



# **TILMedia Suite 3**

**TLK-Thermo GmbH**

**in cooperation with**

**Institut für Thermodynamik**

**Technische Universität Braunschweig**



# TIL Media Substance properties optimized for stable and extremely fast dynamic simulations

- Calculation methods to express thermophysical properties of:



Incompressible Liquids



Ideal Gases



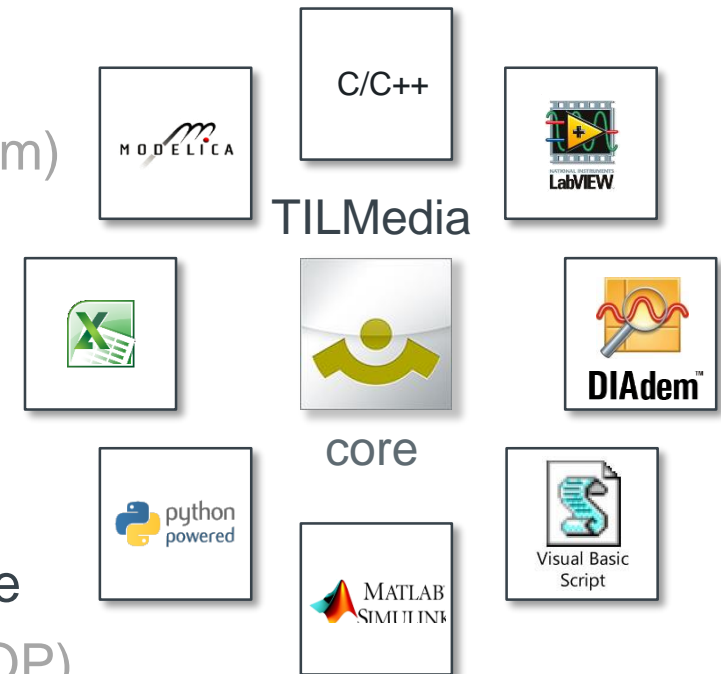
Real Fluids (with vapor liquid equilibrium)

- Mixtures

- Optimized mathematical equations with extremely high calculation speeds and high accuracies

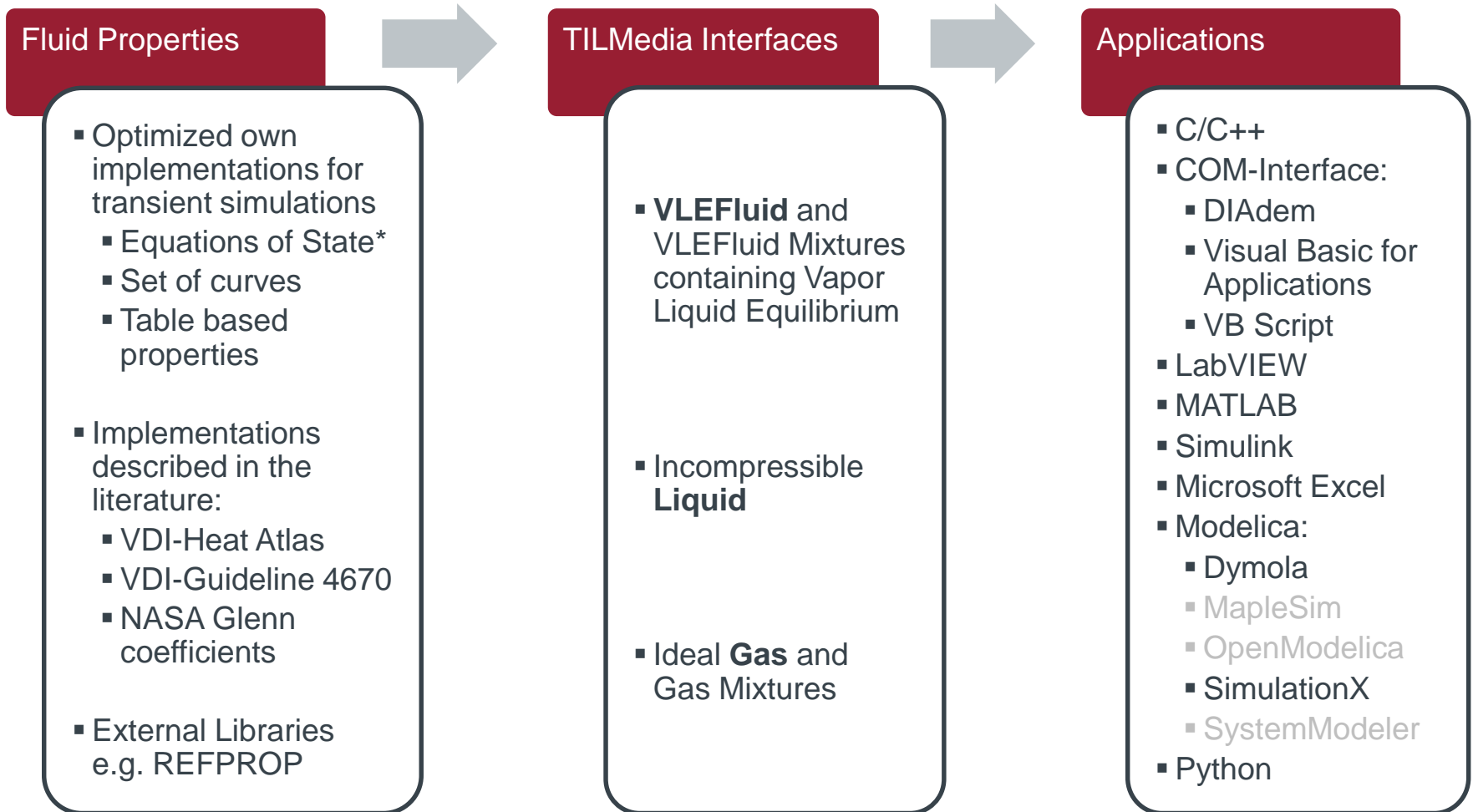
- Several hundreds of substances available (also from external sources e.g. REFPROP)

- TILMedia Suite interfaces one property core for various software





# TILMedia as Interface



\* In the following = EOS

▪ Not implemented jet



# VLEFluids Available thermophysical properties

Interface for fluids containing a Vapor Liquid Equilibrium

The following VLEFluid properties are provided:

- density
- specific enthalpy
- pressure
- specific entropy
- temperature
- mass fraction
- mole fraction
- average molar mass
- steam mass fraction (quality)
- specific isobaric/isochoric heat capacity
- isobaric thermal expansion coefficient
- isothermal compressibility
- speed of sound
- derivative of density with respect to specific enthalpy
- derivative of density with respect to pressure
- derivative of density with respect to mass fraction
- heat capacity ratio / isentropic expansion factor
- molar mass of one component in mixture
- prandtl number
- thermal conductivity
- dynamic viscosity
- surface tension
- Saturation properties, critical properties & properties on dew and bubble line

As a function of:

- $p, h, x_i$
- $p, T, x_i$
- $p, s, x_i$
- $d, T, x_i$

with  $x_i=1$  for a single component



# Available pure VLEFluids – EOS

## Approaches optimized for simulation with EOS

- faster than REFPROP
- not exportable
- moderate memory requirement
- very good accuracy of calculations
- Ammonia
- Argon
- CO<sub>2</sub> (Span/Wagner and GERG)
- Ethanol
- Ethylbenzene
- M-Xylene
- Nitrogen
- Oxygen
- O-Xylene
- Propane
- P-Xylene
- R1234yf
- R125
- R134a (Tillner-Roth/Baehr and Astina/Sato)
- R143a
- R245fa
- R32
- R404A (PPF)
- R407C (PPF)
- R410A (PPF)
- R507A (PPF)
- Water (IAPWS 1995)



# Available pure VLEFluids

## Implementations with sets of curves:

- very fast
- exportable
- low memory requirement
- good accuracy of calculations

- CO<sub>2</sub>
- R1234yf
- R134a
- R407C
- R410A
- Water

## Table based thermophysical properties\*:

- very fast
- exportable
- high memory requirement
- very good accuracy of calculations

- Air (PPF)
- Methane
- R134a
- Table based properties of all REFPROP-Fluids possible on demand.

\* Only available with the modelica library TIL



# Available VLEFluids – REFPROP

## REFPROP (208 media, pure fluids and mixtures):

▪ 1-Butene	▪ DEE	▪ Krypton	▪ Novec7000	▪ R-115	▪ R-407B	▪ R-423A	▪ R-508B
▪ Acetone	▪ Decane	▪ (R-784)	▪ Octane	▪ R-116	▪ R-407C	▪ R-424A	▪ R-509A
▪ Air	▪ DMC	▪ MDN	▪ Orthohydrogen	▪ R-123	▪ R-407D	▪ R-425A	▪ R-510A
▪ Amarillo	▪ DME	▪ MD2M	▪ (R-702)	▪ R-124	▪ R-407E	▪ R-426A	▪ R-512A
▪ Ammonia	▪ Ebenzene	▪ MD3N	▪ Oxygen	▪ R-125	▪ R-407F	▪ R-427A	▪ (Lemy 134)
▪ Argon (R-740)	▪ Ekofisk	▪ MD4N	▪ (R-732)	▪ R-134a	▪ R-408A	▪ R-428A	▪ R-744
▪ Benzene	▪ Ethane	▪ Methane	▪ Oxylene	▪ R-141b	▪ R-409A	▪ R-429A	▪ R-1216
▪ Butane	▪ (R-170)	▪ (R-50)	▪ Parahydrogen	▪ R-142b	▪ R-409B	▪ R-430A	▪ R-1233zd
▪ (R-600)	▪ Ethanol	▪ Methanol	▪ (R-702p)	▪ R-143a	▪ R-410A	▪ R-431A	▪ R-1234yf
▪ C1-CC6	▪ Ethylene	▪ Methyl	▪ Pentane	▪ R-152a	▪ R-410B	▪ R-432A	▪ R-1234ze
▪ C2-Butene	▪ (R-1150)	▪ Linoleate	▪ (R-601)	▪ R-161	▪ R-411A	▪ R-433A	▪ R-C318
▪ C3-CC6	▪ Fluorine	▪ Methyl	▪ Propane	▪ R-218	▪ R-411B	▪ R-434A	▪ RE143a
▪ C4-F10	▪ Gifcoast	▪ Linolenate	▪ (R-290)	▪ R-227ea	▪ R-412A	▪ R-435A	▪ (HFE-143m)
▪ (R-3-1-10)	▪ H2S	▪ MM	▪ Propylene	▪ R-236ea	▪ R-413A	▪ R-436A	▪ RE245cb2
▪ C5-F12	▪ HCL	▪ Methyl Oleate	▪ (R-1270)	▪ R-236fa	▪ R-414A	▪ R-436B	▪ (HFE-245cb2)
▪ (R-4-1-12)	▪ Helium	▪ Methyl	▪ Propyne	▪ R-245ca	▪ R-414B	▪ R-437A	▪ RE245fa2
▪ C11	▪ (R-704)	▪ Palmitate	▪ P-Xylene	▪ R-245fa	▪ R-415A	▪ R-438A	▪ (HFE-245fa2)
▪ C12	▪ Heptane	▪ Methyl	▪ R-11	▪ R-365mfc	▪ R-415B	▪ R-441A	▪ RE347mcc
▪ C-F3-I	▪ Hexane	▪ Stearate	▪ R-12	▪ R-401A	▪ R-416A	▪ R-442A	▪ (HFE-7000)
▪ CO	▪ HighCO2	▪ M-Xylene	▪ R-13	▪ R-401B	▪ R-417A	▪ R-443A	▪ SF6
▪ COS	▪ HighN2	▪ N2O (R-744A)	▪ R-14	▪ R-401C	▪ R-418A	▪ R-444A (AC5)	▪ SO2 (R-764)
▪ Cyclohexane	▪ Hydrogen	▪ Neon (R-720)	▪ R-21	▪ R-402A	▪ R-419A	▪ R-420A	▪ T2-Butene
▪ Cyclopropane	▪ (R-702)	▪ Neopentane	▪ R-22	▪ R-402B	▪ R-420A	▪ R-500	▪ Toluene
▪ D2	▪ I-Butene	▪ NF3	▪ R-23	▪ R-403A	▪ R-421A	▪ R-501	▪ Water
▪ D4	▪ I-Hexane	▪ Ngsample	▪ R-32	▪ R-403B	▪ R-421B	▪ R-502	▪ Xenon
▪ D5	▪ I-Octane	▪ Nitrogen	▪ R-40	▪ R-404A	▪ R-422A	▪ R-503	
▪ D6	▪ I-Pentane	▪ (R-728)	▪ R-41	▪ R-405A	▪ R-422B	▪ R-504	
▪ D2O	▪ (R-601a)	▪ Nonane	▪ R-113	▪ R-406A	▪ R-422C	▪ R-507A	
	▪ Isobutanol	▪ Novec649	▪ R-114	▪ R-407A	▪ R-422D	▪ R-508A	

Table based properties (high speed) can be created of all 208 REFPROP-Fluids on demand.



# VLEFluid mixtures

## Variable mixture calculations:

Fundamental EOS (very detailed, designed for transient simulations):

- Ammonia and Water (Tillner-Roth & Friend)

Cubic EOS:

- Argon
- CO<sub>2</sub>
- Ethanol
- Hydrogen
- Nitrogen
- Oxygen
- R134a
- Water
- and many more (VDI-Wärmeatlas)

All 208 REFPROP-Fluids with each other





# Liquids Available thermophysical properties

Interface for incompressible liquids

The following liquid properties are provided:

- density
- specific enthalpy
- pressure
- specific entropy
- temperature
- specific isobaric heat capacity
- isobaric thermal expansion coefficient
- derivative of density with respect to specific enthalpy
- prandtl number
- thermal conductivity
- dynamic viscosity

As a function of:

- $p, h$
- $p, T$



# Available Liquids

Polynomial fits, 1-dimensional, temperature dependent:

- Addinol XW15
- Glysantin (30%-60%)
- Oil Aral 0W30
- Oil 15W40
- Propylenglykol (30%-50%)
- SHC\_XMP320 (Synthetic gear oil)
- Therminol 59
- Therminol 66
- Therminol D12
- Tyfocor 30
- Tyfocor 45
- Tyfocorl 33
- Water
- Zitrec M10
- Zitrec M20

All liquid mediums listed in VDI-Heat Atlas



# Gases Available thermophysical properties

## Interface for gases and gas-vapor mixtures

The following gas properties are provided:

- density
- specific enthalpy
- pressure
- specific entropy
- temperature
- mass fraction
- mole fraction
- average molar mass
- derivative of density with respect to specific enthalpy
- derivative of density with respect to pressure
- derivative of density with respect to mass fraction
- partial pressure of components
- mass fraction of gasous condensing component
- specific isobaric/isochoric heat capacity
- isobaric thermal expansion coefficient
- isothermal compressibility
- speed of sound
- relative humidity
- prandtl number
- thermal conductivity
- dynamic viscosity
- Saturation properties

As a function of:

- $p, h, xi$
- $p, T, xi$
- $p, s, xi$

with  $xi=1$  for a single component



# Available pure gases

Approaches with EOS (high accuracy and fast):

- Dry Air
- Exhaust Gas
- Diesel Exhaust Gas

VDI-Guideline 4670:

- |            |         |                   |
|------------|---------|-------------------|
| ▪ Dry Air  | ▪ Argon | ▪ Carbon Dioxide  |
| ▪ Nitrogen | ▪ Neon  | ▪ Carbon Monoxide |
| ▪ Oxygen   | ▪ Water | ▪ Sulphur Dioxide |

All 275 mediums listed in VDI-Heat Atlas

All 2024 mediums listed by NASA Glenn coefficients



# Gas mixtures

Variable gas mixture calculations with independent library choice (All pure gases can be mixed with each other):

- Approaches with EOS
- VDI-Guideline 4670
- VDI-Heat Atlas (275 media)
- NASA Glenn coefficients (2024 media)



# Moist Air Available thermophysical properties

Interface for moist air (specialized form of gas mixture)

The following moist air properties are provided:

- density
- specific enthalpy
- pressure
- specific entropy
- temperature
- molar mass
- specific isobaric heat capacity
- specific isochoric heat capacity
- partial pressures
- (saturation) water mass fraction
- (saturation) water content
- (saturation) humidity ratio
- relative humidity
- specific enthalpy 1+x
- specific enthalpy of pure gas
- specific enthalpy of vaporisation
- specific enthalpy of desublimation
- prandtl number
- thermal conductivity
- dynamic viscosity
- freezing point
- speed of sound
- mass fraction
- gaseous mass fraction
- isobaric thermal expansion coefficient
- isothermal compressibility
- density derivative WRT mass fraction
- density derivative WRT pressure
- density derivative WRT specific enthalpy

As a function of:

- $p, h, xi$
- $p, s, xi$
- $p, T, xi$
- $p, T, \text{humRatio}$
- $p, T, \text{phi}$



# Available Moist Air

## Moist Air – TLK and IfT:

- Gas vapor mixture
- Heat capacity of water is considered as constant
- Condensing and ice build up realized by constant enthalpy of evaporation and constant enthalpy of fusion
- Transport properties equal to those of dry air

## VDI-Guideline 4670 for Moist Air and combustion gases:

- Gas vapor mixture
- Condensing and ice build up realized by temperature dependent enthalpy of evaporation and temperature dependent enthalpy of fusion
- Transport properties equal to those of dry air



# Example in Modelica

Input of e.g. medium name, enthalpy and pressure amongst others to calculate all thermophysical properties

Icon  
VLE<sup>ph</sup>  
VLEfluid...

General Advanced Add modifiers

Component

Name vleFluid1

Comment

Model

Path TILMedia.VLEFluid\_ph

Comment VLE-Fluid model describing super-critical, subcooled, superheated fluid including the vapour liquid equilibrium (p, h and xi as independent variables)

Parameters

vleFluidType	TILMedia.CO2	type record of the VLE fluid or VLE fluid mixture
computeTransportProperties	false	=true, if transport properties are calculated
computeVLEAdditionalProperties	false	Compute detailed vapour liquid equilibrium properties
computeVLETransportProperties	false	Compute detailed vapour liquid equilibrium transport properties
deactivateTwoPhaseRegion	false	Deactivate calculation of two phase region
h	h J/kg	Specific enthalpy
p	p Pa	Pressure
xi	vleFluidType.xi_default	Mass Fraction of Component i

OK Info Cancel

Modeling Simulation





# Example in MATLAB

## Anwendungserweiterung

- TILMediaMatlab321x64.dll
- TILMediaMatlab321Win32.dll

## C/C++ Header

- TILMediaHeader4Matlab.h
- portable.h

## Class

- VLEFluid.m
- MoistAir.m
- Liquid.m
- Gas.m

## Function

- loadTILMedia
- checkInputVar
- calcComputeF

% TILMedia-functions:

```
vle = VLEFluid();  
vle = vle.setVLEFluidType(vle_name,1);
```

% Calculation-loop for thermophysical properties:

```
for i=1:length(p_vle)  
    vle = vle.setState_phxi(p_vle(i)*ones(length(h_vle),1),h_vle);  
    T_vle = [T_vle,vle.T];  
    s_vle = [s_vle,vle.s];  
end
```

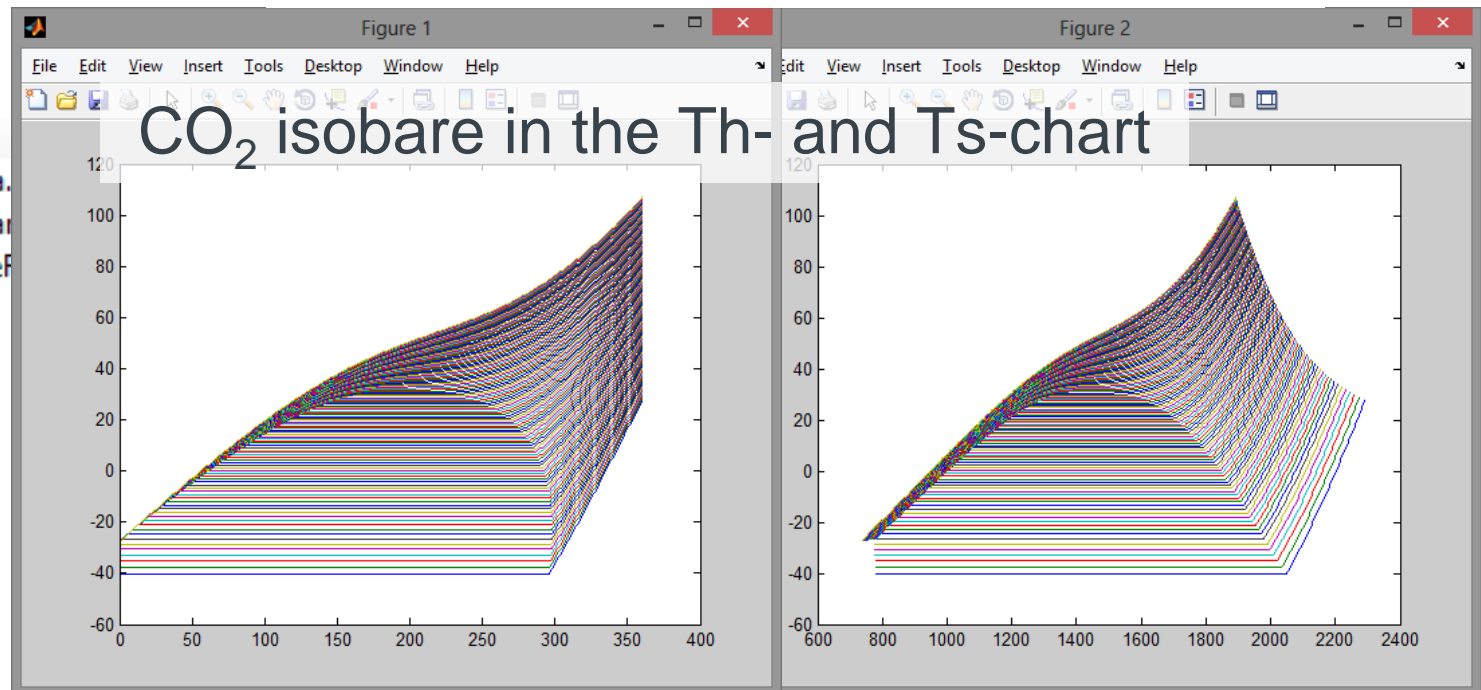
% Choise of substance:

```
vle_name = 'CO2';
```

% Input of enthalpy- and pressure-range:

```
h_vle = [140:1:500]*1e3;
```

```
p_vle = [10:1:120]*1e5;
```



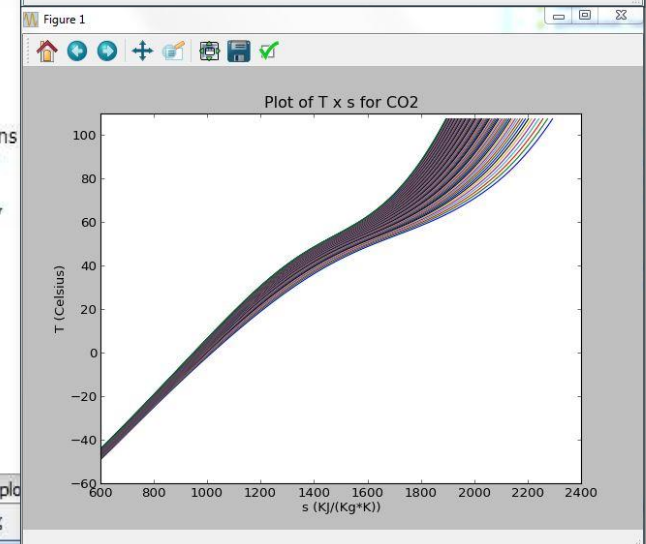
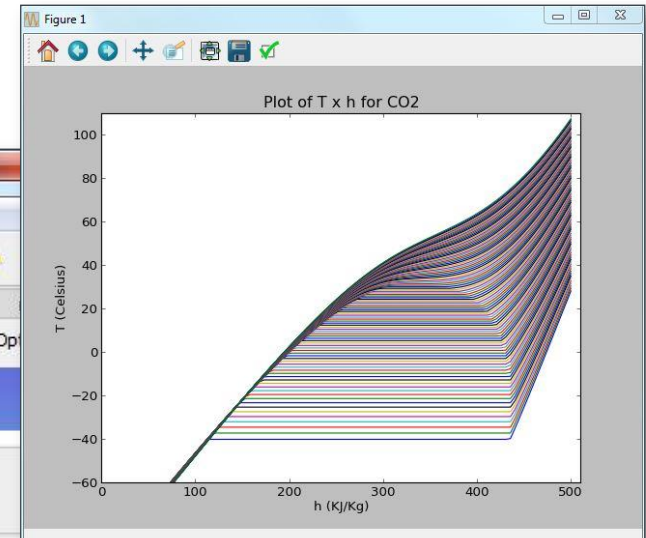
# Example in Python

## CO<sub>2</sub> isobare in the Th- and Ts-chart

```

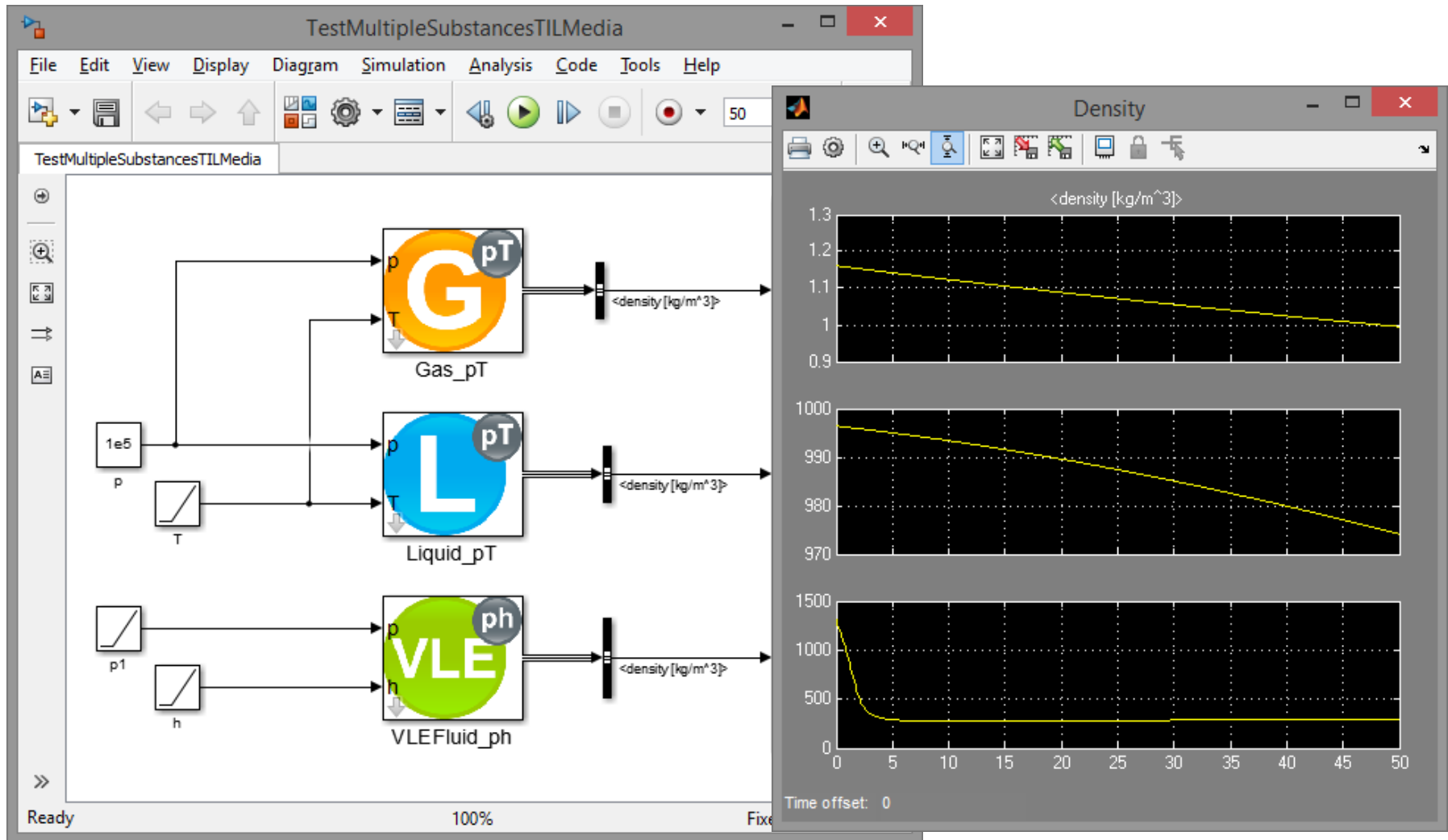
Spyder (Python 2.7)
File Edit Search Source Run Interpreters Tools View ?
Editor - F:\TILMedia\isobar_T_x_h.py
Object inspector
Source Console Object a Op
range
Definition : range(stop)
Type : Function of __builtin__ module
range(stop) -> list of integers
range(start, stop[, step]) -> list of integers
Return a list containing an arithmetic progression of integers. range(i, j) returns [i, i+1, i+2, ..., j-1]; start (!) defaults to 0. When step is given, it specifies the increment (or decrement). For example, range(4) returns [0, 1, 2, 3]. The end point is omitted! These are exactly the valid indices for a list of 4 elements.

1 #isobar.py
2 import TILMedia
3 import numpy as np
4 import pylab as pl
5
6 nrElem_p = 100
7 nrElem_h = 100
8
9 p = np.linspace(10,120,nrElem_p) # bar
10 h = np.linspace(10,500,nrElem_h) # KJ/Kg
11 T = np.zeros(nrElem_h) # Celsius
12 vle = TILMedia.VLEFluid('Refprop.CO2')
13
14 for i_p in range(nrElem_p):
15     for i_h in range(nrElem_h):
16         vle.setState_phxi(p[i_p]*1e5,h[i_h]*1000)
17         T[i_h] = vle.T - 273.15
18     pl.plot(h,T)
19
20 # give plot a title
21 pl.title('Plot of T x h for CO2')
22 # make axis labels
23 pl.xlabel('h (KJ/Kg)')
24 pl.ylabel('T (Celsius)')
25 # set axis limits
26 pl.xlim(0, 510)
27 pl.ylim(-60.0, 110.)
28 # show the plot on the screen
  
```



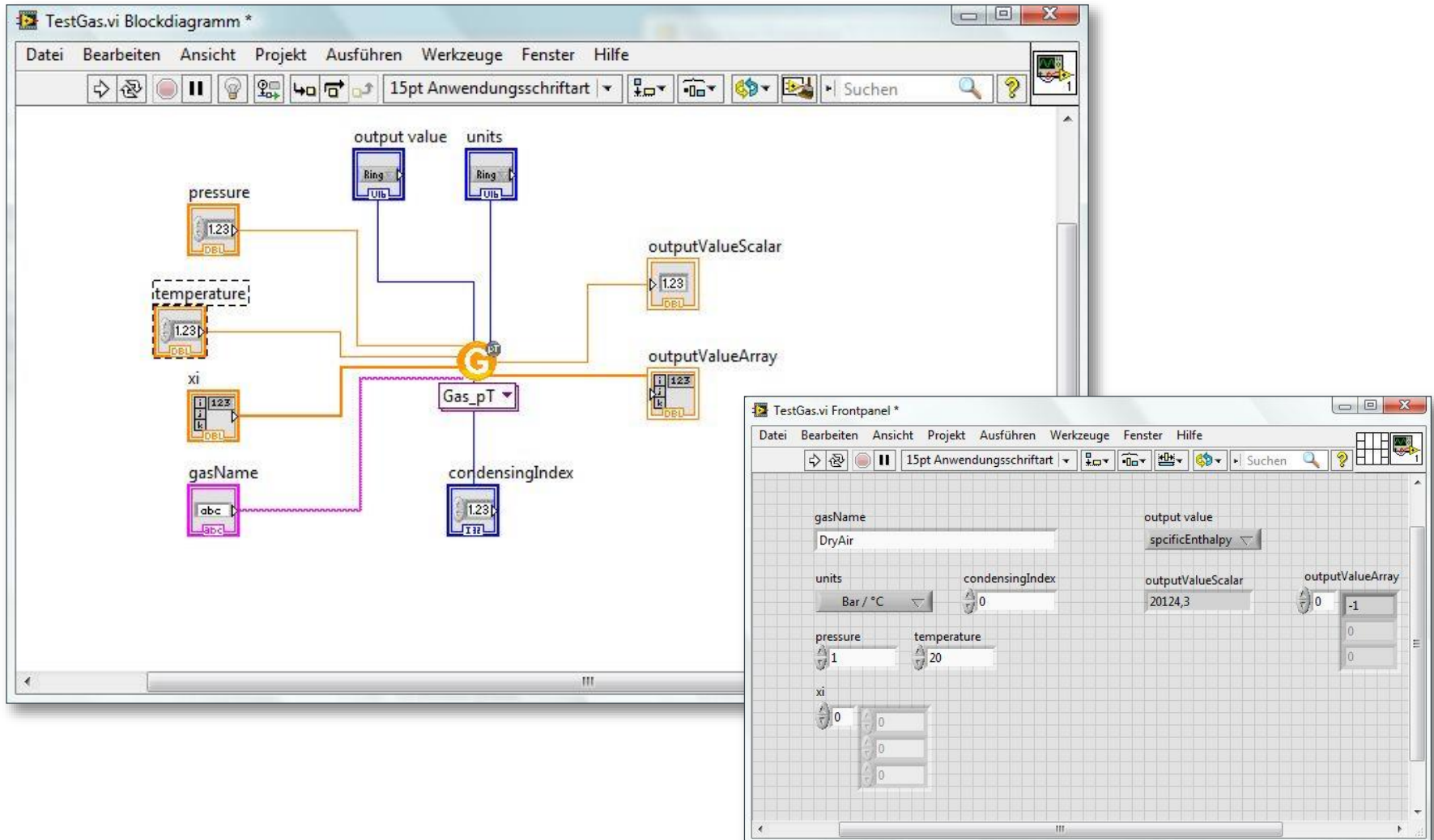


# Example in Simulink





# Example in LabVIEW





# Example in Excel

BEREICHE				
=Liquid_density_T(A3;C5)				
	A	B	C	D
1	Description	Variable	Value	Unit
2	Inputs:			
3	Glystantin_50			
4	Pressure	p	1.01E+05	Pa
5	Temperature	T	300	K
6	Calculation:			
7	Density	rho	=Liquid_density_T(A3;C5)	
8	Entropy	s	-567.3796701	J/(kg K)
9	Enthalpy	h	89323.76775	J/kg
10	Specific isobaric heat capacity	cp		
11	Isobaric thermal expansion coefficient	beta		
12	Dynamic viscosity	eta		
13	Thermal conductivity	lambda		
14	Prandtl number	Pr		

Calculation of density of water-glycol 50:50 at 300 K (and normal pressure)

Funktionsargumente

Liquid\_density\_T

LiquidName  = "Glystantin\_50"

T  = 300

= 1065.506241

'Density' in [kg/m^3] as a function of T.

**LiquidName** Liquid name.

Formelergbnis = 1065.506241

[Hilfe für diese Funktion](#)

OK Abbrechen



# Examples using Windows COM-Interface

## Visual Basic for Applications (VBA) and Visual Basic Script (VBS)

```
'creating two variables
Dim obj, msg As String
'creating a liquid-object
Set obj = CreateObject("TILMedia.Liquid")
'set medium to water-glycol-mixture 50:50
Call obj.setLiquidType("Glysantin_50")
'calculating properties with pressure = 1
    bar and temperature = 300 K
Call obj.setState_pTxi(1e5, 300)
'constructing a message:
msg = "The density of Glysantin_50 is "
    + str(obj.d)
msg = msg + " under the conditions
    pressure = " + str(obj.p)
msg = msg + " and temperature = "
    + str(obj.T)
MsgBox msg 'displaying a message
```

## DIAdem

```
'loading a data file with values for pressure
    and temperature
Call DataFileLoad(CurrentScriptPath
    &"Example_pT.TDM", "TDM", "Load")
'creating four variables
Dim obj, array_p, array_T, chnName_d(0)
'creating a vector-liquid-object
Set obj = CreateObject("TILMedia.VectorLiquid")
'set medium to water-glycol-mixture 50:50
Call obj.setLiquidType("Glysantin_50")
'store pressure values of channel to array
array_p = ChnToValue("channelName_p")
'temperature values from channel to array
array_T = ChnToValue("channelName_T") Call
'calculating properties with pressure and
    temperature array in SI units
obj.setState_pTxi(array_p, array_T)
'set new channel name (vector) to "density"
chnName_d(0) = "density"
'saving density values in channel
Call ArrayToChannels(obj.d, chnName_d, true)
```

red colored = DIAdem-specific functions

# Thank you



If you have any questions,  
don't hesitate to contact us at  
[til@tlk-thermo.com](mailto:til@tlk-thermo.com)



Or your contact person  
**Ingo Frohböse**  
[i.frohboese@tlk-thermo.com](mailto:i.frohboese@tlk-thermo.com)

TLK-Thermo GmbH  
Hans-Sommer-Str. 5  
38106 Braunschweig  
Germany  
[www.tlk-thermo.com](http://www.tlk-thermo.com)

Phone: +49/531/390 76 - 0  
Fax: +49/531/390 76 - 29