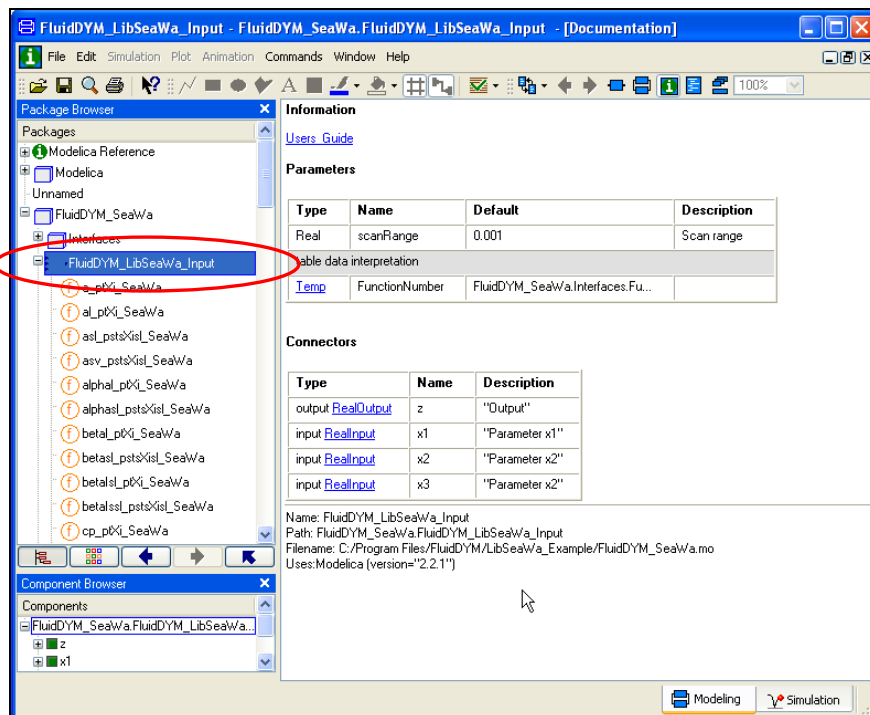


**Figure 2.24:** Selecting the "Users\_Guide"

- The program documentation will be displayed within your default web browser.

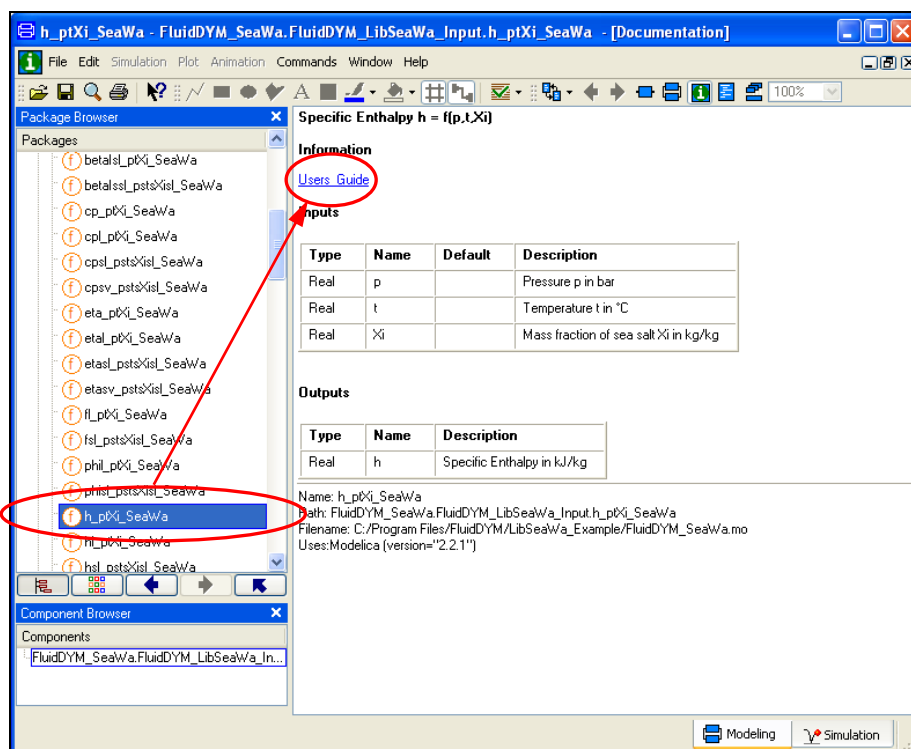
Now, we will show you how to access the help page of a specific property function.

- Make sure Dymola is set to "Modeling-Mode".
- Now click the  button in the Dymola menu bar to choose "Documentation Mode".
- Double-click on the "FluidDYM\_SeaWa\_Input" block on the left (see Figure 2.25).



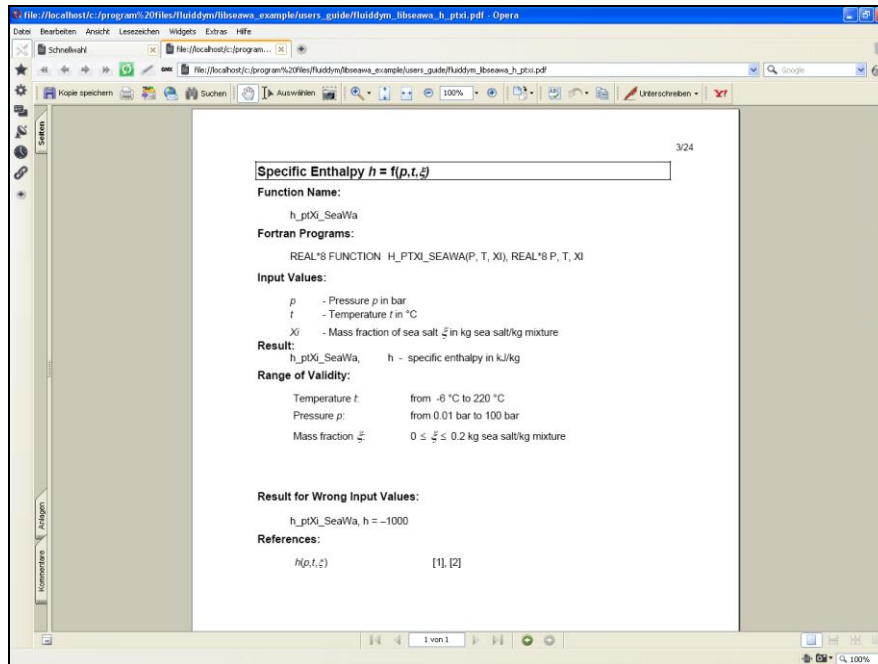
**Figure 2.25:** Selected "FluidDYM\_SeaWa\_Input" Block

- Below "FluidDYM\_SeaWa\_Input" you will see all functions of the LibSeaWa property function (see Figure 2.25).
- Now select a function, e.g. "h\_ptXi\_SeaWa", and then click on "Users\_Guide" (see Figure 2.26).



**Figure 2.26:** Marking the "h\_ptXi\_SeaWa" function and selecting the "Users\_Guide"

- You will now see the help page of the selected function, here "h\_ptXi\_SeaWa", in your default web browser (see Figure 2.27).



**Figure 2.27:** Help page of the function "h\_ptXi\_SeaWa" in the web browser

## 2.3 Removing LibSeaWa in Dymola

In order to remove the property library LibSeaWa from your hard drive in Windows®, click "Start" in the lower task bar, then "Settings" and "Control Panel".

Afterwards double-click on "Add or Remove Programs".

In the list box of the "Add or Remove Programs" menu which appears, select "FluidDYM LibSeaWa" by clicking on it and then clicking the "Change/Remove" button.

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

Confirm the "Perform Uninstall" menu which appears by clicking the "Finish" button.

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

"FluidDYM LibSeaWa" has now been removed.

If LibSeaWa is the only library installed, the directory "FluidDYM" will be removed as well.

### **3. Program Documentation**

**Thermal Diffusivity  $a = f(p, t, \xi)$** 
**Function Name:**

a\_ptXi\_SeaWa

**Fortran Programs:**

```
REAL*8 FUNCTION A_PTXI_SEAWA(P, T, XI), REAL*8 P, T, XI
```

**Input Values:**

$p$  - Pressure  $p$  in bar  
 $t$  - Temperature  $t$  in °C  
 $Xi$  - Mass fraction of sea salt  $\xi$  in kg sea salt/kg mixture

**Result:**

a\_ptXi\_SeaWa,  $a$  - Thermal diffusivity in m<sup>2</sup>/s

**Range of Validity:**

Temperature  $t$ : from 0 °C to 220 °C  
 Pressure  $p$ : from  $p_s(t = 0 \text{ °C}, \xi = 0.2)$  to 100 bar and  $p \geq p_{\text{mel}}(t, \xi)$   
 Mass fraction  $\xi$ :  $0 \leq \xi \leq 0.2$  kg sea salt/kg mixture

**Comments:**

- Thermal diffusivity  $a = \frac{\lambda}{\rho * c_p}$
- This function is not defined in the wet steam region.

**Result for Wrong Input Values:**

a\_ptXi\_SeaWa,  $a = -1000$

**References:**

$a(p, t, \xi)$  [1], [2]



--

**Thermal Diffusivity of Liquid Seawater  $a = f(t, \xi)$** 
**Function Name:**

al\_tXi\_SeaWa

**Fortran Programs:**

```
REAL*8 FUNCTION AL_TXI_SEAWA(T, XI), REAL*8 T, XI
```

**Input Values:**

$t$  - Temperature  $t$  in °C

$X_i$  - Mass fraction of sea salt  $\xi$  in kg sea salt/kg mixture

**Result:**

al\_tXi\_SeaWa, al - Thermal diffusivity of liquid seawater in m<sup>2</sup>/s

**Range of Validity:**

Temperature  $t$ : from 0 °C to 220 °C

Mass fraction  $\xi$ :  $0 \leq \xi \leq 0.2$  kg sea salt/kg mixture

**Comments:**

- Thermal diffusivity  $a = \frac{\lambda}{\rho * c_p}$

**Result for Wrong Input Values:**

al\_tXi\_SeaWa, al = - 1000

**References:**

$a(t, \xi)$  [1]

**Thermal Diffusivity of Saturated Liquid  $a_{sl} = f(p_s, t_s, \xi^{sl})$** 
**Function Name:**

asl\_pstsXisl\_SeaWa

**Fortran Programs:**

REAL\*8 FUNCTION ASL\_PSTXISL\_SEAWA(PS, TS, XISL), REAL\*8 PS, TS, XISL

**Input Values:**

$ps$  - Pressure  $p$  in bar  
 $ts$  - Temperature  $t$  in °C  
 $Xisl$  - Mass fraction of sea salt  $\xi$  in kg sea salt/kg mixture

**Result:**

 asl\_pstsXisl\_SeaWa, asl - Thermal diffusivity of saturated seawater in m<sup>2</sup>/s

**Range of Validity:**

Temperature  $t$ : from 0 °C to 220 °C  
 Mixture pressure  $p$ : from  $p_s(t = 0 \text{ °C}, \xi = 0.2)$  to  $p_s(t = 220 \text{ °C}, \xi = 0)$   
 Mass fraction  $\xi$ :  $0 \leq \xi \leq 0.2$  kg sea salt/kg mixture

**Comments:**

- Thermal diffusivity  $a = \frac{\lambda}{\rho * c_p}$

**Possible input variants:**

$$\begin{aligned}
 a^{sl} &= f(-1000, t_s, \xi^{sl}) \\
 a^{sl} &= f(p_s, -1000, \xi^{sl}) \\
 a^{sl} &= f(p_s, t_s, -1000) \\
 a^{sl} &= f(p_s, t_s, \xi^{sl})
 \end{aligned}$$

**Result for Wrong Input Values:**

asl\_pstsXisl\_SeaWa, asl = -1000

**References:**
 $a(p, t, \xi)$  [1],[2]

**Thermal Diffusivity of Saturated Vapor  $a_{sv} = f(p_s, t_s, \xi^{sl})$** 
**Function Name:**

asv\_pstsXisl\_SeaWa

**Fortran Programs:**

```
REAL*8 FUNCTION ASV_PSTSXISL_SEAWA(PS,TS,XISL), REAL*8 PS, TS,XISL
```

**Input Values:**

$ps$  - Pressure  $p$  in bar  
 $ts$  - Temperature  $t$  in °C  
 $Xisl$  - Mass fraction of sea salt  $\xi$  in kg sea salt/kg mixture

**Result:**

asv\_pstsXisl\_SeaWa, asv - Thermal diffusivity of saturated seawater in m<sup>2</sup>/s

**Range of Validity:**

Temperature  $t$ : from 0 °C to 220 °C  
 Mixture pressure  $p$ : from  $p_s(t = 0 \text{ °C}, \xi = 0.2)$  to  $p_s(t = 220 \text{ °C}, \xi = 0)$   
 Mass fraction  $\xi$ :  $0 \leq \xi \leq 0.2$  kg sea salt/kg mixture

**Comments:**

- Thermal diffusivity  $a = \frac{\lambda}{\rho * c_p}$

**Result for Wrong Input Values:**

asv\_pstsXisl\_SeaWa, asv = -1000

**References:**

$a(p, t, \xi)$  [1],[2]

**Thermal Expansion Coefficient of Liquid Seawater  $\alpha_l = f(p, t, \xi)$** 
**Function Name:**

alpha\_ptXi\_SeaWa

**Fortran Programs:**

```
REAL*8 FUNCTION  ALPHAL_PTXI_SEAWA(P, T, XI), REAL*8 P, T, XI
```

**Input Values:**

$p$  - Pressure  $p$  in bar  
 $t$  - Temperature  $t$  in °C  
 $Xi$  - Mass fraction of sea salt  $\xi$  in kg sea salt/kg mixture

**Result:**

alpha\_ptXi\_SeaWa, alpha - Thermal expansion of liquid seawater in 1/K

**Range of Validity:**

Temperature  $t$ : from 0 °C to 80 °C  
 Pressure  $p$ : from  $p_s(t = 0 \text{ °C}, \xi = 0.12)$  bar to 1000 bar and  $p \geq p_{\text{mel}}(t, \xi)$   
 Mass fraction  $\xi$ :  $0 \leq \xi \leq 0.12$  kg sea salt/kg mixture

**Result for Wrong Input Values:**

alpha\_ptXi\_SeaWa, alpha = -1000

**References:**

$\alpha(p, t, \xi)$  [1]

**Thermal Expansion Coefficient of Saturated Liquid  $a_{sl} = f(p_s, t_s, \xi^{sl})$** 
**Function Name:**

alphasl\_pstsXisl\_SeaWa

**Fortran Programs:**

```
REAL*8 FUNCTION ALPHASL_PSTSXISL_SEAWA(PS, TS, XISL),
REAL*8 PS, TS, XISL
```

**Input Values:**

$ps$  - Pressure  $p$  in bar  
 $ts$  - Temperature  $t$  in °C  
 $Xisl$  - Mass fraction of sea salt  $\xi$  in kg sea salt/kg mixture

**Result:**

alphasl\_pstsXisl\_SeaWa, alphasl - Thermal expansion of saturated seawater in 1/K

**Range of Validity:**

Temperature  $t$ : from 0 °C to 80 °C  
Mixture pressure  $p$ : from  $p_s(t = 0 \text{ °C}, \xi = 0.12)$  to  $p_s(t = 80 \text{ °C}, \xi = 0)$   
Mass fraction  $\xi$ :  $0 \leq \xi \leq 0.12$  kg sea salt/kg mixture

**Possible input variants:**

$$\alpha^{sl} = f(-1000, t_s, \xi^{sl})$$

$$\alpha^{sl} = f(p_s, -1000, \xi^{sl})$$

$$\alpha^{sl} = f(p_s, t_s, -1000)$$

$$\alpha^{sl} = f(p_s, t_s, \xi^{sl})$$

**Result for Wrong Input Values:**

alphasl\_pstsXisl\_SeaWa, alphasl = -1000

**References:**

$\alpha(p, t, \xi)$  [1]

**Haline Contraction Coefficient of Liquid Seawater  $\beta_l = f(p, t, \xi)$** 
**Function Name:**

betal\_ptXi\_SeaWa

**Fortran Programs:**

```
REAL*8 FUNCTION BETAL_PTXI_SEAWA(P, T, XI), REAL*8 P, T, XI
```

**Input Values:**

$p$         - Pressure  $p$  in bar  
 $t$         - Temperature  $t$  in °C  
 $Xi$        - Mass fraction of sea salt  $\xi$  in kg sea salt/kg mixture

**Result:**

betal\_ptXi\_SeaWa, betal - Haline contraction coefficient of liquid seawater in kg/kg

**Range of Validity:**

Temperature  $t$ :            from 0 °C to 80 °C  
 Pressure  $p$ :                from  $p_s(t = 0 \text{ °C}, \xi = 0.12)$  bar to 1000 bar and  $p \geq p_{\text{mel}}(t, \xi)$   
 Mass fraction  $\xi$ :             $0 \leq \xi \leq 0.12$  kg sea salt/kg mixture

**Result for Wrong Input Values:**

betal\_ptXi\_SeaWa, betal = -1000

**References:**

$\beta(p, t, \xi)$  [1]

**Haline Contraction Coefficient of Saturated Liquid  $\beta_{sl} = f(p_s, t_s, \xi^{sl})$** 
**Function Name:**

betasl\_pstsXisl\_SeaWa

**Fortran Programs:**

```
REAL*8 FUNCTION BETASL_PSTSXISL_SEAWA(PS, TS, XISL),
REAL*8 PS, TS, XISL
```

**Input Values:**

$ps$  - Pressure  $p$  in bar  
 $ts$  - Temperature  $t$  in °C  
 $Xisl$  - Mass fraction of sea salt  $\xi$  in kg sea salt/kg mixture

**Result:**

betasl\_pstsXisl\_SeaWa, betasl - Haline contraction coefficient of saturated seawater in 1/K

**Range of Validity:**

Temperature  $t$ : from 0 °C to 80 °C  
Mixture pressure  $p$ : from  $p_s(t = 0 \text{ °C}, \xi = 0.12)$  to  $p_s(t = 80 \text{ °C}, \xi = 0)$   
Mass fraction  $\xi$ :  $0 \leq \xi \leq 0.12$  kg sea salt/kg mixture

**Possible input variants:**

$$\begin{aligned}\beta^{sl} &= f(-1000, t_s, \xi^{sl}) \\ \beta^{sl} &= f(p_s, -1000, \xi^{sl}) \\ \beta^{sl} &= f(p_s, t_s, -1000) \\ \beta^{sl} &= f(p_s, t_s, \xi^{sl})\end{aligned}$$

**Result for Wrong Input Values:**

betasl\_ptXi\_SeaWa, betasl = -1000

**References:**

$\beta(p, t, \xi)$  [1]



**Isentropic Temperature - Pressure Coefficient (Adiabatic Lapse Rate) of Liquid Seawater  $\beta_{\text{sl}} = f(p, t, \xi)$**

**Function Name:**

betalsl\_ptXi\_SeaWa

**Fortran Programs:**

```
REAL*8 FUNCTION BETAISL_PTXI_SEAWA(P, T, XI), REAL*8 P, T, XI
```

**Input Values:**

$p$  - Pressure  $p$  in bar  
 $t$  - Temperature  $t$  in °C  
 $\xi$  - Mass fraction of sea salt  $\xi$  in kg sea salt/kg mixture

**Result:**

betalsl\_ptXi\_SeaWa, betalsl - adiabatic lapse rate in K/kPa

**Range of Validity:**

Temperature  $t$ : from 0 °C to 80 °C  
 Pressure  $p$ : from  $p_s(t = 0 \text{ °C}, \xi = 0.12)$  bar to 1000 bar and  $p \geq p_{\text{mel}}(t, \xi)$   
 Mass fraction  $\xi$ :  $0 \leq \xi \leq 0.12$  kg sea salt/kg mixture

**Result for Wrong Input Values:**

betalsl\_ptXi\_SeaWa, betalsl = -1000

**References:**

$\beta_{\text{sl}}(p, t, \xi)$  [1]

**Isentropic Temperature - Pressure Coefficient (Adiabatic Lapse Rate) of Saturated Liquid  $\beta_{ls}^{sl} = f(p_s, t_s, \xi^{sl})$**

**Function Name:**

betalssl\_pstsXisl\_SeaWa

**Fortran Programs:**

```
REAL*8 FUNCTION BETAISL_PSTSL_SEAWA(PS, TS, XISL),  
REAL*8 PS, TS, XISL
```

**Input Values:**

$p_s$  - Pressure  $p$  in bar  
 $t_s$  - Temperature  $t$  in °C  
 $Xisl$  - Mass fraction of sea salt  $\xi$  in kg sea salt/kg mixture

**Result:**

betalssl\_pstsXisl\_SeaWa, betalssl - Adiabatic lapse rate of saturated seawater in K/kPa

**Range of Validity:**

Temperature  $t$ : from 0 °C to 80 °C  
 Mixture pressure  $p$ : from  $p_s(t = 0 \text{ °C}, \xi = 0.12)$  to  $p_s(t = 80 \text{ °C}, \xi = 0)$   
 Mass fraction  $\xi$ :  $0 \leq \xi \leq 0.12$  kg sea salt/kg mixture

**Possible input variants:**

$$\begin{aligned}\beta_{ls}^{sl} &= f(-1000, t_s, \xi^{sl}) \\ \beta_{ls}^{sl} &= f(p_s, -1000, \xi^{sl}) \\ \beta_{ls}^{sl} &= f(p_s, t_s, -1000) \\ \beta_{ls}^{sl} &= f(p_s, t_s, \xi^{sl})\end{aligned}$$

**Result for Wrong Input Values:**

betalssl\_pstsXisl\_SeaWa, betalssl = -1000

**References:**

$$\beta_{ls}(p, t, \xi) \quad [1]$$

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

cp\_ptXi\_SeaWa

**Fortran Programs:**

```
REAL*8 FUNCTION CP_PTXI_SEAWA(P, T, XI), REAL*8 P, T, XI
```

**Input Values:**

$p$  - Pressure  $p$  in bar  
 $t$  - Temperature  $t$  in °C  
 $\xi$  - Mass fraction of sea salt  $\xi$  in kg sea salt/kg mixture

**Result:**

cp\_ptXi\_SeaWa, cp - Specific isobaric heat capacity in kJ/(kg\*K)

**Range of Validity:**

Temperature  $t$ : from 0 °C to 220 °C  
 Pressure  $p$ : from  $p_s(t = 0 \text{ °C}, \xi = 0.12)$  bar to 1000 bar and  $p \geq p_{\text{mel}}(t, \xi)$   
 Mass fraction  $\xi$ :  $0 \leq \xi \leq 0.2$  kg sea salt/kg mixture

**Comments:**

- This function is not defined in the wet steam region.

**Result for Wrong Input Values:**

cp\_ptXi\_SeaWa, cp = -1000

**References:**

$c_p(p, t, \xi)$  [1], [2]

**Specific Isobaric Heat Capacity of Liquid Seawater  $c_{pl} = f(p, t, \xi)$** 
**Function Name:**

cpl\_ptXi\_SeaWa

**Fortran Programs:**

```
REAL*8 FUNCTION CPL_PTXI_SEAWA(P, T, XI), REAL*8 P, T, XI
```

**Input Values:**

$p$  - Pressure  $p$  in bar  
 $t$  - Temperature  $t$  in °C  
 $Xi$  - Mass fraction of sea salt  $\xi$  in kg sea salt/kg mixture

**Result:**

cpl\_ptXi\_SeaWa, cpl - Specific isobaric heat capacity of liquid seawater in kJ/(kg\*K)

**Range of Validity:**

Temperature  $t$ : from 0 °C to 220 °C  
 Pressure  $p$ : from  $p_s(t = 0 \text{ °C}, \xi = 0.2)$  bar to 1000 bar and  $p \geq p_{\text{mel}}(t, \xi)$   
 Mass fraction  $\xi$ :  $0 \leq \xi \leq 0.2$  kg sea salt/kg mixture

**Result for Wrong Input Values:**

cpl\_ptXi\_SeaWa, cpl = -1000

**References:**

$c_p(p, t, \xi)$  [1], [2]

**Specific Isobaric Heat Capacity of Saturated Liquid  $c_{p_{sl}} = f(p_s, t_s, \xi^{sl})$**

**Function Name:**

cpsl\_pstsXisl\_SeaWa

**Fortran Programs:**

```
REAL*8 FUNCTION CPSL_PSTSXISL_SEAWA(PS, TS, XISL), REAL*8 PS, TS, XISL
```

**Input Values:**

*ps* - Pressure  $p$  in bar  
*ts* - Temperature  $t$  in °C  
*Xisl* - Mass fraction of sea salt  $\xi$  in kg sea salt/kg mixture

**Result:**

cpsl\_pstsXisl\_SeaWa, cpsl - Specific isobaric heat capacity of saturated seawater in kJ/(kg\*K)

**Range of Validity:**

Temperature  $t$ : from 0 °C to 220 °C  
 Mixture pressure  $p$ : from  $p_s(t = 0 \text{ °C}, \xi = 0.2)$  to  $p_s(t = 220 \text{ °C}, \xi = 0)$   
 Mass fraction  $\xi$ :  $0 \leq \xi \leq 0.2$  kg sea salt/kg mixture

**Possible input variants:**

$$c_p^{sl} = f(-1000, t_s, \xi^{sl})$$

$$c_p^{sl} = f(p_s, -1000, \xi^{sl})$$

$$c_p^{sl} = f(p_s, t_s, -1000)$$

$$c_p^{sl} = f(p_s, t_s, \xi^{sl})$$

**Result for Wrong Input Values:**

cpsl\_ptXi\_SeaWa, cpsl = -1000

**References:**

$c_p(p, t, \xi)$  [1],[2]

**Specific Isobaric Heat Capacity of Saturated Vapor  $c_{p_{sv}} = f(p_s, t_s, \xi^{sl})$** 
**Function Name:**

cpsv\_pstsXisl\_SeaWa

**Fortran Programs:**

```
REAL*8 FUNCTION  CPSV_PSTSXISL_SEAWA(PS,TS,XISL)  REAL*8 PS,TS,XISL
```

**Input Values:**

$ps$         - Pressure  $p$  in bar  
 $ts$         - Temperature  $t$  in °C  
 $Xisl$        - Mass fraction of sea salt  $\xi$  in kg sea salt/kg mixture

**Result:**

cpsv\_pstsXisl\_SeaWa,        cpsv - Specific isobaric heat capacity of saturated seawater  
 in kJ/(kg\*K)

**Range of Validity:**

Temperature  $t$ :                from 0 °C to 220 °C  
 Mixture pressure  $p$ :        from  $p_s(t = 0\text{ °C}, \xi = 0.2)$  to  $p_s(t = 220\text{ °C}, \xi = 0)$   
 Mass fraction  $\xi$ :               $0 \leq \xi \leq 0.2$  kg sea salt/kg mixture

**Result for Wrong Input Values:**

cpsv\_pstsXisl\_SeaWa, cpsv = -1000

**References:**

$c_p(p, t, \xi)$                     [1], [2]

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

eta\_ptXi\_SeaWa

**Fortran Programs:**

```
REAL*8 FUNCTION ETA_PTXI_SEAWA(P, T, XI), REAL*8 P, T, XI
```

**Input Values:**

$p$  - Pressure  $p$  in bar  
 $t$  - Temperature  $t$  in °C  
 $Xi$  - Mass fraction of sea salt  $\xi$  in kg sea salt/kg mixture

**Result:**

eta\_ptXi\_SeaWa, eta - Dynamic Viscosity in Pa\*s

**Range of Validity:**

Temperature  $t$ : from 0 °C to 220 °C  
 Pressure  $p$ : from  $p_s(t = 0\text{ °C}, \xi = 0.2)$  to 100 bar and  $p \geq p_{\text{mel}}(t, \xi)$   
 Mass fraction  $\xi$ :  $0 \leq \xi \leq 0.2$  kg sea salt/kg mixture

**Comments:**

- This function is not defined in the wet steam region.

**Result for Wrong Input Values:**

eta\_ptXi\_SeaWa, eta = -1000

**References:**

$\eta(p, t, \xi)$  [1],[2]

**Dynamic Viscosity of Liquid Seawater  $\eta_l = f(t, \xi)$** 
**Function Name:**

etal\_tXI\_SeaWa

**Fortran Programs:**

```
REAL*8 FUNCTION ETAL_PTXI_SEAWA(T, XI), REAL*8 T, XI
```

**Input Values:**

$t$  - Temperature  $t$  in °C

$X_i$  - Mass fraction of sea salt  $\xi$  in kg sea salt/kg mixture

**Result:**

etal\_tXI\_SeaWa, etal - Dynamic Viscosity in Pa\*s

**Range of Validity:**

Temperature  $t$ : from 0 °C to 220 °C

Mass fraction  $\xi$ :  $0 \leq \xi \leq 0.2$  kg sea salt/kg mixture

**Result for Wrong Input Values:**

etal\_tXi\_SeaWa, etal = -1000

**References:**

$\eta(\rho, t, \xi)$  [1], [2]



**Dynamic Viscosity of Saturated Liquid  $\eta_{sl} = f(p_s, t_s, \xi^{sl})$** 
**Function Name:**

etasl\_pstsXisl\_SeaWa

**Fortran Programs:**

```
REAL*8 FUNCTION ETASL_PSTSXISL_SEAWA(PS, TS, XISL), REAL*8 PS, TS, XISL
```

**Input Values:**

$ps$  - Pressure  $p$  in bar  
 $ts$  - Temperature  $t$  in °C  
 $Xisl$  - Mass fraction of sea salt  $\xi$  in kg sea salt/kg mixture

**Result:**

etasl\_pstsXisl\_SeaWa, etasl - Dynamic Viscosity in Pa\*s

**Range of Validity:**

Temperature  $t$ : from 0 °C to 220 °C  
 Mixture pressure  $p$ : from  $p_s(t = 0 \text{ °C}, \xi = 0.2)$  to  $p_s(t = 220 \text{ °C}, \xi = 0)$   
 Mass fraction  $\xi$ :  $0 \leq \xi \leq 0.2$  kg sea salt/kg mixture

**Possible input variants:**

$$\begin{aligned}\eta^{sl} &= f(-1000, t_s, \xi^{sl}) \\ \eta^{sl} &= f(p_s, -1000, \xi^{sl}) \\ \eta^{sl} &= f(p_s, t_s, -1000) \\ \eta^{sl} &= f(p_s, t_s, \xi^{sl})\end{aligned}$$

**Result for Wrong Input Values:**

etasl\_ptXi\_SeaWa, etasl = -1000

**References:**

$\eta(p, t, \xi)$  [1], [2]

**Dynamic Viscosity of Saturated Vapor  $\eta_{sv} = f(p_s, t_s, \xi^{sl})$** 
**Function Name:**

etasv\_pstsXisl\_SeaWa

**Fortran Programs:**

```
REAL*8 FUNCTION ETASV_PSTSXISL_SEAWA(PS,TS,XISL), REAL*8 PS,TS,XISL
```

**Input Values:**

$ps$  - Pressure  $p$  in bar  
 $ts$  - Temperature  $t$  in °C  
 $Xisl$  - Mass fraction of sea salt  $\xi$  in kg sea salt/kg mixture

**Result:**

etasv\_pstsXisl\_SeaWa, etasv - Dynamic Viscosity of saturated seawater in Pa\*s

**Range of Validity:**

Temperature  $t$ : from 0 °C to 220 °C  
 Mixture pressure  $p$ : from  $p_s(t = 0 \text{ °C}, \xi = 0.2)$  to  $p_s(t = 220 \text{ °C}, \xi = 0)$   
 Mass fraction  $\xi$ :  $0 \leq \xi \leq 0.2$  kg sea salt/kg mixture

**Result for Wrong Input Values:**

etasv\_pstsXisl\_SeaWa, etasv = -1000

**References:**

$\eta(p, t, \xi)$  [1], [2]

**Specific Helmholtz Energy of Liquid Seawater  $f_l = f(p, t, \xi)$** 
**Function Name:**

fl\_ptXi\_SeaWa

**Fortran Programs:**

```
REAL*8 FUNCTION FL_PTXI_SEAWA(P, T, XI), REAL*8 P, T, XI
```

**Input Values:**

$p$  - Pressure  $p$  in bar  
 $t$  - Temperature  $t$  in °C  
 $X_i$  - Mass fraction of sea salt  $\xi$  in kg sea salt/kg mixture

**Result:**

fl\_ptXi\_SeaWa, fl - Specific Helmholtz energy in kJ/kg

**Range of Validity:**

Temperature  $t$ : from 0 °C to 80 °C  
 Mixture pressure  $p$ : from  $p_s(t = 0\text{ °C}, \xi = 0.12)$  to 1000 bar and  $p \geq p_{\text{mel}}(t, \xi)$   
 Mass fraction  $\xi$ :  $0 \leq \xi \leq 0.12$  kg sea salt/kg mixture

**Result for Wrong Input Values:**

fl\_ptXi\_SeaWa, fl = -1000

**References:**

$f(p, t, \xi)$  [1]

**Specific Helmholtz Energy of Saturated Liquid  $f_{sl} = f(p_s, t_s, \xi^{sl})$** 
**Function Name:**

fsl\_pstsXisl\_SeaWa

**Fortran Programs:**

```
REAL*8 FUNCTION FSL_PSTSXISL_SEAWA(PS, TS, XISL), REAL*8 PS, TS, XISL
```

**Input Values:**

$ps$  - Pressure  $p$  in bar  
 $ts$  - Temperature  $t$  in °C  
 $Xisl$  - Mass fraction of sea salt  $\xi$  in kg sea salt/kg mixture

**Result:**

fsl\_pstsXisl\_SeaWa, fsl - Specific Helmholtz energy in kJ/kg

**Range of Validity:**

Temperature  $t$ : from 0 °C to 80 °C  
 Mixture pressure  $p$ : from  $p_s(t = 0\text{ °C}, \xi = 0.12)$  to  $p_s(t = 80\text{ °C}, \xi = 0)$   
 Mass fraction  $\xi$ :  $0 \leq \xi \leq 0.12$  kg sea salt/kg mixture

**Possible input variants:**

$$f^{sl} = f(-1000, t_s, \xi^{sl})$$

$$f^{sl} = f(p_s, -1000, \xi^{sl})$$

$$f^{sl} = f(p_s, t_s, -1000)$$

$$f^{sl} = f(p_s, t_s, \xi^{sl})$$

**Result for Wrong Input Values:**

fsl\_pstsXisl\_SeaWa, fsl = -1000

**References:**

$f(p, t, \xi)$  [1]

**Osmotic Coefficient of Liquid Seawater  $\phi_l = f(p, t, \xi)$** 
**Function Name:**

phil\_ptXi\_SeaWa

**Fortran Programs:**

```
REAL*8 FUNCTION PHIL_PTXI_SEAWA(P, T, XI), REAL*8 P, T, XI
```

**Input Values:**

$p$  - Pressure  $p$  in bar  
 $t$  - Temperature  $t$  in °C  
 $X_i$  - Mass fraction of sea salt  $\xi$  in kg sea salt/kg mixture

**Result:**

phil\_ptXi\_SeaWa, phil - Osmotic Coefficient in [-]

**Range of Validity:**

Temperature  $t$ : from 0 °C to 80 °C  
 Pressure  $p$ : from  $p_s(t = 0\text{ °C}, \xi = 0.12)$  to 1000 bar and  $p \geq p_{\text{mel}}(t, \xi)$   
 Mass fraction  $\xi$ :  $0 < \xi \leq 0.12$  kg sea salt/kg mixture

**Result for Wrong Input Values:**

phil\_ptXi\_SeaWa, phil = -1000

**References:**

$\phi(p, t, \xi)$  [1]

**Osmotic Coefficient of Saturated Liquid  $\phi_{sl} = f(p_s, t_s, \xi^{sl})$** 
**Function Name:**

phisl\_pstsXisl\_SeaWa

**Fortran Programs:**

```
REAL*8 FUNCTION PHISL_PSTSXISL_SEAWA(PS, TS, XISL), REAL*8 PS, TS, XISL
```

**Input Values:**

$ps$  - Pressure  $p$  in bar  
 $ts$  - Temperature  $t$  in °C  
 $Xisl$  - Mass fraction of sea salt  $\xi$  in kg sea salt/kg mixture

**Result:**

phisl\_pstsXisl\_SeaWa, phisl - Osmotic Coefficient in [-]

**Range of Validity:**

Temperature  $t$ : from 0 °C to 80 °C  
 Mixture pressure  $p$ : from  $p_s(t = 0 \text{ °C}, \xi = 0.12)$  to  $p_s(t = 80 \text{ °C}, \xi = 0)$   
 Mass fraction  $\xi$ :  $0 < \xi \leq 0.12$  kg sea salt/kg mixture

**Possible input variants:**

$$\begin{aligned}\phi^{sl} &= f(-1000, t_s, \xi^{sl}) \\ \phi^{sl} &= f(p_s, -1000, \xi^{sl}) \\ \phi^{sl} &= f(p_s, t_s, -1000) \\ \phi^{sl} &= f(p_s, t_s, \xi^{sl})\end{aligned}$$

**Result for Wrong Input Values:**

phisl\_pstsXisl\_SeaWa, fsl = - 1000

**References:**

$\phi(p, t, \xi)$  [1]

**Specific Enthalpy  $h = f(p, t, \xi)$** 
**Function Name:**

h\_ptXi\_SeaWa

**Fortran Programs:**

```
REAL*8 FUNCTION H_PTXI_SEAWA(P, T, XI), REAL*8 P, T, XI
```

**Input Values:**

$p$  - Pressure  $p$  in bar  
 $t$  - Temperature  $t$  in °C  
 $Xi$  - Mass fraction of sea salt  $\xi$  in kg sea salt/kg mixture

**Result:**

h\_ptXi\_SeaWa, h - specific enthalpy in kJ/kg

**Range of Validity:**

Temperature  $t$ : from 0 °C to 220 °C  
 Pressure  $p$ : from  $p_s(t = 0\text{ °C}, \xi = 0.2)$  to 1000 bar and  $p \geq p_{\text{mel}}(t, \xi)$   
 Mass fraction  $\xi$ :  $0 \leq \xi \leq 0.2$  kg sea salt/kg mixture

**Result for Wrong Input Values:**

h\_ptXi\_SeaWa, h = -1000

**References:**

$h(p, t, \xi)$  [1], [2]

**Specific Enthalpy of Liquid Seawater  $h_l = f(p, t, \xi)$** 
**Function Name:**

hl\_ptXi\_SeaWa

**Fortran Programs:**

```
REAL*8 FUNCTION HL_PTXI_SEAWA(P, T, XI), REAL*8 P, T, XI
```

**Input Values:**

$p$  - Pressure  $p$  in bar  
 $t$  - Temperature  $t$  in °C  
 $Xi$  - Mass fraction of sea salt  $\xi$  in kg sea salt/kg mixture

**Result:**

hl\_ptXi\_SeaWa, hl - Specific Enthalpy of liquid seawater in kJ/kg

**Range of Validity:**

Temperature  $t$ : from 0 °C to 220 °C  
 Pressure  $p$ : from  $p_s(t = 0\text{ °C}, \xi = 0.2)$  to 1000 bar and  $p \geq p_{\text{mel}}(t, \xi)$   
 Mass fraction  $\xi$ :  $0 \leq \xi \leq 0.2$  kg sea salt/kg mixture

**Result for Wrong Input Values:**

hl\_ptXi\_SeaWa, hl = -1000

**References:**

$h(p, t, \xi)$  [1], [2]



**Specific Enthalpy of Saturated Fluid  $h_{sl} = f(p_s, t_s, \xi^{sl})$** 
**Function Name:**

hsl\_pstsXisl\_SeaWa

**Fortran Programs:**

```
REAL*8 FUNCTION HSL_PSTSXISL_SEAWA(PS, TS, XISL), REAL*8 PS, TS, XISL
```

**Input Values:**

$ps$       - Pressure  $p$  in bar  
 $ts$       - Temperature  $t$  in °C  
 $Xisl$      - Mass fraction of sea salt  $\xi$  in kg sea salt/kg mixture

**Result:**

hsl\_pstsXisl\_SeaWa, hsl - Specific Enthalpy of saturated seawater in kJ/kg

**Range of Validity:**

Temperature  $t$ :                      from 0 °C to 220 °C  
 Mixture pressure  $p$ :                from  $p_s(t = 0 \text{ °C}, \xi = 0.2)$  to  $p_s(t = 220 \text{ °C}, \xi = 0)$   
 Mass fraction  $\xi$ :                       $0 \leq \xi \leq 0.2$  kg sea salt/kg mixture

**Possible input variants:**

$$\begin{aligned}
 h^{sl} &= f(-1000, t_s, \xi^{sl}) \\
 h^{sl} &= f(p_s, -1000, \xi^{sl}) \\
 h^{sl} &= f(p_s, t_s, -1000) \\
 h^{sl} &= f(p_s, t_s, \xi^{sl})
 \end{aligned}$$

**Result for Wrong Input Values:**

hsl\_pstsXisl\_SeaWa, hsl = -1000

**References:**

$h(p, t, \xi)$                                       [1] , [2]

**Specific Enthalpy of Saturated Steam  $h_{sv} = f(p_s, t_s, \xi^{sl})$** 
**Function Name:**

hsv\_pstsXisl\_SeaWa

**Fortran Programs:**

```
REAL*8 FUNCTION HSV_PSTSXISL_SEAWA(PS,TS,XISL), REAL*8 PS, TS,XISL
```

**Input Values:**

$ps$  - Pressure  $p$  in bar  
 $ts$  - Temperature  $t$  in °C  
 $Xisl$  - Mass fraction of sea salt  $\xi$  in kg sea salt/kg mixture

**Result:**

hsv\_pstsXisl\_SeaWa, hsv - Specific Enthalpy of saturated seawater in kJ/kg

**Range of Validity:**

Temperature  $t$ : from 0 °C to 220 °C  
Mixture pressure  $p$ : from  $p_s(t = 0 \text{ °C}, \xi = 0.2)$  to  $p_s(t = 220 \text{ °C}, \xi = 0)$   
Mass fraction  $\xi$ :  $0 \leq \xi \leq 0.2$  kg sea salt/kg mixture

**Result for Wrong Input Values:**

hsv\_pstsXisl\_SeaWa, hsv = -1000

**References:**

$h(p, t, \xi)$  [1] , [2]

**Isentropic Exponent  $\kappa = f(p, t, \xi)$** 
**Function Name:**

kappa\_ptXi\_SeaWa

**Fortran Programs:**

```
REAL*8 FUNCTION KAPPA_PTXI_SEAWA(P, T, XI), REAL*8 P, T, XI
```

**Input Values:**

$p$  - Pressure  $p$  in bar  
 $t$  - Temperature  $t$  in °C  
 $Xi$  - Mass fraction of sea salt  $\xi$  in kg sea salt/kg mixture

**Result:**

kappa\_ptXi\_SeaWa, kappa - Isentropic Exponent in [-]

**Range of Validity:**

Temperature  $t$ : from 0 °C to 80 °C  
 Pressure  $p$ : from  $p_s(t = 0\text{ °C}, \xi = 0.12)$  to 1000 bar and  $p \geq p_{\text{mel}}(t, \xi)$   
 Mass fraction  $\xi$ :  $0 \leq \xi \leq 0.12$  kg sea salt/kg mixture

**Comments:**

- Isentropic Exponent  $\kappa = \frac{w^2}{p^* v}$
- This function is not defined in the wet steam region.

**Result for Wrong Input Values:**

kappa\_ptXi\_SeaWa, kappa = -1000

**References:**

$\kappa(p, t, \xi)$  [1]

**Isentropic Exponent of Liquid Seawater  $\kappa_l = f(p, t, \xi)$** 
**Function Name:**

kappal\_ptXi\_SeaWa

**Fortran Programs:**

```
REAL*8 FUNCTION KAPPAL_PTXI_SEAWA(P, T, XI), REAL*8 P, T, XI
```

**Input Values:**

$p$  - Pressure  $p$  in bar  
 $t$  - Temperature  $t$  in °C  
 $Xi$  - Mass fraction of sea salt  $\xi$  in kg sea salt/kg mixture

**Result:**

kappal\_ptXi\_SeaWa, kappal - Isentropic Exponent of liquid seawater in [-]

**Range of Validity:**

Temperature  $t$ : from 0 °C to 80 °C  
 Pressure  $p$ : from  $p_s(t = 0 \text{ °C}, \xi = 0.12)$  to 1000 bar and  $p \geq p_{\text{mel}}(t, \xi)$   
 Mass fraction  $\xi$ :  $0 \leq \xi \leq 0.12$  kg sea salt/kg mixture

**Comments:**

- Isentropic Exponent  $\kappa = \frac{w^2}{p^* v}$

**Result for Wrong Input Values:**

kappal\_ptXi\_SeaWa, kappal = -1000

**References:**

$\kappa(p, t, \xi)$  [1]

**Isentropic Exponent of Saturated Fluid  $\kappa_{sl} = f(p_s, t_s, \xi^{sl})$** 
**Function Name:**

kappasl\_pstsXisl\_SeaWa

**Fortran Programs:**

```
REAL*8 FUNCTION KAPPASL_PSTSL_XISL_SEAWA(PS, TS, XISL)
REAL*8 PS, TS, XISL
```

**Input Values:**

$ps$  - Pressure  $p$  in bar  
 $ts$  - Temperature  $t$  in °C  
 $Xisl$  - Mass fraction of sea salt  $\xi$  in kg sea salt/kg mixture

**Result:**

kappasl\_pstsXisl\_SeaWa, kappasl - Isentropic Exponent of saturated seawater in [-]

**Range of Validity:**

Temperature  $t$ : from 0 °C to 80 °C  
 Pressure  $p$ : from  $p_s(t = 0 \text{ °C}, \xi = 0.12)$  to  $p_s(t = 80 \text{ °C}, \xi = 0)$   
 Mass fraction  $\xi$ :  $0 \leq \xi \leq 0.12$  kg sea salt/kg mixture

**Comments:**

- Isentropic Exponent  $\kappa = \frac{w^2}{p^* v}$

**Possible input variants:**

$$\begin{aligned}\kappa^{sl} &= f(-1000, t_s, \xi^{sl}) \\ \kappa^{sl} &= f(p_s, -1000, \xi^{sl}) \\ \kappa^{sl} &= f(p_s, t_s, -1000) \\ \kappa^{sl} &= f(p_s, t_s, \xi^{sl})\end{aligned}$$

**Result for Wrong Input Values:**

kappasl\_pstsXisl\_SeaWa, kappasl = -1000

**References:**

$\kappa(p, t, \xi)$  [1]

**Isentropic Exponent of Saturated Vapor  $\kappa_{sv} = f(p_s, t_s, \xi^{sl})$** 
**Function Name:**

kappasv\_pstsXisl\_SeaWa

**Fortran Programs:**

```
REAL*8 FUNCTION KAPPASV_PSTSL_SEAWA(PS,TS,XISL), REAL*8 PS,TS,XISL
```

**Input Values:**

$p_s$       - Pressure  $p$  in bar  
 $t_s$       - Temperature  $t$  in °C  
 $Xisl$      - Mass fraction of sea salt  $\xi$  in kg sea salt/kg mixture

**Result:**

kappasv\_pstsXisl\_SeaWa, kappasv - Isentropic Exponent of saturated seawater in [-]

**Range of Validity:**

Temperature  $t$ :                      from 0 °C to 80 °C  
 Pressure  $p$ :                            from  $p_s(t = 0 \text{ °C}, \xi = 0.12)$  to  $p_s(t = 80 \text{ °C}, \xi = 0)$   
 Mass fraction  $\xi$ :                       $0 \leq \xi \leq 0.12$  kg sea salt/kg mixture

**Comments:**

- Isentropic Exponent  $\kappa = \frac{w^2}{p^* v}$

**Result for Wrong Input Values:**

kappasv\_pstsXisl\_SeaWa, kappasv = -1000

**References:**

$\kappa(p, t, \xi)$                                       [1]

**Isothermal Compressibility of Liquid Seawater  $\kappa_{TI} = f(p, t, \xi)$** 
**Function Name:**

kappaTI\_ptXi\_SeaWa

**Fortran Programs:**

```
REAL*8 FUNCTION KAPPATL_PTXI_SEAWA(P, T, XI), REAL*8 P, T, XI
```

**Input Values:**

$p$  - Pressure  $p$  in bar  
 $t$  - Temperature  $t$  in °C  
 $Xi$  - Mass fraction of sea salt  $\xi$  in kg sea salt/kg mixture

**Result:**

kappaTI\_ptXi\_SeaWa, kappaTI - Isothermal Compressibility of liquid seawater in [1/kPa]

**Range of Validity:**

Temperature  $t$ : from 0 °C to 80 °C  
 Pressure  $p$ : from  $p_s(t = 0 \text{ °C}, \xi = 0.12)$  to 1000 bar and  $p \geq p_{\text{mel}}(t, \xi)$   
 Mass fraction  $\xi$ :  $0 \leq \xi \leq 0.12$  kg sea salt/kg mixture

**Result for Wrong Input Values:**

kappaTI\_ptXi\_SeaWa, kappaTI = -1000

**References:**

$\kappa_T(p, t, \xi)$  [1]

**Isothermal Compressibility of Saturated Fluid  $\kappa_{T\text{sl}} = f(p_s, t_s, \xi^{\text{sl}})$** 
**Function Name:**

kappaTsl\_pstsXisl\_SeaWa

**Fortran Programs:**

```
REAL*8 FUNCTION KAPPATSL_PSTSXISL_SEAWA(PS, TS, XISL)
      REAL*8 PS, TS, XISL
```

**Input Values:**

*ps*        - Pressure  $p$  in bar  
*ts*        - Temperature  $t$  in °C  
*Xisl*      - Mass fraction of sea salt  $\xi$  in kg sea salt/kg mixture

**Result:**

kappaTsl\_pstsXisl\_SeaWa, kappaTsl - Isothermal Compressibility of saturated fluid in [1/kPa]

**Range of Validity**

Temperature  $t$ :                      from 0 °C to 80 °C  
 Pressure  $p$ :                        from  $p_s(t = 0\text{ °C}, \xi = 0.12)$  to  $p_s(t = 80\text{ °C}, \xi = 0)$   
 Mass fraction  $\xi$ :                     $0 \leq \xi \leq 0.12$  kg sea salt/kg mixture

**Possible input variants:**

$$\begin{aligned}\kappa_T^{\text{sl}} &= f(-1000, t_s, \xi^{\text{sl}}) \\ \kappa_T^{\text{sl}} &= f(p_s, -1000, \xi^{\text{sl}}) \\ \kappa_T^{\text{sl}} &= f(p_s, t_s, -1000) \\ \kappa_T^{\text{sl}} &= f(p_s, t_s, \xi^{\text{sl}})\end{aligned}$$

**Result for Wrong Input Values:**

kappaTsl\_pstsXisl\_SeaWa, kappaTsl = -1000

**References:**

$\kappa_T(p, t, \xi)$                                       [1]



**Isothermal Compressibility of Liquid Seawater  $\kappa_{ls} = f(p, t, \xi)$** 
**Function Name:**

kappalsl\_ptXi\_SeaWa

**Fortran Programs:**

```
REAL*8 FUNCTION KAPPAISL_PTXI_SEAWA(P, T, XI), REAL*8 P, T, XI
```

**Input Values:**

$p$  - Pressure  $p$  in bar  
 $t$  - Temperature  $t$  in °C  
 $Xi$  - Mass fraction of sea salt  $\xi$  in kg sea salt/kg mixture

**Result:**

kappalsl\_ptXi\_SeaWa, kappalsl - Isentropic Compressibility of liquid seawater in [1/kPa]

**Range of Validity:**

Temperature  $t$ : from 0 °C to 80 °C  
 Pressure  $p$ : from  $p_s(t = 0 \text{ °C}, \xi = 0.12)$  to 1000 bar and  $p \geq p_{\text{mel}}(t, \xi)$   
 Mass fraction  $\xi$ :  $0 \leq \xi \leq 0.12$  kg sea salt/kg mixture

**Result for Wrong Input Values:**

kappalsl\_ptXi\_SeaWa, kappalsl = -1000

**References:**
 $\kappa_{ls}(p, t, \xi)$  [1]

**Isothermal Compressibility of Saturated Seawater  $\kappa_{Is\ sl} = f(p_s, t_s, \xi^{sl})$** 
**Function Name:**

kappalssl\_pstsXisl\_SeaWa

**Fortran Programs:**

```
REAL*8 FUNCTION KAPPAISL_PSTSEXISL_SEAWA(PS, TS, XISL)
REAL*8 PS, TS, XISL
```

**Input Values:**

$p_s$  - Pressure  $p$  in bar  
 $t_s$  - Temperature  $t$  in °C  
 $Xisl$  - Mass fraction of sea salt  $\xi$  in kg sea salt/kg mixture

**Result:**

kappalssl\_pstsXisl\_SeaWa, kappalssl - Isentropic Compressibility of saturated seawater in [1/kPa]

**Range of Validity:**

Temperature  $t$ : from 0 °C to 80 °C  
 Pressure  $p$ : from  $p_s(t = 0\text{ °C}, \xi = 0.12)$  to  $p_s(t = 80\text{ °C}, \xi = 0)$   
 Mass fraction  $\xi$ :  $0 \leq \xi \leq 0.12$  kg sea salt/kg mixture

**Possible input variants:**

$$\begin{aligned}\kappa_{Is}^{sl} &= f(-1000, t_s, \xi^{sl}) \\ \kappa_{Is}^{sl} &= f(p_s, -1000, \xi^{sl}) \\ \kappa_{Is}^{sl} &= f(p_s, t_s, -1000) \\ \kappa_{Is}^{sl} &= f(p_s, t_s, \xi^{sl})\end{aligned}$$

**Result for Wrong Input Values:**

kappalssl\_pstsXisl\_SeaWa, kappalssl = -1000

**References:**

$\kappa_{Is}(p, t, \xi)$  [1]

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

lambda\_ptXi\_SeaWa

**Fortran Programs:**

```
REAL*8 FUNCTION LAMBDA_PTXI_SEAWA(P, T, XI), REAL*8 P, T, XI
```

**Input Values:**

$p$  - Pressure  $p$  in bar  
 $t$  - Temperature  $t$  in °C  
 $Xi$  - Mass fraction of sea salt  $\xi$  in kg sea salt/kg mixture

**Result:**

lambda\_ptXi\_SeaWa, lambda - Thermal Conductivity in W/(m\*K)

**Range of Validity:**

Temperature  $t$ : from 0 °C to 220 °C  
 Pressure  $p$ : from  $p_s(t = 0 \text{ °C}, \xi = 0.2)$  to 100 bar and  $p \geq p_{\text{mel}}(t, \xi)$   
 Mass fraction  $\xi$ :  $0 \leq \xi \leq 0.2$  kg sea salt/kg mixture

**Comments:**

- This function is not defined in the wet steam region.

**Result for Wrong Input Values:**

lambda\_ptXi\_SeaWa, lambda = -1000

**References:**

$\lambda(p, t, \xi)$  [3], [4], [15]

**Thermal Conductivity of Liquid Seawater  $\lambda_l = f(t, \xi)$** 
**Function Name:**

lambdal\_tXi\_SeaWa

**Fortran Programs:**

```
REAL*8 FUNCTION LAMBDAL_TXI_SEAWA(T, XI), REAL*8 T, XI
```

**Input Values:**

$t$  - Temperature  $t$  in °C

$X_i$  - Mass fraction of sea salt  $\xi$  in kg sea salt/kg mixture

**Result:**

lambdal\_tXi\_SeaWa, lambdal - Thermal Conductivity of liquid seawater in W/(m\*K)

**Range of Validity:**

Temperature  $t$ : from 0 °C to 220 °C

Mass fraction  $\xi$ :  $0 \leq \xi \leq 0.2$  kg sea salt/kg mixture

**Result for Wrong Input Values:**

lambdal\_tXi\_SeaWa, lambdal = -1000

**References:**

$\lambda(p, t, \xi)$  [3], [4], [15]

**Thermal Conductivity of Saturated Fluid  $\lambda_{sl} = f(p_s, t_s, \xi^{sl})$** 
**Function Name:**

lambdasl\_pstsXisl\_SeaWa

**Fortran Programs:**

```

      REAL*8 FUNCTION  LAMBDA_SL_PSTSL_SEAWA(PS, TS, XISL)
      REAL*8 PS, TS, XISL
      ps      - Pressure  $p$  in bar
      ts      - Temperature  $t$  in °C
      Xisl    - Mass fraction of sea salt  $\xi$  in kg sea salt/kg mixture
  
```

**Result:**

lambdasl\_pstsXisl\_SeaWa, lambdasl - Thermal Conductivity of saturated fluid in W/(m\*K)

**Range of Validity:**

Temperature  $t$ : from 0 °C to 220 °C  
 Mixture pressure  $p$ : from  $p_s(t = 0\text{ °C}, \xi = 0.2)$  to  $p_s(t = 220\text{ °C}, \xi = 0)$   
 Mass fraction  $\xi$ :  $0 \leq \xi \leq 0.2$  kg sea salt/kg mixture

**Possible input variants:**

$$\lambda^{sl} = f(-1000, t_s, \xi^{sl})$$

$$\lambda^{sl} = f(p_s, -1000, \xi^{sl})$$

$$\lambda^{sl} = f(p_s, t_s, -1000)$$

$$\lambda^{sl} = f(p_s, t_s, \xi^{sl})$$

**Result for Wrong Input Values:**

lambdasl\_pstsXisl\_SeaWa, lambdasl = -1000

**References:**

$\lambda(p, t, \xi)$  [3], [4], [15]

**Thermal Conductivity of Saturated Vapor  $\lambda_{sv} = f(p_s, t_s, \xi^{sl})$** 
**Function Name:**

lambdasv\_pstsXisl\_SeaWa

**Fortran Programs:**

```
REAL*8 FUNCTION LAMBDA SV_PST SXISL_SEAWA(PS,TS,XISL)
REAL*8 PS,TS,XISL
```

**Input Values:**

$ps$  - Pressure  $p$  in bar  
 $ts$  - Temperature  $t$  in °C  
 $Xisl$  - Mass fraction of sea salt  $\xi$  in kg sea salt/kg mixture

**Result:**

lambdasvXisl\_psts\_SeaWa, lambdasv - Thermal Conductivity of saturated seawater in W/(m\*K)

**Range of Validity:**

Temperature  $t$ : from 0 °C to 220 °C  
Mixture pressure  $p$ : from  $p_s(t = 0 \text{ °C}, \xi = 0.2)$  to  $p_s(t = 220 \text{ °C}, \xi = 0)$   
Mass fraction  $\xi$ :  $0 \leq \xi \leq 0.2$  kg sea salt/kg mixture

**Result for Wrong Input Values:**

lambdasv\_pstsXisl\_SeaWa, lambdasv = -1000

**References:**

$\lambda(p, t, \xi)$  [3], [4], [15]

**Relative Chemical Potential of Liquid Seawater  $\mu_l = f(p, t, \xi)$** 
**Function Name:**

myl\_ptXi\_SeaWa

**Fortran Programs:**

```
REAL*8 FUNCTION MYL_PTXI_SEAWA(P, T, XI), REAL*8 P, T, XI
```

**Input Values:**

$p$         - Pressure  $p$  in bar  
 $t$         - Temperature  $t$  in °C  
 $Xi$        - Mass fraction of sea salt  $\xi$  in kg sea salt/kg mixture

**Result:**

myl\_ptXi\_SeaWa, myl - Relative chemical potential of liquid seawater in [kJ/kg]

**Range of Validity:**

Temperature  $t$ :                      from 0 °C to 80 °C  
 Pressure  $p$ :                            from  $p_s(t = 0 \text{ °C}, \xi = 0.12)$  to 1000 bar and  $p \geq p_{\text{mel}}(t, \xi)$   
 Mass fraction  $\xi$ :                       $0 \leq \xi \leq 0.12$  kg sea salt/kg mixture

**Result for Wrong Input Values:**

myl\_ptXi\_SeaWa, myl = -1000

**References:**

$\mu(p, t, \xi)$  [1]

**Relative Chemical Potential of Saturated Fluid  $\mu_{sl} = f(p_s, t_s, \xi^{sl})$** 
**Function Name:**

mysl\_pstsXisl\_SeaWa

**Fortran Programs:**

```
REAL*8 FUNCTION MYSL_PSTXISL_SEAWA(PS, TS, XISL)
REAL*8 PS, TS, XISL
```

**Input Values:**

*ps*        - Pressure  $p$  in bar  
*ts*        - Temperature  $t$  in °C  
*Xisl*      - Mass fraction of sea salt  $\xi$  in kg sea salt/kg mixture

**Result:**

mysl\_pstsXisl\_SeaWa, mysl - Relative chemical potential of saturated seawater in [kJ/kg]

**Range of Validity:**

Temperature  $t$ :                      from 0 °C to 80 °C  
Mixture pressure  $p$ :                from  $p_s(t = 0 \text{ °C}, \xi = 0.12)$  to  $p_s(t = 80 \text{ °C}, \xi = 0)$   
Mass fraction  $\xi$ :                       $0 \leq \xi \leq 0.12$  kg sea salt/kg mixture

**Possible input variants:**

$$\begin{aligned}\mu^{sl} &= f(-1000, t_s, \xi^{sl}) \\ \mu^{sl} &= f(p_s, -1000, \xi^{sl}) \\ \mu^{sl} &= f(p_s, t_s, -1000) \\ \mu^{sl} &= f(p_s, t_s, \xi^{sl})\end{aligned}$$

**Result for Wrong Input Values:**

mysl\_pstsXisl\_SeaWa, mysl = -1000

**References:**

$$\mu(p, t, \xi) \quad [1]$$



**Chemical Potential of H<sub>2</sub>O of Liquid Seawater  $\mu_{wl} = f(p, t, \xi)$** 
**Function Name:**

mywl\_ptXi\_SeaWa

**Fortran Programs:**

```
REAL*8 FUNCTION MYWL_PTXI_SEAWA(P, T, XI), REAL*8 P, T, XI
```

**Input Values:**

$p$         - Pressure  $p$  in bar  
 $t$         - Temperature  $t$  in °C  
 $Xi$        - Mass fraction of sea salt  $\xi$  in kg sea salt/kg mixture

**Result:**

mywl\_ptXi\_SeaWa, mywl - Chemical potential of H<sub>2</sub>O of liquid seawater in [kJ/kg]

**Range of Validity:**

Temperature  $t$ :                      from 0 °C to 80 °C  
 Pressure  $p$ :                            from  $p_s(t = 0 \text{ °C}, \xi = 0.12)$  to 1000 bar and  $p \geq p_{\text{mel}}(t, \xi)$   
 Mass fraction  $\xi$ :                       $0 \leq \xi \leq 0.12$  kg sea salt/kg mixture

**Result for Wrong Input Values:**

mywl\_ptXi\_SeaWa, mywl = -1000

**References:**

$\mu_w(p, t, \xi)$                                       [1]

**Chemical Potential of H<sub>2</sub>O of Saturated Fluid  $\mu_{w\,sl} = f(p_s, t_s, \xi^{sl})$** 
**Function Name:**

mywsl\_pstsXisl\_SeaWa

**Fortran Programs:**

```
REAL*8 FUNCTION MYWSL_PSTSXISL_SEAWA(PS, TS, XISL)
REAL*8 PS, TS, XISL
```

**Input Values:**

*ps*        - Pressure  $p$  in bar  
*ts*        - Temperature  $t$  in °C  
*Xisl*      - Mass fraction of sea salt  $\xi$  in kg sea salt/kg mixture

**Result:**

mywsl\_pstsXisl\_SeaWa, mywsl - Relative chemical potential of H<sub>2</sub>O of saturated seawater in [kJ/kg]

**Range of Validity:**

Temperature  $t$ :                      from 0 °C to 80 °C  
Mixture pressure  $p$ :                from  $p_s(t = 0\text{ °C}, \xi = 0.12)$  to  $p_s(t = 80\text{ °C}, \xi = 0)$   
Mass fraction  $\xi$ :                       $0 \leq \xi \leq 0.12$  kg sea salt/kg mixture

**Possible input variants:**

$$\begin{aligned}\mu_w^{sl} &= f(-1000, t_s, \xi^{sl}) \\ \mu_w^{sl} &= f(p_s, -1000, \xi^{sl}) \\ \mu_w^{sl} &= f(p_s, t_s, -1000) \\ \mu_w^{sl} &= f(p_s, t_s, \xi^{sl})\end{aligned}$$

**Result for Wrong Input Values:**

mywsl\_pstsXisl\_SeaWa, mywsl = -1000

**References:**

$$\mu_w(p, t, \xi) \quad [1]$$

**Chemical Potential of Sea Salt of Liquid Seawater  $\mu_{\text{Salt I}} = f(p, t, \xi)$** 
**Function Name:**

mySaltI\_ptXi\_SeaWa

**Fortran Programs:**

```
REAL*8 FUNCTION MYSALTPTXI_SEAWA(P, T, XI), REAL*8 P, T, XI
```

**Input Values:**

$p$  - Pressure  $p$  in bar  
 $t$  - Temperature  $t$  in °C  
 $X_i$  - Mass fraction of sea salt  $\xi$  in kg sea salt/kg mixture

**Result:**

mySaltI\_ptXi\_SeaWa, mysaltI - Chemical potential of sea salt of liquid seawater  
in [kJ/kg]

**Range of Validity:**

Temperature  $t$ : from 0 °C to 80 °C  
Pressure  $p$ : from  $p_s(t = 0 \text{ °C}, \xi = 0.12)$  to 1000 bar and  $p \geq p_{\text{mel}}(t, \xi)$   
Mass fraction  $\xi$ :  $0 \leq \xi \leq 0.12$  kg sea salt/kg mixture

**Result for Wrong Input Values:**

mySaltI\_ptXi\_SeaWa, mySaltI = -1000

**References:**

$\mu_S(p, t, \xi)$  [1]

**Chemical Potential of Sea Salt of Saturated Liquid  $\mu_{\text{Salt sl}} = f(p_s, t_s, \xi^{\text{sl}})$** 
**Function Name:**

mySaltsl\_pstsXisl\_SeaWa

**Fortran Programs:**

```

      REAL*8 FUNCTION  MYSALTSL_PSTSLXISL_SEAWA(PS, TS, XISL)
      REAL*8 PS, TS, XISL
      ps      - Pressure p in bar
      ts      - Temperature t in °C
      Xisl    - Mass fraction of sea salt  $\xi$  in kg sea salt/kg mixture
  
```

**Result:**

mySaltsl\_pstsXisl\_SeaWa, mySaltsl - Relative chemical potential of H2O of saturated seawater in [kJ/kg]

**Range of Validity:**

Temperature *t*: from 0 °C to 80 °C  
 Mixture pressure *p*: from  $p_s(t = 0\text{ °C}, \xi = 0.12)$  to  $p_s(t = 80\text{ °C}, \xi = 0)$   
 Mass fraction  $\xi$ :  $0 \leq \xi \leq 0.12$  kg sea salt/kg mixture

**Possible input variants:**

$$\begin{aligned} \mu_{\text{Salt}}^{\text{sl}} &= f(-1000, t_s, \xi^{\text{sl}}) \\ \mu_{\text{Salt}}^{\text{sl}} &= f(p_s, -1000, \xi^{\text{sl}}) \\ \mu_{\text{Salt}}^{\text{sl}} &= f(p_s, t_s, -1000) \\ \mu_{\text{Salt}}^{\text{sl}} &= f(p_s, t_s, \xi^{\text{sl}}) \end{aligned}$$

**Result for Wrong Input Values:**

mySaltsl\_pstsXisl\_SeaWa, mySaltsl = -1000

**References:**

$$\mu_s(p, t, \xi) \quad [1]$$

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

ny\_ptXi\_SeaWa

**Fortran Programs:**

```
REAL*8 FUNCTION NY_PTXI_SEAWA(P, T, XI), REAL*8 P, T, XI
```

**Input Values:**

$p$  - Pressure  $p$  in bar  
 $t$  - Temperature  $t$  in °C  
 $Xi$  - Mass fraction of sea salt  $\xi$  in kg sea salt/kg mixture

**Result:**

ny\_ptXi\_SeaWa, ny - Kinematic Viscosity in m<sup>2</sup>/s

**Range of Validity:**

Temperature  $t$ : from 0 °C to 220 °C  
 Pressure  $p$ : from  $p_s(t = 0 \text{ °C}, \xi = 0.2)$  to 100 bar and  $p \geq p_{\text{mel}}(t, \xi)$   
 Mass fraction  $\xi$ :  $0 \leq \xi \leq 0.2$  kg sea salt/kg mixture

**Comments:**

- Kinematic Viscosity  $\nu = \frac{\eta}{\rho}$
- This function is not defined in the wet steam region.

**Result for Wrong Input Values:**

ny\_ptXi\_SeaWa, ny = -1000

**References:**

$\nu(p, t, \xi)$  [1], [2]

**Kinematic Viscosity of Liquid Seawater  $\nu_l = f(t, \xi)$** 
**Function Name:**

nyl\_tXi\_SeaWa

**Fortran Programs:**

REAL\*8 FUNCTION NYL\_TXI\_SEAWA(T, XI), REAL\*8 T, XI

**Input Values:**
 $t$  - Temperature  $t$  in °C

 $X_i$  - Mass fraction of sea salt  $\xi$  in kg sea salt/kg mixture

**Result:**

 nyl\_tXi\_SeaWa, nyl - Kinematic Viscosity of liquid seawater in m<sup>2</sup>/s

**Range of Validity:**

 Temperature  $t$ : from 0 °C to 220 °C

 Mass fraction  $\xi$ :  $0 \leq \xi \leq 0.2$  kg sea salt/kg mixture

**Comments:**

 - Kinematic Viscosity  $\nu = \frac{\eta}{\rho}$ 
**Result for Wrong Input Values:**

nyl\_tXi\_SeaWa, nyl = -1000

**References:**
 $\nu(p, t, \xi)$  [1],[2]

**Kinematic Viscosity of Saturated Liquid  $\nu_{sl} = f(p_s, t_s, \xi^{sl})$** 
**Function Name:**

nysl\_pstsXisl\_SeaWa

**Fortran Programs:**

```
REAL*8 FUNCTION NYSL_PSTSXISL_SEAWA(PS, TS, XISL), REAL*8 PS, TS, XISL
```

**Input Values:**

$ps$  - Pressure  $p$  in bar  
 $ts$  - Temperature  $t$  in °C  
 $Xisl$  - Mass fraction of sea salt  $\xi$  in kg sea salt/kg mixture

**Result:**

nysl\_pstsXisl\_SeaWa, nysl - Kinematic Viscosity of saturated seawater in m<sup>2</sup>/s

**Range of Validity:**

Temperature  $t$ : from 0 °C to 220 °C  
 Mixture pressure  $p$ : from  $p_s(t = 0 \text{ °C}, \xi = 0.2)$  to  $p_s(t = 220 \text{ °C}, \xi = 0)$   
 Mass fraction  $\xi$ :  $0 \leq \xi \leq 0.2$  kg sea salt/kg mixture

**Comments:**

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

**Possible input variants:**

$$\begin{aligned}\nu^{sl} &= f(-1000, t_s, \xi^{sl}) \\ \nu^{sl} &= f(p_s, -1000, \xi^{sl}) \\ \nu^{sl} &= f(p_s, t_s, -1000) \\ \nu^{sl} &= f(p_s, t_s, \xi^{sl})\end{aligned}$$

**Result for Wrong Input Values:**

nysl\_pstsXisl\_SeaWa, nysl = -1000

**References:**

$\nu(p, t, \xi)$  [1],[2]

**Kinematic Viscosity of Saturated Vapor  $\nu_{sv} = f(p_s, t_s, \xi^{sl})$** 
**Function Name:**

nysv\_pstsXisl\_SeaWa

**Fortran Programs:**

```
REAL*8 FUNCTION ASV_PSTSXISL_SEAWA(PS,TS,XISL), REAL*8 PS,TS,XISL
```

**Input Values:**

$ps$  - Pressure  $p$  in bar  
 $ts$  - Temperature  $t$  in °C  
 $Xisl$  - Mass fraction of sea salt  $\xi$  in kg sea salt/kg mixture

**Result:**

nysv\_pstsXisl\_SeaWa, nysv - Kinematic Viscosity of saturated seawater in m<sup>2</sup>/s

**Range of Validity:**

Temperature  $t$ : from 0 °C to 220 °C  
 Mixture pressure  $p$ : from  $p_s(t = 0\text{ °C}, \xi = 0.2)$  to  $p_s(t = 220\text{ °C}, \xi = 0)$   
 Mass fraction  $\xi$ :  $0 \leq \xi \leq 0.2$  kg sea salt/kg mixture

**Comments:**

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

**Result for Wrong Input Values:**

nysv\_pstsXisl\_SeaWa, nysv = -1000

**References:**

$\nu(p, t, \xi)$  [1],[2]



**Boiling Pressure  $p_s = f(t_s, \xi^{sl})$** 
**Function Name:**

ps\_tsXisl\_SeaWa

**Fortran Programs:**

```
REAL*8 FUNCTION PS_TSXISL_SEAWA( TS, XISL), REAL*8 TS, XISL
```

**Input Values:**

$t_s$  - Saturated Temperature  $t_s$  in °C  
 $X_i$  - Mass fraction of sea salt  $\xi$  in kg sea salt/kg mixture

**Result:**

ps\_tsXisl\_SeaWa, ps - Saturation Pressure in bar

**Range of Validity:**

Temperature  $t$  from 0 °C to 220 °C  
 Mass fraction  $\xi$ :  $0 \leq \xi \leq 0.2$  kg sea salt/kg mixture

**Result for Wrong Input Values:**

ps\_tsXisl\_SeaWa, ps = -1000

**References:**

$p_s(t_s, \xi_{sl})$  [1], [2],[4]

**Freezing Pressure  $p_{\text{mel}} = f(t, \xi)$** 
**Function Name:**

pmel\_tXi\_SeaWa

**Fortran Programs:**

```
REAL*8 FUNCTION PMEL_TXI_SEAWA( T, XI), REAL*8 T, XI
```

**Input Values:**

$t$  - Temperature  $t$  in °C

$Xi$  - Mass fraction of sea salt  $\xi$  in kg sea salt/kg mixture

**Result:**

pmel\_tXi\_SeaWa, pmel - Melting Pressure in bar

**Range of Validity:**

Temperature  $t$ : from 0 °C to 0.01 °C

Mass fraction  $\xi$ :  $0 \leq \xi \leq 0.12$  kg sea salt/kg mixture

**Comments:**

- If  $p_{\text{mel}} > 1000$  bar then pmel\_tXi\_SeaWa, pmel = -1000
- If  $p_{\text{mel}} < p_{\text{tr}}(\xi)$  then pmel\_tXi\_SeaWa, pmel = -1000

**Result for Wrong Input Values:**

pmel\_tXi\_SeaWa, pmel = -1000

**References:**

$p_{\text{mel}}(t, \xi)$  [1], [4], [5]

**Triple Point Pressure  $p_{tr} = f(\xi)$** 
**Function Name:**

ptr\_Xi\_SeaWa

**Fortran Programs:**

```
REAL*8 FUNCTION PTR_XI_SEAWA( XI), REAL*8 XI
```

**Input Values:**

$X_i$  - Mass fraction of sea salt  $\xi$  in kg sea salt/kg mixture

**Result:**

ptr\_Xi\_SeaWa, ptr - Triplepoint Pressure in bar

**Range of Validity:**

Mass fraction  $\xi$ :  $0 \leq \xi \leq 0.12$  kg sea salt/kg mixture

**Result for Wrong Input Values:**

ptr\_Xi\_SeaWa, ptr = -1000

**References:**

$p_{tr}(\xi)$  [1], [4], [5]

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

Pr\_ptXi\_SeaWa

**Fortran Programs:**

REAL\*8 FUNCTION PR\_PTXI\_SEAWA(P, T, XI), REAL\*8 P, T, XI

**Input Values:**

$p$  - Pressure  $p$  in bar  
 $t$  - Temperature  $t$  in °C  
 $Xi$  - Mass fraction of sea salt  $\xi$  in kg sea salt/kg mixture

**Result:**

Pr\_ptXi\_SeaWa, Pr - Prandtl Number in [-]

**Range of Validity:**

Temperature  $t$ : from 0 °C to 220 °C  
 Pressure  $p$ : from  $p_s(t = 0\text{ °C}, \xi = 0.2)$  to 100 bar and  $p \geq p_{\text{mel}}(t, \xi)$   
 Mass fraction  $\xi$ :  $0 \leq \xi \leq 0.2$  kg sea salt/kg mixture

**Comments:**

- Prandtl Number  $Pr = \frac{\nu}{a}$
- This function is not defined in the wet steam region.

**Result for Wrong Input Values:**

Pr\_ptXi\_SeaWa, Pr = -1000

**References:**

$Pr(p, t, \xi)$  [1], [2], [3]

**Prandtl Number of Liquid Seawater  $Pr_l = f(t, \xi)$** 
**Function Name:**

Prl\_tXi\_SeaWa

**Fortran Programs:**

```
REAL*8 FUNCTION PRL_TXI_SEAWA(P, T, XI), REAL*8 P, T, XI
```

**Input Values:**

$p$  - Pressure  $p$  in bar  
 $t$  - Temperature  $t$  in °C  
 $X_i$  - Mass fraction of sea salt  $\xi$  in kg sea salt/kg mixture

**Result:**

Prl\_ptXi\_SeaWa, Prl - Prandtl Number of liquid seawater in [-]

**Range of Validity:**

Temperature  $t$ : from 0 °C to 220 °C  
 Mass fraction  $\xi$ :  $0 \leq \xi \leq 0.2$  kg sea salt/kg mixture

**Comments:**

- Prandtl Number  $Pr = \frac{V}{a}$

**Result for Wrong Input Values:**

Prl\_tXi\_SeaWa, Prl = -1000

andtl Number **References:**

$Pr(p, t, \xi)$  [1], [2], [3]

**Prandtl Number of Saturated Liquid  $Pr_{sl} = f(p_s, t_s, \xi^{sl})$** 
**Function Name:**

Prsl\_pstsXisl\_SeaWa

**Fortran Programs:**

REAL\*8 FUNCTION PRSL\_PSTSXISL\_SEAWA(PS, TS, XISL), REAL\*8 PS, TS, XISL

**Input Values:**

$ps$  - Pressure  $p$  in bar  
 $ts$  - Temperature  $t$  in °C  
 $Xisl$  - Mass fraction of sea salt  $\xi$  in kg sea salt/kg mixture

**Result:**

Prsl\_pstsXisl\_SeaWa, Prsl - Prandtl Number of saturated seawater in [-]

**Range of Validity:**

Temperature  $t$ : from 0 °C to 220 °C  
 Mixture pressure  $p$ : from  $p_s(t = 0 \text{ °C}, \xi = 0.2)$  to  $p_s(t = 220 \text{ °C}, \xi = 0)$   
 Mass fraction  $\xi$ :  $0 \leq \xi \leq 0.2$  kg sea salt/kg mixture

**Possible input variants:**

$$Pr^{sl} = f(-1000, t_s, \xi^{sl})$$

$$Pr^{sl} = f(p_s, -1000, \xi^{sl})$$

$$Pr^{sl} = f(p_s, t_s, -1000)$$

$$Pr^{sl} = f(p_s, t_s, \xi^{sl})$$

**Comments:**

- Prandtl Number  $Pr = \frac{\nu}{a}$

**Result for Wrong Input Values:**

Prsl\_ptXi\_SeaWa, Prsl = -1000

**References:**

$Pr(p, t, \xi)$  [1], [2], [3]

**Prandtl Number of Saturated Vapor  $Pr_{sv} = f(p_s, t_s, \xi^{sl})$** 
**Function Name:**

Prsv\_pstsXisl\_SeaWa

**Fortran Programs:**

```
REAL*8 FUNCTION PRSV_PSTXISL_SEAWA(PS,TS), REAL*8 PS,TS,XISL
```

**Input Values:**

$ps$  - Pressure  $p$  in bar  
 $ts$  - Temperature  $t$  in °C  
 $Xisl$  - Mass fraction of sea salt  $\xi$  in kg sea salt/kg mixture

**Result:**

Prsv\_pstsXisl\_SeaWa, Prsv - Prandtl Number of saturated seawater in m<sup>2</sup>/s

**Range of Validity:**

Temperature  $t$ : from 0 °C to 220 °C  
 Mixture pressure  $p$ : from  $p_s(t = 0 \text{ °C}, \xi = 0.2)$  to  $p_s(t = 220 \text{ °C}, \xi = 0)$   
 Mass fraction  $\xi$ :  $0 \leq \xi \leq 0.2$  kg sea salt/kg mixture

**Comments:**

- Prandtl Number  $Pr = \frac{\nu}{a}$

**Result for Wrong Input Values:**

Prsv\_pstsXisl\_SeaWa, Prsv = -1000

**References:**

$Pr(p, t, \xi)$  [1], [2], [3]

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

rho\_ptXi\_SeaWa

**Fortran Programs:**

```
REAL*8 FUNCTION RHO_PTXI_SEAWA(P, T, XI), REAL*8 P, T, XI
```

**Input Values:**

$p$  - Pressure  $p$  in bar  
 $t$  - Temperature  $t$  in °C  
 $Xi$  - Mass fraction of sea salt  $\xi$  in kg sea salt/kg mixture

**Result:**

rho\_ptXi\_SeaWa, rho - Density in kg/m³

**Range of Validity:**

Temperature  $t$ : from 0 °C to 220 °C  
 Pressure  $p$ : from  $p_s(t = 0 \text{ °C}, \xi = 0.12)$  to 1000 bar and  $p \geq p_{\text{mel}}(t, \xi)$   
 Mass fraction  $\xi$ :  $0 \leq \xi \leq 0.2$  kg sea salt/kg mixture

**Result for Wrong Input Values:**

rho\_ptXi\_SeaWa, rho = -1000

**References:**

$\rho(p, t, \xi)$  [1], [2]



**Density of Liquid Seawater  $\rho_l = f(p, t, \xi)$** 
**Function Name:**

rhoI\_ptXi\_SeaWa

**Fortran Programs:**

```
REAL*8 FUNCTION RHOL_PTXI_SEAWA(P, T, XI), REAL*8 P, T, XI
```

**Input Values:**

$p$  - Pressure  $p$  in bar  
 $t$  - Temperature  $t$  in °C  
 $Xi$  - Mass fraction of sea salt  $\xi$  in kg sea salt/kg mixture

**Result:**

rhoI\_ptXi\_SeaWa, rhoI - Density of liquid seawater in kg/m³

**Range of Validity:**

Temperature  $t$ : from 0 °C to 220 °C  
 Pressure  $p$ : from  $p_s(t = 0\text{ °C}, \xi = 0.12)$  to 1000 bar and  $p \geq p_{\text{mel}}(t, \xi)$   
 Mass fraction  $\xi$ :  $0 \leq \xi \leq 0.2$  kg sea salt/kg mixture

**Result for Wrong Input Values:**

rhoI\_ptXi\_SeaWa, rhoI = -1000

**References:**

$\rho(p, t, \xi)$  [1], [2], [3]

**Density of Saturated Liquid  $\rho_{sl} = f(p_s, t_s, \xi^{sl})$** 
**Function Name:**

rhosl\_pstsXisl\_SeaWa

**Fortran Programs:**

REAL\*8 FUNCTION RHOSL\_PSTSXISL\_SEAWA(PS, TS, XISL), REAL\*8 PS, TS, XISL

**Input Values:**

$p_s$  - Pressure  $p$  in bar  
 $t_s$  - Temperature  $t$  in °C  
 $Xisl$  - Mass fraction of sea salt  $\xi$  in kg sea salt/kg mixture

**Result:**

rhosl\_pstsXisl\_SeaWa, rhosl - Density of saturated seawater in kg/m³

**Range of Validity:**

Temperature  $t$ : from 0 °C to 220 °C  
 Pressure  $p$ : from  $p_s(t = 0 \text{ °C}, \xi = 0.12)$  to  $p_s(t = 220 \text{ °C}, \xi = 0)$   
 Mass fraction  $\xi$ :  $0 \leq \xi \leq 0.2$  kg sea salt/kg mixture

**Possible input variants:**

$$\begin{aligned}\rho^{sl} &= f(-1000, t_s, \xi^{sl}) \\ \rho^{sl} &= f(p_s, -1000, \xi^{sl}) \\ \rho^{sl} &= f(p_s, t_s, -1000) \\ \rho^{sl} &= f(p_s, t_s, \xi^{sl})\end{aligned}$$

**Result for Wrong Input Values:**

rhosl\_pstsXisl\_SeaWa, rhosl = -1000

**References:**

$\rho(p, t, \xi)$  [1], [2], [3]

**Density of Saturated Vapor  $\rho_{sv} = f(p_s, t_s, \xi^{sl})$** 
**Function Name:**

rhosv\_pstsXisl\_SeaWa

**Fortran Programs:**

```
REAL*8 FUNCTION RHOSV_PSTSL_SEAWA(P,TS,XISL), REAL*8 P,TS,XISL
```

**Input Values:**

$p_s$  - Pressure  $p$  in bar  
 $t_s$  - Temperature  $t$  in °C  
 $Xisl$  - Mass fraction of sea salt  $\xi$  in kg sea salt/kg mixture

**Result:**

rhosv\_pstsXisl\_SeaWa, rhosv - Density of saturated seawater in kg/m³

**Range of Validity:**

Temperature  $t$ : from 0 °C to 220 °C  
 Pressure  $p$ : from  $p_s(t = 0 \text{ °C}, \xi = 0.2)$  to  $p_s(t = 220 \text{ °C}, \xi = 0)$   
 Mass fraction  $\xi$ :  $0 \leq \xi \leq 0.2$  kg sea salt/kg mixture

**Result for Wrong Input Values:**

rhosv\_pstsXisl\_SeaWa, rhosv = -1000

**References:**

$\rho(p, t, \xi)$  [1], [2], [3]

**Specific Entropy  $s = f(p, t, \xi)$** 
**Function Name:**

s\_ptXi\_SeaWa

**Fortran Programs:**

```
REAL*8 FUNCTION S_PTXI_SEAWA(P, T, XI), REAL*8 P, T, XI
```

**Input Values:**

$p$  - Pressure  $p$  in bar  
 $t$  - Temperature  $t$  in °C  
 $Xi$  - Mass fraction of sea salt  $\xi$  in kg sea salt/kg mixture

**Result:**

s\_ptXi\_SeaWa,  $s$  - specific entropy in kJ/(kg\*K)

**Range of Validity:**

Temperature  $t$ : from 0 °C to 220 °C  
 Pressure  $p$ : from  $p_s(t = 0 \text{ °C}, \xi = 0.2)$  to 1000 bar and  $p \geq p_{\text{mel}}(t, \xi)$   
 Mass fraction  $\xi$ :  $0 \leq \xi \leq 0.2$  kg sea salt/kg mixture

**Result for Wrong Input Values:**

s\_ptXi\_SeaWa,  $s = -1000$

**References:**

$s(p, t, \xi)$  [1], [2]

**Specific Entropy of Liquid Seawater  $s_l = f(p, t, \xi)$** 
**Function Name:**

sl\_ptXi\_SeaWa

**Fortran Programs:**

```
REAL*8 FUNCTION SL_PTXI_SEAWA(P, T, XI), REAL*8 P, T, XI
```

**Input Values:**

$p$  - Pressure  $p$  in bar  
 $t$  - Temperature  $t$  in °C  
 $Xi$  - Mass fraction of sea salt  $\xi$  in kg sea salt/kg mixture

**Result:**

sl\_ptXi\_SeaWa, sl - Specific Entropy of liquid seawater in kJ/(kg\*K)

**Range of Validity:**

Temperature  $t$ : from 0 °C to 220 °C  
 Pressure  $p$ : from  $p_s(t = 0 \text{ °C}, \xi = 0.2)$  to 1000 bar and  $p \geq p_{\text{mel}}(t, \xi)$   
 Mass fraction  $\xi$ :  $0 \leq \xi \leq 0.2$  kg sea salt/kg mixture

**Result for Wrong Input Values:**

sl\_ptXi\_SeaWa, sl = -1000

**References:**

$s(p, t, \xi)$  [1], [2]

**Specific Entropy of Saturated Liquid  $s_{sl} = f(p_s, t_s, \xi^{sl})$** 
**Function Name:**

ssl\_pstsXisl\_SeaWa

**Fortran Programs:**

REAL\*8 FUNCTION SSL\_PSTXISL\_SEAWA(PS, TS, XISL), REAL\*8 PS, TS, XISL

**Input Values:**

$ps$  - Pressure  $p$  in bar  
 $ts$  - Temperature  $t$  in °C  
 $Xisl$  - Mass fraction of sea salt  $\xi$  in kg sea salt/kg mixture

**Result:**

ssl\_pstsXisl\_SeaWa, ssl - Specific Entropy of saturated seawater in kJ/(kg\*K)

**Range of Validity:**

Temperature  $t$ : from 0 °C to 220 °C  
 Pressure  $p$ : from  $p_s(t = 0 \text{ °C}, \xi = 0.2)$  to  $p_s(t = 220 \text{ °C}, \xi = 0)$   
 Mass fraction  $\xi$ :  $0 \leq \xi \leq 0.2$  kg sea salt/kg mixture

**Possible input variants:**

$$s^{sl} = f(-1000, t_s, \xi^{sl})$$

$$s^{sl} = f(p_s, -1000, \xi^{sl})$$

$$s^{sl} = f(p_s, t_s, -1000)$$

$$s^{sl} = f(p_s, t_s, \xi^{sl})$$

**Result for Wrong Input Values:**

ssl\_pstsXi\_SeaWa, ssl = -1000

**References:**

$s(p, t, \xi)$  [1] , [2]

**Specific Entropy of Saturated Vapor  $s_{sv} = f(p_s, t_s, \xi^{sl})$** 
**Function Name:**

ssv\_pstsXisl\_SeaWa

**Fortran Programs:**

```
REAL*8 FUNCTION SSV_PSTSXISL_SEAWA(PS,TS,XISL), REAL*8 PS,TS,XISL
```

**Input Values:**

$ps$  - Pressure  $p$  in bar  
 $ts$  - Temperature  $t$  in °C  
 $Xisl$  - Mass fraction of sea salt  $\xi$  in kg sea salt/kg mixture

**Result:**

ssv\_pstsXisl\_SeaWa, ssv - Specific Entropy of saturated seawater in kJ/(kg\*K)

**Range of Validity:**

Temperature  $t$ : from 0 °C to 220 °C  
 Pressure  $p$ : from  $p_s(t = 0 \text{ °C}, \xi = 0.2)$  to  $p_s(t = 220 \text{ °C}, \xi = 0)$   
 Mass fraction  $\xi$ :  $0 \leq \xi \leq 0.2$  kg sea salt/kg mixture

**Result for Wrong Input Values:**

ssv\_pstsXisl\_SeaWa, ssv = -1000

**References:**

$s(p, t, \xi)$  [1] , [2]

**Boiling Temperature  $t_s = f(p_s, \xi^{sl})$** 
**Function Name:**

ts\_psXisl\_SeaWa

**Fortran Programs:**

```
REAL*8 FUNCTION TS_PSXISL_SEAWA( PS, XISL), REAL*8 PS, XISL
```

**Input Values:**

*ps* - Pressure  $p$  in bar

*Xisl* - Mass fraction of sea salt  $\xi$  in kg sea salt/kg mixture

**Result:**

ts\_psXisl\_SeaWa, ts - Saturation Temperature in °C

**Range of Validity:**

Pressure  $p$ : from  $p_s(t = 0\text{ °C}, \xi = 0.2)$  to  $p_s(t = 220\text{ °C}, \xi = 0)$

Mass fraction  $\xi$ :  $0 \leq \xi \leq 0.2$  kg sea salt/kg mixture

**Result for Wrong Input Values:**

ts\_psXisl\_SeaWa, ts = -1000

**References:**

$t_s(p_s, \xi_{sl})$  [1], [2], [4]



**Freezing Temperature  $t_{\text{mel}} = f(p, \xi)$** 
**Function Name:**

tmel\_pXi\_SeaWa

**Fortran Programs:**

```
REAL*8 FUNCTION TMEL_PXI_SEAWA( P, XI), REAL*8 P, XI
```

**Input Values:**

$p$  - Pressure  $p$  in bar

$Xi$  - Mass fraction of sea salt  $\xi$  in kg sea salt/kg mixture

**Result:**

tmel\_pXi\_SeaWa, tmel - Melting Temperature in °C

**Range of Validity:**

Pressure  $p$ : from  $p_s(t = 0\text{ °C}, \xi = 0.12)$  to 1000 bar

Mass fraction  $\xi$ :  $0 \leq \xi \leq 0.12$  kg sea salt/kg mixture

**Comments:**

- If  $t_{\text{mel}}(p, \xi) < -12.15\text{ °C}$  then tmel\_pXi\_SeaWa, tmel = -1000

**Result for Wrong Input Values:**

tmel\_pXi\_SeaWa, tmel = -1000

**References:**

$t_{\text{mel}}(p, \xi)$  [1], [4], [5]

**Triple Point Temperature  $t_{tr} = f(\xi)$** 
**Function Name:**

ttr\_Xi\_SeaWa

**Fortran Programs:**

```
REAL*8 FUNCTION TTR_XI_SEAWA( XI), REAL*8 XI
```

**Input Values:**

$X_i$  - Mass fraction of sea salt  $\xi$  in kg sea salt/kg mixture

**Result:**

ttr\_Xi\_SeaWa, ttr - Triplepoint Temperature in °C

**Range of Validity:**

Mass fraction  $\xi$ :  $0 \leq \xi \leq 0.12$  kg sea salt/kg mixture

**Result for Wrong Input Values:**

ttr\_Xi\_SeaWa, ttr = -1000

**References:**

$t_{tr}(\xi)$  [1], [4], [5]

**Region**  $Region = f(p, t, \xi)$ 
**Function Name:**

Region\_ptXi\_SeaWa

**Fortran Programs:**

```
REAL*8 FUNCTION REGION_PTXI_SEAWA(P, T, XI), REAL*8 P, T, XI
```

**Input Values:**

$p$  - Pressure  $p$  in bar  
 $t$  - Temperature  $t$  in °C  
 $Xi$  - Mass fraction of sea salt  $\xi$  in kg sea salt/kg mixture

**Result:**

Region\_ptXi\_SeaWa, Region - Region in [-]

**Range of Validity:**

Temperature  $t$ : from 0 °C to 220 °C  
 Pressure  $p$ : from  $p_s(t = 0 \text{ °C}, \xi = 0.2)$  to 1000 bar and  $p \geq p_{\text{mel}}(t, \xi)$   
 Mass fraction  $\xi$ :  $0 \leq \xi \leq 0.2$  kg sea salt/kg mixture

**Comments:**

- Region : 1 subcooled liquid calculated from IAPWS Industrial Formulation 2013
- 2 wet steam region calculated from IAPWS Industrial Formulation 2013
- 3 superheated steam calculated from IAPWS Industrial Formulation 2013
- 11 subcooled liquid calculated from Fichtner-Handbook
- 22 wet steam region calculated from Fichtner-Handbook
- 33 superheated steam calculated from Fichtner-Handbook

**Result for Wrong Input Values:**

Region\_ptXi\_SeaWa, Region = -1000

**References:**

Region( $p, t, \xi$ ) [1], [2], [4]

**Region**  $Region = f(p, h, \xi)$ 
**Function Name:**

Region\_phXi\_SeaWa

**Fortran Programs:**

REAL\*8 FUNCTION REGION\_PHXI\_SEAWA(P, H, XI), REAL\*8 P, H, XI

**Input Values:**

$p$  - Pressure  $p$  in bar  
 $h$  - Specific Enthalpy  $h$  in kJ/kg  
 $Xi$  - Mass fraction of sea salt  $\xi$  in kg sea salt/kg mixture

**Result:**

Region\_phXi\_SeaWa, Region - Region in [-]

**Range of Validity:**

Pressure  $p$ : from  $p_s(t = 0\text{ °C}, \xi = 0.2)$  to 1000 bar and  $p \geq p_{\text{mel}}(t, \xi)$   
 Enthalpy  $h$ : from  $h_{\text{min}}$  kJ/kg to  $h_{\text{max}}$  kJ/kg  
 Mass fraction  $\xi$ :  $0 \leq \xi \leq 0.2$  kg sea salt/kg mixture

**Comments:**

- Region : 1 subcooled liquid calculated from IAPWS Industrial Formulation 2013
- 2 wet steam region calculated from IAPWS Industrial Formulation 2013
- 3 superheated steam calculated from IAPWS Industrial Formulation 2013
- 11 subcooled liquid calculated from Fichtner-Handbook
- 22 wet steam region calculated from Fichtner-Handbook
- 33 superheated steam calculated from Fichtner-Handbook

**Result for Wrong Input Values:**

Region\_phXi\_SeaWa, Region = -1000

**References:**

$Region(p, h, \xi)$  [1], [2], [4]

**Region**  $Region = f(p, s, \xi)$ 
**Function Name:**

Region\_psXi\_SeaWa

**Fortran Programs:**

REAL\*8 FUNCTION REGION\_P SXI\_SEAWA(P, S, XI), REAL\*8 P, S, XI

**Input Values:**

$p$  - Pressure  $p$  in bar  
 $s$  - Specific Entropy  $s$  in kJ/(kg\*K)  
 $Xi$  - Mass fraction of sea salt  $\xi$  in kg sea salt/kg mixture

**Result:**

Region\_psXi\_SeaWa, Region - Region in [-]

**Range of Validity:**

Pressure  $p$ : from  $p_s(t = 0\text{ °C}, \xi = 0.2)$  to 1000 bar and  $p \geq p_{\text{mel}}(t, \xi)$   
 Entropy  $s$ : from  $s_{\text{min}}$  kJ/(kg K) to  $s_{\text{max}}$  kJ/(kg K)  
 Mass fraction  $\xi$ :  $0 \leq \xi \leq 0.2$  kg sea salt/kg mixture

**Comments:**

- Region : 1 subcooled liquid calculated from IAPWS Industrial Formulation 2013
- 2 wet steam region calculated from IAPWS Industrial Formulation 2013
- 3 superheated steam calculated from IAPWS Industrial Formulation 2013
- 11 subcooled liquid calculated from Fichtner-Handbook
- 22 wet steam region calculated from Fichtner-Handbook
- 33 superheated steam calculated from Fichtner-Handbook

**Result for Wrong Input Values:**

Region\_psXi\_SeaWa, Region = -1000

**References:**

$Region(p, s, \xi)$  [1], [2], [4]

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

t\_phXi\_SeaWa

**Fortran Programs:**

```
REAL*8 FUNCTION T_PHXI_SEAWA(P, H, XI), REAL*8 P, H, XI
```

**Input Values:**

$p$  - Pressure  $p$  in bar  
 $h$  - Specific Enthalpy  $h$  in kJ/kg  
 $Xi$  - Mass fraction of sea salt  $\xi$  in kg sea salt/kg mixture

**Result:**

t\_phXi\_SeaWa, t - Temperature in °C

**Range of Validity:**

Pressure  $p$ : from  $p_s(t = 0\text{ °C}, \xi = 0.2)$  to 1000 bar and  $p \geq p_{\text{mel}}(t, \xi)$   
 Enthalpy  $h$ : from  $h_{\text{min}}$  kJ/kg to  $h_{\text{max}}$  kJ/kg  
 Mass fraction  $\xi$ :  $0 \leq \xi \leq 0.2$  kg sea salt/kg mixture

**Result for Wrong Input Values:**

t\_phXi\_SeaWa, t = -1000

**References:**

$t(p, h, \xi)$  [1], [2], [4]

**Backward Function: Temperature  $t = f(p, s, \xi)$** 
**Function Name:**

t\_psXi\_SeaWa

**Fortran Programs:**

```
REAL*8 FUNCTION T_PSI_SEAWA(P, S, XI), REAL*8 P, S, XI
```

**Input Values:**

$p$  - Pressure  $p$  in bar  
 $s$  - Specific Entropy  $h$  in kJ/(kg\*K)  
 $Xi$  - Mass fraction of sea salt  $\xi$  in kg sea salt/kg mixture

**Result:**

t\_psXi\_SeaWa, t - Temperature in °C

**Range of Validity:**

Pressure  $p$ : from  $p_s(t = 0\text{ °C}, \xi = 0.2)$  to 1000 bar and  $p \geq p_{\text{mel}}(t, \xi)$   
 Entropy  $s$ : from  $s_{\text{min}}$  kJ/(kg K) to  $s_{\text{max}}$  kJ/(kg K)  
 Mass fraction  $\xi$ :  $0 \leq \xi \leq 0.2$  kg sea salt/kg mixture

**Result for Wrong Input Values:**

t\_psXi\_SeaWa, t = -1000

**References:**

$t(p, s, \xi)$  [1], [2], [4]

**Specific Internal Energy  $u = f(p, t, \xi)$** 
**Function Name:**

u\_ptXi\_SeaWa

**Fortran Programs:**

```
REAL*8 FUNCTION U_PTXI_SEAWA(P, T, XI), REAL*8 P, T, XI
```

**Input Values:**

$p$  - Pressure  $p$  in bar  
 $t$  - Temperature  $t$  in °C  
 $Xi$  - Mass fraction of sea salt  $\xi$  in kg sea salt/kg mixture

**Result:**

u\_ptXi\_SeaWa,  $u$  - Specific internal energy in kJ/kg

**Range of Validity:**

Temperature  $t$ : from 0 °C to 220 °C  
 Pressure  $p$ : from  $p_s(t = 0 \text{ °C}, \xi = 0.2)$  to 1000 bar and  $p \geq p_{\text{mel}}(t, \xi)$   
 Mass fraction  $\xi$ :  $0 \leq \xi \leq 0.2$  kg sea salt/kg mixture

**Comments:**

- Specific internal energy  $u = h - (p * v)$

**Result for Wrong Input Values:**

u\_ptXi\_SeaWa,  $u = -1000$

**References:**

$u(p, t, \xi)$  [1], [2]



**Specific Internal Energy of Liquid Seawater  $u_l = f(p, t, \xi)$** 
**Function Name:**

ul\_ptXi\_SeaWa

**Fortran Programs:**

```
REAL*8 FUNCTION UL_PTXI_SEAWA(P, T, XI), REAL*8 P, T, XI
```

**Input Values:**

$p$  - Pressure  $p$  in bar  
 $t$  - Temperature  $t$  in °C  
 $X_i$  - Mass fraction of sea salt  $\xi$  in kg sea salt/kg mixture

**Result:**

ul\_ptXi\_SeaWa, ul - Specific internal energy of liquid seawater in kJ/kg

**Range of Validity:**

Temperature  $t$ : from 0 °C to 220 °C  
 Pressure  $p$ : from  $p_s(t = 0 \text{ °C}, \xi = 0.2)$  to 1000 bar and  $p \geq p_{\text{mel}}(t, \xi)$   
 Mass fraction  $\xi$ :  $0 \leq \xi \leq 0.2$  kg sea salt/kg mixture

**Comments:**

- Specific internal energy  $u = h - (p * v)$

**Result for Wrong Input Values:**

ul\_ptXi\_SeaWa, ul = -1000

**References:**

$u(p, t, \xi)$  [1] , [2]

**Specific Internal Energy of Saturated Liquid  $u_{sl} = f(p_s, t_s, \xi^{sl})$** 
**Function Name:**

usl\_pstsXisl\_SeaWa

**Fortran Programs:**

REAL\*8 FUNCTION USL\_PSTSXISL\_SEAWA(PS, TS, XISL), REAL\*8 PS, TS, XISL

**Input Values:**

$ps$       - Pressure  $p$  in bar  
 $ts$       - Temperature  $t$  in °C  
 $Xisl$      - Mass fraction of sea salt  $\xi$  in kg sea salt/kg mixture

**Result:**

usl\_pstsXisl\_SeaWa, usl - Specific internal energy of saturated seawater in kJ/kg

**Range of Validity:**

Temperature  $t$ :                      from 0 °C to 220 °C  
 Mixture pressure  $p$ :                from  $p_s(t = 0 \text{ °C}, \xi = 0.2)$  to  $p_s(t = 220 \text{ °C}, \xi = 0)$   
 Mass fraction  $\xi$ :                       $0 \leq \xi \leq 0.2$  kg sea salt/kg mixture

**Possible input variants:**

$$\begin{aligned}
 u^{sl} &= f(-1000, t_s, \xi^{sl}) \\
 u^{sl} &= f(p_s, -1000, \xi^{sl}) \\
 u^{sl} &= f(p_s, t_s, -1000) \\
 u^{sl} &= f(p_s, t_s, \xi^{sl})
 \end{aligned}$$

**Comments:**

- Specific internal energy  $u = h - (p * v)$

**Result for Wrong Input Values:**

usl\_pstsXisl\_SeaWa, usl = -1000

**References:**

$u(p, t, \xi)$                                       [1] , [2]

**Specific Internal Energy of Saturated Vapor  $u_{sv} = f(p_s, t_s, \xi^{sl})$** 
**Function Name:**

usv\_pstsXisl\_SeaWa

**Fortran Programs:**

```
REAL*8 FUNCTION USV_PSTSXISL_SEAWA(PS,TS,XISL), REAL*8 PS,TS,XISL
```

**Input Values:**

$ps$  - Pressure  $p$  in bar  
 $ts$  - Temperature  $t$  in °C  
 $Xisl$  - Mass fraction of sea salt  $\xi$  in kg sea salt/kg mixture

**Result:**

usv\_pstsXisl\_SeaWa, usv - Specific internal energy of saturated seawater in kJ/kg

**Range of Validity:**

Temperature  $t$ : from 0 °C to 220 °C  
 Mixture pressure  $p$ : from  $p_s(t = 0 \text{ °C}, \xi = 0.2)$  to  $p_s(t = 220 \text{ °C}, \xi = 0)$   
 Mass fraction  $\xi$ :  $0 \leq \xi \leq 0.2$  kg sea salt/kg mixture

**Comments:**

- Specific internal energy  $u = h - (p * v)$

**Result for Wrong Input Values:**

usv\_pstsXisl\_SeaWa, usv = -1000

**References:**

$u(p, t, \xi)$  [1] , [2]

**Specific Volume  $v = f(p, t, \xi)$** 
**Function Name:**

v\_ptXi\_SeaWa

**Fortran Programs:**

```
REAL*8 FUNCTION V_PTXI_SEAWA(P, T, XI), REAL*8 P, T, XI
```

**Input Values:**

$p$  - Pressure  $p$  in bar  
 $t$  - Temperature  $t$  in °C  
 $Xi$  - Mass fraction of sea salt  $\xi$  in kg sea salt/kg mixture

**Result:**

v\_ptXi\_SeaWa,  $v$  - Specific volume in m³/kg

**Range of Validity:**

Temperature  $t$ : from 0 °C to 220 °C  
 Pressure  $p$ : from  $p_s(t = 0 \text{ °C}, \xi = 0.2)$  to 1000 bar and  $p \geq p_{\text{mel}}(t, \xi)$   
 Mass fraction  $\xi$ :  $0 \leq \xi \leq 0.2$  kg sea salt/kg mixture

**Result for Wrong Input Values:**

v\_ptXi\_SeaWa,  $v = -1000$

**References:**

$v(p, t, \xi)$  [1], [2]

**Specific Volume of Liquid Seawater  $v_l = f(p, t, \xi)$** 
**Function Name:**

vl\_ptXi\_SeaWa

**Fortran Programs:**

```
REAL*8 FUNCTION VL_PTXI_SEAWA(P, T, XI), REAL*8 P, T, XI
```

**Input Values:**

$p$         - Pressure  $p$  in bar  
 $t$         - Temperature  $t$  in °C  
 $X_i$        - Mass fraction of sea salt  $\xi$  in kg sea salt/kg mixture

**Result:**

vl\_ptXi\_SeaWa, vl - Specific volume of liquid seawater in m<sup>3</sup>/kg

**Range of Validity:**

Temperature  $t$ :                      from 0 °C to 220 °C  
 Pressure  $p$ :                            from  $p_s(t = 0 \text{ °C}, \xi = 0.2)$  to 1000 bar and  $p \geq p_{\text{mel}}(t, \xi)$   
 Mass fraction  $\xi$ :                       $0 \leq \xi \leq 0.2$  kg sea salt/kg mixture

**Result for Wrong Input Values:**

vl\_ptXi\_SeaWa, vl = -1000

**References:**

$v(p, t, \xi)$                                       [1] , [2]

**Specific Volume of Saturated Liquid  $v_{sl} = f(p_s, t_s, \xi^{sl})$** 
**Function Name:**

vsl\_pstsXisl\_SeaWa

**Fortran Programs:**

```
REAL*8 FUNCTION VSL_PSTSXISL_SEAWA(PS, TS, XISL), REAL*8 PS, TS, XISL
```

**Input Values:**

$ps$  - Pressure  $p$  in bar  
 $ts$  - Temperature  $t$  in °C  
 $Xisl$  - Mass fraction of sea salt  $\xi$  in kg sea salt/kg mixture

**Result:**

vsl\_pstsXisl\_SeaWa, vsl - Specific volume of saturated seawater in m³/kg

**Range of Validity:**

Temperature  $t$ : from 0 °C to 220 °C  
 Mixture pressure  $p$ : from  $p_s(t = 0 \text{ °C}, \xi = 0.2)$  to  $p_s(t = 220 \text{ °C}, \xi = 0)$   
 Mass fraction  $\xi$ :  $0 \leq \xi \leq 0.2$  kg sea salt/kg mixture

**Possible input variants:**

$$v^{sl} = f(-1000, t_s, \xi^{sl})$$

$$v^{sl} = f(p_s, -1000, \xi^{sl})$$

$$v^{sl} = f(p_s, t_s, -1000)$$

$$v^{sl} = f(p_s, t_s, \xi^{sl})$$

**Result for Wrong Input Values:**

vsl\_pstsXisl\_SeaWa, vsl = -1000

**References:**

$v(p, t, \xi)$  [1] , [2]

**Specific Volume of Saturated Vapor  $v_{sv} = f(p_s, t_s, \xi^{sl})$** 
**Function Name:**

vsv\_pstsXisl\_SeaWa

**Fortran Programs:**

```
REAL*8 FUNCTION VSV_PSTSXISL_SEAWA(PS,TS,XISL), REAL*8 PS,TS,XISL
```

**Input Values:**

$ps$  - Pressure  $p$  in bar  
 $ts$  - Temperature  $t$  in °C  
 $Xisl$  - Mass fraction of sea salt  $\xi$  in kg sea salt/kg mixture

**Result:**

vsv\_pstsXisl\_SeaWa, vsv - Specific volume of saturated seawater in m³/kg

**Range of Validity:**

Temperature  $t$ : from 0 °C to 220 °C  
 Mixture pressure  $p$ : from  $p_s(t = 0 \text{ °C}, \xi = 0.2)$  to  $p_s(t = 220 \text{ °C}, \xi = 0)$   
 Mass fraction  $\xi$ :  $0 \leq \xi \leq 0.2$  kg sea salt/kg mixture

**Result for Wrong Input Values:**

vsv\_pstsXisl\_SeaWa, vsv = -1000

**References:**

$v(p, t, \xi)$  [1] , [2]

**Speed of Sound  $w = f(p, t, \xi)$** 
**Function Name:**

w\_ptXi\_SeaWa

**Fortran Programs:**

```
REAL*8 FUNCTION W_PTXI_SEAWA(P, T, XI), REAL*8 P, T, XI
```

**Input Values:**

$p$  - Pressure  $p$  in bar  
 $t$  - Temperature  $t$  in °C  
 $X_i$  - Mass fraction of sea salt  $\xi$  in kg sea salt/kg mixture

**Result:**

w\_ptXi\_SeaWa,  $w$  - Speed of Sound in m/s

**Range of Validity:**

Temperature $t$ :	from 0 °C to 80 °C
Pressure $p$ :	from $p_s(t = 0 \text{ °C}, \xi = 0.12)$ to 1000 bar and $p \geq p_{\text{mel}}(t, \xi)$
Mass fraction $\xi$ :	$0 \leq \xi \leq 0.12$ kg sea salt/kg mixture

**Comments:**

- This function is not defined in the wet steam region.
- If the input values are outside the range of validity of [1] the result is -1000 (for high salinity, high pressures and temperatures above 65 °C)

**Result for Wrong Input Values:**

w\_ptXi\_SeaWa,  $w = -1000$

**References:**

$w(p, t, \xi)$  [1]



**Speed of Sound of Liquid  $w_l = f(p, t, \xi)$** 
**Function Name:**

wl\_ptXi\_SeaWa

**Fortran Programs:**

```
REAL*8 FUNCTION WL_PTXI_SEAWA(P, T, XI), REAL*8 P, T, XI
```

**Input Values:**

$p$  - Pressure  $p$  in bar  
 $t$  - Temperature  $t$  in °C  
 $X_i$  - Mass fraction of sea salt  $\xi$  in kg sea salt/kg mixture

**Result:**

wl\_ptXi\_SeaWa, wl - Speed of Sound of liquid in m/s

**Range of Validity:**

Temperature $t$ :	from 0 °C to 80 °C
Pressure $p$ :	from $p_s(t = 0 \text{ °C}, \xi = 0.12)$ to 1000 bar and $p \geq p_{\text{mel}}(t, \xi)$
Mass fraction $\xi$ :	$0 \leq \xi \leq 0.12$ kg sea salt/kg mixture

**Comments:**

- If the input values are outside the range of validity of [1] the result is -1000 (for high salinity, high pressures and temperatures above 65 °C)

**Result for Wrong Input Values:**

wl\_ptXi\_SeaWa, wl = -1000

**References:**

$w(p, t, \xi)$  [1]

**Speed of Sound of Saturated Liquid  $w_{sl} = f(p_s, t_s, \xi^{sl})$** 
**Function Name:**

wsl\_pstsXisl\_SeaWa

**Fortran Programs:**

REAL\*8 FUNCTION WSL\_PSTSXISL\_SEAWA(PS, TS, XISL), REAL\*8 PS, TS, XISL

**Input Values:**

$ps$  - Pressure  $p$  in bar  
 $ts$  - Temperature  $t$  in °C  
 $Xisl$  - Mass fraction of sea salt  $\xi$  in kg sea salt/kg mixture

**Result:**

wsl\_pstsXisl\_SeaWa, wsl - Sound speed of saturated seawater in m/s

**Range of Validity:**

Temperature  $t$ : from 0 °C to 80 °C  
 Mixture pressure  $p$ : from  $p_s(t = 0 \text{ °C}, \xi = 0.12)$  to  $p_s(t = 80 \text{ °C}, \xi = 0)$   
 Mass fraction  $\xi$ :  $0 \leq \xi \leq 0.12$  kg sea salt/kg mixture

**Possible input variants:**

$$w^{sl} = f(-1000, t_s, \xi^{sl})$$

$$w^{sl} = f(p_s, -1000, \xi^{sl})$$

$$w^{sl} = f(p_s, t_s, -1000)$$

$$w^{sl} = f(p_s, t_s, \xi^{sl})$$

**Comments:**

- If the input values are outside the range of validity of [1] the result is -1000 (for high salinity, high pressures and temperatures above 65 °C)

**Result for Wrong Input Values:**

wsl\_pstsXisl\_SeaWa, wsl = -1000

**References:**

$w(p, t, \xi)$  [1]

**Speed of Sound of Saturated Vapor  $w_{sv} = f(p_s, t_s, \xi^{sl})$** 
**Function Name:**

wsv\_pstsXisl\_SeaWa

**Fortran Programs:**

```
REAL*8 FUNCTION WSV_PSTSXISL_SEAWA(PS,TS,XISL), REAL*8 PS,TS,XISL
```

**Input Values:**

$ps$  - Pressure  $p$  in bar  
 $ts$  - Temperature  $t$  in °C  
 $Xisl$  - Mass fraction of sea salt  $\xi$  in kg sea salt/kg mixture

**Result:**

wsv\_pstsXisl\_SeaWa, wsv - Specific volume of saturated seawater in m<sup>3</sup>/kg

**Range of Validity:**

Temperature  $t$ : from 0 °C to 80 °C  
 Mixture pressure  $p$ : from  $p_s(t = 0 \text{ °C}, \xi = 0.12)$  to  $p_s(t = 80 \text{ °C}, \xi = 0)$   
 Mass fraction  $\xi$ :  $0 \leq \xi \leq 0.12$  kg sea salt/kg mixture

**Comments:**

- If the input values are outside the range of validity of [1] the result is –1000 (for high salinity, high pressures and temperatures above 65 °C)

**Result for Wrong Input Values:**

wsv\_pstsXisl\_SeaWa, wsv = –1000

**References:**

$w(p, t, \xi)$  [1]

**Vapor Fraction  $x = f(p, t, \xi)$** 
**Function Name:**

x\_ptXi\_SeaWa

**Fortran Programs:**

```
REAL*8 FUNCTION X_PTXI_SEAWA(P, T, XI), REAL*8 P, T, XI
```

**Input Values:**

$p$  - Pressure  $p$  in bar  
 $t$  - Temperature  $t$  in °C  
 $Xi$  - Mass fraction of sea salt  $\xi$  in kg sea salt/kg mixture

**Result:**

x\_ptXi\_SeaWa,  $x$  - Vapor Fraction in kg/kg

**Range of Validity:**

Temperature  $t$ : from 0 °C to 220 °C  
 Pressure  $p$ : from  $p_s(t = 0 \text{ °C}, \xi = 0.2)$  to 1000 bar and  $p \geq p_{\text{mel}}(t, \xi)$   
 Mass fraction  $\xi$ :  $0 \leq \xi \leq 0.2$  kg sea salt/kg mixture

**Comments:**

- This function is defined in the wet steam region.

**Result for Wrong Input Values:**

x\_ptXi\_SeaWa,  $x = -1000$

**References:**

$x(p, t, \xi)$  [1]

**Mass Fraction of Sea Salt of Saturated Liquid  $\xi_{sl} = f(p_s, t_s)$** 
**Function Name:**

Xisl\_psts\_SeaWa

**Fortran Programs:**

```
REAL*8 FUNCTION XISL_PSTS_SEAWA(PS, TS), REAL*8 PS, TS
```

**Input Values:**

$ps$      - Pressure  $p$  in bar  
 $ts$      - Temperature  $t$  in °C

**Result:**

Xisl\_psts\_SeaWa, Xisl - Mass fraction of sea salt of saturated liquid  
in kg sea salt/kg mixture

**Range of Validity:**

Temperature  $t$ :                      from 0 °C to 220 °C  
Mixture pressure  $p$ :                from  $p_s(t = 0 \text{ °C}, \xi = 0.2)$  to  $p_s(t = 220 \text{ °C}, \xi = 0)$

**Result for Wrong Input Values:**

Xisl\_psts\_SeaWa, Xisl = -1000

**References:**

$\xi_{sl}(p_s, t_s)$                                       [1]

**Mass Fraction of Sea Salt of Saturated Vapor  $\xi_{sv} = f(p_s, t_s)$** 
**Function Name:**

Xisv\_psts\_SeaWa

**Fortran Programs:**

```
REAL*8 FUNCTION XISV_PSTS_SEAWA(PS, TS), REAL*8 PS, TS
```

**Input Values:**

$ps$  - Pressure  $p$  in bar  
 $ts$  - Temperature  $t$  in °C

**Result:**

Xisv\_psts\_SeaWa, Xisv - Mass fraction of sea salt of saturated steam  
in kg sea salt/kg mixture

**Range of Validity:**

Temperature  $t$ : from 0 °C to 220 °C  
Mixture pressure  $p$ : from  $p_s(t = 0 \text{ °C}, \xi = 0.2)$  to  $p_s(t = 220 \text{ °C}, \xi = 0)$

**Result for Wrong Input Values:**

Xisv\_psts\_SeaWa, Xisv = -1000

**References:**

$\xi_{sv}(p_s, t_s)$  [1]

## 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
  - IAPWS-IF97-S03rev
  - IAPWS-IF97-S04
  - IAPWS-IF97-S05
- IAPWS Revised Advisory Note No. 3 on Thermodynamic Derivatives (2008)

#### Library LibSBTL\_IF97 Library LibSBTL\_95

Extremely fast property calculations according to the IAPWS Guideline 2015 Spline-based Table Look-up Method (SBTL) applied to the Industrial Formulation IAPWS-IF97 and to the Scientific Formulation IAPWS-95 for Computational Fluid Dynamics and simulating non-stationary processes

### Humid Combustion Gas Mixtures

#### Library LibHuGas

Model: Ideal mixture of the real fluids:  
 $\text{CO}_2$  - Span, Wagner     $\text{H}_2\text{O}$  - IAPWS-95  
 $\text{O}_2$  - Schmidt, Wagner     $\text{N}_2$  - Span et al.  
 Ar - Tegeler et al.  
 and of the ideal gases:  
 $\text{SO}_2$ , CO, Ne  
 (Scientific Formulation of Bucker 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

### 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	$\text{H}_2\text{O}$	$\text{F}_2$	Propane
$\text{N}_2$	$\text{SO}_2$	$\text{NH}_3$	Iso-Butane
$\text{O}_2$	$\text{H}_2$	Methane	n-Butane
CO	$\text{H}_2\text{S}$	Ethane	Benzene
$\text{CO}_2$	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](http://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 Bucker and Wagner (2006)

#### n-Butane

#### Library LibButane\_n

Formulation of Bucker 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

$\text{C}_2\text{H}_6\text{O}_2$	Ethylene glycol
$\text{C}_3\text{H}_8\text{O}_2$	Propylene glycol
$\text{C}_2\text{H}_5\text{OH}$	Ethanol
$\text{CH}_3\text{OH}$	Methanol
$\text{C}_3\text{H}_8\text{O}_3$	Glycerol
$\text{K}_2\text{CO}_3$	Potassium carbonate
$\text{CaCl}_2$	Calcium chloride
$\text{MgCl}_2$	Magnesium chloride
$\text{NaCl}$	Sodium chloride
$\text{C}_2\text{H}_3\text{KO}_2$	Potassium acetate
$\text{CHKO}_2$	Potassium formate
$\text{LiCl}$	Lithium chloride
$\text{NH}_3$	Ammonia

Formulation of the International Institute of Refrigeration (IIR 2010)

**Ethanol****Library LibC2H5OH**

Formulation of  
Schroeder (2012)

**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 **H<sub>2</sub>S** **Library LibH2S**

Nitrous oxide **N<sub>2</sub>O** **Library LibN2O**

Sulfur dioxide **SO<sub>2</sub>** **Library LibSO2**

Acetone  $C_3H_6O$  **Library LibC3H6O**

Formulation of Lemmon and Span (2006)

**For more information please contact:**

KCE-ThermoFluidProperties UG (limited liability) & Co. KG  
Professor Hans-Joachim Kretschmar

Wallotstr. 3  
01307 Dresden, Germany

Internet: [www.thermofluidprop.com](http://www.thermofluidprop.com)

E-mail: [info@thermofluidprop.com](mailto:info@thermofluidprop.com)

Phone: +49-351-27597860

Mobile: +49-172-7914607

Fax: +49-3222-4262250

**The following thermodynamic and transport properties can be calculated<sup>a</sup>:****Thermodynamic Properties**

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

**Transport Properties**

- Dynamic viscosity  $\eta$
- Kinematic viscosity  $\nu$
- Thermal conductivity  $\lambda$
- Prandtl number  $Pr$

**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**

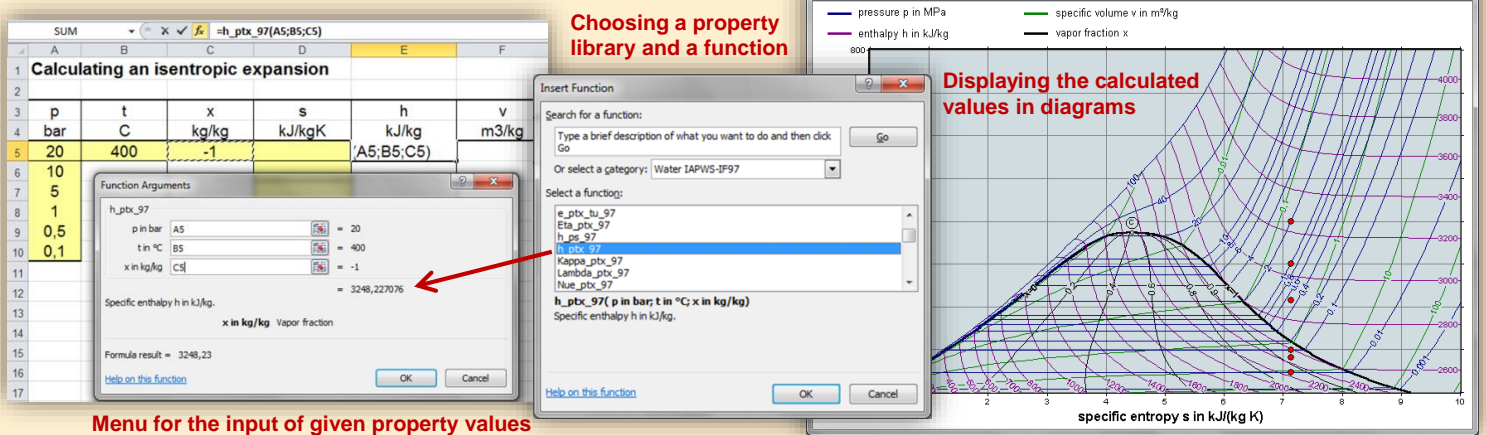
- Partial derivatives can be calculated.

<sup>a</sup> 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<sup>Graphics</sup> for Excel<sup>®</sup>



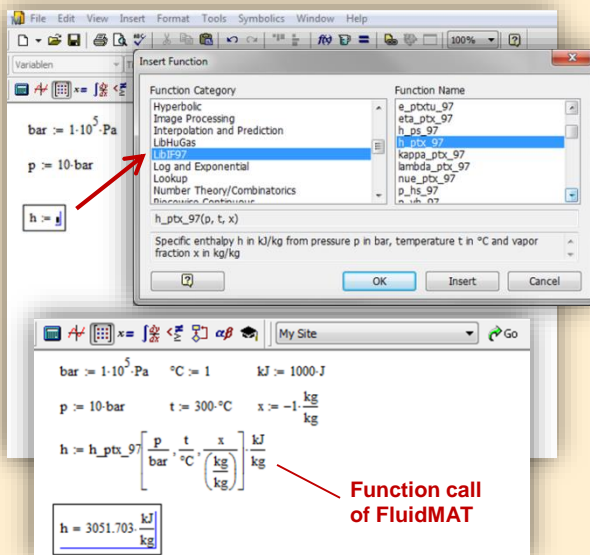
**Choosing a property library and a function**

**Menu for the input of given property values**

**Displaying the calculated values in diagrams**

## Add-In FluidMAT for Mathcad<sup>®</sup>

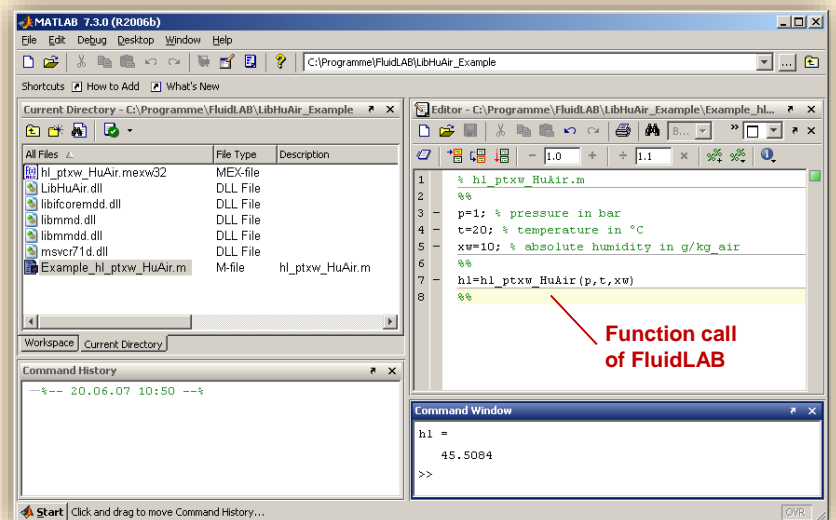
The property libraries can be used in Mathcad<sup>®</sup>.



**Function call of FluidMAT**

## Add-In FluidLAB for MATLAB<sup>®</sup>

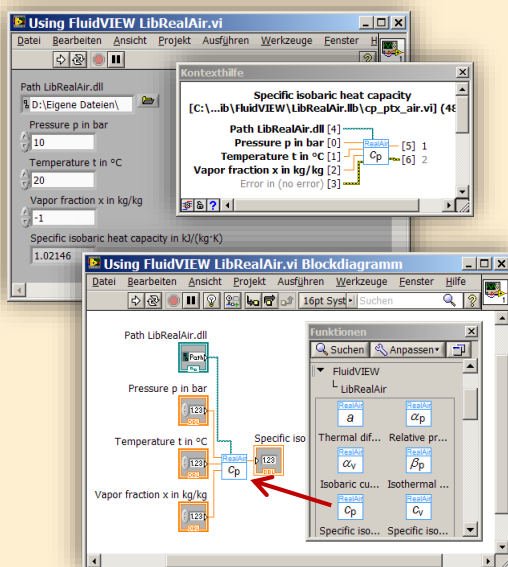
Using the Add-In FluidLAB the property functions can be called in MATLAB<sup>®</sup>.



**Function call of FluidLAB**

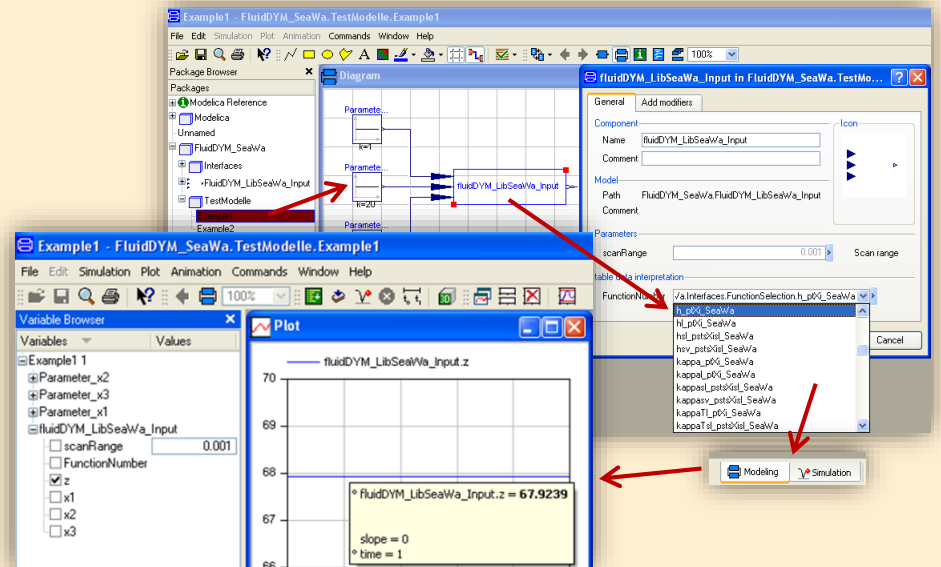
## Add-On FluidVIEW for LabVIEW<sup>™</sup>

The property functions can be calculated in LabVIEW<sup>™</sup>.

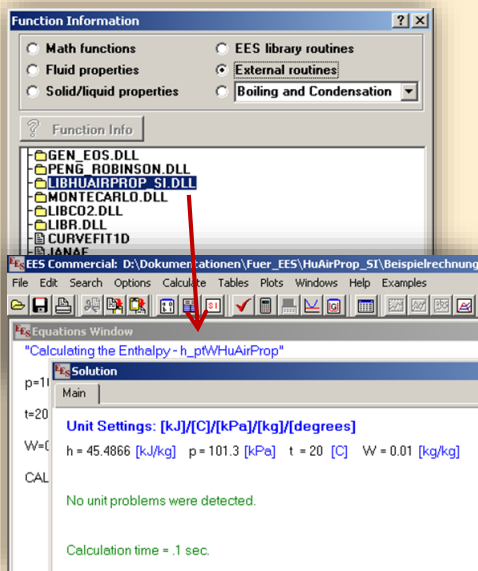


## Add-In FluidDYM for DYMOLA<sup>®</sup> (Modelica) and SimulationX<sup>®</sup>

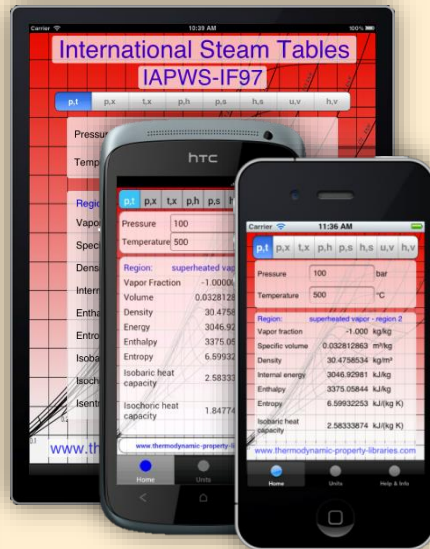
The property functions can be called in DYMOLA<sup>®</sup> and SimulationX<sup>®</sup>.



## Add-In 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](http://www.thermofluidprop.com)

**Zittau's Fluid Property Calculator**

Fluid:

Function:

Unit System:

Enter given values: [Range of validity](#)

Pressure p:  bar

Temperature t:  °C

Vapor fraction x:  kg/kg

**Calculate / Recalculate**

**Result:**

Specific enthalpy h = 3097.38 kJ/kg

For further information on property libraries available for EXCEL®, MATLAB®, Mathcad®, Engineering Equation Solver®, DYMOLA® (Modelica), SimulationX®, and LabView® click [here](#).

An App for calculating steam properties on iPhone, iPad, and iPod touch can be found [here](#) PDF with the [description](#).

© Zittau/Göeritz University of Applied Sciences  
Faculty of Mechanical Engineering  
Department of Technical Thermodynamics  
Prof. Hans-Joachim Kretzschmar  
Dr. Ines Stöcker  
Programmer: Joachim Posselt

Tel.: +49-3583-61-1946 or -1981  
Fax: +49-3583-61-1946  
E-mail: [info@thermofluidprop.com](mailto:info@thermofluidprop.com)  
[www.thermofluidprop.com](http://www.thermofluidprop.com)  
[www.thermofluidprop.com](http://www.thermofluidprop.com)  
[www.thermofluidprop.com](http://www.thermofluidprop.com)

## Property Software for Pocket Calculators

### FluidCasio



fx 9750 G II    CFX 9850 fx-GG20    CFX 9860 G Graph 85    ALGEBRA FX 2.0

### FluidHP



HP 48    HP 49

### FluidTI



TI Nspire CX CAS    TI 83  
TI Nspire CAS    TI 84  
TI 89

TI Voyage 200

TI 92

## For more information please contact:

KCE-ThermoFluidProperties UG (limited liability) & Co. KG  
Professor Hans-Joachim Kretzschmar

Wallotstr. 3  
01307 Dresden, Germany

Internet: [www.thermofluidprop.com](http://www.thermofluidprop.com)  
E-mail: [info@thermofluidprop.com](mailto:info@thermofluidprop.com)  
Phone: +49-351-27597860  
Mobile: +49-172-7914607  
Fax: +49-3222-4262250

The following thermodynamic and transport properties<sup>a</sup> 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  $\rho$
- Specific volume  $v$
- Enthalpy  $h$
- Internal energy  $u$
- Entropy  $s$
- Exergy  $e$
- Isobaric heat capacity  $c_p$
- Isochoric heat capacity  $c_v$
- Isentropic exponent  $\kappa$
- Speed of sound  $w$
- Surface tension  $\sigma$

### Transport Properties

- Dynamic viscosity  $\eta$
- Kinematic viscosity  $\nu$
- Thermal conductivity  $\lambda$
- Prandtl number  $Pr$

### 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 can be calculated.

<sup>a</sup> Not all of these property functions are available in all property libraries.

## 5. References

- [1] IAPWS Advisory Note No. 5: Industrial Calculation of the Thermodynamic Properties of Seawater. Available from [www.iapws.org](http://www.iapws.org) (2013)
- [2] Hömig, H.E:  
Seawater an Seawater Distillation. Fichtner-Handbook,  
Vulkan-Verlag, Essen (1978)
- [3] Jamieson, D.T; Tudhope, J. S.:  
Physical Properties of Sea Water Solutions: Thermal Conductivity  
Desalination 8, 393-401 (1970)
- [4] Feistel, R.:  
A Gibbs Function for Seawater Thermodynamics for -6°C to 80°C and Salinity  
up to 120g/kg. Deep Sea Research (2008)
- [5] IAPWS. Release on the IAPWS Formulation 1995 for the Thermodynamic  
Properties of Ordinary Water Substances for General and Scientific Use.  
Available from [www.iapws.org](http://www.iapws.org) (1995)
- [6] IAPWS. Release on an Equation of State for H<sub>2</sub>O Ice Ih.  
Available from [www.iapws.org](http://www.iapws.org) (2006)
- [7] IAPWS. Release on Revised Advisory Note No.3, Thermodynamic Derivatives  
from IAPWS Formulations.  
Available from [www.iapws.org](http://www.iapws.org) (2008)
- [8] IAPWS. Revised Release on the IAPWS Industrial Formulation 1997 for the  
Thermodynamic Properties of Water and Steam IAPWS-IF97.  
Available from [www.iapws.org](http://www.iapws.org) (2007)
- [9] Wagner, W.; Kretzschmar, H.-J.:  
International Steam Tables.Springer-Verlag (2008)
- [10] Hellriegel, T.:  
ITERA\_PRO Iterationsroutine.  
Fachgebiet Technische Thermodynamik, Hochschule Zittau/Görlitz (2008)
- [11] Miyagawa, K.:  
IAPWS-IF97 for the Water Part of the EOS of SeaWater. (2008)
- [12] Kretzschmar, H.-J.; Stöcker, I.; Jähne, I.: Programm FluidEXL Graphics für  
Excel® Hochschule Zittau/Görlitz (FH), Fachbereich Maschinenwesen (1998 -  
2008)
- [13] Salomo, B.:  
Berechnung der thermodynamischen Stoffeigenschaften von  
Kohlendioxid/Wasser-Gemischen für die Modellierung von Energie-  
umwandlungsprozessen mit CO<sub>2</sub>-Abscheidung Anbindung an MATLAB.  
Diplomarbeit, Zittau (2007)
- [14] IAPWS. Revised Release on the Pressure along the Melting and Sublimation  
Curves of Ordinary Water Substance. Available from [www.iapws.org](http://www.iapws.org) (2008)

## 6. Satisfied Customers

Date: 05/2018

The following companies and institutions use the property libraries

- FluidEXL *Graphics* for Excel®
- FluidLAB for MATLAB®
- FluidMAT for Mathcad®
- FluidEES for Engineering Equation Solver® EES
- FluidDYM for Dymola® (Modelica) and SimulationX®
- FluidVIEW for LabVIEW™.

### 2018

Universität Madrid, Madrid, Spanien	05/2018
HS Zittau/ Görlitz, Fakultät Wirtschaft, Zittau	05/2018
HS Niederrhein, Krefeld	05/2018
GRS, Köln	03/2018
RONAL AG, Härklingen, Schweiz	02/2018
Ingenieurbüro Leipert, Riegelsberg	02/2018
AIXPROCESS, Aachen	02/2018
KRONES, Neutraubling	02/2018
Doosan Lentjes, Ratingen	01/2018

### 2017

Compact Kältetechnik, Dresden	12/2017
Endress + Hauser Messtechnik GmbH +Co. KG, Hannover	12/2017
TH Mittelhessen, Gießen	11/2017
Haarslev Industries, Sønderød, Denmark	11/2017
Hochschule Zittau/Görlitz, Fachgebiet Energiesystemtechnik	11/2017
ATESTEO, Alsdorf	10/2017
Wijbenga, PC Geldermalsen, Netherlands	10/2017
Fels-Werke GmbH, Elbingerode	10/2017
KIT Karlsruhe, Institute für Neutronenphysik und Reaktortechnik	09/2017
Air-Consult, Jena	09/2017
Papierfabrik Koehler, Oberkirch	09/2017
ZWILAG, Würenlingen, Switzerland	09/2017
TLK-Thermo Universität Braunschweig, Braunschweig	08/2017
Fichtner IT Consulting AG, Stuttgart	07/2017
Hochschule Ansbach, Ansbach	06/2017
RONAL, Härkingen, Switzerland	06/2017
BORSIG Service, Berlin	06/2017

BOGE Kompressoren, Bielefeld	06/2017
STEAG Energy Services, Zwingenberg	06/2017
CES clean energy solutions, Wien, Austria	04/2017
Princeton University, Princeton, USA	04/2017
B2P Bio-to-Power, Wadersloh	04/2017
TU Dresden, Institute for Energy Engineering, Dresden	04/2017
SAINT-GOBAIN, Vaujours, France	03/2017
TU Bergakademie Freiberg, Chair of Thermodynamics, Freiberg	03/2017
SCHMIDT + PARTNER, Therwil, Switzerland	03/2017
KAESER Kompressoren, Gera	03/2017
F&R, Praha, Czech Republic	03/2017
ULT Umwelt-Lufttechnik, Löbau	02/2017
JS Energie & Beratung, Erding	02/2017
Kelvion Braze PHE, Nobitz-Wilchwitz	02/2017
MTU Aero Engines, München	02/2017
Hochschule Zittau/Görlitz, IPM	01/2017
CombTec ProCE, Zittau	01/2017
SHELL Deutschland Oil, Wesseling	01/2017
MARTEC Education Center, Frederikshaven, Denmark	01/2017
SynErgy Thermal Management, Krefeld	01/2017

## 2016

BOGE Druckluftsysteme, Bielefeld	12/2016
BFT Planung, Aachen	11/2016
Midiplan, Bietigheim-Bissingen	11/2016
BBE Barnich IB	11/2016
Wenisch IB,	11/2016
INL, Idaho Falls	11/2016
TU Kältetechnik, Dresden	11/2016
Kopf SynGas, Sulz	11/2016
INTVEN, Bellevue (USA)	11/2016
DREWAG Dresden, Dresden	10/2016
AGO AG Energie+Anlagen, Kulmbach	10/2016
Universität Stuttgart, ITW, Stuttgart	09/2016
Pöry Deutschland GmbH, Dresden	09/2016
Siemens AG, Erlangen	09/2016
BASF über Fichtner IT Consulting AG	09/2016
B+B Engineering GmbH, Magdeburg	09/2016
Wilhelm Büchner Hochschule, Pfungstadt	08/2016



Webasto Thermo & Comfort SE, Gliching	08/2016
TU Dresden, Dresden	08/2016
Endress+Hauser Messtechnik GmbH+Co. KG, Hannover	08/2016
D + B Kältetechnik, Althausen	07/2016
Fichtner IT Consulting AG, Stuttgart	07/2016
AB Electrolux, Krakow, Poland	07/2016
ENEXIO Germany GmbH, Herne	07/2016
VPC GmbH, Vetschau/Spreewald	07/2016
INWAT, Lodz, Poland	07/2016
E.ON SE, Düsseldorf	07/2016
Planungsbüro Waidhas GmbH, Chemnitz	07/2016
EEB Enerko, Aldershoven	07/2016
IHEBA Naturenergie GmbH & Co. KG, Pfaffenhofen	07/2016
SSP Kälteplaner AG, Wolfertschwenden	07/2016
EEB ENERKO Energiewirtschaftliche Beratung GmbH, Berlin	07/2016
BOGE Kompressoren Otto BOGE GmbH & Co KG, Bielefeld	06/2016
Universidad Carlos III de Madrid, Madrid, Spain	04/2016
INWAT, Lodzi, Poland	04/2016
Planungsbüro Waidhas GmbH, Chemnitz	04/2016
STEAG Energy Services GmbH, Laszlo Küppers, Zwingenberg	03/2016
WULFF & UMAG Energy Solutions GmbH, Husum	03/2016
FH Bielefeld, Bielefeld	03/2016
EWT Eckert Wassertechnik GmbH, Celle	03/2016
ILK Institut für Luft- und Kältetechnik GmbH, Dresden	02/2016, 06/2016 (2x)
IEV KEMA - DNV GV – Energie, Dresden	02/2016
Allborg University, Department of Energie, Aalborg, Denmark	02/2016
G.A.M. Heat GmbH, Gräfenhainichen	02/2016
Institut für Luft- und Kältetechnik, Dresden	02/2016, 05/2016, 06/2016
Bosch, Stuttgart	02/2016
INL Idaho National Laboratory, Idaho, USA	11/2016, 01/2016
Friedl ID, Wien, Austria	01/2016
Technical University of Dresden, Dresden	01/2016

## 2015

EES Enerko, Aachen	12/2015
Ruldolf IB, Strau, Austria	12/2015
Allborg University, Department of Energie, Aalborg, Denmark	12/2015
University of Lyubljana, Slovenia	12/2015
Steinbrecht IB, Berlin	11/2015
Universidad Carlos III de Madrid, Madrid, Spain	11/2015
STEAK, Essen	11/2015

Bosch, Lohmar	10/2015
Team Turbo Machines, Rouen, France	09/2015
BTC – Business Technology Consulting AG, Oldenburg	07/2015
KIT Karlsruhe Institute of Technology, Eggenstein-Leopoldshafen	07/2015
ILK, Dresden	07/2015
Schniewindt GmbH & Co. KG, Neuenwalde	08/2015

## 2014

PROJEKTPLAN, Dohna	04/2014
Technical University of Vienna, Austria	04/2014
MTU Aero Engines AG, Munich	04/2014
GKS, Schweinfurt	03/2014
Technical University of Nuremberg	03/2014
EP-E, Niederstetten	03/2014
Rückert NatUrgas GmbH, Lauf	03/2014
YESS-World, South Korea	03/2014
ZAB, Dessau	02/2014
KIT-TVT, Karlsruhe	02/2014
Stadtwerke Neuburg	02/2014
COMPAREX, Leipzig for RWE Essen	02/2014
Technical University of Prague, Czech Republic	02/2014
HS Augsburg	02/2014
Envi-con, Nuremberg	01/2014
DLR, Stuttgart	01/2014
Doosan Lentjes, Ratingen	01/2014
Technical University of Berlin	01/2014
Technical University of Munich	01/2014
Technical University of Braunschweig	01/2014
M&M Turbinentechnik, Bielefeld	01/2014

## 2013

TRANTER-GmbH, Artern	12/2013
SATAKE, Shanghai, China	12/2013
VOITH, Kunshan, China	12/2013
ULT, Löbau	12/2013
MAN, Copenhagen, Dänemark	11/2013
DREWAG, Dresden	11/2013
Haarslev Industries, Herlev, Dänemark	11/2013
STEAG, Herne	11/2013, 12/2013
Ingersoll-Rand, Oberhausen	11/2013
Wilhelm-Büchner HS, Darmstadt	10/2013

IAV, Chemnitz	10/2013
Technical University of Regensburg	10/2013
PD-Energy, Bitterfeld	09/2013
Thermofin, Heinsdorfergrund	09/2013
SHI, New Jersey, USA	09/2013
M&M Turbinentechnik, Bielefeld	08/2013
BEG-BHV, Bremerhaven	08/2013
TIG-Group, Husum	08/2013
COMPAREX, Leipzig	08/2013, 11/2013
for RWE Essen	12/2013
University of Budapest, Hungary	08/2013
Siemens, Frankenthal	08/2013, 10/2013
	11/2013
VGB, Essen	07/2013, 11/2013
Brunner Energieberatung, Zurich, Switzerland	07/2013
Technical University of Deggendorf	07/2013
University of Maryland, USA	07/2013, 08/2013
University of Princeton, USA	07/2013
NIST, Boulder, USA	06/2013
IGUS GmbH, Dresden	06/2013
BHR Bilfinger, Essen	06/2013
SÜDSALZ, Bad Friedrichshall	06/2013, 12/2013
Technician School of Berlin	05/2013
KIER, Gajeong-ro, Südkorea	05/2013
Schwing/Stetter GmbH, Memmingen	05/2013
Vattenfall, Berlin	05/2013
AUTARK, Kleinmachnow	05/2013
STEAG, Zwingenberg	05/2013
Hochtief, Düsseldorf	05/2013
University of Stuttgart	04/2013
Technical University -Bundeswehr, Munich	04/2013
Rerum Cognitio Forschungszentrum, Frankfurt	04/2013
Kältetechnik Dresden + Bremen, Alfhausen	04/2013
University Auckland, New Zealand	04/2013
MASDAR Institut, Abu Dhabi, United Arab Emirates	03/2013
Simpelkamp, Dresden	02/2013
VEO, Eisenhüttenstadt	02/2013
ENTEC, Auerbach	02/2013
Caterpillar, Kiel	02/2013
Technical University of Wismar	02/2013
Technical University of Dusseldorf	02/2013



ILK, Dresden	01/2013, 08/2013
Fichtner IT, Stuttgart	01/2013, 11/2013
Schnepf Ingeniuerbüro, Nagold	01/2013
Schütz Engineering, Wadgassen	01/2013
Endress & Hauser, Reinach, Switzerland	01/2013
Oschatz GmbH, Essen	01/2013
frischli Milchwerke, Rehburg-Loccum	01/2013

## 2012

Voith, Bayreuth	12/2012
Technical University of Munich	12/2012
Dillinger Huette	12/2012
University of Stuttgart	11/2012
Siemens, Muehlheim	11/2012
Sennheiser, Hannover	11/2012
Oschatz GmbH, Essen	10/2012
Fichtner IT, Stuttgart	10/2012, 11/2012
Helbling Technik AG, Zurich, Switzerland	10/2012
University of Duisburg	10/2012
Rerum Cognitio Forschungszentrum, Frankfurt	09/2012
Pöry Deutschland GmbH, Dresden	08/2012
Extracciones, Guatemala	08/2012
RWE, Essen	08/2012
Weghaus Consulting Engineers, Wuerzburg	08/2012
GKS, Schweinfurt	07/2012
COMPAREX, Leipzig for RWE Essen	07/2012
GEA, Nobitz	07/2012
Meyer Werft, Papenburg	07/2012
STEAG, Herne	07/2012
GRS, Cologne	06/2012
Fichtner IT Consult, Chennai, India	06/2012
Siemens, Freiburg	06/2012
Nikon Research of America, Belmont, USA	06/2012
Niederrhein University of Applied Sciences, Krefeld	06/2012
STEAG, Zwingenberg	06/2012
Mainova, Frankfurt on Main via Fichtner IT Consult	05/2012
Endress & Hauser	05/2012
PEU, Espenheim	05/2012
Luzern University of Applied Sciences, Switzerland	05/2012

BASF, Ludwigshafen (general license) via Fichtner IT Consult	05/2012
SPX Balcke-Dürr, Ratingen	05/2012, 07/2012
Gruber-Schmidt, Wien, Austria	04/2012
Vattenfall, Berlin	04/2012
ALSTOM, Baden	04/2012
SKW, Piesteritz	04/2012
TERA Ingegneria, Trento, Italy	04/2012
Siemens, Erlangen	04/2012, 05/2012
LAWI Power, Dresden	04/2012
Stadtwerke Leipzig	04/2012
SEITZ, Wetzikon, Switzerland	03/2012, 07/2012
M & M, Bielefeld	03/2012
Sennheiser, Wedemark	03/2012
SPG, Montreuil Cedex, France	02/2012
German Destillation, Sprendlingen	02/2012
Lopez, Munguia, Spain	02/2012
Endress & Hauser, Hannover	02/2012
Palo Alto Research Center, USA	02/2012
WIPAK, Walsrode	02/2012
Freudenberg, Weinheim	01/2012
Fichtner, Stuttgart	01/2012
airinotec, Bayreuth	01/2012, 07/2012
University Auckland, New Zealand	01/2012
VPC, Vetschau	01/2012
Franken Guss, Kitzingen	01/2012

## 2011

XRG-Simulation, Hamburg	12/2011
Smurfit Kappa PPT, AX Roermond, Netherlands	12/2011
AWTEC, Zurich, Switzerland	12/2011
eins-energie, Bad Elster	12/2011
BeNow, Rodenbach	11/2011
Luzern University of Applied Sciences, Switzerland	11/2011
GMVA, Oberhausen	11/2011
CCI, Karlsruhe	10/2011
W.-Büchner University of Applied Sciences, Pfungstadt	10/2011
PLANAIR, La Sagne, Switzerland	10/2011
LAWI, Dresden	10/2011
Lopez, Munguia, Spain	10/2011
University of KwaZulu-Natal, Westville, South Africa	10/2011

Voith, Heidenheim	09/2011
SpgBe Montreal, Canada	09/2011
SPG TECH, Montreuil Cedex, France	09/2011
Voith, Heidenheim-Mergelstetten	09/2011
MTU Aero Engines, Munich	08/2011
MIBRAG, Zeitz	08/2011
RWE, Essen	07/2011
Fels, Elingerode	07/2011
Weihenstephan University of Applied Sciences	07/2011, 09/2011 10/2011
Forschungszentrum Juelich	07/2011
RWTH Aachen University	07/2011, 08/2011
INNEO Solutions, Ellwangen	06/2011
Caliqua, Basel, Switzerland	06/2011
Technical University of Freiberg	06/2011
Fichtner IT Consulting, Stuttgart	05/2011, 06/2011, 08/2011
Salzgitter Flachstahl, Salzgitter	05/2011
Helbling Beratung & Bauplanung, Zurich, Switzerland	05/2011
INEOS, Cologne	04/2011
Enseleit Consulting Engineers, Siebigerode	04/2011
Witt Consulting Engineers, Stade	03/2011
Helbling, Zurich, Switzerland	03/2011
MAN Diesel, Copenhagen, Denmark	03/2011
AGO, Kulmbach	03/2011
University of Duisburg	03/2011, 06/2011
CCP, Marburg	03/2011
BASF, Ludwigshafen	02/2011
ALSTOM Power, Baden, Switzerland	02/2011
Universität der Bundeswehr, Munich	02/2011
Calorifer, Elgg, Switzerland	01/2011
STRABAG, Vienna, Austria	01/2011
TUEV Sued, Munich	01/2011
ILK Dresden	01/2011
Technical University of Dresden	01/2011, 05/2011 06/2011, 08/2011

## 2010

Umweltinstitut Neumarkt	12/2010
YIT Austria, Vienna, Austria	12/2010
MCI Innsbruck, Austria	12/2010

University of Stuttgart	12/2010
HS Cooler, Wittenburg	12/2010
Visteon, Novi Jicin, Czech Republic	12/2010
CompuWave, Brunntal	12/2010
Stadtwerke Leipzig	12/2010
MCI Innsbruck, Austria	12/2010
EVONIK Energy Services, Zwingenberg	12/2010
Caliqua, Basel, Switzerland	11/2010
Shanghai New Energy Resources Science & Technology, China	11/2010
Energieversorgung Halle	11/2010
Hochschule für Technik Stuttgart, University of Applied Sciences	11/2010
Steinmueller, Berlin	11/2010
Amberg-Weiden University of Applied Sciences	11/2010
AREVA NP, Erlangen	10/2010
MAN Diesel, Augsburg	10/2010
KRONES, Neutraubling	10/2010
Vaillant, Remscheid	10/2010
PC Ware, Leipzig	10/2010
Schubert Consulting Engineers, Weißenberg	10/2010
Fraunhofer Institut UMSICHT, Oberhausen	10/2010
Behringer Consulting Engineers, Tagmersheim	09/2010
Saacke, Bremen	09/2010
WEBASTO, Neubrandenburg	09/2010
Concordia University, Montreal, Canada	09/2010
Compañía Eléctrica de Sochagota, Bogota, Colombia	08/2010
Hannover University of Applied Sciences	08/2010
ERGION, Mannheim	07/2010
Fichtner IT Consulting, Stuttgart	07/2010
TF Design, Matieland, South Africa	07/2010
MCE, Berlin	07/2010, 12/2010
IPM, Zittau/Goerlitz University of Applied Sciences	06/2010
TUEV Sued, Dresden	06/2010
RWE IT, Essen	06/2010
Glen Dimplex, Kulmbach	05/2010, 07/2010
	10/2010
Hot Rock, Karlsruhe	05/2010
Darmstadt University of Applied Sciences	05/2010
Voith, Heidenheim	04/2010
CombTec, Zittau	04/2010
University of Glasgow, Great Britain	04/2010
Universitaet der Bundeswehr, Munich	04/2010

Technical University of Hamburg-Harburg	04/2010
Vattenfall Europe, Berlin	04/2010
HUBER Consulting Engineers, Berching	04/2010
VER, Dresden	04/2010
CCP, Marburg	03/2010
Offenburg University of Applied Sciences	03/2010
Technical University of Berlin	03/2010
NIST Boulder CO, USA	03/2010
Technical University of Dresden	02/2010
Siemens Energy, Nuremberg	02/2010
Augsburg University of Applied Sciences	02/2010
ALSTOM Power, Baden, Switzerland	02/2010, 05/2010
MIT Massachusetts Institute of Technology Cambridge MA, USA	02/2010
Wieland Werke, Ulm	01/2010
Siemens Energy, Goerlitz	01/2010, 12/2010
Technical University of Freiberg	01/2010
ILK, Dresden	01/2010, 12/2010
Fischer-Uhrig Consulting Engineers, Berlin	01/2010

## 2009

ALSTOM Power, Baden, Schweiz	01/2009, 03/2009 05/2009
Nordostschweizerische Kraftwerke AG, Doettingen, Switzerland	02/2009
RWE, Neurath	02/2009
Brandenburg University of Technology, Cottbus	02/2009
Hamburg University of Applied Sciences	02/2009
Kehrein, Moers	03/2009
EPP Software, Marburg	03/2009
Bernd Münstermann, Telgte	03/2009
Suedzucker, Zeitz	03/2009
CPP, Marburg	03/2009
Gelsenkirchen University of Applied Sciences	04/2009
Regensburg University of Applied Sciences	05/2009
Gatley & Associates, Atlanta, USA	05/2009
BOSCH, Stuttgart	06/2009, 07/2009
Dr. Nickolay, Consulting Engineers, Gommersheim	06/2009
Ferrostal Power, Saarlouis	06/2009
BHR Bilfinger, Essen	06/2009
Intraserv, Wiesbaden	06/2009
Lausitz University of Applied Sciences, Senftenberg	06/2009
Nuernberg University of Applied Sciences	06/2009

Technical University of Berlin	06/2009
Fraunhofer Institut UMSICHT, Oberhausen	07/2009
Bischoff, Aurich	07/2009
Fichtner IT Consulting, Stuttgart	07/2009
Techsoft, Linz, Austria	08/2009
DLR, Stuttgart	08/2009
Wienstrom, Vienna, Austria	08/2009
RWTH Aachen University	09/2009
Vattenfall, Hamburg	10/2009
AIC, Chemnitz	10/2009
Midiplan, Bietigheim-Bissingen	11/2009
Institute of Air Handling and Refrigeration ILK, Dresden	11/2009
FZD, Rossendorf	11/2009
Techgroup, Ratingen	11/2009
Robert Sack, Heidelberg	11/2009
EC, Heidelberg	11/2009
MCI, Innsbruck, Austria	12/2009
Saacke, Bremen	12/2009
ENERKO, Aldenhoven	12/2009

## 2008

Pink, Langenwang	01/2008
Fischer-Uhrig, Berlin	01/2008
University of Karlsruhe	01/2008
MAAG, Kuesnacht, Switzerland	02/2008
M&M Turbine Technology, Bielefeld	02/2008
Lentjes, Ratingen	03/2008
Siemens Power Generation, Goerlitz	04/2008
Evonik, Zwingenberg (general EBSILON program license)	04/2008
WEBASTO, Neubrandenburg	04/2008
CFC Solutions, Munich	04/2008
RWE IT, Essen	04/2008
Rerum Cognitio, Zwickau	04/2008, 05/2008
ARUP, Berlin	05/2008
Research Center, Karlsruhe	07/2008
AWECO, Neukirch	07/2008
Technical University of Dresden,	07/2008
Professorship of Building Services	
Technical University of Cottbus,	07/2008, 10/2008
Chair in Power Plant Engineering	
Ingersoll-Rand, Unicov, Czech Republic	08/2008

Technip Benelux BV, Zoetermeer, Netherlands	08/2008
Fennovoima Oy, Helsinki, Finland	08/2008
Fichtner Consulting & IT, Stuttgart	09/2008
PEU, Espenhain	09/2008
Poyry, Dresden	09/2008
WINGAS, Kassel	09/2008
TUEV Sued, Dresden	10/2008
Technical University of Dresden, Professorship of Thermic Energy Machines and Plants	10/2008, 11/2008
AWTEC, Zurich, Switzerland	11/2008
Siemens Power Generation, Erlangen	12/2008

## 2007

Audi, Ingolstadt	02/2007
ANO Abfallbehandlung Nord, Bremen	02/2007
TUEV NORD SysTec, Hamburg	02/2007
VER, Dresden	02/2007
Technical University of Dresden, Chair in Jet Propulsion Systems	02/2007
Redacom, Nidau, Switzerland	02/2007
Universität der Bundeswehr, Munich	02/2007
Maxxtec, Sinsheim	03/2007
University of Rostock, Chair in Technical Thermodynamics	03/2007
AGO, Kulmbach	03/2007
University of Stuttgart, Chair in Aviation Propulsions	03/2007
Siemens Power Generation, Duisburg	03/2007
ENTHAL Haustechnik, Rees	05/2007
AWECO, Neukirch	05/2007
ALSTOM, Rugby, Great Britain	06/2007
SAAS, Possendorf	06/2007
Grenzebach BSH, Bad Hersfeld	06/2007
Reichel Engineering, Haan	06/2007
Technical University of Cottbus, Chair in Power Plant Engineering	06/2007
Voith Paper Air Systems, Bayreuth	06/2007
Egger Holzwerkstoffe, Wismar	06/2007
Tissue Europe Technologie, Mannheim	06/2007
Dometic, Siegen	07/2007
RWTH Aachen University, Institute for Electrophysics	09/2007
National Energy Technology Laboratory, Pittsburg, USA	10/2007
Energieversorgung Halle	10/2007
AL-KO, Jettingen	10/2007
Grenzebach BSH, Bad Hersfeld	10/2007

Wiesbaden University of Applied Sciences, Department of Engineering Sciences	10/2007
Endress+Hauser Messtechnik, Hannover	11/2007
Munich University of Applied Sciences, Department of Mechanical Engineering	11/2007
Rerum Cognitio, Zwickau	12/2007
Siemens Power Generation, Erlangen	11/2007
University of Rostock, Chair in Technical Thermodynamics	11/2007, 12/2007

## 2006

STORA ENSO Sachsen, Eilenburg	01/2006
Technical University of Munich, Chair in Energy Systems	01/2006
NUTEC Engineering, Bisikon, Switzerland	01/2006, 04/2006
Conwel eco, Bochov, Czech Republic	01/2006
Offenburg University of Applied Sciences	01/2006
KOCH Transporttechnik, Wadgassen	01/2006
BEG Bremerhavener Entsorgungsgesellschaft	02/2006
Deggendorf University of Applied Sciences, Department of Mechanical Engineering and Mechatronics	02/2006
University of Stuttgart,	02/2006
Department of Thermal Fluid Flow Engines	
Technical University of Munich,	02/2006
Chair in Apparatus and Plant Engineering	
Energietechnik Leipzig (company license),	02/2006
Siemens Power Generation, Erlangen	02/2006, 03/2006
RWE Power, Essen	03/2006
WAETAS, Pobershau	04/2006
Siemens Power Generation, Goerlitz	04/2006
Technical University of Braunschweig,	04/2006
Department of Thermodynamics	
EnviCon & Plant Engineering, Nuremberg	04/2006
Brassel Engineering, Dresden	05/2006
University of Halle-Merseburg,	05/2006
Department of USET Merseburg incorporated society	
Technical University of Dresden,	05/2006
Professorship of Thermic Energy Machines and Plants	
Fichtner Consulting & IT Stuttgart (company licenses and distribution)	05/2006
Suedzucker, Ochsenfurt	06/2006
M&M Turbine Technology, Bielefeld	06/2006
Feistel Engineering, Volkach	07/2006
ThyssenKrupp Marine Systems, Kiel	07/2006



Caliqua, Basel, Switzerland (company license)	09/2006
Atlas-Stord, Rodovre, Denmark	09/2006
Konstanz University of Applied Sciences, Course of Studies Construction and Development	10/2006
Siemens Power Generation, Duisburg	10/2006
Hannover University of Applied Sciences, Department of Mechanical Engineering	10/2006
Siemens Power Generation, Berlin	11/2006
Zikesch Armaturentechnik, Essen	11/2006
Wismar University of Applied Sciences, Seafaring Department	11/2006
BASF, Schwarzheide	12/2006
Enertech Energie und Technik, Radebeul	12/2006

## 2005

TUEV Nord, Hannover	01/2005
J.H.K Plant Engineering and Service, Bremerhaven	01/2005
Electrowatt-EKONO, Zurich, Switzerland	01/2005
FCIT, Stuttgart	01/2005
Energietechnik Leipzig (company license)	02/2005, 04/2005
	07/2005
eta Energieberatung, Pfaffenhofen	02/2005
FZR Forschungszentrum, Rossendorf/Dresden	04/2005
University of Saarbruecken	04/2005
Technical University of Dresden	04/2005
Professorship of Thermic Energy Machines and Plants	
Grenzebach BSH, Bad Hersfeld	04/2005
TUEV Nord, Hamburg	04/2005
Technical University of Dresden, Waste Management	05/2005
Siemens Power Generation, Goerlitz	05/2005
Duesseldorf University of Applied Sciences, Department of Mechanical Engineering and Process Engineering	05/2005
Redacom, Nidau, Switzerland	06/2005
Dumas Verfahrenstechnik, Hofheim	06/2005
Alensys Engineering, Erkner	07/2005
Stadtwerke Leipzig	07/2005
SaarEnergie, Saarbruecken	07/2005
ALSTOM ITC, Rugby, Great Britain	08/2005
Technical University of Cottbus, Chair in Power Plant Engineering	08/2005
Vattenfall Europe, Berlin (group license)	08/2005
Technical University of Berlin	10/2005
Basel University of Applied Sciences, Department of Mechanical Engineering, Switzerland	10/2005

Midiplan, Bietigheim-Bissingen	11/2005
Technical University of Freiberg, Chair in Hydrogeology	11/2005
STORA ENSO Sachsen, Eilenburg	12/2005
Energieversorgung Halle (company license)	12/2005
KEMA IEV, Dresden	12/2005

## 2004

Vattenfall Europe (group license)	01/2004
TUEV Nord, Hamburg	01/2004
University of Stuttgart, Institute of Thermodynamics and Heat Engineering	02/2004
MAN B&W Diesel A/S, Copenhagen, Denmark	02/2004
Siemens AG Power Generation, Erlangen	02/2004
Ulm University of Applied Sciences	03/2004
Visteon, Kerpen	03/2004, 10/2004
Technical University of Dresden, Professorship of Thermic Energy Machines and Plants	04/2004
Rerum Cognitio, Zwickau	04/2004
University of Saarbruecken	04/2004
Grenzebach BSH, Bad Hersfeld	04/2004
SOFBID Zwingenberg (general EBSILON program license)	04/2004
EnBW Energy Solutions, Stuttgart	05/2004
HEW-Kraftwerk, Tiefstack	06/2004
h s energieanlagen, Freising	07/2004
FCIT, Stuttgart	08/2004
Physikalisch Technische Bundesanstalt (PTB), Braunschweig	08/2004
Mainova Frankfurt	08/2004
Rietschle Energieplaner, Winterthur, Switzerland	08/2004
MAN Turbo Machines, Oberhausen	09/2004
TUEV Sued, Dresden	10/2004
STEAG Kraftwerk, Herne	10/2004, 12/2004
University of Weimar	10/2004
energeticals (e-concept), Munich	11/2004
SorTech, Halle	11/2004
Enertech EUT, Radebeul (company license)	11/2004
Munich University of Applied Sciences	12/2004
STORA ENSO Sachsen, Eilenburg	12/2004
Technical University of Cottbus, Chair in Power Plant Engineering	12/2004
Freudenberg Service, Weinheim	12/2004

## 2003

Paper Factory, Utzenstorf, Switzerland	01/2003
MAB Plant Engineering, Vienna, Austria	01/2003

Wulff Energy Systems, Husum	01/2003
Technip Benelux BV, Zoetermeer, Netherlands	01/2003
ALSTOM Power, Baden, Switzerland	01/2003, 07/2003
VER, Dresden	02/2003
Rietschle Energieplaner, Winterthur, Switzerland	02/2003
DLR, Leupholdhausen	04/2003
Emden University of Applied Sciences, Department of Technology	05/2003
Pettersson+Ahrends, Ober-Moerlen	05/2003
SOFBID ,Zwingenberg (general EBSILON program license)	05/2003
Ingenieurbuero Ostendorf, Gummersbach	05/2003
TUEV Nord, Hamburg	06/2003
Muenstermann GmbH, Telgte-Westbevern	06/2003
University of Cali, Colombia	07/2003
Atlas-Stord, Rodovre, Denmark	08/2003
ENERKO, Aldenhoven	08/2003
STEAG RKB, Leuna	08/2003
eta Energieberatung, Pfaffenhofen	08/2003
exergie, Dresden	09/2003
AWTEC, Zurich, Switzerland	09/2003
Energie, Timelkam, Austria	09/2003
Electrowatt-EKONO, Zurich, Switzerland	09/2003
LG, Annaberg-Buchholz	10/2003
FZR Forschungszentrum, Rossendorf/Dresden	10/2003
EnviCon & Plant Engineering, Nuremberg	11/2003
Visteon, Kerpen	11/2003
VEO Vulkan Energiewirtschaft Oderbruecke, Eisenhuettenstadt	11/2003
Stadtwerke Hannover	11/2003
SaarEnergie, Saarbruecken	11/2003
Fraunhofer-Gesellschaft, Munich	12/2003
Erfurt University of Applied Sciences, Department of Supply Engineering	12/2003
SorTech, Freiburg	12/2003
Mainova, Frankfurt	12/2003
Energieversorgung Halle	12/2003

## 2002

Hamilton Medical AG, Rhaezuens, Switzerland	01/2002
Bochum University of Applied Sciences, Department of Thermo- and Fluid Dynamics	01/2002
SAAS, Possendorf/Dresden	02/2002
Siemens, Karlsruhe (general license for the WinIS information system)	02/2002

FZR Forschungszentrum, Rossendorf/Dresden	03/2002
CompAir, Simmern	03/2002
GKS Gemeinschaftskraftwerk, Schweinfurt	04/2002
ALSTOM Power Baden, Switzerland (group licenses)	05/2002
InfraServ, Gendorf	05/2002
SoftSolutions, Muehlhausen (company license)	05/2002
DREWAG, Dresden (company license)	05/2002
SOFBID, Zwingenberg	06/2002
(general EBSILON program license)	
Kleemann Engineering, Dresden	06/2002
Caliqua, Basel, Switzerland (company license)	07/2002
PCK Raffinerie, Schwedt (group license)	07/2002
Fischer-Uhrig Engineering, Berlin	08/2002
Fichtner Consulting & IT, Stuttgart	08/2002
(company licenses and distribution)	
Stadtwerke Duisburg	08/2002
Stadtwerke Hannover	09/2002
Siemens Power Generation, Goerlitz	10/2002
Energieversorgung Halle (company license)	10/2002
Bayer, Leverkusen	11/2002
Dillinger Huette, Dillingen	11/2002
G.U.N.T. Geraetebau, Barsbuettel	12/2002
(general license and training test benches)	
VEAG, Berlin (group license)	12/2002

## 2001

ALSTOM Power, Baden, Switzerland	01/2001, 06/2001 12/2001
KW2 B. V., Amersfoot, Netherlands	01/2001, 11/2001
Eco Design, Saitamaken, Japan	01/2001
M&M Turbine Technology, Bielefeld	01/2001, 09/2001
MVV Energie, Mannheim	02/2001
Technical University of Dresden, Department of Power Machinery and Plants	02/2001
PREUSSAG NOELL, Wuerzburg	03/2001
Fichtner Consulting & IT Stuttgart	04/2001
(company licenses and distribution)	
Muenstermann GmbH, Telgte-Westbevern	05/2001
SaarEnergie, Saarbruecken	05/2001
Siemens, Karlsruhe	08/2001
(general license for the WinIS information system)	
Neusiedler AG, Ulmerfeld, Austria	09/2001

h s energieranlagen, Freising	09/2001
Electrowatt-EKONO, Zurich, Switzerland	09/2001
IPM Zittau/Goerlitz University of Applied Sciences (general license)	10/2001
eta Energieberatung, Pfaffenhofen	11/2001
ALSTOM Power Baden, Switzerland	12/2001
VEAG, Berlin (group license)	12/2001

## 2000

SOFBID, Zwingenberg	01/2000
(general EBSILON program license)	
AG KKK - PGW Turbo, Leipzig	01/2000
PREUSSAG NOELL, Wuerzburg	01/2000
M&M Turbine Technology, Bielefeld	01/2000
IBR Engineering Reis, Nittendorf-Undorf	02/2000
GK, Hannover	03/2000
KRUPP-UHDE, Dortmund (company license)	03/2000
UMAG W. UDE, Husum	03/2000
VEAG, Berlin (group license)	03/2000
Thinius Engineering, Erkrath	04/2000
SaarEnergie, Saarbruecken	05/2000, 08/2000
DVO Data Processing Service, Oberhausen	05/2000
RWTH Aachen University	06/2000
VAUP Process Automation, Landau	08/2000
Knuerr-Lommatec, Lommatzsch	09/2000
AVACON, Helmstedt	10/2000
Compania Electrica, Bogota, Colombia	10/2000
G.U.N.T. Geraetebau, Barsbuettel	11/2000
(general license for training test benches)	
Steinhaus Informationssysteme, Datteln	12/2000
(general license for process data software)	

## 1999

Bayernwerk, Munich	01/1999
DREWAG, Dresden (company license)	02/1999
KEMA IEV, Dresden	03/1999
Regensburg University of Applied Sciences	04/1999
Fichtner Consulting & IT, Stuttgart	07/1999
(company licenses and distribution)	
Technical University of Cottbus, Chair in Power Plant Engineering	07/1999
Technical University of Graz, Department of Thermal Engineering, Austria	11/1999
Ostendorf Engineering, Gummersbach	12/1999

## 1998

Technical University of Cottbus, Chair in Power Plant Engineering	05/1998
Fichtner Consulting & IT (CADIS information systems) Stuttgart (general KPRO program license)	05/1998
M&M Turbine Technology Bielefeld	06/1998
B+H Software Engineering Stuttgart	08/1998
Alfa Engineering, Switzerland	09/1998
VEAG Berlin (group license)	09/1998
NUTEC Engineering, Bisikon, Switzerland	10/1998
SCA Hygiene Products, Munich	10/1998
RWE Energie, Neurath	10/1998
Wilhelmshaven University of Applied Sciences	10/1998
BASF, Ludwigshafen (group license)	11/1998
Energieversorgung, Offenbach	11/1998

**1997**

Gerb, Dresden	06/1997
Siemens Power Generation, Goerlitz	07/1997