KCE
ThermoFluidProperties

# Property Library for Mixtures of Water/Lithium Bromide

## FluidEXL*Graphics* with LibWaLi for Excel®

Prof. Dr. Hans-Joachim Kretzschmar
Prof. Dr. Matthias Kunick
Dr. Sebastian Herrmann
M.Eng Martin Suender
Ines Jaehne
S. Hasch

# Software for the Calculation of the Properties of Mixtures of Water/Lithium Bromide Including DLL and Add-In for Excel®

# FluidEXL*Graphics*
# LibWaLi

## Contents


0. Package Contents
   - 0.1 Zip-files for 32-bit Office®
   - 0.2 Zip-files for 64-bit Office®
1. Property Functions
2. Application of FluidEXL in Excel®
   - 2.1 Installing FluidEXL
   - 2.2 Registering FluidEXL as Add-In in Excel®
   - 2.3 Licensing the LibWaLi Property Library
   - 2.4 Example calculation
   - 2.5 The FluidEXL Help System
   - 2.6 Removing FluidEXL
3. Program Documentation
4. Property Libraries for Calculating Heat Cycles, Boilers, Turbines, and Refrigerators
5. References
6. Satisfied Customers


---

© KCE-ThermoFluidProperties
Prof. Dr. Hans-Joachim Kretzschmar
Haager Weg 6, 92224 Amberg, Germany
Phone: +49-9621-1762047
Mobile: +49-172-7914607
Fax: +49-3222-1095810
Email: info@thermofluidprop.com
Internet: www.thermofluidprop.com

# 0 Package Contents

## 0.1 Zip files for 64-bit Office®

The following zip file has been delivered for your computer running a 64-bit Office® version:

CD_FluidEXL_Graphics_Eng_LibWaLi_x64.zip

including the following folders an files:

\FLUFT\

\Formulation97\

FluidEXL_Graphics_LibWaLi_Docu_Eng.pdf

FluidEXL_Graphics_Eng.xla

LC.dll

LibWaLi.dll

LibWaLi.chm.

## 0.2 Zip files for 32-bit Office®

The following zip file has been delivered for your computer running a 32-bit Office® version:

CD_FluidEXL_Graphics_Eng_LibWaLi.zip

including the following folders and files:

\FLUFT\

\Formulation97\

FluidEXL_Graphics_LibWaLi_Docu_Eng.pdf

FluidEXL_Graphics_Eng.xla

LC.dll

LibWaLi.dll

LibWaLi.chm.

# 1. Property Functions

| Functional Dependence | Function Name | Call from Fortran program | Property or Function | Unit of the result |
|---|---|---|---|---|
| $c_p = \mathrm{f}(p,t,\xi)$ | cp_ptxi_WaLi | CP_PTXI_WALI(P,T,XI) | Specific isobaric heat capacity | kJ/(kg K) |
| $c'_p = \mathrm{f}(p_s,t_s,\xi')$ | cpl_pstsxil_WaLi | CPL_WALI(PS,TS,XIL) | Specific isobaric heat capacity of saturated liquid | kJ/(kg K) |
| $c''_p = \mathrm{f}(p_s,t_s,\xi')$ | cpv_pstsxil_WaLi | CPV_WALI(PS,TS,XIL) | Specific isobaric heat capacity of saturated steam | kJ/(kg K) |
| $\eta = \mathrm{f}(p,t,\xi)$ | eta_ptxi_WaLi | ETA_PTXI_WALI(P,T,XI) | Dynamic viscosity | Pa s |
| $\eta' = \mathrm{f}(p_s,t_s,\xi')$ | etal_pstsxil_WaLi | ETAL_WALI(PS,TS,XIL) | Dynamic viscosity of saturated liquid | Pa s |
| $\eta'' = \mathrm{f}(p_s,t_s,\xi')$ | etav_pstsxil_WaLi | ETAV_WALI(PS,TS,XIL) | Dynamic viscosity of saturated steam | Pa s |
| $h = \mathrm{f}(p,t,\xi)$ | h_ptxi_WaLi | H_PTXI_WALI(P,T,XI) | Specific enthalpy | kJ/kg |
| $h' = \mathrm{f}(p_s,t_s,\xi')$ | hl_pstsxil_WaLi | HL_WALI(PS,TS,XIL) | Specific enthalpy of saturated liquid | kJ/kg |
| $h'' = \mathrm{f}(p_s,t_s,\xi')$ | hv_pstsxil_WaLi | HV_WALI(PS,TS,XIL) | Specific enthalpy of saturated steam | kJ/kg |
| $h_{sol} = \mathrm{f}(\xi)$ | hsol_xi_WaLi | HSOL_XI_WALI(XI) | Specific enthalpy at the crystallization barrier | kJ/kg |
| $\lambda = \mathrm{f}(p,t,\xi)$ | lam_ptxi_WaLi | LAM_PTXI_WALI(P,T,XI) | Thermal conductivity | W/(m K) |
| $\lambda' = \mathrm{f}(p_s,t_s,\xi')$ | laml_pstsxil_WaLi | LAML_WALI(PS,TS,XIL) | Thermal conductivity of saturated liquid | W/(m K) |
| $\lambda'' = \mathrm{f}(p_s,t_s,\xi')$ | lamv_pstsxil_WaLi | LAMV_WALI(PS,TS,XIL) | Thermal conductivity of saturated steam | W/(m K) |
| $\nu = \mathrm{f}(p,t,\xi)$ | ny_ptxi_WaLi | NY_PTXI_WALI(P,T,XI) | Kinematic viscosity | $m^2/s$ |
| $\nu' = \mathrm{f}(p_s,t_s,\xi')$ | nyl_pstsxil_WaLi | NYL_WALI(PS,TS,XIL) | Kinematic viscosity of saturated liquid | $m^2/s$ |
| $\nu'' = \mathrm{f}(p_s,t_s,\xi')$ | nyv_pstsxil_WaLi | NYV_WALI(PS,TS,XIL) | Kinematic viscosity of saturated steam | $m^2/s$ |
| $Pr = \mathrm{f}(p,t,\xi)$ | Pr_ptxi_WaLi | PR_PTXI_WALI(P,T,XI) | *Prandtl*-Number | - |

| Functional Dependence | Function Name | Call from Fortran program | Property or Function | Unit of the result |
|---|---|---|---|---|
| $Pr' = f(p_s,t_s,\xi')$ | Prl_pstsxil_WaLi | PRL_WALI(PS,TS,XIL) | *Prandtl*-Number of saturated liquid | - |
| $Pr'' = f(p_s,t_s,\xi')$ | Prv_pstsxil_WaLi | PRV_WALI(PS,TS,XIL) | *Prandtl*-Number of saturated steam | - |
| $p_s = f(t_s,\xi')$ | ps_tsxil_WaLi | PS_TSXIL_WALI(TS,XIL) | Vapor pressure | bar |
| $p_{sol} = f(t)$ | psol_t_WaLi | PSOL_T_WALI(T) | Pressure at the crystallization barrier | bar |
| $Region = f(p,t,\xi)$ | region_ptxi_WaLi | REGION_PTXI_WALI(P,T,XI) | Phase region from pressure, temperature and mass fraction of $H_2O$ | 1 – Subcooled fluid<br>2 – Two phase region<br>3 – Gas phase |
| $Region = f(p,h,\xi)$ | region_phxi_WaLi | REGION_PHXI_WALI(P,H,XI) | Phase region from pressure, enthalpy and mass fraction of $H_2O$ | |
| $Region = f(p,s,\xi)$ | region_psxi_WaLi | REGION_PSXI_WALI(P,S,XI) | Phase region from pressure, entropy and mass fraction of $H_2O$ | |
| $s = f(p,t,\xi)$ | s_ptxi_WaLi | S_PTXI_WALI(P,T,XI) | Specific entropy | kJ/(kg K) |
| $s' = f(p_s,t_s,\xi')$ | sl_pstsxil_WaLi | SL_WALI(PS,TS,XIL) | Specific entropy of saturated liquid | kJ/(kg K) |
| $s'' = f(p_s,t_s,\xi')$ | sv_pstsxil_WaLi | SV_WALI(PS,TS,XIL) | Specific entropy of saturated steam | kJ/(kg K) |
| $t = f(p,h,\xi)$ | t_phxi_WaLi | T_PHXI_WALI(P,H,XI) | Backward function: Temperature from pressure, enthalpy and mass fraction of $H_2O$ | °C |
| $t = f(p,s,\xi)$ | t_psxi_WaLi | T_PSXI_WALI(P,S,XI) | Backward function: Temperature from pressure, entropy and mass fraction of $H_2O$ | °C |
| $t_s = f(p_s,\xi')$ | ts_psxil_WaLi | TS_PSXIL_WALI(PS,XIL) | Saturation temperature | °C |

| **Functional Dependence** | **Function Name** | **Call from Fortran program** | **Property or Function** | **Unit of the result** |
|---|---|---|---|---|
| $t_{sol} = f(p)$ | tsol_p_WaLi | TSOL_P_WALI(P) | Temperature at the crystallization barrier | °C |
| $v = f(p,t,\xi)$ | v_ptxi_WaLi | V_PTXI_WALI(P,T,XI) | Specific volume | $m^3/kg$ |
| $v' = f(p_s,t_s,\xi')$ | vl_pstsxil_WaLi | VL_WALI(PS,TS,XIL) | Specific volume of saturated liquid | $m^3/kg$ |
| $v'' = f(p_s,t_s,\xi')$ | vv_pstsxil_WaLi | VV_WALI(PS,TS,XIL) | Specific volume of saturated steam | $m^3/kg$ |
| $\xi' = f(p_s,t_s)$ | xil_psts_WaLi | XIL_PSTS_WALI(PS,TS) | Mass fraction $H_2O$ of saturated liquid | kg/kg |
| $\xi_{sol} = f(p)$ | xisol_p_WaLi | XISOL_P_WALI(P) | Mass fraction $H_2O$ at the crystallization barrier | kg/kg |
| $\xi'' = f(p_s,t_s)$ | xiv_psts_WaLi | XIV_PSTS_WALI(PS,TS) | Mass fraction $H_2O$ of saturated steam | kg/kg |

**Units:**

| | |
|---|---|
| $t$ | in °C |
| $p$ | in bar |
| $\xi$ | in (kg $H_2O$)/(kg mixture) |

## Important hints for the calculation of wet steam

The wet steam region is calculated automatically by the subprograms, which are valid within the entire range of state.

It is necessary to define two parameters for the functions of saturated liquids (...′) and saturated steam (...″).

- either $t_s$ and $\xi'$
- or $p_s$ and $\xi'$
- or $p_s$ and $t_s$

and –1 must be entered as a pro-forma value for the value which is not given. When $p_s$, $t_s$, and $\xi'$ are entered, the program will consider $p_s$, $t_s$, and $\xi'$ to be appropriate to represent the saturation curve $p_s$. If this is not the case the calculation for the quantity of the chosen function to be calculated results in –1000.

## Range of validity

| | |
|---|---|
| Temperature range: | from 0 °C to 210 °C |
| Pressure range: | from 0.00074 bar to 10 bar |
| Composition range: | from 0.3 to 1.0 (kg $H_2O$)/(kg mixture) |

## Reference state

| | |
|---|---|
| Water: | Triple point for saturated liquid |
| | $h_{H2O}$ = 0.000611783 kJ/kg and |
| | $s_{H2O}$ = 0 |
| | at $p_{tr}$ = 0.00611657 bar and $t_{tr}$ = 0.01 °C |
| Mixture of water/lithium bromide: | saturated liquid |
| | $h_{0.5}$ = −0.0209415 kJ/kg and |
| | $s_{0.5}$ = −0,0000780433 kJ/(kg K) |
| | at $t$ = 0 °C and $\xi$ = 0.5 kg $H_2O$ / kg |

***Note.***

*If the calculation results in –1000, the values entered represent a state point outside the range of validity of LibWaLi. For further information on each function and its range of validity see Chapter 3. The same information may also be accessed via the online help pages.*

# 2. Application of FluidEXL*Graphics* in Excel®

The FluidEXL*Graphics* Add-In has been developed to calculate thermodynamic properties in Excel® more conveniently. Within Excel®, it enables the direct call of functions relating to Water and Steam from the LibWaLi property program library.

## 2.1 Installing FluidEXL*Graphics*

Complete the following steps for initial installation of FluidEXL*Graphics*.
Before you begin, it is best to uninstall any older version of FluidEXL*Graphics*.

The installation routine for 32-bit and 64-bit versions of Excel is similar. The following instructions are valid for both versions.
After you have downloaded and extracted the zip-file:

CD_FluidEXL_Graphics_LibWaLi_x64_Eng.zip (for 64 bit version)

or

CD_FluidEXL_Graphics_LibWaLi_Eng.zip" (for 32 bit version).

you will see the folder

\CD_FluidEXL_Graphics_LibWaLi_x64_Eng\ (for 64 bit version)

or

\CD_FluidEXL_Graphics_LibWaLi_Eng\ (for 32 bit version)

in your Windows Explorer, Total Commander etc.

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

Within this folder you will see the following folders and files:

\FLUFT\
\Formulation97\
FluidEXL_Graphics_Eng.xla
FluidEXL_Graphics_LibWaLi_Docu_Eng
LC.dll
LibWaLi.dll
LibWaLi.chm
Reg_.reg

Now, please copy the following folders and files

\FLUFT\
\Formulation97\
FluidEXL_Graphics_Eng.xla
LibWaLi.dll
LibWaLi.chm
LC.dll

into the folder

C:\Users\[your name]\AppData\Roaming\Microsoft\AddIns\,

where [your name] is your name in the Windows system.

If this folder is not found, follow the next section anyway.

## 2.2 Registering FluidEXL*Graphics* as Add-In in Excel®

After installation in Windows®, FluidEXL*Graphics* must be registered in Excel® as an Add-In. To do this, start Excel® and carry out the following steps:

- Click the "File" button in the upper left hand corner of Excel® (see Fig. 2.1)

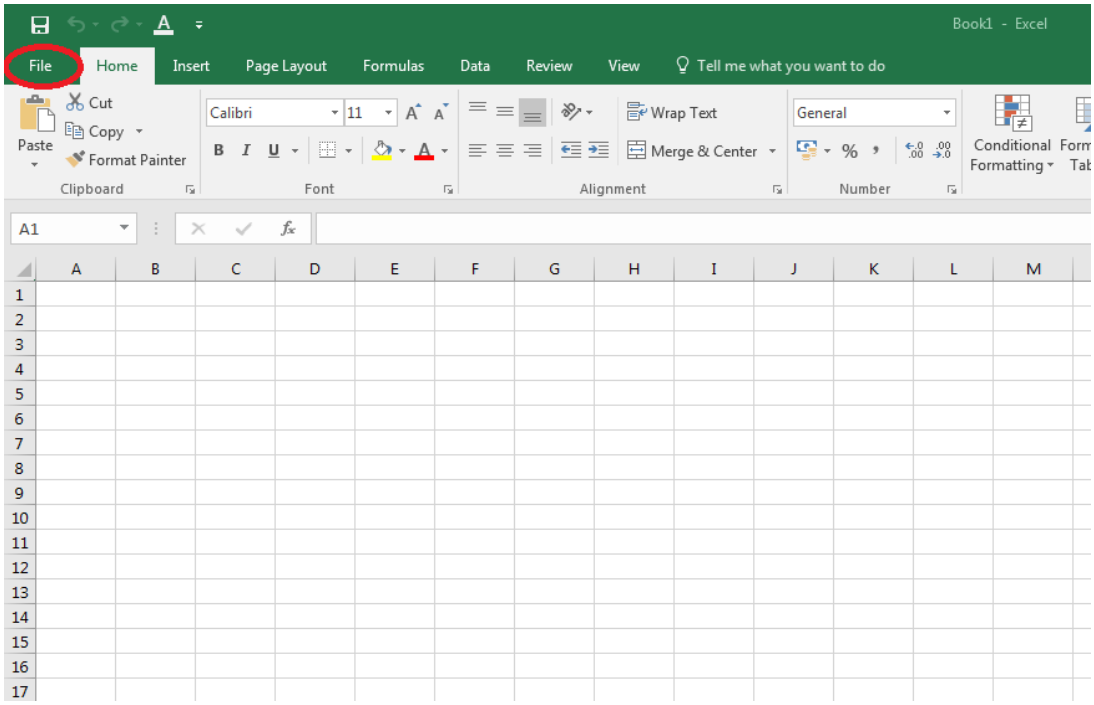



**Figure 2.1:** Registering FluidEXL*Graphics* as Add-In in Excel® 2016

- Click on the "Options" button in the menu which appears (see Fig. 2.2)

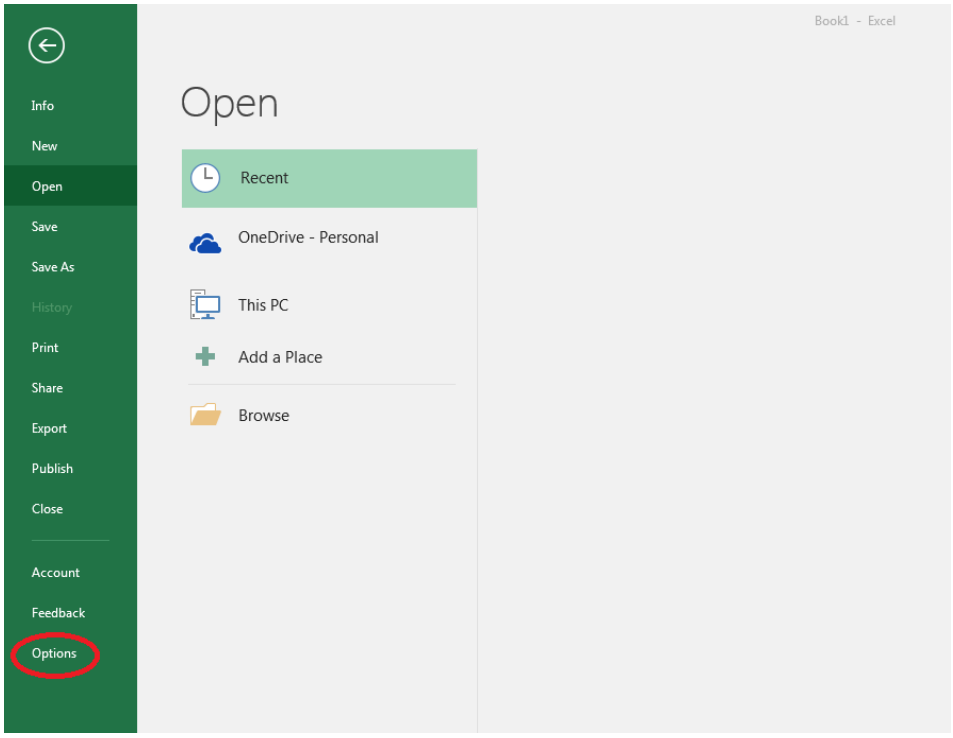



**Figure 2.2:** Registering FluidEXL*Graphics* as Add-In in Excel® 2016

- Click on "Add-Ins" in the next menu (Fig. 2.3)

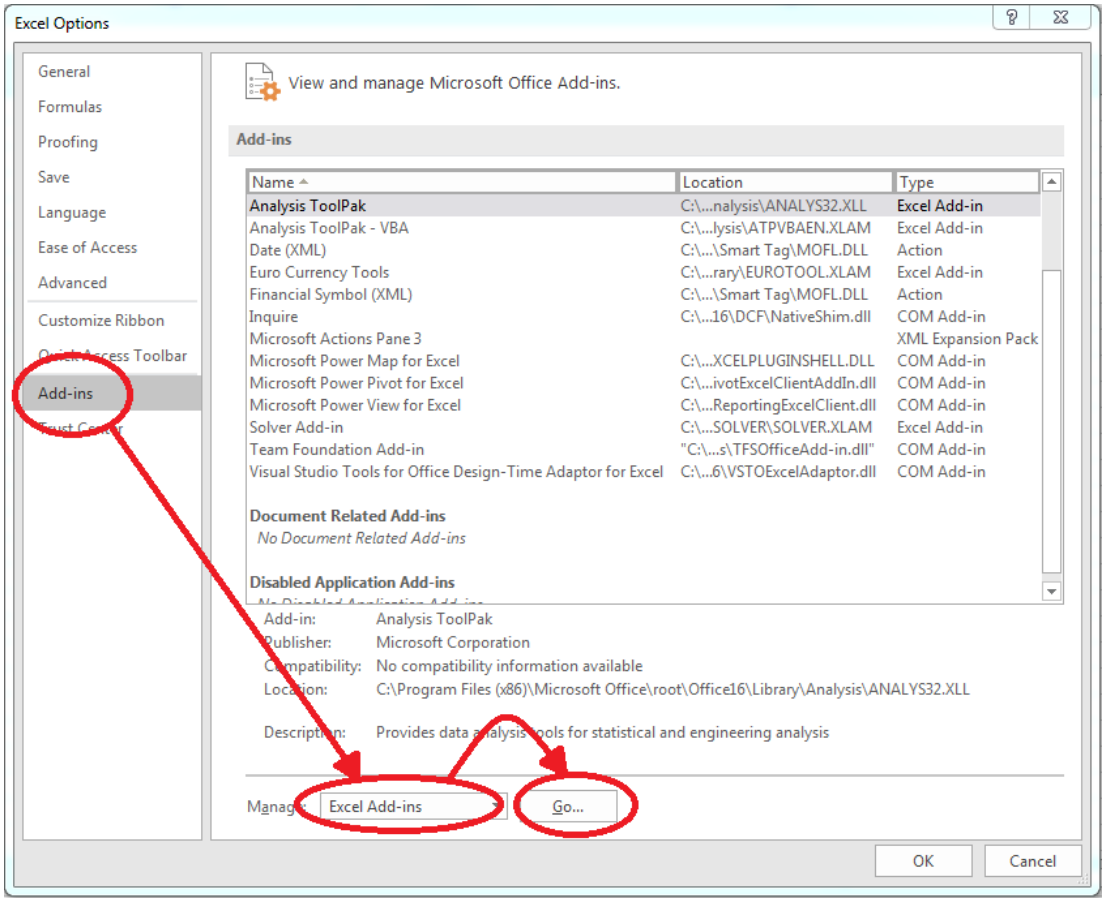



**Figure 2.3:** Dialog window "Excel Options"

- Select "Excel Add-ins" next to "Manage:" in the lower area of the menu
- Then click the "Go…" button
- Click "Browse" in the following window (Fig. 2.4)

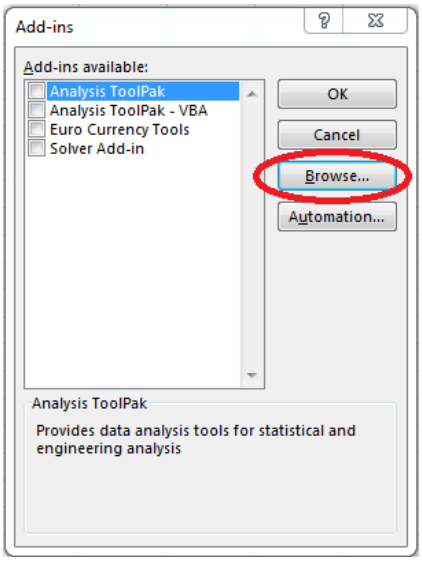



**Figure 2.4:** Dialog window "Add-ins"

- Excel opens the AddIns folder. This is usually
  C:\Users\[your name]\AppData\Roaming\Microsoft\AddIns\.
- If the FluidEXL files have already been copied to this directory in section 2.1, please skip the following indented section. If not, follow the indented instructions to successful paste the needed files for the FluidEXL Add-In:

  In the upper part of the "Browse" window the correct Add-In path is displayed (see Figure 2.5). Please note that not the entire path is displayed.

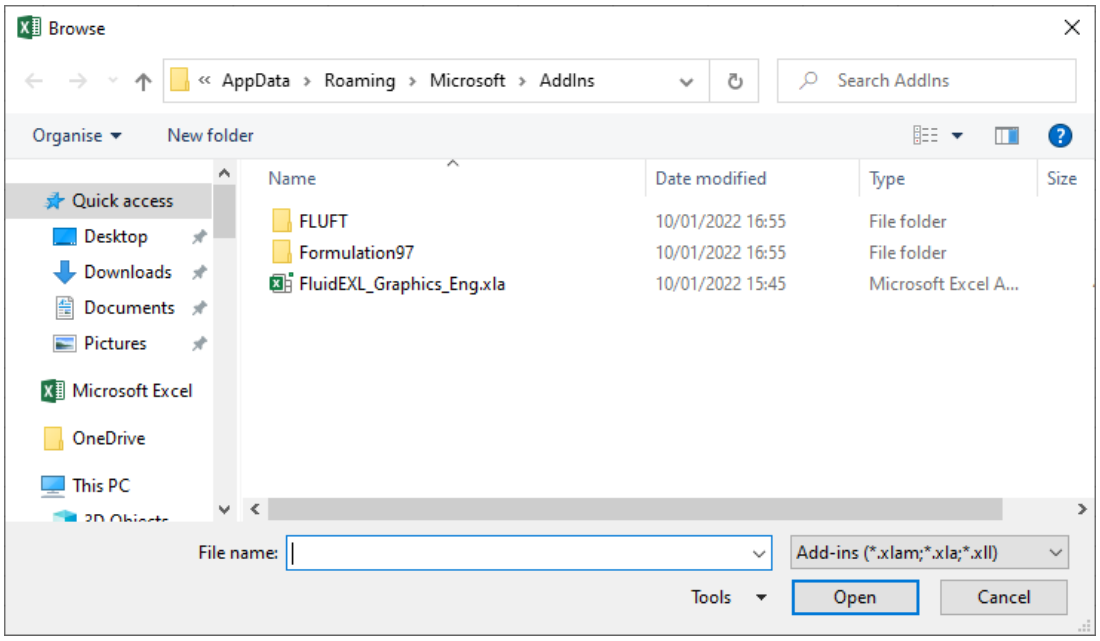



  **Figure 2.5:** "Browse"-Window

  Please copy this path and paste it into your file manager.
  Now, please copy the following directories and files:
  \FLUFT\
  \Formulation97\
  LC.dll
  FluidEXL_Graphics_Eng.xla
  LibWaLi.dll
  LibWaLi.chm
  from the delivered CD-folder into this folder.

- Click "FluidEXL_Graphics_Eng.xla" in this folder (see Fig. 2.5) and click "OK."
- Now, "FluidEXL Graphics Eng" will be shown in the list of Add-ins (see Fig. 2.6).
  (If a checkmark is in the box next to the name "FluidEXL Graphics Eng“, this Add-In will automatically be loaded whenever Excel starts. This will continue to occur unless the checkmark is removed from the box by clicking on it.)
- In order to register the Add-In click the "OK" button in the "Add-ins" window (see Fig. 2.6).

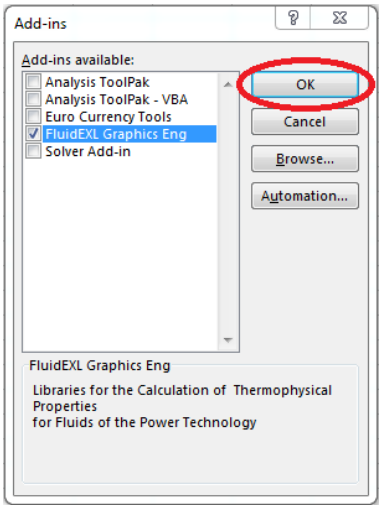



**Figure 2.6:** Dialog window "Add-Ins"

In order to use FluidEXL*Graphics* in the following example, click on the menu item "Add-Ins" shown in Fig. 2.7.

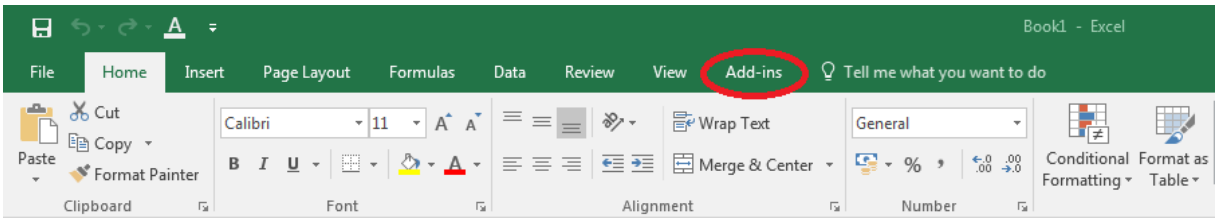



**Figure 2.7:** Menu item "Add-Ins"

In the upper menu region of Excel®, the FluidEXL*Graphics* menu bar will appear as marked with the red circle in Fig. 2.8.

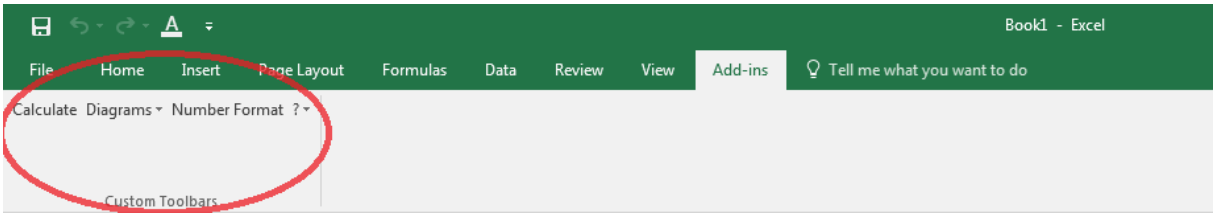



**Figure 2.8:** FluidEXL*Graphics* menu bar

The Installation of FluidEXL*Graphics* in Excel® is now complete.

An example calculation of "LibWaLi" DLL library property functions can be found in chapter 2.4.

## 2.3 Licensing the LibWaLi Property LibWaLirary

The licensing procedure has to be carried out when Excel® starts up and a FluidEXL*Graphics* prompt message appears. In this case, you will see the "License Information" window (see figure below).

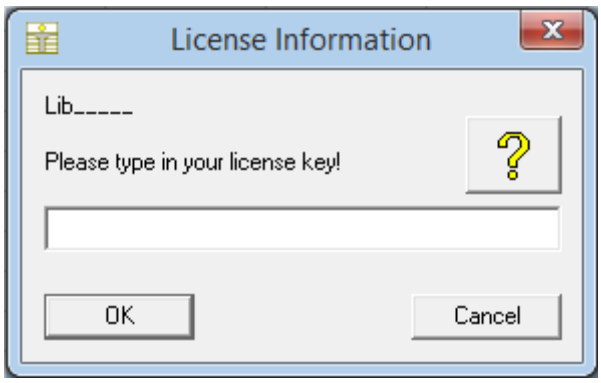



**Figure 2.12:** "License Information" window

Here you will have to type in the license key. 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:

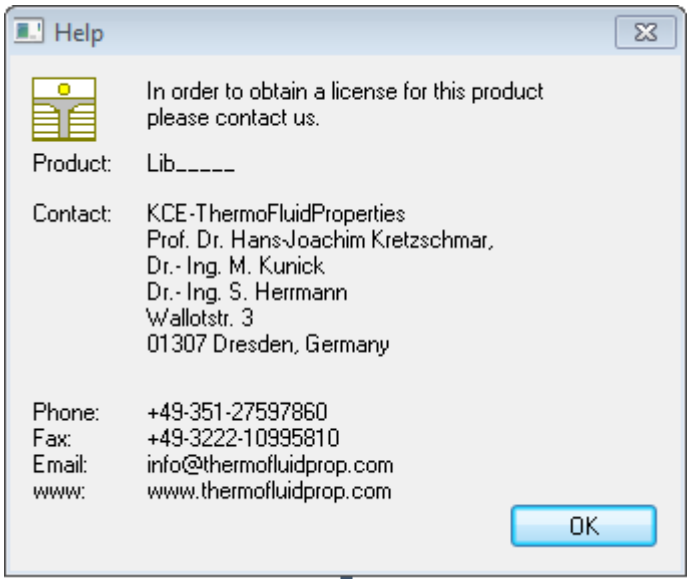



**Figure 2.13:** "Help" window

If you do not enter a valid license it is still possible to start Excel® by clicking "Cancel" twice. In this case, the LibWaLi property library will display the result "–11111111" for every calculation. The "License Information" window will appear every time you start Excel® unless you uninstall FluidEXL$^{Graphics}$ according to the description in section 2.6 of this User's Guide.
Should you not wish to license the LibWaLi property library, you have to delete the files

LibWaLi.dll
LibWaLi.chm

in the installation folder of FluidEXL$^{Graphics}$ (the standard being)

C:\Program Files\FluidEXL_Graphics_Eng

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

***Note:***

*The product name "LibWaLi_ _ _ _ _ in Figure 2.12 and 2.13 stands for the LibWaLirary you are installing. In this case it is the LibWaLi library.*

## 2.4 Example calculation

Now we will calculate, step by step, the specific enthalpy *h* as a function of pressure *p*, temperature *t*, and the mass fraction of water $\xi$, using FluidEXL*Graphics*. Please carry out the following steps:

- Start Excel®
- Enter a value for *p* in bar in a cell
  (Range of validity: $p$ = 0.00074 bar ... 10 bar)
  ⇒ e. g.: Enter the value 1 into cell A2
- Enter a value for *t* in °C in a cell
  (Range of validity: $t$ = 0 °C ... 210 °C)
  ⇒ e. g.: Enter the value 25 into cell B2
- Enter a value for $\xi$ in kg Water/kg mixture into a cell
  (Range of validity: $\xi$ = 0.3 kg $H_2O$/kg ... 1.0 kg $H_2O$/kg)
  ⇒ e.g.: Enter the value 0.6 into cell C2
- Click the cell in which the enthalpy *h* in kJ/kg is to be displayed.
  ⇒ e.g.: Click cell D2
- Click "Calculate" in the menu bar of FluidEXL*Graphics*
  Now, the "Insert Function" window appears (see next figure).

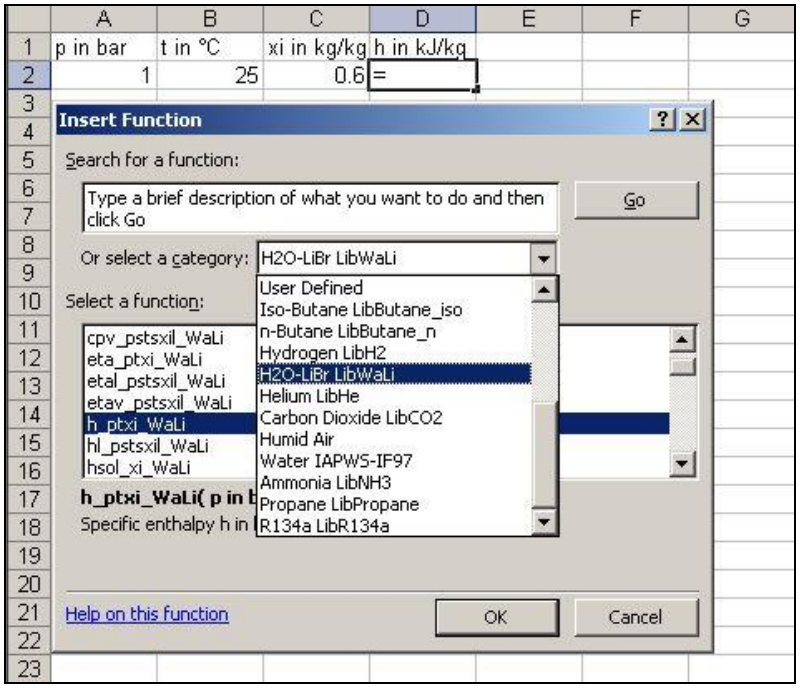



**Image 2.14:** Choice of library and function name

- In the drop-down menu in the middle, next to "Or select a category:" locate and click the library "H2O-LiBr LibWaLi"
- Below this drop-down menu under "Select a function:" choose and click the function h_ptx_WaLi
  Here it is possible to get more information on the range of validity, measuring units, error responses, etc. by clicking the "Help on this function" button.
- Click the "OK" button.
  The window shown in the next figure will now appear.

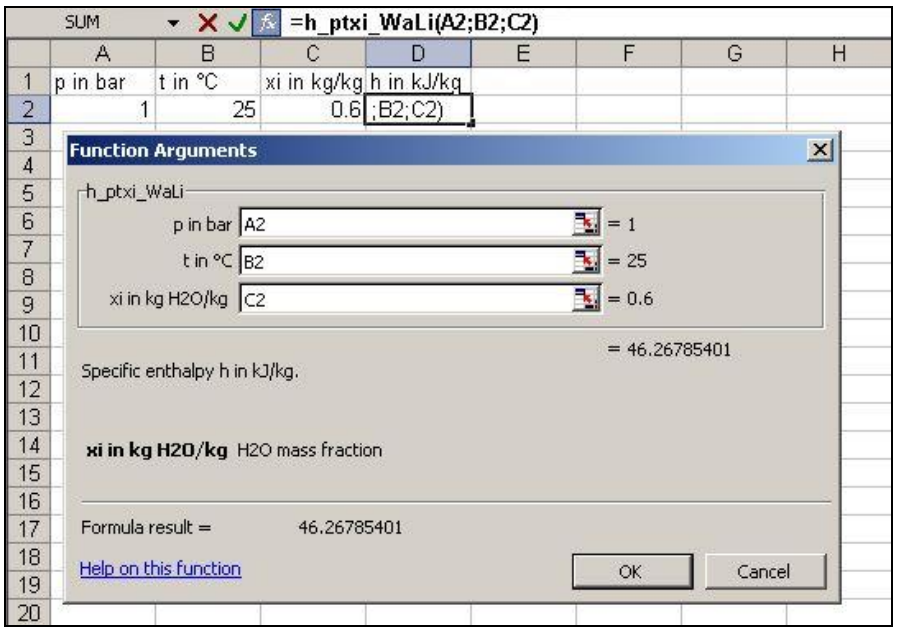



**Image 2.15:** Enter the given values for the property calculation

- The Cursor is now situated on the line next to "p in bar". You can now enter the value for *p* either by clicking the cell with the value for *p*, by entering the name of the cell with the value for *p*, or by entering the value for *p* directly.
- Situate the cursor next to "t in °C" and enter the value for *t* by clicking the cell with the value for *t*, by entering the name of the cell with the value for *t*, or by entering the value for *t* directly.
- Situate the cursor next to "xi in kg H2O/kg"and enter the value for $\xi$ by clicking the cell with the value for $\xi$, oby entering the name of the cell with the value for $\xi$, or by entering the value for $\xi$ directly.
- Click the "OK" button

The result for *h* in kJ/kg appears in the chosen cell.

⇒ The result of this example must be h = 46.26785401 kJ/kg

The calculation of $h = f(p,t,\xi)$ has been carried out. You can now arbitrarily change the values for *p*, *t*, or $\xi$ in the appropriate cells. The specific enthalpy is recalculated and updated every time you change the data. This shows that the Excel® data flow and the DLL calculations are working together successfully.

**Note:**

If the calculation results in –1000, this indicates that the values entered are located outside the range of validity of LibWaLi. More detailed information on each function and its range of validity is available in Chapter 3.

For further property functions calculable in FluidEXL*Graphics*, see the function table in Chapter 1.

## Number Formats

When using FluidEXL*Graphics* you have the option of choosing special number formats in advance.
Changes can be made as follows:

- Click the cell or select and click on the cells you wish to format.
  (In empty cells the new format will be applied once a value has been entered.)

- Click "Number Format" in the FluidEXL*Graphics* menu bar.
- Select the desired number format in the dialog box which appears:

  "STD – Standard": Insignificant zeros behind the decimal point are not shown.

  "FIX – Fixed Number of Digits": All set decimal places are shown, including insignificant zeros.

  "SCI – Scientific Format": Numbers are always shown in the exponential form with the set number of decimal places.
- Set the "Number of decimal places" by entering the number into the appropriate window.
- Confirm this by clicking the "OK" button.

As an example, the table below shows the three formats for the number 1.230 adjusted for three decimal places:

| | |
|---|---|
| STD | 1.23 |
| FIX | 1.230 |
| SCI | 1.230E+00 |

This formatting can also be applied to cells which have already been calculated.

## 2.5 The FluidEXL*Graphics* Help System

As mentioned earlier, FluidEXL*Graphics* also provides detailed help functions.

Information on individual property functions may be accessed via the following steps:

- Click "Calculate" in the FluidEXL*Graphics* menu bar.
- Click on the “LibWaLi” library under "Or select a category:" in the "Insert Function" window which will appear.
- Click the "Help on this function" button in the lower left-hand edge of the "Insert Function" window.

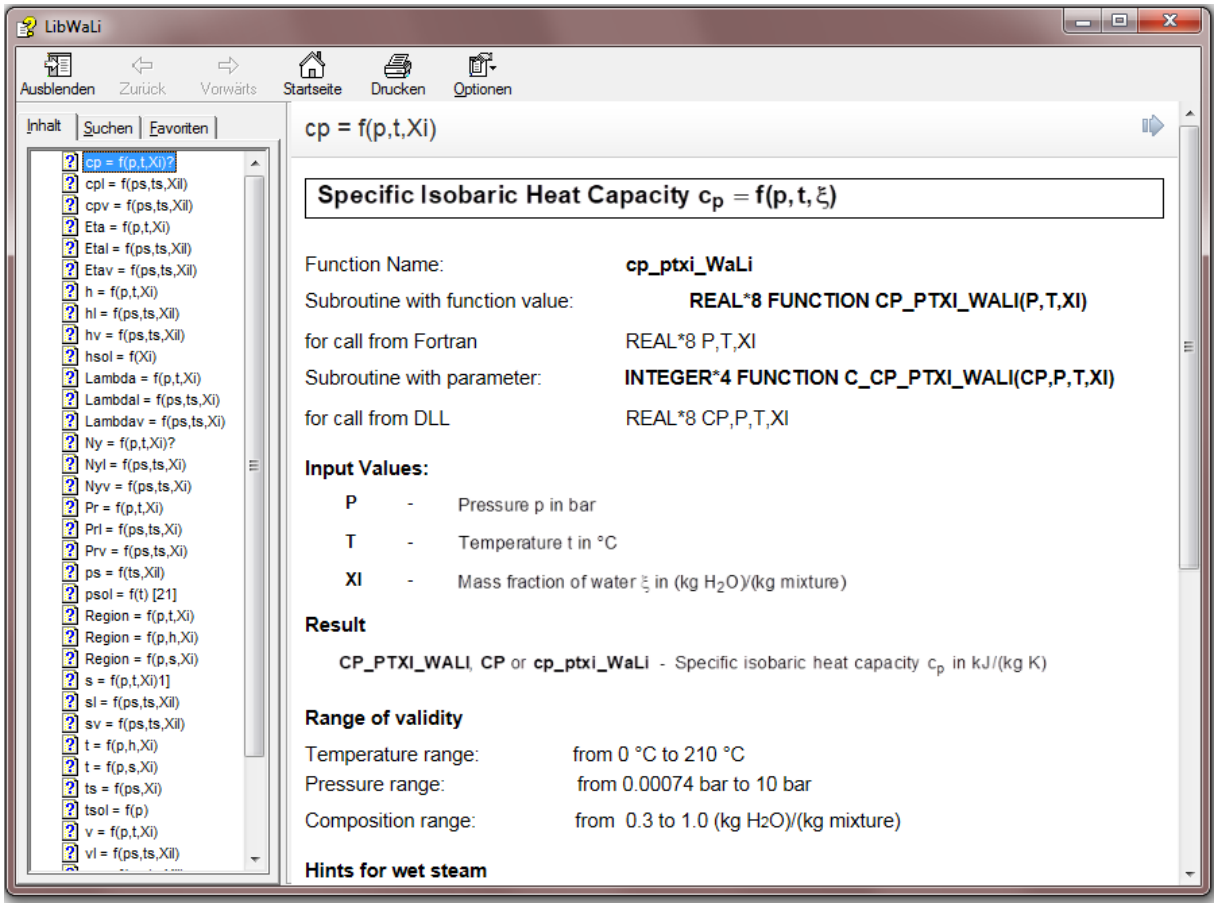



**Figure 2.16:** Help Window

If the LibWaLi.chm function help cannot be found, you will be redirected to a Microsoft® help website by your standard browser. In this case, the LibWaLi.chm file has to be copied into the folder of FluidEXL*Graphics*, in the standard case

C:\Program Files\FluidEXL_Graphics_Eng

to use the help system.

## 2.6 Removing FluidEXLGraphics

### 2.6.1 Removing LibWaLi Library

Should you wish to remove only the LibWaLi library, delete the files

LibWaLi.dll
LibWaLi.chm

in the directory selected for the installation of FluidEXL*Graphics*, in the standard case,

C:\Program Files\FluidEXL_Graphics_Eng

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

### 2.6.2 Unregistering and uninstalling FluidEXL*Graphics* as Add-In in versions of Excel® from 2007 onwards (for earlier versions see 2.6.3)

In order to unregister the FluidEXL*Graphics* Add-In in versions of Excel® from 2007 onwards start Excel® and carry out the following commands:

- Click the “File” button in the upper left corner of Excel®
- Click on the "Options" button in the menu which appears

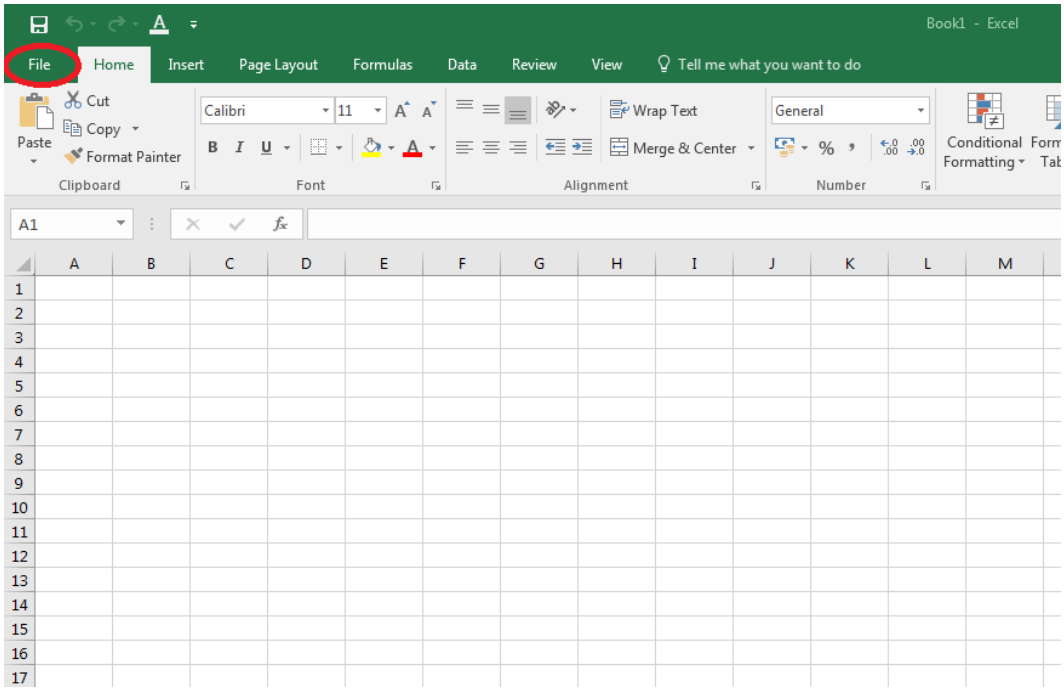



**Figure 2.17:** Unregistering FluidEXL*Graphics* as Add-In in Excel® 2016

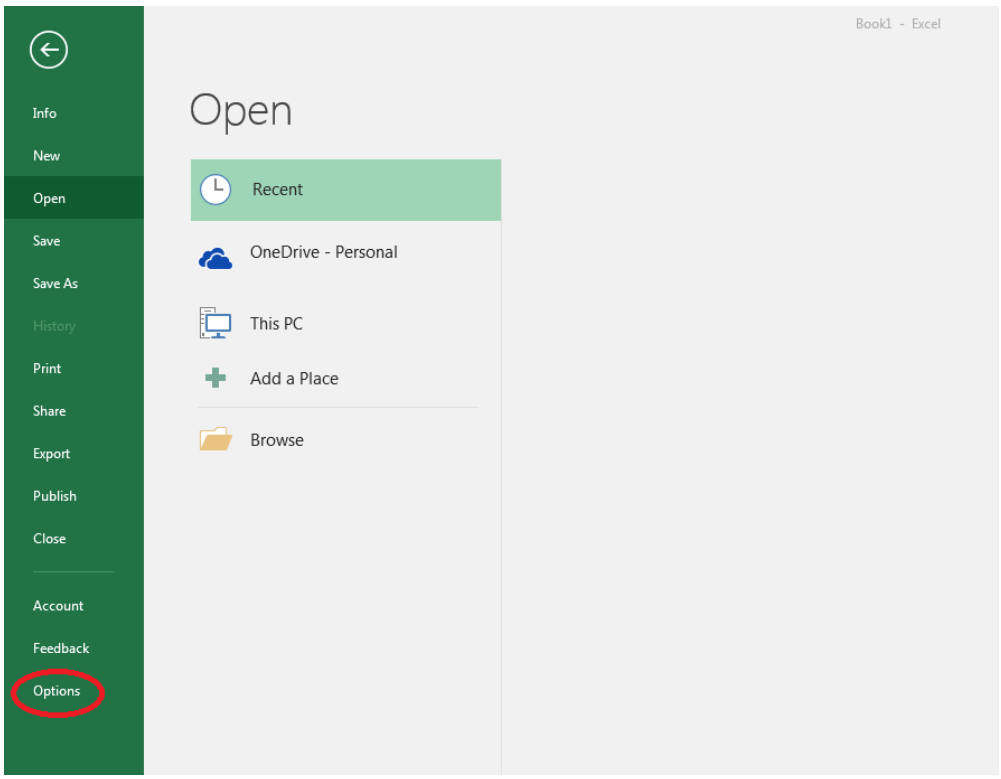



**Figure 2.18** Unregistering FluidEXL*Graphics* as Add-In in Excel® 2016

- Click on "Add-Ins" in the next menu (Figure 2.19)

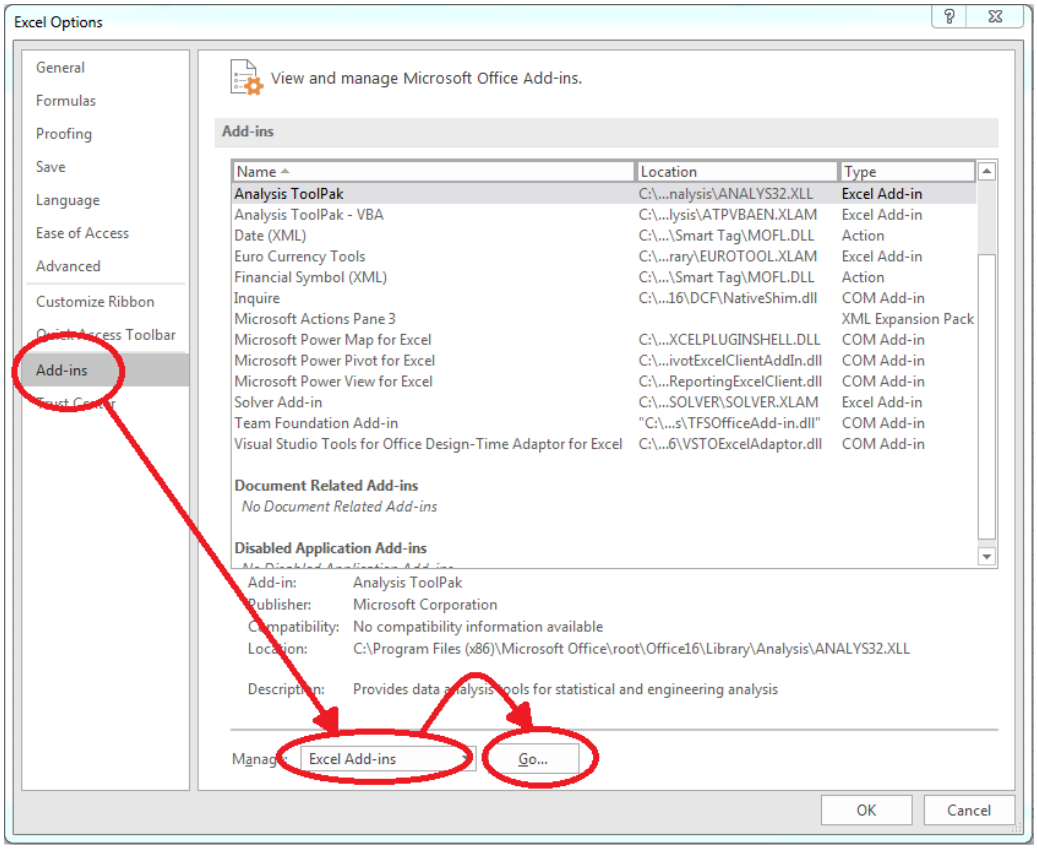



**Figure 2.19:** Dialog window "Add-Ins"

- If it is not shown in the list automatically, chose and click "Excel Add-ins" next to "Manage:" in the lower area of the menu

- Afterwards click the "Go…" button
- Remove the checkmark in front of
  "FluidEXL Graphics Eng"
  in the window which now appears. Click the "OK" button to confirm your entry.

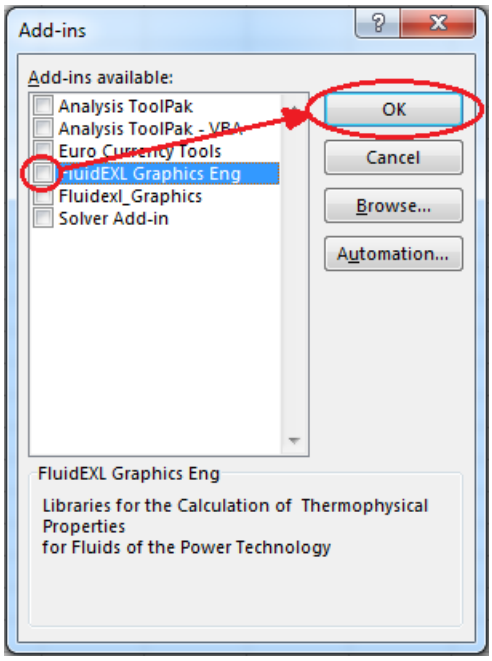



**Figure 2.20:** Dialog window "Add-Ins"

In order to remove FluidEXL*Graphics* from Windows and the hard drive, click "Start" in the Windows task bar, select "Settings" and click "Control Panel."

Now, double click on "Add or Remove Programs."

In the list box of the "Add or Remove Programs" window that appears, select
"FluidEXL Graphics Eng"
by clicking on it and then clicking the "Add/Remove…" button.

Click "Automatic" in the following dialog box and then the "Next >" button.

Click "Finish" in the "Perform Uninstall" window.

Answer the question of whether all shared components should be removed with "Yes to All." Finally, close the "Add or Remove Programs" and "Control Panel" windows.

Now FluidEXL*Graphics* has been completely removed from your computer.

### 2.6.3 Unregistering and uninstalling FluidEXL*Graphics* as Add-In in Excel®, versions 2003 or earlier

To remove FluidEXL*Graphics* completely, proceed as follows: First the registration of
FluidEXL_Graphics_Eng.xla
has to be cancelled in Excel®.

In order to do this, click "Tools" in the upper menu bar of Excel® and here "Add-Ins…". Untick the box on the left-hand side of
"FluidEXL Graphics Eng"
in the window that appears and click the "OK" button. The additional menu bar of FluidEXL*Graphics* disappears from the upper part of the Excel® window. Afterwards, we

recommend closing Excel®.

If the FluidEXL*Graphics* menu bar does not disappear, take the following steps:
Click "View" in the upper menu bar of Excel®, then "Toolbars" and then "Customize…" in the list box which appears.

"FluidEXL Graphics Eng"

is situated at the bottom of the "Toolbars" entries, which must be selected by clicking on it. Delete the entry by clicking "Delete". You will be asked whether you really want to delete the toolbar – click "OK".

Within the next step delete the files

LibWaLi.dll
LibWaLi.chm

in the directory selected for the installation of FluidEXL*Graphics*, in the standard case,

C:\Program Files\FluidEXL_Graphics_Eng

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

In order to remove FluidEXL*Graphics* from Windows and the hard drive, click "Start" in the Windows task bar, select "Settings" and click "Control Panel". Now double-click on "Add or Remove Programs". In the list box of the "Add/Remove Programs" window that appears select

"FluidEXL Graphics Eng"

by clicking on it and click the "Add/Remove…" button. In the following dialog box, click "Automatic" and then "Next >". Click "Finish" in the "Perform Uninstall" window. Answer the question whether all shared components shall be removed with "Yes to All". Finally, close the "Add/Remove Programs" and "Control Panel" windows.

Now FluidEXL*Graphics* has been removed.

# 3. Program Documentation

## Specific Isobaric Heat Capacity $c_p = f(p,t,\xi)$

Function Name: **cp_ptxi_WaLi**

Subroutine with function value: **REAL*8 FUNCTION CP_PTXI_WALI(P,T,XI)**
for call from Fortran REAL*8 P,T,XI

### Input Values:

**P** - Pressure p in bar

**T** - Temperature t in °C

**XI** - Mass fraction of water $\xi$ in (kg $H_2O$)/(kg mixture)

### Result

**CP_PTXI_WALI**, **CP** or **cp_ptxi_WaLi** - Specific isobaric heat capacity $c_p$ in kJ/(kg K)

### Range of validity

| | |
|---|---|
| Temperature range: | from 0 °C to 210 °C |
| Pressure range: | from 0.00074 bar to 10 bar |
| Composition range: | from 0.3 to 1.0 (kg $H_2O$)/(kg mixture) |

### Hints for wet steam

In the wet steam region the calculation is carried out for saturated liquid and saturated steam only. The value -1000 will be shown for the wet steam region lying between.

### Results for wrong input values

Result **CP_PTXI_WALI = -1000**, **CP = -1000** or **cp_ptxi_WaLi = -1000** for input values:

$p > 10$ bar or $p < 0.00074$ bar or
$t > 210$ °C or $t < 0$ °C or
$\xi_s > \xi_{sol}(t)$ or
$\xi > 1.0$ kg/kg or $\xi < 0.3$ kg/kg
Points in the wet steam region between saturated liquid and saturated steam

**References:** [1], [2], [4]

# Specific Isobaric Heat Capacity of Saturated Liquid $c_p^l = f(p_s, t_s, \xi^l)$

Function Name: **cpl_pstsxil_WaLi**

Subroutine with function value: **REAL*8 FUNCTION CPL_WALI(PS,TS,XIL)**
for call from Fortran REAL*8 PS,TS,XIL

## Input Values:

**PS** - Vapor pressure $p_s$ in bar

**TS** - Saturation temperature $t_s$ in °C

**XIL** - Mass fraction of water as saturated liquid $\xi^l$ in (kg $H_2O$)/(kg mixture)

## Result

**CPL_WALI**, **CPL** or **cpl_pstsxil_WaLi** - Specific isobaric heat capacity of the saturated liquid $c_p^l$ in kJ / (kg K)

## Range of validity

Temperature range: from 0 °C to 210 °C

Pressure range: from 0.00074 bar to 10 bar

Composition range: from 0.3 to 1.0 (kg $H_2O$)/(kg mixture)

## Hints of the input variants

For calculating the specific isobaric heat capacity of the saturated liquid it is necessary to define two parameters (either $t_S$ and $\xi'$ or $p_S$ and $\xi'$ or $p_S$ and $t_S$) and –1 must be entered as a pro-forma value for the value which is not given. When $p_S$, $t_S$, and $\xi'$ are entered, the program considers $p_S$, $t_S$, and $\xi'$ to be appropriate to represent the vapor pressure curve. If it is not the case the calculation for the quantity of the chosen function to be calculated results in –1000.

Possible input variants: $c_p^l = f(-1, t_s, \xi^l)$

$c_p^l = f(p_s, -1, \xi^l)$

$c_p^l = f(p_s, t_s, -1)$

$c_p^l = f(p_s, t_s, \xi^l)$

## Results for wrong input values

Result **CPL_WALI = −1000**, **CPL = −1000** or **cpl_pstsxil_WaLi = −1000** for input values:

$p_s > 10$ bar or $p_s < 0.00074$ bar or

$t_s > 210$ °C or $t_s < 0$ °C or

$t_s > t_{sol}(p_s)$ or

$\xi^l > 1.0$ kg/kg or $\xi^l < 0.3$ kg/kg

**References:** [1], [2], [4]

# Specific Isobaric Heat Capacity of Saturated Steam

## $c_p^v = f(p_s, t_s, \xi^l)$

Function Name: **cpv_pstsxil_WaLi**

Subroutine with function value: **REAL*8 FUNCTION CPV_WALI(PS,TS,XIL)**
for call from Fortran REAL*8 PS,TS,XIL

### Input Values:

**PS** - Vapor pressure $p_s$ in bar

**TS** - Saturation temperature $t_s$ in °C

**XIL** - Mass fraction of water as saturated liquid $\xi^l$ in (kg $H_2O$)/(kg mixture)

### Result

**CPV_WALI**, **CPV** or **cpv_pstsxil_WaLi** - Specific isobaric heat capacity of saturated steam $c_p^v$ in kJ/(kg K)

### Range of validity

Temperature range: from 0 °C to 210 °C

Pressure range: from 0.00074 bar to 10 bar

Composition range: from 0.3 to 1.0 (kg $H_2O$)/(kg mixture)

### Hints of the input variants

For calculating the specific isobaric heat capacity of saturated steam it is necessary to define two parameters (either $t_S$ and $\xi'$ or $p_S$ and $\xi'$ or $p_S$ and $t_S$) and –1 must be entered as a pro-forma value for the value which is not given. When $p_S$, $t_S$, and $\xi'$ are entered, the program considers $p_S$, $t_S$, and $\xi'$ to be appropriate to represent the vapor pressure curve. If it is not the case the calculation for the quantity of the chosen function to be calculated results in –1000.

Possible input variants:

$c_p^v = f(-1, t_s, \xi^l)$

$c_p^v = f(p_s, -1, \xi^l)$

$c_p^v = f(p_s, t_s, -1)$

$c_p^v = f(p_s, t_s, \xi^l)$

### Results for wrong input values

Result **CPV_WALI = −1000**, **CPV = −1000** or **cpv_pstsxil_WaLi = −1000** for input values:

$p_s > 10$ bar or $p_s < 0.00074$ bar or

$t_s > 210$ °C or $t_s < 0$ °C or

$t_s > t_{sol}(p_s)$ or

$\xi^l > 1.0$ kg/kg or $\xi^l < 0.3$ kg/kg

**References:** [1], [4]

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

Function Name: **eta_ptxi_WaLi**

Subroutine with function value: **REAL*8 FUNCTION ETA_PTXI_WALI(P,T,XI)**
for call from Fortran REAL*8 P,T,XI

## Input Values:

**P** - Pressure p in bar

**T** - Temperature t in °C

**XI** - Mass fraction of water $\xi$ in (kg $H_2O$)/(kg mixture)

## Result

**ETA_PTXI_WALI**, **ETA** or **eta_ptxi_WaLi** – Dynamic viscosity $\eta$ in Pa s

## Range of validity

| | |
|---|---|
| Temperature range: | from 0 °C to 210 °C |
| Pressure range: | from 0.00074 bar to 10 bar |
| Composition range: | from 0.3 to 1.0 (kg $H_2O$)/(kg mixture) |

## Hints for wet steam

In the wet steam region the calculation is carried out for saturated liquid and saturated steam only. The value -1000 will be shown for the wet steam region lying between.

## Results for wrong input values

Result **ETA_PTXI_WALI = -1000**, **ETA = -1000** or **eta_ptxi_PROP = -1000** for input values:

$p > 10$ bar or $p < 0.00074$ bar or
$t > 210$ °C or $t < 0$ °C or
$\xi_s > \xi_{sol}(t)$ or
$\xi > 1.0$ kg/kg or $\xi < 0.3$ kg/kg
Points in the wet steam region between saturated liquid and saturated steam

**References:** [1], [3], [4]

# Dynamic Viscosity of Saturated Liquid $\eta^l = f(p_s, t_s, \xi^l)$

Function Name: **etal_pstsxil_WaLi**

Subroutine with function value: **REAL*8 FUNCTION ETAL_WALI(PS,TS,XIL)**
for call from Fortran REAL*8 PS,TS,XIL

## Input Values:

**PS** - Vapor pressure $p_s$ in bar

**TS** - Saturation temperature $t_s$ in °C

**XIL** - Mass fraction of water as saturated liquid $\xi^l$ in (kg $H_2O$)/(kg mixture)

## Result

**ETAL_WALI**, **ETAL** or **etal_pstsxil_WaLi** - Dynamic viscosity of saturated liquid $\eta^l$ in Pa s

## Range of validity

Temperature range: from 0 °C to 210 °C

Pressure range: from 0.00074 bar to 10 bar

Composition range: from 0.3 to 1.0 (kg $H_2O$)/(kg mixture)

## Hints of the input variants

For calculating the dynamic viscosity of the saturated liquid it is necessary to define two parameters (either $t_S$ and $\xi'$ or $p_S$ and $\xi'$ or $p_S$ and $t_S$) and –1 must be entered as a pro-forma value for the value which is not given. When $p_S$, $t_S$, and $\xi'$ are entered, the program considers $p_S$, $t_S$, and $\xi'$ to be appropriate to represent the vapor pressure curve. If it is not the case the calculation for the quantity of the chosen function to be calculated results in –1000.

Possible input variants:

$\eta^l = f(-1, t_s, \xi^l)$

$\eta^l = f(p_s, -1, \xi^l)$

$\eta^l = f(p_s, t_s, -1)$

$\eta^l = f(p_s, t_s, \xi^l)$

## Results for wrong input values

Result **ETAL_WALI = −1000**, **ETAL = −1000** or **etal_pstsxil_WaLi = −1000** for input values:

$p_s > 10$ bar or $p_s < 0.00074$ bar or

$t_s > 210$ °C or $t_s < 0$ °C or

$t_s > t_{sol}(p_s)$ or

$\xi^l > 1.0$ kg/kg or $\xi^l < 0.3$ kg/kg

**References:** [1], [3], [4]

# Dynamic Viscosity of Saturated Steam $\eta^v = f(p_s, t_s, \xi^l)$

Function Name: **etav_pstsxil_WaLi**

Subroutine with function value: **REAL*8 FUNCTION ETAV_WALI(PS,TS,XIL)**
for call from Fortran REAL*8 PS,TS,XIL

## Input Values:

**PS** - Vapor pressure $p_s$ in bar

**TS** - Saturation temperature $t_s$ in °C

**XIL** - Mass fraction of water as saturated liquid $\xi^l$ in (kg $H_2O$)/(kg mixture)

## Result

**ETAV_WALI**, **ETAV** or **etav_pstsxil_WaLi** - Dynamic viscosity of saturated steam $\eta^v$ in Pa s

## Range of validity

Temperature range: from 0 °C to 210 °C

Pressure range: from 0.00074 bar to 10 bar

Composition range: from 0.3 to 1.0 (kg $H_2O$)/(kg mixture)

## Hints of the input variants

For calculating the dynamic viscosity of saturated steam it is necessary to define two parameters (either $t_S$ and $\xi'$ or $p_S$ and $\xi'$ or $p_S$ and $t_S$) and–1 must be entered as a pro-forma value for the value which is not given. When $p_S$, $t_S$, and $\xi'$ are entered, the program considers $p_S$, $t_S$, and $\xi'$ to be appropriate to represent the vapor pressure curve. If it is not the case the calculation for the quantity of the chosen function to be calculated results in –1000.

Possible input variants:

$$\eta^v = f(-1, t_s, \xi^l)$$
$$\eta^v = f(p_s, -1, \xi^l)$$
$$\eta^v = f(p_s, t_s, -1)$$
$$\eta^v = f(p_s, t_s, \xi^l)$$

## Results for wrong input values

Result **ETAV_WALI = −1000**, **ETAV = −1000** or **etav_pstsxil_WaLi = −1000** for input values:

$p_s > 10$ bar or $p_s < 0.00074$ bar or

$t_s > 210$ °C or $t_s < 0$ °C or

$t_s > t_{sol}(p_s)$ or

$\xi^l > 1.0$ kg/kg or $\xi^l < 0.3$ kg/kg

**References:** [1], [4]

# Specific Enthalpy $h = f(p,t,\xi)$

Function Name: **h_ptxi_WaLi**

Subroutine with function value: **REAL*8 FUNCTION H_PTXI_WALI(P,T,XI)**
for call from Fortran REAL*8 P,T,XI

## Input Values:

**P** - Pressure p in bar

**T** - Temperature t in °C

**XI** - Mass fraction of water $\xi$ in (kg $H_2O$)/(kg mixture)

## Result

**H_PTXI_WALI**, **H** or **h_ptxi_WaLi** – Specific enthalpy h in kJ/kg

## Range of validity

Temperature range: from 0 °C to 210 °C

Pressure range: from 0.00074 bar to 10 bar

Composition range: from 0.3 to 1.0 (kg $H_2O$)/(kg mixture)

## Hints for wet steam

The wet steam region is calculated automatically by the subprograms:

## Results for wrong input values

Result **H_PTXI_WALI = −1000**, **H = −1000** or **h_ptxi_WaLi = −1000** for input values:

p > 10 bar or p < 0.00074 bar or
t > 210 °C or t < 0 °C or
$\xi_s > \xi_{sol}(t)$ or
$\xi$ > 1.0 kg/kg or $\xi$ < 0.3 kg/kg

**References:** [1], [2], [4]

# Specific Enthalpy of Saturated Liquid $h^l = f(p_s, t_s, \xi^l)$

Function Name: **hl_pstsxil_WaLi**

Subroutine with function value: **REAL*8 FUNCTION HL_WALI(PS,TS,XIL)**
for call from Fortran REAL*8 PS,TS,XIL

## Input Values:

**PS** - Vapor pressure $p_s$ in bar

**TS** - Saturation temperature $t_s$ in °C

**XIL** - Mass fraction of water as saturated liquid $\xi^l$ in (kg $H_2O$)/(kg mixture)

## Result

**HL_WALI**, **HL** or **hl_pstsxil_WaLi** - Specific enthalpy of saturated liquid $h^l$ in kJ/kg

## Range of validity

Temperature range: from 0 °C to 210 °C

Pressure range: from 0.00074 bar to 10 bar

Composition range: from 0.3 to 1.0 (kg $H_2O$)/(kg mixture)

## Hints of the input variants

For calculating the specific enthalpy of the saturated liquid it is necessary to define two parameters (either $t_S$ and $\xi'$ or $p_S$ and $\xi'$ or $p_S$ and $t_S$) and –1 must be entered as a pro-forma value for the value which is not given. When $p_S$, $t_S$, and $\xi'$ are entered, the program considers $p_S$, $t_S$, and $\xi'$ to be appropriate to represent the vapor pressure curve. If it is not the case the calculation for the quantity of the chosen function to be calculated results in –1000.

Possible input variants:

$h^l = f(-1, t_s, \xi^l)$

$h^l = f(p_s, -1, \xi^l)$

$h^l = f(p_s, t_s, -1)$

$h^l = f(p_s, t_s, \xi^l)$

## Results for wrong input values

Result **HL_WALI = −1000**, **HL = −1000** or **hl_pstsxil_WaLi = −1000** for input values:

$p_s > 10$ bar or $p_s < 0.00074$ bar or

$t_s > 210$ °C or $t_s < 0$ °C or

$t_s > t_{sol}(p_s)$ or

$\xi^l > 1.0$ kg/kg or $\xi^l < 0.3$ kg/kg

**References:** [1], [2], [4]

# Specific Enthalpy of Saturated Steam $h^v = f(p_s, t_s, \xi^l)$

Function Name: **hv_pstsxil_WaLi**

Subroutine with function value: **REAL*8 FUNCTION HV_WALI(PS,TS,XIL)**
for call from Fortran REAL*8 PS,TS,XIL

## Input Values:

**PS** - Vapor pressure $p_s$ in bar

**TS** - Saturation temperature $t_s$ in °C

**XIL** - Mass fraction of water as saturated liquid $\xi^l$ in (kg $H_2O$)/(kg mixture)

## Result

**HV_WALI**, **HV** or **hv_pstsxil_WaLi** - Specific enthalpy of saturated steam $h^v$ in kJ/kg

## Range of validity

Temperature range: from 0 °C to 210 °C

Pressure range: from 0.00074 bar to 10 bar

Composition range: from 0.3 to 1.0 (kg $H_2O$)/(kg mixture)

## Hints of the input variants

For calculating the specific enthalpy of saturated steam it is necessary to define two parameters (either $t_S$ and $\xi'$ or $p_S$ and $\xi'$ or $p_S$ and $t_S$) and –1 must be entered as a pro-forma value for the value which is not given. When $p_S$, $t_S$, and $\xi'$ are entered, the program considers $p_S$, $t_S$, and $\xi'$ to be appropriate to represent the vapor pressure curve. If it is not the case the calculation for the quantity of the chosen function to be calculated results in –1000.

Possible input variants:

$$h^v = f(-1, t_s, \xi^l)$$
$$h^v = f(p_s, -1, \xi^l)$$
$$h^v = f(p_s, t_s, -1)$$
$$h^v = f(p_s, t_s, \xi^l)$$

## Results for wrong input values

Result **HV_WALI = −1000**, **HV = −1000** or **hv_pstsxil_WaLi = −1000** for input values:

$p_s > 10$ bar or $p_s < 0.00074$ bar or

$t_s > 210$ °C or $t_s < 0$ °C or

$t_s > t_{sol}(p_s)$ or

$\xi^l > 1.0$ kg/kg or $\xi^l < 0.3$ kg/kg

**References:** [1], [4]

# Specific Enthalpy at the Crystallization Barrier $h_{sol} = f(\xi)$

Function Name: **hsol_xi_WaLi**

Subroutine with function value: **REAL*8 FUNCTION HSOL_XI_WALI(XI)**
for call from Fortran REAL*8 XI

## Input Values:

**XI** - Mass fraction of water $\xi$ in (kg $H_2O$)/(kg mixture)

## Result

**HSOL_XI_WALI**, **HSOL** or **hsol_xi_WaLi** - Specific enthalpy at the crystallization barrier $h_{sol}$ in kJ/kg

## Range of validity

Composition range: from 0.3 to 0.45 (kg $H_2O$)/(kg mixture)

## Explanation of the function

This function calculates the specific enthalpy at the crystallization barrier. If the actual specific enthalpy calculated in the liquid phase of the mixture lies below the enthalpy calculated by the function then the Lithium Bromide crystallizes out. This case is recognised by the other functions. A calculation of properties below the crystallization barrier is not possible.

## Results for wrong input values

Result **HSOL_XI_WALI = -1000**, **HSOL = -1000** or **hsol_xi_WaLi = -1000** for input values:

$\xi > 0.45$ kg/kg or $\xi < 0.3$ kg/kg

## References: [4]

# Thermal Conductivity $\lambda = f(p,t,\xi)$

Function Name: **lam_ptxi_WaLi**

Subroutine with function value: **REAL*8 FUNCTION LAM_PTXI_WALI(P,T,XI)**
for call from Fortran REAL*8 P,T,XI

## Input Values:

**P** - Pressure p in bar

**T** - Temperature t in °C

**XI** - Mass fraction of water $\xi$ in (kg $H_2O$)/(kg mixture)

## Result

**LAM_PTXI_WALI**, **LAM** or **lam_ptxi_WaLi** – Thermal conductivity $\lambda$ in W/m K

## Range of validity

| | |
|---|---|
| Temperature range: | from 0 °C to 210 °C |
| Pressure range: | from 0.00074 bar to 10 bar |
| Composition range: | from 0.3 to 1.0 (kg $H_2O$)/(kg mixture) |

## Hints for wet steam

In the wet steam region the calculation is carried out for saturated liquid and saturated steam only. The value -1000 will be shown for the wet steam region lying between.

## Results for wrong input values

Result **LAM_PTXI_WALI = -1000**, **LAM = -1000** or **lam_ptxi_WaLi = -1000** for input values:

p > 10 bar or p < 0.00074 bar or
t > 210 °C or t < 0 °C oder
$\xi_s > \xi_{sol}(t)$ or

$\xi$ > 1.0 kg/kg or $\xi$ < 0.3 kg/kg
Points in the wet steam region between saturated liquid and saturated steam

**References:** [1], [3], [4]

# Thermal Conductivity of Saturated Liquid $\lambda^l = f(p_s, t_s, \xi^l)$

Function Name: **laml_pstsxil_WaLi**

Subroutine with function value: **REAL*8 FUNCTION LAML_WALI(PS,TS,XIL)**
for call from Fortran REAL*8 PS,TS,XIL

## Input Values:

**PS** - Vapor pressure $p_s$ in bar

**TS** - Saturation temperature $t_s$ in °C

**XIL** - Mass fraction of water as saturated liquid $\xi^l$ in (kg $H_2O$)/(kg mixture)

## Result

**LAML_WALI**, **LAML** or **laml_pstsxil_WaLi** - Thermal conductivity of saturated liquid $\lambda^l$ in W / (m K)

## Range of validity

Temperature range: from 0 °C to 210 °C

Pressure range: from 0.00074 bar to 10 bar

Composition range: from 0.3 to 1.0 (kg $H_2O$)/(kg mixture)

## Hints of the input variants

For calculating the thermal conductivity of the saturated liquid it is necessary to define two parameters (either $t_S$ and $\xi'$ or $p_S$ and $\xi'$ or $p_S$ and $t_S$) and –1 must be entered as a pro-forma value for the value which is not given. When $p_S$, $t_S$, and $\xi'$ are entered, the program considers $p_S$, $t_S$, and $\xi'$ to be appropriate to represent the vapor pressure curve. If it is not the case the calculation for the quantity of the chosen function to be calculated results in –1000.

Possible input variants:

$\lambda^l = f(-1, t_s, \xi^l)$

$\lambda^l = f(p_s, -1, \xi^l)$

$\lambda^l = f(p_s, t_s, -1)$

$\lambda^l = f(p_s, t_s, \xi^l)$

## Results for wrong input values

Result **LAML_WALI = −1000**, **LAML = −1000** or **laml_pstsxil_WaLi = −1000** for input values:

$p_s > 10$ bar or $p_s < 0.00074$ bar or

$t_s > 210$ °C or $t_s < 0$ °C or

$t_s > t_{sol}(p_s)$ or

$\xi^l > 1.0$ kg/kg or $\xi^l < 0.3$ kg/kg

**References:** [1], [3], [4]

# Thermal Conductivity of Saturated Steam $\lambda^v = f(p_s, t_s, \xi^l)$

Function Name: **lamv_pstsxil_WaLi**

Subroutine with function value: **REAL*8 FUNCTION LAMV_WALI(PS,TS,XIL)**
for call from Fortran REAL*8 PS,TS,XIL

## Input Values:

**PS** - Vapor pressure $p_s$ in bar

**TS** - Saturation temperature $t_s$ in °C

**XIL** - Mass fraction of water as saturated liquid $\xi^l$ in (kg $H_2O$)/(kg mixture)

## Result

**LAMV_WALI**, **LAMV** or **lamv_pstsxil_WaLi** - Thermal conductivity of saturated steam $\lambda^v$ in W / (m K)

## Range of validity

Temperature range: from 0 °C to 210 °C

Pressure range: from 0.00074 bar to 10 bar

Composition range: from 0.3 to 1.0 (kg $H_2O$)/(kg mixture)

## Hints of the input variants

For calculating the thermal conductivity of saturated steam it is necessary to define two parameters (either $t_S$ and $\xi'$ or $p_S$ and $\xi'$ or $p_S$ and $t_S$) and –1 must be entered as a pro-forma value for the value which is not given. When $p_S$, $t_S$, and $\xi'$ are entered, the program considers $p_S$, $t_S$, and $\xi'$ to be appropriate to represent the vapor pressure curve. If it is not the case the calculation for the quantity of the chosen function to be calculated results in –1000.

Possible input variants:

$$\lambda^v = f(-1, t_s, \xi^l)$$

$$\lambda^v = f(p_s, -1, \xi^l)$$

$$\lambda^v = f(p_s, t_s, -1)$$

$$\lambda^v = f(p_s, t_s, \xi^l)$$

## Results for wrong input values

Result **LAMV_WALI = −1000**, **LAMV = −1000** or **lamv_pstsxil_WaLi = −1000** for input values:

$p_s > 10$ bar or $p_s < 0.00074$ bar or

$t_s > 210$ °C or $t_s < 0$ °C or

$t_s > t_{sol}(p_s)$ or

$\xi^l > 1.0$ kg/kg or $\xi^l < 0.3$ kg/kg

**References:** [1], [4]

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

Function Name: **ny_ptxi_WaLi**

Subroutine with function value: **REAL*8 FUNCTION NY_PTXI_WALI(P,T,XI)**
for call from Fortran REAL*8 P,T,XI

## Input Values:

**P** - Pressure p in bar

**T** - Temperature t in °C

**XI** - Mass fraction of water $\xi$ in (kg $H_2O$)/(kg mixture)

## Result

**NY_PTXI_WALI**, **NY** or **ny_ptxi_WaLi** - Kinematic viscosity $\nu = \eta \cdot v$ in $m^2/s$

## Range of validity

| | |
|---|---|
| Temperature range: | from 0 °C to 210 °C |
| Pressure range: | from 0.00074 bar to 10 bar |
| Composition range: | from 0.3 to 1.0 (kg $H_2O$)/(kg mixture) |

## Hints for wet steam

In the wet steam region the calculation is carried out for saturated liquid and saturated steam only. The value -1000 will be shown for the wet steam region lying between.

## Results for wrong input values

Result **NY_PTXI_WALI = -1000**, **NY = -1000** or **ny_ptxi_WaLi = -1000** for input values:

p > 10 bar or p < 0.00074 bar or
t > 210 °C or t < 0 °C oder
$\xi_s > \xi_{sol}(t)$ or
$\xi$ > 1.0 kg/kg or $\xi$ < 0.3 kg/kg
Points in the wet steam region between saturated liquid and saturated steam

**References:** [1], [2], [3], [4]

# Kinematic Viscosity of Saturated Liquid $\nu^l = f(p_s, t_s, \xi^l)$

Function Name: **nyl_pstsxil_WaLi**

Subroutine with function value: **REAL*8 FUNCTION NYL_WALI(PS,TS,XIL)**
for call from Fortran REAL*8 PS,TS,XIL

## Input Values:

**PS** - Vapor pressure $p_s$ in bar

**TS** - Saturation temperature $t_s$ in °C

**XIL** - Mass fraction of water as saturated liquid $\xi^l$ in (kg $H_2O$)/(kg mixture)

## Result

**NYL_WALI**, **NYL** or **nyl_pstsxil_WaLi** - Kinematic viscosity of saturated liquid $\nu^l = \eta^l \cdot v^l$ in $m^2/s$

## Range of validity

Temperature range: from 0 °C to 210 °C

Pressure range: from 0.00074 bar to 10 bar

Composition range: from 0.3 to 1.0 (kg $H_2O$)/(kg mixture)

## Hints of the input variants

For calculating the kinematic viscosity of the saturated liquid it is necessary to define two parameters (either $t_S$ and $\xi'$ or $p_S$ and $\xi'$ or $p_S$ and $t_S$) and –1 must be entered as a pro-forma value for the value which is not given. When $p_S$, $t_S$, and $\xi'$ are entered, the program considers $p_S$, $t_S$, and $\xi'$ to be appropriate to represent the vapor pressure curve. If it is not the case the calculation for the quantity of the chosen function to be calculated results in –1000.

Possible input variants:

$\nu^l = f(-1, t_s, \xi^l)$

$\nu^l = f(p_s, -1, \xi^l)$

$\nu^l = f(p_s, t_s, -1)$

$\nu^l = f(p_s, t_s, \xi^l)$

## Results for wrong input values

Result **NYL_WALI = −1000**, **NYL = −1000** or **nyl_pstsxil_WaLi = −1000** for input values:

$p_s > 10$ bar or $p_s < 0.00074$ bar or

$t_s > 210$ °C or $t_s < 0$ °C or

$t_s > t_{sol}(p_s)$ or

$\xi^l > 1.0$ kg/kg or $\xi^l < 0.3$ kg/kg

**References:** [1], [2], [3], [4]

# Kinematic Viscosity of Saturated Steam $\nu^v = f(p_s, t_s, \xi^l)$

Function Name: **nyv_pstsxil_WaLi**

Subroutine with function value: **REAL*8 FUNCTION NYV_WALI(PS,TS,XIL)**
for call from Fortran REAL*8 PS,TS,XIL

## Input Values:

**PS** - Vapor pressure $p_s$ in bar

**TS** - Saturation temperature $t_s$ in °C

**XIL** - Mass fraction of water as saturated liquid $\xi^l$ in (kg $H_2O$)/(kg mixture)

## Result

**NYV_WALI**, **NYV** or **nyv_pstsxil_WaLi** - Kinematic viscosity of saturated steam
$\nu^v = \eta^v \cdot v^v$ in $m^2/s$

## Range of validity

Temperature range: from 0 °C to 210 °C

Pressure range: from 0.00074 bar to 10 bar

Composition range: from 0.3 to 1.0 (kg $H_2O$)/(kg mixture)

## Hints of the input variants

For calculating the kinematic viscosity of the saturated liquid it is necessary to define two parameters (either $t_S$ and $\xi'$ or $p_S$ and $\xi'$ or $p_S$ and $t_S$) and –1 must be entered as a pro-forma value for the value which is not given. When $p_S$, $t_S$, and $\xi'$ are entered, the program considers $p_S$, $t_S$, and $\xi'$ to be appropriate to represent the vapor pressure curve. If it is not the case the calculation for the quantity of the chosen function to be calculated results in –1000.

Possible input variants:

$$\nu^v = f(-1, t_s, \xi^l)$$
$$\nu^v = f(p_s, -1, \xi^l)$$
$$\nu^v = f(p_s, t_s, -1)$$
$$\nu^v = f(p_s, t_s, \xi^l)$$

## Results for wrong input values

Result **NYV_WALI = -1000**, **NYV = -1000** or **nyv_pstsxil_WaLi = -1000** for input values:

$p_s > 10$ bar or $p_s < 0.00074$ bar or

$t_s > 210$ °C or $t_s < 0$ °C or

$t_s > t_{sol}(p_s)$ or

$\xi^l > 1.0$ kg/kg or $\xi^l < 0.3$ kg/kg

**References:** [1], [4]

# Prandtl-Number $Pr = f(p,t,\xi)$

Function Name: **Pr_ptxi_WaLi**

Subroutine with function value: **REAL*8 FUNCTION PR_PTXI_WALI(P,T,XI)**
for call from Fortran REAL*8 P,T,XI

## Input Values:

**P** - Pressure p in bar

**T** - Temperature t in °C

**XI** - Mass fraction of water $\xi$ in (kg $H_2O$)/(kg mixture)

## Result

**PR_PTXI_WALI**, **PR** or **Pr_ptxi_WALI** - Prandtl-Number $Pr = \frac{\eta \cdot c_p}{\lambda}$

## Range of validity

| | |
|---|---|
| Temperature range: | from 0 °C to 210 °C |
| Pressure range: | from 0.00074 bar to 10 bar |
| Composition range: | from 0.3 to 1.0 (kg $H_2O$)/(kg mixture) |

## Hints for wet steam

In the wet steam region the calculation is carried out for saturated liquid and saturated steam only. The value -1000 will be shown for the wet steam region lying between.

## Results for wrong input values

Result **PR_PTXI_WALI = -1000**, **PR = -1000** or **Pr_ptxi_WaLi = -1000** for input values:

$p > 10$ bar or $p < 0.00074$ bar or
$t > 210$ °C or $t < 0$ °C or
$\xi_s > \xi_{sol}(t)$ or
$\xi > 1.0$ kg/kg or $\xi < 0.3$ kg/kg
Points in the wet steam region between saturated liquid and saturated steam

**References:** [1], [2], [3], [4]

# Prandtl-Number of Saturated Liquid $Pr^l = f(p_s, t_s, \xi^l)$

Function Name: **Prl_pstsxil_WaLi**

Subroutine with function value: **REAL*8 FUNCTION PRL_WALI(PS,TS,XIL)**
for call from Fortran REAL*8 PS,TS,XIL

## Input Values:

**PS** - Vapor pressure $p_s$ in bar

**TS** - Saturation temperature $t_s$ in °C

**XIL** - Mass fraction of water as saturated liquid $\xi^l$ in (kg $H_2O$)/(kg mixture)

## Result

**PRL_WALI**, **PRL** or **Prl_pstsxil_WaLi** - Prandtl-Number of saturated liquid $Pr^l = \frac{\eta^l \cdot c_p{}^l}{\lambda^l}$

## Range of validity

Temperature range: from 0 °C to 210 °C

Pressure range: from 0.00074 bar to 10 bar

Composition range: from 0.3 to 1.0 (kg $H_2O$)/(kg mixture)

## Hints of the input variants

For calculating the Prandtl-Number of the saturated liquid it is necessary to define two parameters (either $t_S$ and $\xi'$ or $p_S$ and $\xi'$ or $p_S$ and $t_S$) and –1 must be entered as a pro-forma value for the value which is not given. When $p_S$, $t_S$, and $\xi'$ are entered, the program considers $p_S$, $t_S$, and $\xi'$ to be appropriate to represent the vapor pressure curve. If it is not the case the calculation for the quantity of the chosen function to be calculated results in –1000.

Possible input variants:

$Pr^l = f(-1, t_s, \xi^l)$

$Pr^l = f(p_s, -1, \xi^l)$

$Pr^l = f(p_s, t_s, -1)$

$Pr^l = f(p_s, t_s, \xi^l)$

## Results for wrong input values

Result **PRL_WALI = −1000**, **PRL = −1000** or **Prl_pstsxil_WaLi = −1000** for input values:

$p_s > 10$ bar or $p_s < 0.00074$ bar or

$t_s > 210$ °C or $t_s < 0$ °C or

$t_s > t_{sol}(p_s)$ or

$\xi^l > 1.0$ kg/kg or $\xi^l < 0.3$ kg/kg

**References:** [1], [2], [3], [4]

# Prandtl-Number of Saturated Steam $Pr^v = f(p_s, t_s, \xi^l)$

Function Name: **Prv_pstsxil_WaLi**

Subroutine with function value: **REAL*8 FUNCTION PRV_WALI(PS,TS,XIL)**
for call from Fortran REAL*8 PS,TS,XIL

## Input Values:

**PS** - Vapor pressure $p_s$ in bar

**TS** - Saturation temperature $t_s$ in °C

**XIL** - Mass fraction of water as saturated liquid $\xi^l$ in (kg $H_2O$)/(kg mixture)

## Result

**PRV_WALI**, **PRV** or **Prv_pstsxil_WaLi** - Prandtl-Number of saturated steam $Pr^v = \frac{\eta^v \cdot c_p{}^v}{\lambda^v}$

## Range of validity

Temperature range: from 0 °C to 210 °C

Pressure range: from 0.00074 bar to 10 bar

Composition range: from 0.3 to 1.0 (kg $H_2O$)/(kg mixture)

## Hints of the input variants

For calculating the Prandtl-Number of saturated steam it is necessary to define two parameters (either $t_S$ and $\xi'$ or $p_S$ and $\xi'$ or $p_S$ and $t_S$) and –1 must be entered as a pro-forma value for the value which is not given. When $p_S$, $t_S$, and $\xi'$ are entered, the program considers $p_S$, $t_S$, and $\xi'$ to be appropriate to represent the vapor pressure curve. If it is not the case the calculation for the quantity of the chosen function to be calculated results in –1000.

Possible input variants:

$Pr^v = f(-1, t_s, \xi^l)$

$Pr^v = f(p_s, -1, \xi^l)$

$Pr^v = f(p_s, t_s, -1)$

$Pr^v = f(p_s, t_s, \xi^l)$

## Results for wrong input values

Result **PRV_WALI = −1000**, **PRV = −1000** or **Prv_pstsxil_WaLi = −1000** for input values:

$p_s > 10$ bar or $p_s < 0.00074$ bar or

$t_s > 210$ °C or $t_s < 0$ °C or

$t_s > t_{sol}(p_s)$ or

$\xi^l > 1.0$ kg/kg or $\xi^l < 0.3$ kg/kg

**References:** [1], [4]

# Vapor Pressure $p_S = f(t_S, \xi^l)$

Function Name: **ps_tsxil_WaLi**

Subroutine with function value: **REAL*8 FUNCTION PS_TSXIL_WALI(TS,XIL)**
for call from Fortran REAL*8 TS,XIL

## Input Values:

**TS** - Saturation temperature $t_s$ in °C

**XIL** - Mass fraction of water as saturated liquid $\xi^l$ in (kg $H_2O$)/(kg mixture)

## Result

**PS_TSXIL_WALI, PS or ps_tsxil_WaLi** - Vapor pressure $p_s$ in bar

## Range of validity

Temperature range: from 0 °C to 210 °C

Composition range: from 0.3 to 1.0 (kg $H_2O$)/(kg mixture)

## Results for wrong input values

Result **PS_TSXIL_WALI = −1000**, **PS = −1000** or **ps_tsxil_WALI = −1000** for input values:

$t_s > 210$ °C or $t_s < 0$ °C or

$\xi^l > 1.0$ kg/kg or $\xi^l < 0.3$ kg/kg

$\xi^l < \xi_{sol}(t_s)$ or

**References:** [1], [2], [4]

# Pressure at the Crystallization Barrier $p_{sol} = f(t)$

Function Name: **psol_t_WaLi**

Subroutine with function value: **REAL*8 FUNCTION PSOL_T_WALI(T)**
for call from Fortran REAL*8 T

## Input Values:

**T** - Temperature t in °C

## Result

**PSOL_T_WALI**, **PSOL** or **psol_t_WaLi** - Pressure at the Crystallization barrier $p_{sol}$ in bar

## Range of validity

Temperature range: from 0 °C to 93.58 °C

## Explanation of the function

This function calculates the pressure at the crystallization barrier. If the actual pressure calculated in the liquid phase of the mixture lies below the pressure calculated by the function then the Lithium Bromide crystallizes out. This case is recognised by the other functions. A calculation of properties below the crystallization barrier is not possible. Please note that this function puts out expedient results only if a liquid phase is calculated since there is pure water in the gas phase.

## Results for wrong input values

Result **PSOL_T_WALI = −1000**, **PSOL = −1000** or **psol_t_WaLi = −1000** for input values:

t > 93.58 °C or t < 0 °C

**References:** [1], [2], [4]

# Phase Region Region = f(p,t,ξ)

| | |
|---|---|
| Function Name: | **region_ptxi_WaLi** |
| Subroutine with function value:<br>for call from Fortran | **INTEGER*4 FUNCTION REGION_PTXI_WALI(P,T,XI)**<br>REAL*8 P,T,XI |

## Input Values:

**P** - Pressure p in bar

**T** - Temperature t in °C

**XI** - Mass fraction of water ξ in (kg $H_2O$)/(kg mixture)

## Result

**REGION_PTXI_WALI**, **REGION** or **region_ptxi_WaLi** – Phase region

## Range of validity

| | |
|---|---|
| Temperature range: | from 0 °C to 210 °C |
| Pressure range: | from 0.00074 bar to 10 bar |
| Composition range: | from 0.3 to 1.0 (kg $H_2O$)/(kg mixture) |

## Explanation of the function

This function defines the phase region at hand. The results have the following meaning:

1 – Subcooled fluid
2 – Two phase region
3 – Gas phase

If a point of state is entered below the crystallization barrier the result will be – 2000.

## Results for wrong input values

Result **REGION_PTXI_WALI = −1000**, **REGION = −1000** or **region_ptxi_WaLi = −1000** for input values:

p > 10 bar or p < 0.00074 bar or
t > 210 °C or t < 0 °C oder
ξ > 1.0 kg/kg or ξ < 0.3 kg/kg

Result **REGION_PTXI_WALI = −2000, REGION = −2000** or **region_ptxi_WaLi = −2000** for input values:

$\xi < \xi_{sol}(t)$

**References:** [1], [2], [4]

# Phase Region Region = f(p,h,ξ)

Function Name: **region_phxi_WaLi**

Subroutine with function value: **INTEGER*4 FUNCTION REGION_PHXI_WALI(P,H,XI)**
for call from Fortran REAL*8 P,H,XI

## Input Values:

**P** - Pressure p in bar

**H** - Specific enthalpy h in kJ/kg

**XI** - Mass fraction of water ξ in (kg $H_2O$)/(kg mixture)

## Result

**REGION_PHXI_WALI**, **REGION** or **region_phxi_WaLi** – Phase region

## Range of validity

Temperature range: from 0 °C to 210 °C

Pressure range: from 0.00074 bar to 10 bar

Composition range: from 0.3 to 1.0 (kg $H_2O$)/(kg mixture)

## Explanation of the function

This function defines the phase region at hand. The results have the following meaning:

1 – Subcooled fluid
2 – Two phase region
3 – Gas phase

If a point of state is entered below the crystallization barrier the result will be – 2000.

## Results for wrong input values

Result **REGION_PHXI_WALI = −1000**, **REGION = −1000** or **region_phxi_WaLi = −1000** for input values:

p > 10 bar or p < 0.00074 bar or
h > 2899.46775 kJ/kg or h < 14 kJ/kg or
ξ > 1.0 kg/kg or ξ < 0.3 kg/kg

Result **REGION_PHXI_WALI = -2000, REGION = -2000** or **region_phxi_WaLi = -2000** for input values:

$h < h_{sol}(\xi)$

**References:** [1], [2], [4]

# Phase Region Region = f(p,s,ξ)

Function Name: **region_psxi_WaLi**

Subroutine with function value: **INTEGER*4 FUNCTION REGION_PSXI_WALI(P,S,XI)**
for call from Fortran REAL*8 P,H,XI

## Input Values:

**P** - Pressure p in bar

**S** - Specific entropy s in kJ/(kg K)

**XI** - Mass fraction of water ξ in (kg $H_2O$)/(kg mixture)

## Result

**REGION_PSXI_WALI**, **REGION** or **region_psxi_WaLi** – Phase region

## Range of validity

Temperature range: from 0 °C to 210 °C

Pressure range: from 0.00074 bar to 10 bar

Composition range: from 0.3 to 1.0 (kg $H_2O$)/(kg mixture)

## Explanation of the function

This function defines the phase region at hand. The results have the following meaning:

1 – Subcooled fluid
2 – Two phase region
3 – Gas phase

If a point of state is entered below the crystallization barrier the result will be – 2000.

## Results for wrong input values

Result **REGION_PHXI_WALI = −1000**, **REGION = −1000** or **region_phxi_WaLi = −1000** for input values:

p > 10 bar or p < 0.00074 bar or
at intermediate result t > 210 °C or t < 0 °C or
ξ > 1.0 kg/kg or ξ < 0.3 kg/kg

Result **REGION_PSXI_WALI = -2000, REGION = -2000** or **region_psxi_WaLi = -2000** for input values:

$\xi < \xi_{sol}(p)$

**References:** [1], [2], [4]

# Specific Entropy s = f(p,t,ξ)

| | |
|---|---|
| Function Name: | **s_ptxi_WaLi** |
| Subroutine with function value:<br>for call from Fortran | **REAL*8 FUNCTION S_PTXI_WALI(P,T,XI)**<br>REAL*8 P,T,X |

## Input Values:

**P** - Pressure p in bar

**T** - Temperature t in °C

**XI** - Mass fraction from water ξ in (kg $H_2O$)/(kg mixture)

## Result

**S_PTXI_WALI**, **S** or **s_ptxi_WaLi** - Specific entropy s in kJ/kg K

## Range of validity

| | |
|---|---|
| Temperature range: | from 0 °C to 210 °C |
| Pressure range: | from 0.00074 bar to 10 bar |
| Composition range: | from 0.3 to 1.0 (kg $H_2O$)/(kg mixture) |

## Hints for wet steam

The wet steam region is calculated automatically by the subprograms:

## Results for wrong input values

Result **S_PTXI_WALI = −1000**, **S = −1000** or **s_ptxi_WaLi = −1000** for input values:

p > 10 bar or p < 0.00074 bar or
t > 210 °C or t < 0 °C oder
$\xi < \xi_{sol}(t)$
ξ > 1.0 kg/kg or ξ < 0.3 kg/kg

**References:** [1], [2], [4]

# Specific Entropy of Saturated Liquid $s^l = f(p_s, t_s, \xi^l)$

Function Name: **sl_pstsxil_WaLi**

Subroutine with function value: **REAL*8 FUNCTION SL_WALI(PS,TS,XIL)**
for call from Fortran REAL*8 PS,TS,XIL

## Input Values:

**PS** - Vapor pressure $p_s$ in bar

**TS** - Saturation temperature $t_s$ in °C

**XIL** - Mass fraction of water as saturated liquid $\xi^l$ in (kg $H_2O$)/(kg mixture)

## Result

**SL_WALI**, **SL** or **sl_pstsxil_WaLi** - Specific entropy of the saturated liquid $s^l$ in kJ/(kg K)

## Range of validity

Temperature range: from 0 °C to 210 °C

Pressure range: from 0.00074 bar to 10 bar

Composition range: from 0.3 to 1.0 (kg $H_2O$)/(kg mixture)

## Hints of the input variants

For calculating the specific entropy of the saturated liquid it is necessary to define two parameters (either $t_S$ and $\xi'$ or $p_S$ and $\xi'$ or $p_S$ and $t_S$) and –1 must be entered as a pro-forma value for the value which is not given. When $p_S$, $t_S$, and $\xi'$ are entered, the program considers $p_S$, $t_S$, and $\xi'$ to be appropriate to represent the vapor pressure curve. If it is not the case the calculation for the quantity of the chosen function to be calculated results in –1000.

Possible input variants:

$s^l = f(-1, t_s, \xi^l)$

$s^l = f(p_s, -1, \xi^l)$

$s^l = f(p_s, t_s, -1)$

$s^l = f(p_s, t_s, \xi^l)$

## Results for wrong input values

Result **SL_WALI = −1000**, **SL = −1000** or **sl_pstsxil_WaLi = −1000** for input values:

$p_s > 10$ bar or $p_s < 0.00074$ bar or

$t_s > 210$ °C or $t_s < 0$ °C or

$t_s > t_{sol}(p_s)$ or

$\xi^l > 1.0$ kg/kg or $\xi^l < 0.3$ kg/kg

**References:** [1], [2], [4]

# Specific Entropy of Saturated Steam $s^v = f(p_s, t_s, \xi^l)$

Function Name: **sv_pstsxil_WaLi**

Subroutine with function value: **REAL*8 FUNCTION SV_WALI(PS,TS,XIL)**
for call from Fortran REAL*8 PS,TS,XIL

## Input Values:

**PS** - Vapor pressure $p_s$ in bar

**TS** - Saturation temperature $t_s$ in °C

**XIL** - Mass fraction of water as saturated liquid $\xi^l$ in (kg $H_2O$)/(kg mixture)

## Result

**SV_WALI**, **SV** or **sv_pstsxil_WaLi** - Specific entropy of saturated steam $s^v$ in kJ/(kg K)

## Range of validity

Temperature range: from 0 °C to 210 °C

Pressure range: from 0.00074 bar to 10 bar

Composition range: from 0.3 to 1.0 (kg $H_2O$)/(kg mixture)

## Hints of the input variants

For calculating the specific entropy of saturated steam it is necessary to define two parameters (either $t_S$ and $\xi'$ or $p_S$ and $\xi'$ or $p_S$ and $t_S$) and –1 must be entered as a pro-forma value for the value which is not given. When $p_S$, $t_S$, and $\xi'$ are entered, the program considers $p_S$, $t_S$, and $\xi'$ to be appropriate to represent the vapor pressure curve. If it is not the case the calculation for the quantity of the chosen function to be calculated results in –1000.

Possible input variants:

$$s^v = f(-1, t_s, \xi^l)$$
$$s^v = f(p_s, -1, \xi^l)$$
$$s^v = f(p_s, t_s, -1)$$
$$s^v = f(p_s, t_s, \xi^l)$$

## Results for wrong input values

Result **SV_WALI = −1000**, **SV = −1000** or **sv_pstsxil_WaLi = −1000** for input values:

$p_s > 10$ bar or $p_s < 0.00074$ bar or

$t_s > 210$ °C or $t_s < 0$ °C or

$t_s > t_{sol}(p_s)$ or

$\xi^l > 1.0$ kg/kg or $\xi^l < 0.3$ kg/kg

**References:** [1], [4]

# Backward Function: Temperature t = f(p,h,ξ)

| | |
|---|---|
| Function Name: | **t_phxi_WaLi** |
| Subroutine with function value:<br>for call from Fortran | **REAL*8 FUNCTION T_PHXI_WALI(P,H,XI)**<br>REAL*8 P,H,XI |

## Input Values:

- **P** - Pressure p in bar
- **H** - Specific enthalpy h in kJ/kg
- **XI** - Mass fraction of water ξ in (kg $H_2O$)/(kg mixture)

## Result

**T_PHXI_WALI**, **T** or **t_phxi_WaLi** – Temperature t in °C

## Range of validity

| | |
|---|---|
| Temperature range: | from 0 °C to 210 °C |
| Pressure range: | from 0.00074 bar to 10 bar |
| Composition range: | from 0.3 to 1.0 (kg $H_2O$)/(kg mixture) |

## Hints for wet steam

The wet steam region is calculated automatically by the subprograms:

## Results for wrong input values

Result **T_PHXI_WALI = −1000**, **T = −1000** or **t_phxi_WaLi = −1000** for input values:

p > 10 bar or p < 0.00074 bar or
at calculation result t > 210 °C or t < 0 °C or
ξ > 1.0 kg/kg or ξ < 0.3 kg/kg or
$h < h_{sol}(\xi)$

**References:** [1], [2], [4]

# Backward Function: Temperature t = f(p,s,ξ)

| | |
|---|---|
| Function Name: | **t_psxi_WaLi** |
| Subroutine with function value:<br>for call from Fortran | **REAL*8 FUNCTION T_PSXI_WALI(P,S,XI)**<br>REAL*8 P,S,XI |

## Input Values:

- **P** - Pressure p in bar
- **S** - Specific entropy s in kJ/(kg K)
- **XI** - Mass fraction of water ξ in (kg $H_2O$)/(kg mixture)

## Result

**T_PSXI_WALI**, **T** or **t_psxi_WaLi** – Temperature *t* in °C

## Range of validity

| | |
|---|---|
| Temperature range: | from 0 °C to 210 °C |
| Pressure range: | from 0.00074 bar to 10 bar |
| Composition range: | from 0.3 to 1.0 (kg $H_2O$)/(kg mixture) |

## Hints for wet steam

The wet steam region is calculated automatically by the subprograms:

## Results for wrong input values

Result **T_PSXI_WALI = −1000**, **T = −1000** or **t_psxi_WaLi = −1000** for input values:

p > 10 bar or p < 0.00074 bar or
at calculation result t > 210 °C oder t < 0 °C oder
$\xi < \xi_{sol}(t)$
$\xi$ > 1.0 kg/kg or $\xi$ < 0.3 kg/kg

**References:** [1], [2], [4]

# Saturation Temperature $t_S = f(p_S, \xi^l)$

Function Name: **ts_psxil_WaLi**

Subroutine with function value: **REAL*8 FUNCTION TS_PSXIL_WALI(PS,XIL)**
for call from Fortran REAL*8 P,XIL

## Input Values:

**PS** - Vapor pressure $p_S$ in bar

**XIL** - Mass fraction of water as saturated liquid $\xi^l$ in (kg $H_2O$)/(kg mixture)

## Result

**TS_PSXIL_WALI**, **TS** or **ts_psxil_WaLi** - Saturation temperature $t_S$ in °C

## Range of validity

Pressure range: from 0.00074 bar to 10 bar

Composition range: from 0.3 to 1.0 (kg $H_2O$)/(kg mixture)

## Results for wrong input values

Result **TS_PSXIL_WALI = −1000**, **TS = −1000** or **ts_psxil_WaLi = −1000** for input values:

$p_S > 10$ bar or $p_S < 0.00074$ bar or

at calculation result $t_S > 210$ °C or $t_S < 0$ °C or

$\xi < \xi_{sol}(t)$

$\xi^l > 1.0$ kg/kg or $\xi^l < 0.3$ kg/kg

**References:** [1], [2], [4]

# Temperature at the Crystallization Barrier $t_{sol} = f(p)$

Function Name: **tsol_p_WaLi**

Subroutine with function value: **REAL*8 FUNCTION TSOL_P_WALI(P)**
for call from Fortran REAL*8 P

## Input Values:

**P** - Pressure p in bar

## Result

**TSOL_P_WALI**, **TSOL** or **tsol_p_WaLi** - Temperature at the crystallization barrier $t_{sol}$ in °C

## Range of validity

Pressure range: from 0.00074 bar to 0.0373 bar

## Explanation of the function

This function calculates the temperature at the crystallization barrier. If the actual temperature calculated in the liquid phase of the mixture lies above the temperature calculated by the function then the Lithium Bromide crystallizes out. This case is recognised by the other functions. A calculation of properties below the crystallization barrier is not possible. Please note that this function puts out expedient results only if a liquid phase is calculated since there is pure water in the gas phase.

## Results for wrong input values

Result **TSOL_P_WALI, = −1000**, **TSOL = −1000** or **tsol_p_WaLi = −1000** for input values:

p < 0.0373 bar or p > 0.00074 bar

**References:** [1], [2], [4]

# Specific Volume v = f(p,t,ξ)

Function Name: **v_ptxi_WaLi**

Subroutine with function value: **REAL*8 FUNCTION V_PTXI_WALI(P,T,XI)**
for call from Fortran REAL*8 P,T,XI

## Input Values:

**P** - Pressure p in bar

**T** - Temperature t in °C

**XI** - Mass fraction of water ξ in (kg $H_2O$)/(kg mixture)

## Result

**V_PTXI_WALI**, **V** or **v_ptxi_WaLi** - Specific volume v in $m^3$/kg

## Range of validity

Temperature range: from 0 °C to 210 °C

Pressure range: from 0.00074 bar to 10 bar

Composition range: from 0.3 to 1.0 (kg $H_2O$)/(kg mixture)

## Hints for wet steam

The wet steam region is calculated automatically by the subprograms:

## Results for wrong input values

Result **V_PTXI_WALI = −1000**, **V = −1000** or **v_ptxi_WaLi = −1000** for input values:

p > 10 bar or p < 0.00074 bar or
t > 210 °C or t < 0 °C oder
$\xi < \xi_{sol}(t)$

ξ > 1.0 kg/kg or ξ < 0.3 kg/kg

**References:** [1], [2], [4]

# Specific Volume of Saturated Liquid $v^l = f(p_s, t_s, \xi^l)$

Function Name: **vl_pstsxil_WaLi**

Subroutine with function value: **REAL*8 FUNCTION VL_WALI(PS,TS,XIL)**
for call from Fortran REAL*8 PS,TS,XIL

## Input Values:

**PS** - Vapor pressure $p_s$ in bar

**TS** - Saturation temperature $t_s$ in °C

**XIL** - Mass fraction of water as saturated liquid $\xi^l$ in (kg $H_2O$)/(kg mixture)

## Result

**VL_WALI**, **VL** or **vl_pstsxil_WaLi** - Specific volume of saturated liquid $v^l$ in $m^3$/kg

## Range of validity

Temperature range: from 0 °C to 210 °C

Pressure range: from 0.00074 bar to 10 bar

Composition range: from 0.3 to 1.0 (kg $H_2O$)/(kg mixture)

## Hints of the input variants

For calculating the specific volume of the saturated liquid it is necessary to define two parameters (either $t_S$ and $\xi'$ or $p_S$ and $\xi'$ or $p_S$ and $t_S$) and –1 must be entered as a pro-forma value for the value which is not given. When $p_S$, $t_S$, and $\xi'$ are entered, the program considers $p_S$, $t_S$, and $\xi'$ to be appropriate to represent the vapor pressure curve. If it is not the case the calculation for the quantity of the chosen function to be calculated results in –1000.

Possible input variants:

$$v^l = f(-1, t_s, \xi^l)$$
$$v^l = f(p_s, -1, \xi^l)$$
$$v^l = f(p_s, t_s, -1)$$
$$v^l = f(p_s, t_s, \xi^l)$$

## Results for wrong input values

Result **VL_WALI = −1000**, **VL = −1000** or **vl_pstsxil_WaLi = −1000** for input values:

$p_s > 10$ bar or $p_s < 0.00074$ bar or

$t_s > 210$ °C or $t_s < 0$ °C or

$t_s > t_{sol}(p_s)$ or

$\xi^l > 1.0$ kg/kg or $\xi^l < 0.3$ kg/kg

**References:** [1], [2], [4]

# Specific Volume of Saturated Steam $v^v = f(p_s, t_s, \xi^l)$

Function Name: **vv_pstsxil_WaLi**

Subroutine with function value: **REAL*8 FUNCTION VV_WALI(PS,TS,XIL)**
for call from Fortran REAL*8 PS,TS,XIL

## Input Values:

**PS** - Vapor pressure $p_s$ in bar

**TS** - Saturation temperature $t_s$ in °C

**XIL** - Mass fraction of water as saturated liquid $\xi^l$ in (kg $H_2O$)/(kg mixture)

## Result

**VV_WALI**, **VV** or **vv_pstsxil_WaLi** - Specific volume of saturated steam $v^v$ in $m^3$/kg

## Range of validity

Temperature range: from 0 °C to 210 °C

Pressure range: from 0.00074 bar to 10 bar

Composition range: from 0.3 to 1.0 (kg $H_2O$)/(kg mixture)

## Hints of the input variants

For calculating the specific volume of saturated steam it is necessary to define two parameters (either $t_S$ and $\xi'$ or $p_S$ and $\xi'$ or $p_S$ and $t_S$) and –1 must be entered as a pro-forma value for the value which is not given. When $p_S$, $t_S$, and $\xi'$ are entered, the program considers $p_S$, $t_S$, and $\xi'$ to be appropriate to represent the vapor pressure curve. If it is not the case the calculation for the quantity of the chosen function to be calculated results in –1000.

Possible input variants:

$$v^v = f(-1, t_s, \xi^l)$$
$$v^v = f(p_s, -1, \xi^l)$$
$$v^v = f(p_s, t_s, -1)$$
$$v^v = f(p_s, t_s, \xi^l)$$

## Results for wrong input values

Result **VV_WALI = −1000**, **VV = −1000** or **vv_pstsxil_WaLi = −1000** for input values:

$p_s > 10$ bar or $p_s < 0.00074$ bar or

$t_s > 210$ °C or $t_s < 0$ °C or

$t_s > t_{sol}(p_s)$ or

$\xi^l > 1.0$ kg/kg or $\xi^l < 0.3$ kg/kg

**References:** [1], [2], [4]

# Mass Fraction $H_2O$ of Saturated Liquid $\xi^l = f(p_s, t_s)$

Function Name: **xil_psts_WaLi**

Subroutine with function value: **REAL*8 FUNCTION XIL_PSTS_WALI(PS,TS)**
for call from Fortran REAL*8 PS,TS

## Input Values:

**PS** - Vapor pressure $p_s$ in bar

**TS** - Saturation temperature $t_s$ in °C

## Result

**XIL_PSTS_WALI**, **XIL** or **xil_psts_WaLi** - Mass fraction $H_2O$ of saturated liquid $\xi^l$ in kg $H_2O$ / kg

## Range of validity

Temperature range: from 0 °C to 210 °C

Pressure range: from 0.00074 bar to 10 bar

## Results for wrong input values

Result **XIL_PSTS_WALI = −1000**, **XIL = −1000** or **xil_psts_WaLi = −1000** for input values:

$p_s > 10$ bar or $p_s < 0.00074$ bar or

$t_s > 210$ °C or $t_s < 0$ °C or

$t_s > t_{sol}(p_s)$ or

**References:** [1], [2], [4]

# Mass Fraction $H_2O$ at the Crystallization Barrier $\xi_{sol} = f(p)$

| | |
|---|---|
| Function Name: | **xisol_p_WaLi** |
| Subroutine with function value:<br>for call from Fortran | **REAL*8 FUNCTION XISOL_P_WALI(P)**<br>REAL*8 P |

## Input Values:

**P** - Pressure p in bar

## Result

**XISOL_P_WALI**, **XISOL** or **xisol_p_WaLi** - Mass fraction $H_2O$ at the crystallization barrier $\xi_{sol}$ in kg $H_2O$ /kg

## Range of validity

Pressure range: from 0.00074 bar to 0.0373 bar

## Explanation of the function

This function calculates the mass fraction $H_2O$ at the crystallization barrier. If the actual mass fraction $H_2O$ calculated in the liquid phase of the mixture lies below the mass fraction calculated by the function then the Lithium Bromide crystallizes out. This case is recognised by the other functions. A calculation of properties below the crystallization barrier is not possible.

## Results for wrong input values

Result **XISOL_P_WALI, = −1000**, **XISOL = −1000** or **xisol_p_WaLi = −1000** for input values:

p < 0.0373 bar or p > 0.00074 bar

**References:** [1], [2], [4]

# Mass Fraction $H_2O$ at the Crystallization Barrier $\xi_{sol} = f(t)$

Function Name: **xisol_t_WaLi**

Subroutine with function value: **REAL*8 FUNCTION XISOL_T_WALI(T)**
for call from Fortran REAL*8 T

## Input Values:

**T** - Temperature t in °C

## Result

**XISOL_T_WALI**, **XISOL** or **xisol_t_WaLi** - Mass fraction $H_2O$ at the crystallization barrier $\xi_{sol}$ in kg $H_2O$ /kg

## Range of validity

Temperature range: from 0 °C to 93.58 °C

## Explanation of the function

This function calculates the mass fraction $H_2O$ at the crystallization barrier. If the actual mass fraction $H_2O$ calculated in the liquid phase of the mixture lies below the mass fraction calculated by the function then the Lithium Bromide crystallizes out. This case is recognised by the other functions. A calculation of properties below the crystallization barrier is not possible.

## Results for wrong input values

Result **XISOL_T_WALI = −1000**, **XISOL = −1000** or **xisol_t_WaLi = −1000** for input values:

t > 93.58 °C or t < 0 °C

**References:** [1], [2], [4]

# Mass Fraction $H_2O$ of Saturated Steam $\xi^v = f(p_s, t_s)$

Function Name: **xiv_psts_WaLi**

Subroutine with function value: **REAL*8 FUNCTION XIV_PSTS_WALI(PS,TS)**
for call from Fortran REAL*8 PS,TS

### Input Values:

**PS** - Vapor pressure $p_s$ in bar

**TS** - Saturation temperature $t_s$ in °C

### Result

**XIV_PSTS_WALI**, **XIV** or **xiv_psts_WaLi** - Mass fraction $H_2O$ of saturated steam $\xi^v$ in kg $H_2O$ / kg

### Range of validity

Temperature range: from 0 °C to 210 °C

Pressure range: from 0.00074 bar to 10 bar

### Explanation of the function

This function calculates the mass fraction of water in the gas phase. Since only very few Lithium Bromide enters the gas phase, the program library acts on the assumption that there is only water in the gas phase. This is also the case for all other functions. For there is only water in the saturated steam, the function will always calculate a result of – 1 within the range of validity.

### Results for wrong input values

Result **XIV_PSTS_WALI = −1000**, **XIV = −1000** or **xiv_psts_WaLi = −1000** for input values:

$p_s > 10$ bar or $p_s < 0.00074$ bar or

$t_s > 210$ °C or $t_s < 0$ °C or

**References:** [1], [2]

KCE ThermoFluidProperties

# KCE-ThermoFluidProperties
# www.thermofluidprop.com

Hochschule Zittau/Görlitz
UNIVERSITY OF APPLIED SCIENCES

## Property Libraries for Calculating Heat Cycles, Boilers, Turbines and Refrigerators

### Water and Steam

**Library LibIF97**

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

**Library LibIF97_META**

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

### Humid Combustion Gas Mixtures

**Library LibHuGas**

Model: Ideal mixture of the real fluids:

$CO_2$ - Span, Wagner  $H_2O$ - IAPWS-95
$O_2$ - Schmidt, Wagner  $N_2$ - Span et al.
Ar - Tegeler et al.

and of the ideal gases:

$SO_2$, CO, Ne
(Scientific Formulation of Bücker et al.)

Consideration of:

- Dissociation from VDI 4670
- Poynting effect

### Humid Air

**Library LibHuAir**

Model: Ideal mixture of the real fluids:

- Dry air from Lemmon et al.
- Steam, water and ice from IAPWS-IF97 and IAPWS-06

Consideration of:

- Condensation and freezing of steam
- Dissociation from VDI 4670
- Poynting effect from ASHRAE RP-1485

### Extremely Fast Property Calculations

Spline-Based Table Look-up Method (SBTL)

**Library LibSBTL_IF97**
**Library LibSBTL_95**
**Library LibSBTL_HuAir**

For steam, water, humid air, carbon dioxide and other fluids and mixtures according IAPWS Guideline 2015 for Computational Fluid Dynamics (CFD), real-time and non-stationary simulations

### Carbon Dioxide Including Dry Ice

**Library LibCO2**

Formulation of Span and Wagner (1996)

### Seawater

**Library LibSeaWa**

IAPWS Industrial Formulation 2013

### Ice

**Library LibICE**

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

### Ideal Gas Mixtures

**Library LibIdGasMix**

Model: Ideal mixture of the ideal gases:

| | | | |
|---|---|---|---|
| Ar | NO | He | Propylene |
| Ne | $H_2O$ | $F_2$ | Propane |
| $N_2$ | $SO_2$ | $NH_3$ | Iso-Butane |
| $O_2$ | $H_2$ | Methane | n-Butane |
| CO | $H_2S$ | Ethane | Benzene |
| $CO_2$ | OH | Ethylene | Methanol |
| Air | | | |

Consideration of:

- Dissociation from the VDI Guideline 4670

**Library LibIDGAS**

Model: Ideal gas mixture from VDI Guideline 4670

Consideration of:

- Dissociation from the VDI Guideline 4670

### Humid Air

**Library ASHRAE LibHuAirProp**

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

Consideration of:

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

www.ashrae.org/bookstore

### Dry Air Including Liquid Air

**Library LibRealAir**

Formulation of Lemmon et al. (2000)

### Refrigerants

**Ammonia**

**Library LibNH3**

Formulation of Tillner-Roth et al. (1993)

**R134a**

**Library LibR134a**

Formulation of Tillner-Roth and Baehr (1994)

**Iso-Butane**

**Library LibButane_Iso**

Formulation of Bücker and Wagner (2006)

**n-Butane**

**Library LibButane_n**

Formulation of Bücker and Wagner (2006)

### Mixtures for Absorption Processes

**Ammonia/Water Mixtures**

**Library LibAmWa**

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

Helmholtz energy equation for the mixing term

(also useable for calculating the Kalina Cycle)

**Water/Lithium Bromide Mixtures**

**Library LibWaLi**

Formulation of Kim and Infante Ferreira (2004)

Gibbs energy equation for the mixing term

### Liquid Coolants

**Liquid Secondary Refrigerants**

**Library LibSecRef**

Liquid solutions of water with

| | |
|---|---|
| $C_2H_6O_2$ | Ethylene glycol |
| $C_3H_8O_2$ | Propylene glycol |
| $C_2H_5OH$ | Ethanol |
| $CH_3OH$ | Methanol |
| $C_3H_8O_3$ | Glycerol |
| $K_2CO_3$ | Potassium carbonate |
| $CaCl_2$ | Calcium chloride |
| $MgCl_2$ | Magnesium chloride |
| NaCl | Sodium chloride |
| $C_2H_3KO_2$ | Potassium acetate |
| $CHKO_2$ | Potassium formate |
| LiCl | Lithium chloride |
| $NH_3$ | Ammonia |

Formulation of the International Institute of Refrigeration (IIR 2010)

### Ethanol

**Library LibC2H5OH**

Formulation of
Schroeder et al. (2014)

### Methanol

**Library LibCH3OH**

Formulation of
de Reuck and Craven (1993)

### Propane

**Library LibPropane**

Formulation of
Lemmon et al. (2009)

### Siloxanes as ORC Working Fluids

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

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

Tetradecamethylhexasiloxane $C_{14}H_{42}O_5Si_6$ **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 $H_2S$ **Library LibH2S**

Nitrous oxide $N_2O$ **Library LibN2O**

Sulfur dioxide $SO_2$ **Library LibSO2**

Acetone $C_3H_6O$ **Library LibC3H6O**

Formulation of Lemmon and Span (2006)

KCE
ThermoFluidProperties


**For more information please contact:**

KCE-ThermoFluidProperties UG & Co. KG
Prof. Dr. Hans-Joachim Kretzschmar
Wallotstr. 3
01307 Dresden, Germany

Internet: www.thermofluidprop.com
Email: info@thermofluidprop.com
Phone: +49-351-27597860
Mobile: +49-172-7914607
Fax: +49-3222-1095810


## The following thermodynamic and transport properties can be calculated[a]:

### Thermodynamic Properties

- Vapor pressure $p_s$
- Saturation temperature $T_s$
- Density $\rho$
- 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 $Pr$
- Thermal diffusivity $a$

### Backward Functions

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

### Thermodynamic Derivatives

- Partial derivatives used in process modeling can be calculated.

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

**www.thermofluidprop.com**

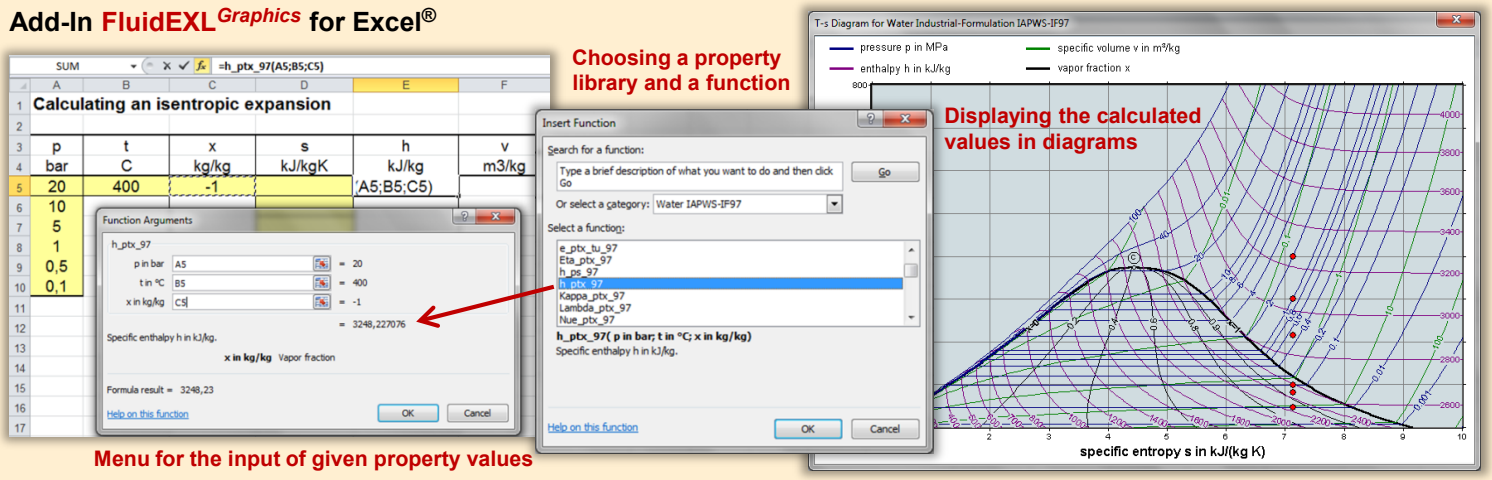
# KCE-ThermoFluidProperties
# www.thermofluidprop.com

KCE ThermoFluidProperties

Hochschule Zittau/Görlitz
UNIVERSITY OF APPLIED SCIENCES

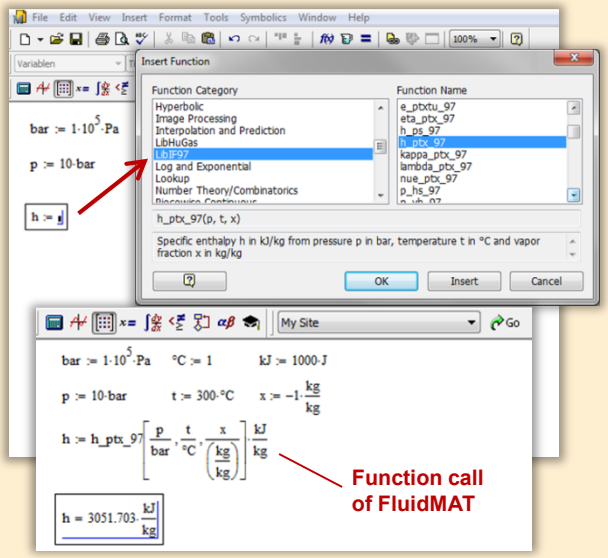
## Property Software for Calculating Heat Cycles, Boilers, Turbines and Refrigerators

### Add-In FluidEXL*Graphics* for Excel®



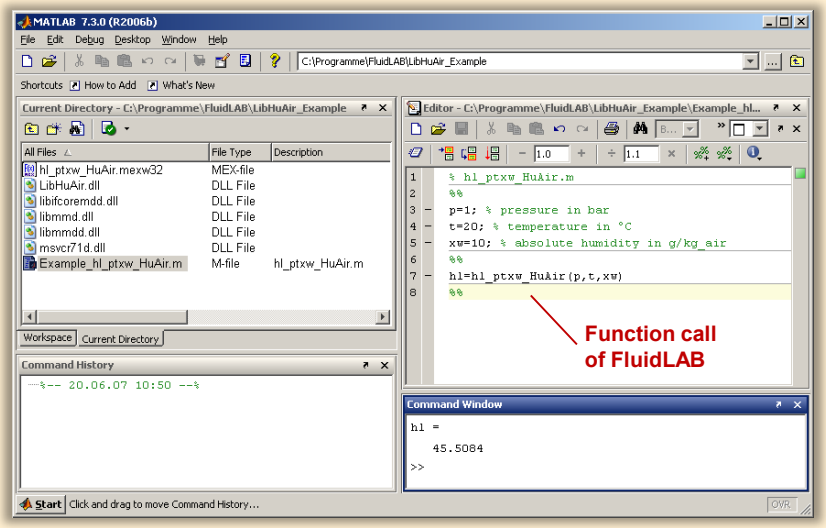

### Add-On FluidMAT for Mathcad®
### Add-On FluidPRIME for Mathcad Prime®

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



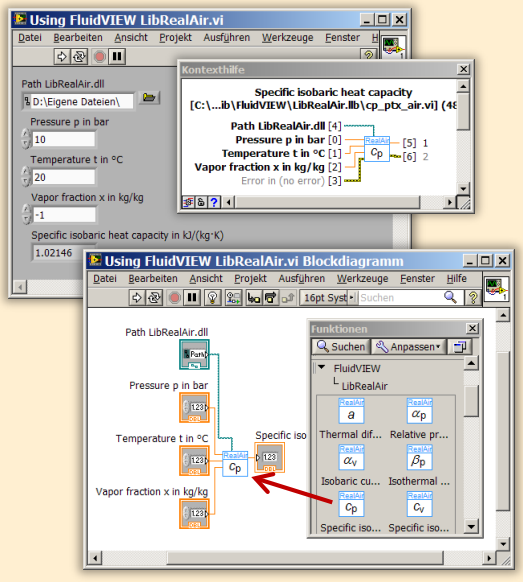

### Add-On FluidLAB for MATLAB® and SIMULINK®

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



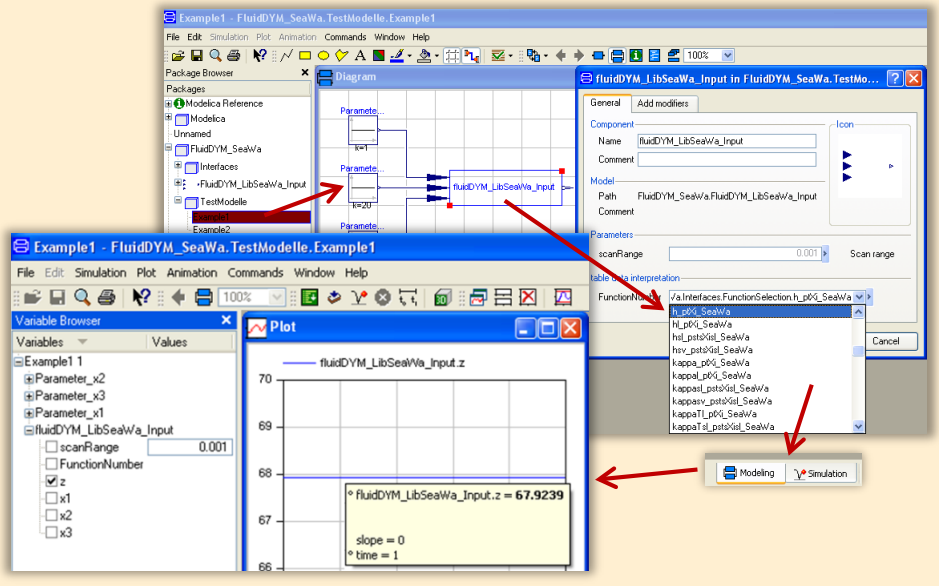

### Add-On FluidVIEW for LabVIEW™

The property functions can be calculated in LabVIEW™.

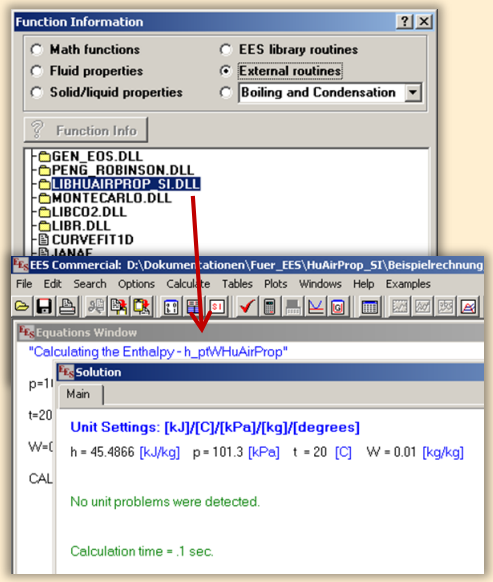



### Add-On FluidDYM for DYMOLA® (Modelica) and SimulationX®

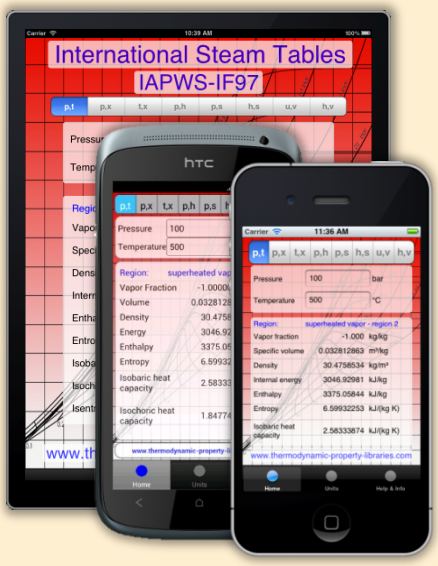
The property functions can be called in DYMOLA® and SimulationX®.



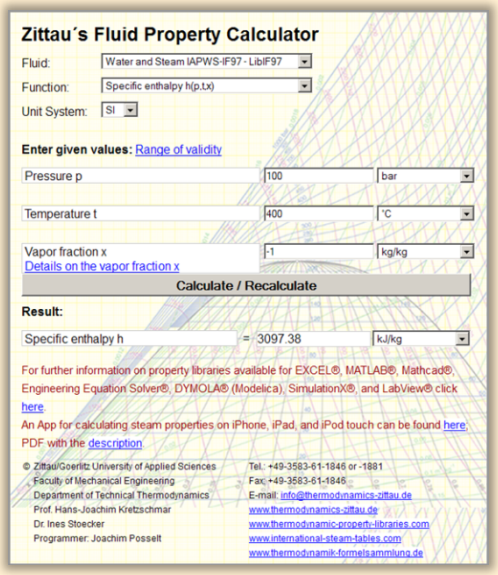
## Add-On FluidEES for Engineering Equation Solver®




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




## Online Property Calculator at www.thermofluidprop.com




## Property Software for Pocket Calculators

### 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 Nspire CAS

TI 83
TI 84
TI 89

TI Voyage 200

TI 92

## For more information please contact:

KCE ThermoFluidProperties

KCE-ThermoFluidProperties UG & Co. KG
Prof. Dr. Hans-Joachim Kretzschmar
Wallotstr. 3
01307 Dresden, Germany

Internet: www.thermofluidprop.com
Email: info@thermofluidprop.com
Phone: +49-351-27597860
Mobile: +49-172-7914607
Fax: +49-3222-1095810

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

### Thermodynamic Properties

- Vapor pressure $p_s$
- Saturation temperature $T_s$
- Density $\rho$
- 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 $Pr$
- Thermal diffusivity $a$

### Backward Functions

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

### Thermodynamic Derivatives

- Partial derivatives used in process modeling can be calculated.

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

**www.thermofluidprop.com**

# 5. References

[1] Wagner, W.; Kruse, A.:
Zustandsgrößen von Wasser und Wasserdampf.
Springer-Verlag, Berlin (1998)

[2] Kim D. S.; Infante Ferreira, C. A.:
A Gibbs Energy Equation for LiBr/$H_2O$ Solutions.
Proceedings of the 6th IIR Gustav Lorentzen Conference on Natural Working Fluids, Glasgow (U.K.), 2004

[3] Kim D. S.; Infante Ferreira, C. A.:
Solar Absorption Cooling.
1st progress report for NOVEM, Oct, 2003

[4] Vollmer, D.:
Programm zur Berechnung thermodynamischer Eigenschaften des Arbeitsstoffpaares Wasser/Lithiumbromid.
Institut für Luft- und Kältetechnik Dresden, intern

# 6. Satisfied Customers

Period from 2018 to 2022

The following companies and institutions use the property libraries:

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

**2022**

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

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

**2021**

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

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

**2020**

| | |
|---|---|
| Drill Cool, Bakersfield CA, USA | 12/2020 |
| Manders, The Netherlands | |
| RWE Supply & Tranding, Essen | |
| NEOWAT Lodz, Poland | |
| University of Duisburg-Essen, Duisburg | 11/2020 |
| Stellenbosch University, South Africa | |

| | |
|---|---|
| University De France-COMTe, France | |
| RWE, Essen | |
| STEAG, Herne | |
| Isenmann Ingenierbüro | |
| University of Stuttgart, ITLR, Stuttgart | |
| Norsk Energi, Oslo, Norway | |
| TGM Kanis, Nürnberg | |
| Stadtwerke Neuburg | 10/2020 |
| Smurfit Kappa, Roermond, The Netherlands | |
| RWE, Essen | |
| Hochschule Zittau/Görlitz, Wirtschaftsingenieurwesen | |
| Stadtwerke, Neuburg | |
| ILK, Dresden | |
| ATESTEO, Alsdorf | |
| Hochschule Zittau/Görlitz, Maschinenwesen | |
| TH Nürnberg, Verfahrenstechnik | |
| Drill Cool, Bakersfield CA,USA | 09/2020 |
| RWE, Essen | |
| 2Meyers Ingenieurbüro, Nürnberg | |
| FELUWA, Mürlenbach | |
| Stadtwerke Neuburg | |
| Caverion, Wien, Austria | |
| GMVA Niederrhein, Oberhausen | |
| INWAT Lodz, Poland | |
| Troche Ingenieurbüro, Hayingen | 08/2020 |
| CEA Saclay, France | |
| VPC, Vetschau | 07/2020 |
| FSK System-Kälte-Klima, Dortmund | |
| Exergie Etudes, Sarl, Switzerland | |
| AWG Wuppertal | |
| STEAG Energy Services, Zwingenberg | |
| Hochschule Braunschweig | 06/2020 |
| DBI, Leipzig | |
| GOHL-KTK, Dumersheim | |
| TU Dresden, Energieverfahrenstechnik | |
| BASF SE, ESI/EE, Ludwigshafen | |
| Wärme Hamburg | |
| Ruchti Ingenieurbüro, Uster, Switzerland | |
| IWB, Basel, Switzerland | |
| Midiplan, Bietingen-Bissingen | 05/2020 |
| Knieschke, Ingenieurbüro | |
| RWE, Essen | |
| Leser, Hamburg | |
| AGRANA, Gmünd, Austria | |
| EWT Wassertechnik, Celle | |
| Hochschule Darmstadt | 04/2020 |
| MTU München CCP | |
| HAW Hamburg | 03/2020 |

| | |
|---|---|
| Hanon, Novi Jicin, Czech Republic | |
| TU Dresden, Kältetechnik | |
| MAN, Copenhagen, Denmark | |
| EnerTech, Radebeul | 02/2020 |
| LEAG, Cottbus | |
| B+B Enginering Magdeburg | |
| Hochschule Offenburg | |
| WIB, Dennheritz | 01/2020 |
| Universität Duisburg-Essen, Strömungsmaschinen | |
| Kältetechnik Dresen-Bremen | |
| TH Ingolstadt | |
| Vattenfall AB, Jokkmokk, Sweden | |
| Fraunhofer UMSICHT | |

**2019**

| | |
|---|---|
| PEU Leipzig, Rötha | 12/2019 |
| MB-Holding, Vestenbergsgreuth | |
| RWE, Essen | |
| Georg-Büchner-Hochschule, Darmstadt | 11/2019 |
| EEB ENERKO, Aldenhoven | |
| Robert Benoufa Energietechnik, Wiesloch | |
| Kehrein & Kubanek Klimatechnik, Moers | 10/2019 |
| Hanon Systems Autopal Services, Hluk, Czech Republic | |
| CEA Saclay, Gif Sur Yvette cedex, France | |
| Saudi Energy Efficiency Center SEEC, Riyadh, Saudi Arabia | |
| VPC, Vetschau | 09/2019 |
| jGanser PM + Engineering, Forchheim | |
| Endress+Hauser Flowtec AG, Reinach, Switzerland | |
| Ruchti IB, Uster, Switzerland | |
| ZWILAG Zwischenlager Würenlingen, Switzerland | 08/2019 |
| Hochschule Zittau/Görlitz, Faculty Maschinenwesen | |
| Stadtwerke Neubrandenburg | |
| Physikalisch Technische Bundesanstalt PTB, Braunschweig | |
| GMVA Oberhausen | 07/2019 |
| Endress+Hauser Flowtec AG, Reinach, Switzerland | |
| WARNICA, Waterloo, Canada | |
| MIBRAG, Zeitz | 06/2019 |
| Pöyry, Zürich, Switzerland | |
| RWTH Aachen, Institut für Strahlantriebe und Turbomaschinen | |
| Midiplan, Bietigheim-Bissingen | |
| GKS Schweinfurt | |
| HS Zittau/Görlitz, Wirtschaftswissenschaften und Wirtschaftsingenieurwesen | |
| ILK Dresden | |
| HZDR Helmholtz Zentrum Dresden-Rossendorf | |
| TH Köln, Technische Gebäudeausrüstung | 05/2019 |
| IB Knittel, Braunschweig | |
| Norsk Energi, Oslo, Norway | |

| | |
|---|---|
| STEAG, Essen | |
| Stora Enso, Eilenburg | |
| IB Lücke, Paderborn | |
| Haarslev, Sonderso, Denmark | |
| MAN Augsburg | |
| Wieland Werke, Ulm | 04/2019 |
| Fels-Werke, Elbingerode | |
| Univ. Luxembourg, Luxembourg | |
| BTU Cottbus, Power Engineering | 03/2009 |
| Eins-Energie Sachsen, Schwarzenberg | |
| TU Dresden, Kälte- und Kryotechnik | |
| ITER, St. Paul Lez Durance Cedex, France | |
| Fraunhofer UMSICHT, Oberhausen | |
| Comparex Leipzig for Spedition Thiele HEMMERSBACH | |
| Rückert NaturGas, Lauf/Pegnitz | |
| BASF, Basel, Switzerland | 02/2019 |
| Stadtwerke Leipzig | |
| Maerz Ofenbau Zürich, Switzerland | |
| Hanon Systems Germany, Kerpen | |
| Thermofin, Heinsdorfergrund | 01/2019 |
| BSH Berlin | |

**2018**

| | |
|---|---|
| Jaguar Energy, Guatemala | 12/2018 |
| WEBASTO, Gilching | |
| Smurfit Kappa, Oosterhout, Netherlands | |
| Univ. BW München | |
| RAIV, Liberec for VALEO, Prague, Czech Republic | 11/2018 |
| VPC Group Vetschau | |
| SEITZ, Wetzikon, Switzerland | |
| MVV, Mannheim | 10/2018 |
| IB Troche | |
| KANIS Turbinen, Nürnberg | |
| TH Ingolstadt, Institut für neue Energiesysteme | |
| IB Kristl & Seibt, Graz, Austria | 09/2018 |
| INEOS, Köln | |
| IB Lücke, Paderborn | |
| Südzucker, Ochsenfurt | 08/2018 |
| K&K Turbinenservice, Bielefeld | 07/2018 |
| OTH Regensburg, Elektrotechnik | |
| Comparex Leipzig for LEAG, Berlin | 06/2018 |
| Münstermann, Telgte | 05/2018 |
| TH Nürnberg, Verfahrenstechnik | |
| Universität Madrid, Madrid, Spanien | |
| HS Zittau/Görlitz, Wirtschaftsingenieurwesen | |
| HS Niederrhein, Krefeld | |
| Wilhelm-Büchner HS, Pfungstadt | 03/2018 |

| | |
|---|---|
| GRS, Köln | |
| WIB, Dennheritz | |
| RONAL AG, Härklingen, Schweiz | 02/2018 |
| Ingenieurbüro Leipert, Riegelsberg | |
| AIXPROCESS, Aachen | |
| KRONES, Neutraubling | |
| Doosan Lentjes, Ratingen | 01/2018 |