

Thermolib

Thermodynamic Systems Library, Release 5.4

Simulation toolbox for the Design and Development
of Thermodynamic Systems in MATLAB[®]/Simulink[®]

USER MANUAL

How to contact EUTech Scientific Engineering GmbH:

Address: EUTech Scientific Engineering GmbH
Dennewartstraße 25-27
D-52068 Aachen
Germany

Phone: +49(0)-241-963-2380

Fax: +49(0)-241-963-2389

Email: sales@thermolib.de
support@thermolib.de

Web : www.eutech-scientific.de



EUTech Scientific Engineering GmbH is a
MathWorks Connections Partner.

Thermolib User Manual, December 2016

Copyright © 2004 – 2016 by EUTech Scientific Engineering GmbH, Germany.

The software described in this document is furnished under a license agreement. The software may be used or copied only under the terms of the license agreement. No part of this manual may be photocopied or reproduced in any form without prior written consent from EUTech Scientific Engineering GmbH.

Trademarks

MATLAB® / Simulink® are registered trademarks of MathWorks, Inc.

Excel® is a registered trademark of Microsoft Corporation.

TABLE OF CONTENTS

1	TECHNICAL CONVENTIONS.....	10
1.1	ABBREVIATIONS AND SYMBOLS	10
1.2	THERMOLIB UNITS	13
2	MEDIA REPRESENTATION IN SIMULINK	15
2.1	FLOW BUS	15
2.2	STATE BUS	16
3	THERMODYNAMIC BACKGROUND	18
3.1	MASS CONSERVATION	18
3.2	ENERGY CONSERVATION.....	18
3.3	THERMODYNAMIC PROPERTIES	19
3.3.1	Gas Phase	19
3.3.1.1	Ideal Gas Model	19
3.3.1.2	Real Gas modeling using Peng-Robinson EOS	20
3.3.1.3	Calculations for Water	21
3.3.2	Liquid Phase	22
3.4	VAPOR-LIQUID EQUILIBRIUM	23
3.5	MODELING OF MIXTURES	25
3.6	CHEMICAL EQUILIBRIUM.....	25
3.7	SUPERCRITICAL STATES.....	26
3.8	LIMITATIONS.....	26
4	CHEMICAL MEDIA DATABASE.....	27
4.1	WHAT ARE CHEMICAL MEDIA DATA?	27
4.2	UNDERSTANDING THE CHEMICAL MEDIA DATABASE.....	27
4.3	CHEMICAL MEDIA DATA IN MATLAB WORKSPACE	28
4.4	CHEMICAL MEDIA DATA IN MS EXCEL.....	31
4.5	WHERE DO I GET THE DATA?	36
4.6	HOW DO THE DATA COME INTO MY SIMULINK MODEL?	37

5	MODELING PRESSURE FEEDBACK.....	40
5.1	WHEN DO I NEED PRESSURE FEEDBACK?	40
5.2	HOW TO CREATE MODELS WITH PRESSURE FEEDBACK?	40
5.3	HOW THE PRESSURE EQUATIONS ARE SOLVED BY THERMOLIB?	41
6	THERMODYNAMIC BALANCING	43
6.1	WHY DO YOU NEED MODEL BALANCING?	43
6.2	MODEL BALANCING BASICS	44
6.3	THE BALANCE BLOCKS CONCEPT	45
6.4	MODEL BALANCING IN THERMOLIB - HOW TO	46
6.5	BALANCE RESULTS IN EXCEL	49
6.6	USING THE BALANCE BLOCKS	50
7	USING THERMOLIB MATLAB FUNCTIONS	57
7.1	CREATING SPECIES DIAGRAMS	57
8	BLOCK REFERENCE.....	59
8.1	BALANCE.....	59
8.1.1	Energy Balance.....	59
8.1.2	Energy Balance_Stored.....	60
8.1.3	Flow Balance.....	62
8.1.4	Flow Balance_Stored.....	63
8.2	COMPONENTS	65
8.2.1	Electric.....	65
8.2.1.1	Inverter DC -> AC.....	65
8.2.1.2	Li-Ion cell	67
8.2.2	Heat and Mass Transfer	72
8.2.2.1	Gas Dryer	72
8.2.2.2	Heat Exchanger NTU	75
8.2.2.3	Heater.....	82
8.2.2.4	Humidity Exchanger	85
8.2.2.5	Thermal Mass.....	88
8.2.3	Hydraulics	91
8.2.3.1	3-Way Valve	91

8.2.3.2	Valve	94
8.2.3.3	Mixer.....	103
8.2.3.4	Pump.....	108
8.2.3.5	Compressor.....	114
8.2.3.6	Turbine	124
8.2.4	Tanks	129
8.2.4.1	Liquid Tank.....	129
8.2.4.2	Tank	134
8.2.5	Hydraulics with Pressure Feedback.....	141
8.2.5.1	Pump Pressure Feedback.....	141
8.2.5.2	Compressor Pressure Feedback.....	145
8.2.5.3	Turbine Pressure Feedback	151
8.2.5.4	Downstream	154
8.2.5.5	Splitter	155
8.2.6	Reactors	158
8.2.6.1	C8H18-Autothermal Reformer.....	158
8.2.6.2	CH4-Autothermal Reformer.....	158
8.2.6.3	CH4-Burner	158
8.2.6.4	CH4-Reformer	158
8.2.6.5	Equilibrium Reactor	158
8.2.6.6	Lambda Burner.....	164
8.2.6.7	Preferential Oxidation Reactor	167
8.2.6.8	Reactor.....	172
8.2.6.9	Shift Reactor.....	175
8.2.7	Stacks	175
8.2.7.1	Current-Controlled Stack with Cooling	175
8.2.7.2	Current-Controlled Stack	181
8.2.7.3	Fixed Conversion Rate Stack with Cooling.....	181
8.2.7.4	Fixed Conversion Rate Stack.....	187
8.2.7.5	SOFC Stack	187
8.2.7.6	DMFC Stack	193
8.2.7.7	PEMFC Stack Analytic	197
8.3	CONTROLLERS	201
8.3.1	PI-Controller	201
8.3.2	PID-Controller	203
8.4	MISCELLANEOUS	206
8.4.1	Convert FlowBus to StateBus	206
8.4.2	Divide.....	206
8.4.3	Hold T p x psi dynamic.....	208

8.4.4	Hold T p x psi with IC.....	210
8.4.5	Initial Condition.....	210
8.4.6	LimitNdot.....	213
8.4.7	Replace Compound.....	213
8.4.8	SimStepSize	214
8.4.9	Sensor	215
8.4.10	Unit Conversion	217
8.5	SETUP	218
8.5.1	Model Setup	218
8.6	SINKS.....	224
8.6.1	FlowDisplay.....	224
8.6.2	Measurement Point.....	226
8.6.3	State Plot.....	229
8.7	SOURCES.....	233
8.7.1	Flow Ground.....	233
8.7.2	Mixture Source.....	234
8.7.3	Power Source.....	236
8.7.4	Pure Substance Source.....	238
8.7.5	Wet Air	241
8.8	THERMODYNAMIC STATES AND PROPERTIES	243
8.8.1	Density.....	243
8.8.2	Viscosity.....	244
8.8.3	Thermal Conductivity	246
8.8.4	Reynolds.....	248
8.8.5	Gas Properties	250
8.8.6	H-p-State with Heat Exchange	252
8.8.7	H-p-State with Reactions.....	255
8.8.8	Liquid Properties	258
8.8.9	Mass Flow Rate.....	262
8.8.10	Molar Weight	264
8.8.11	Pressure Loss	265
8.8.12	Relative Humidity.....	266

8.8.13	Saturation Pressures	269
8.8.14	Saturation Temperatures	271
8.8.15	T-p-H-State	273
8.8.16	T-p-State	274
8.8.17	T-p-Vapor Fractions	275
8.8.18	T-p-x-State	277
8.8.19	Volume Flow	279
8.8.20	T-p-S-State	281
8.8.21	T-S-State	283
8.8.22	T-rho State	285
8.8.23	S-p-State	287
8.8.24	U-rho State	288
8.9	THERMODYNAMIC STATE CHANGES	291
8.9.1	Isobaric Enthalpy Change	291
8.9.2	Isobaric Entropy Change	292
8.9.3	Isobaric Temperature Change	293
8.9.4	Isobaric Evaporation	294
8.9.5	Isobaric Condensation	295
8.9.6	Isentropic Pressure Change	295
8.9.7	Isentropic Temperature Change	297
8.9.8	Isothermal Entropy Change	298
8.9.9	Adiabatic Pressure Change	299
8.10	COMPATIBILITY BLOCKS	300
8.10.1	FClib - 4-1-8-4	300
8.10.1.1	Upstream	300
9	COMMAND REFERENCE	303
9.1	THERMOLIB_SETUP	303
9.2	THERMOLIB_VERSION	303
9.3	THERMOLIB_OPEN	304
9.4	TH_CREATEFEEDBACK	304
9.5	TH_EXPORTBALANCE	306

9.6	TH_MODIFYCONSTANTS.....	307
10	COMMAND LINE FUNCTIONS.....	310
10.1	TH_DENSITY.....	310
10.2	TH_HPSTATE	312
10.3	TH_PENGROBINSON	314
10.4	TH_SATURATIONPRESSURE	315
10.5	TH_SATURATIONTEMPERATURE	317
10.6	TH_SPSTATE	318
10.7	TH_TPHSTATE	320
10.8	TH_TPSTATE.....	322
10.9	TH_TpVAPORFRACTIONS.....	324
10.10	TH_TpXSTATE.....	326
10.11	TH_TRHOSTATE	328
10.12	TH_URHOSTATE.....	330
11	REFERENCES	333

LIST OF FIGURES

Figure 1: Detail of "ChemData" sheet in Chemical Media Database ChemicalMediaData.xls.	32
Figure 2: Constants in Chemical Media Database	32
Figure 3: Adding a new Species to the database	33
Figure 4: Adding a new property to the database.....	34
Figure 5: Adding a new constant to the database.....	34
Figure 6: Adding a new element to the database	35
Figure 7: Adding new element to species database	35
Figure 8: Model Setup dialog	38
Figure 9: Simple air flow network	40
Figure 10: Pressure network with two branches.....	41
Figure 11: Mass balance in Excel	49
Figure 12: Mass balance in ASCII	50
Figure 13: A user defined balance space using balancing blocks.....	52
Figure 14: Principle of internal representation of heat exchanger.....	77
Figure 15: GUI for automatic creation of pressure feedback connection	305

1 TECHNICAL CONVENTIONS

This section gives some conventions about the symbols and units used in Thermolib.

1.1 ABBREVIATIONS AND SYMBOLS

Symbol	Description	Unit
\dot{C}	Heat capacity rate	W/K
C	Heat capacity ratio	-
C	Capacitance	F
c_p	Specific heat capacity	J/(g K)
$ctrl$	Fraction of maximum mass flow	-
D	Hydraulic diameter	m
F	Frequency	Hz
f_{power}	Power factor	W/(g/s)
f_i	Fugacity of component i	Pa
G	Total Gibb's function	J
H	Total Enthalpy	J
\dot{H}	Total Enthalpy flow	J/s
h	Specific enthalpy	J/kg
h_m	Molar Enthalpy	J/mol
I	Current	A
k	Pressure Loss Factor	Pa/(g/s) ²
k	Thermal conductivity	W/(m K)
K	Heat transfer coefficient	W/(m ² K)
KA	Heat transfer rate	W/K
L	Moles of Liquid	mol

Symbol	Description	Unit
\dot{m}	Total mass flow	g/s
\dot{n}	Total molar flow	mol/s
N	Number of Transfer Units	-
n	Polytropic exponent	-
n_i	Amount of substance of species i	mol
P	Mechanical or Electrical Power	W
p	Pressure	Pa
p_c	Critical Pressure	Pa
Δp	Pressure Loss	Pa
pos	Valve Position	-
PR	Pressure Ratio	-
\dot{Q}	Heat Flow	W
R	Universal gas constant (= 8.3145 [J/mol.K])	J/(mol K)
R	Resistance	Ω
RPM	Rounds per minute	RPM
S	Total entropy	J/K
\dot{S}	Entropy flow	J/(s K)
s	Specific entropy	J/(K g)
s_m	Molar entropy	J/(K mol)
SOC	State of charge	-
T	Absolute Temperature	K
T_c	Critical temperature	K
U	Voltage	V
U	Total internal energy	J
UA	Overall heat exchange coefficient	W/K

Symbol	Description	Unit
v	Velocity	m/s
V	Volume	m ³
V	Moles of vapor	mol
$VisG$	Viscosity for gas	Pa.s
$VisL$	Viscosity for liquid	Pa.s
x	Vapor fraction	-
x_i	Mole fraction of species i in liquid phase	mol/mol
y_i	Mole fraction of species i in gas phase	mol/mol
Z	Compressibility factor	-
ε	Effectiveness	-
η	Efficiency	-
ρ	Density	g/m ³
τ	Characteristic time	s
Ψ_i	Mole fraction of the species i	-
ω	Acentric factor	-

Subscripts:

Symbol	Description
0	State at reference temperature and reference pressure
an	Anode
c	Cold
c	Critical
$cath$	Cathode

Symbol	Description
<i>el</i>	Electric
<i>env</i>	Environment
<i>f</i>	Liquid phase
<i>f</i>	Formation
<i>fg</i>	Evaporation
<i>g</i>	Gas phase
<i>gas</i>	Gas phase
<i>h</i>	Hot
<i>ig</i>	Ideal gas
<i>in</i>	Incoming
<i>l</i>	Liquid phase
<i>m</i>	Molar
<i>max</i>	Maximal
<i>mch</i>	Mechanic
<i>min</i>	Minimal
<i>NASA</i>	Calculated with NASA Polynomials
<i>out</i>	Outgoing
<i>ref</i>	Reference state
<i>s</i>	Isentropic state change
<i>sat</i>	Saturated
<i>stored</i>	Stored
<i>TM</i>	Thermal mass

1.2 THERMOLIB UNITS

Thermolib uses SI units, where it is not stated or configured by the user for any special unit.

Standard units are K, J, W, m, mol. The only exception is the mass, which is given in g, rather than in kg.

2 MEDIA REPRESENTATION IN SIMULINK

Simulink is signal orientated graphical programming environment for dynamic simulations. Blocks are connected via signals. To represent all thermodynamic information in an easy to access way, Thermolib uses Simulink Busses to represent media flows and thermodynamic states.

2.1 FLOW BUS

In models that are based on the **Thermolib**, the signal flow describing the medium flow between the blocks is described by a so-called flow bus (FB).

The flow bus contains the primary information on the flowing media. Apart from the total molar flow and the chemical composition, it contains the thermodynamic properties temperature, pressure and vapor fraction.

The flow bus is structured as follows:

Signal Name	Name	Symbol	Unit
ndot	Total molar flow	\dot{n}	mol/s
T	Temperature	T	K
p	Pressure	p	Pa
Hdot	Enthalpy flow	\dot{H}	W
Sdot	Entropy flow	\dot{S}	W/ K
Gdot	Gibbs energy rate	\dot{G}	W
Cpdot	Heat capacity rate	c_p	W/ K
x	Vapor fraction of all compounds (vector)	x	mol/mol
psi	Molar fractions of all compounds (vector)	ψ	mol/mol

The signals x and psi are vectors of the same length. The length is the same as the number of selected compounds to work with. You can select the compounds in the Model Setup block.

For all of these compounds, coefficients for calculating the heat capacity, enthalpy and entropy are given in an Excel file (ChemicalMediaData.xls). The user can easily extend this

file or the resulting ChemicalMediaData.mat for simulations with other compounds. For more details on how to define new compounds, see Chapter 4.

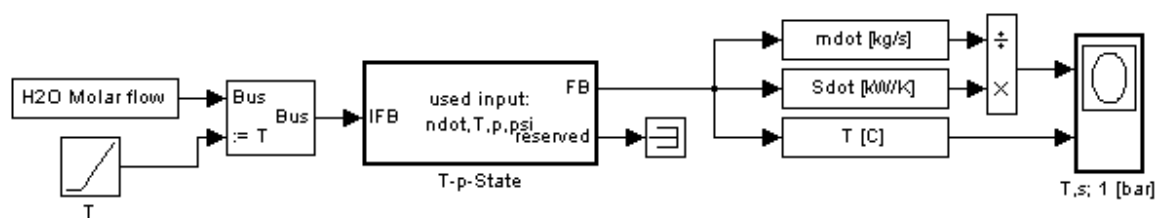
There are concepts of a complete flow bus (FB) and an incomplete flow bus (IFB). The properties of a complete flow bus are thermodynamically consistent, e.g., calculating the vapor fractions from T , p and H simply yields the vapor fractions x that are already present in the FB.

An incomplete flow bus is more or less a vector of values, and no guarantee is given that they are consistent in the above sense. On the contrary, in most cases they are not.

In a nutshell, a block that outputs a FB ensures that the result is a consistent flow bus and that all properties are valid.

If you change a signal property on a FB, then this thermodynamic state becomes inconsistent, so the FB becomes a IFB. Use then the blocks of section "Thermodynamic States and Properties" to update the other signals on the bus as needed.

The following picture shows how that works.



The source block on the left creates a consistent Flow Bus. The Bus assignment changes only the temperature on the bus and makes the State "inconsistent", as e. g. the temperature now does not longer corresponds with the enthalpy. Now a T-p-State block is used to update the FB and make the values consistent using "ndot,T,p,psi" of the IFB.

The blocks on the right are sensor blocks, which are finally used to create a T-S diagram.

2.2 STATE BUS

Unlike the flow bus, the state bus describes a current state of a portion of media, e. g. a control volume in a tank. The majority of flows in Thermolib are described with the flow bus; the state bus is mainly used to describe the current state of tanks: mass, temperature and pressure. To display the state bus, the flow display can be used as long as the user selects 'State Bus' in the mask.

The state bus is structured as follows:

Signal Name	Name	Symbol	Unit
-------------	------	--------	------

n	Total number of moles	n	mol
T	Temperature	T	K
p	Pressure	p	Pa
H	Enthalpy	H	J
S	Entropy	S	J/ K
G	Gibbs energy	G	J
C_p	Heat capacity	c_p	J/ K
U	Internal Energy	U	J
x	Vapor fraction of all compounds	x	mol/mol
ψ	Molar fractions of all compounds	ψ	mol/mol

3 THERMODYNAMIC BACKGROUND

This chapter gives a short introduction into the thermodynamic background of Thermolib. It explains the main concepts used to model thermodynamic processes in Thermolib.

3.1 MASS CONSERVATION

Mass is a conserved quantity (except in nuclear reactions), as is the mass of any individual compound if there is no chemical reaction occurring in the system. Let us write the conservation equation:

$$\frac{dm}{dt} = \sum \dot{m}_{out} - \sum \dot{m}_{in}$$

All components blocks in Thermolib except Tanks do not accumulate mass. If there is no mass accumulation in the control volume, as in a steady state, steady flow process this simplifies to:

$$\sum \dot{m}_{out} - \sum \dot{m}_{in} = 0$$

3.2 ENERGY CONSERVATION

Lot of components blocks in Thermolib have inputs and outputs like "Qdot" or "P_mch" representing energy flows across the border of the block.

If such a signal has a value > 0 at a time, then currently the energy flow is from outside of the block to inside of the block. The energy/power/heat is then added to the flow or to internal energy storages.

If such a signal has a value < 0 at a time, then currently the energy flow is from inside of the block to outside of the block. The energy/power/heat is drawn from the flow or from internal energy storages.

In general it holds:

$$\frac{dU}{dt} = \sum \dot{H}_{in,i} - \sum \dot{H}_{out,j} + \sum \dot{Q}_k + \sum P_m$$

as the energy conservation law (1st law of thermodynamics). Kinetic and potential energies of the incoming and outgoing flows are neglected in Thermolib and in the equation above.

Where $\dot{H}_{in,i}$ is the enthalpy flow of the input flow i , $\dot{H}_{out,j}$ is the enthalpy flow of the output flow j , \dot{Q}_k is heat flow k , P_m is mechanical power m and U is the internal energy of the component.

That means for a compressor the P_{mch} will be usually positive, because the mechanical energy at the shaft is added to the fluid, while for a turbine it is negative as the power is released by the fluid.

For a hot component like a reactor the \dot{Q} signal (heat exchange with environment) will be usually negative, as this heat is leaving the reactor (suppose a cold environment is configured).

In an adiabatic chemical reactor in steady state the input and output total enthalpies (thermal + enthalpy of formation) are the same. But due to the different concentrations of the compounds at inlet and outlet, there might be very different temperatures at inlet and outlet.

3.3 THERMODYNAMIC PROPERTIES

This chapter describes how gas and liquid phase are modeled in Thermolib.

3.3.1 Gas Phase

For the gas phase, the modified ideal gas law

$$Z \times R \times T = p \times v$$

is used to calculate density, enthalpy and entropy departure of real gases at a given temperature and pressure.

In Model Setup block you can choose between two options:

- Ideal Gas ($Z=1$)
- Real Gas (Z calculated using Peng-Robinson equation of state)

3.3.1.1 Ideal Gas Model

In case of ideal gas selection in Model Setup block, enthalpy, entropy and heat capacity at a constant pressure are calculated with the help of NASA polynomials.

The NASA polynomials are:

$$\frac{c_{p,m}^{ig}}{R} = A + B \times T + C \times T^2 + D \times T^3 + E \times T^4$$

Pressure dependence of entropy is considered assuming reference pressure of $p_{ref} = 0.1$ MPa as:

$$s_m^{ig} = s_{m,NASA} - R \ln \left(\frac{p}{p_{ref}} \right)$$

We use total enthalpy and entropy, it means:

$$\begin{aligned} h_m^{ig}(T_{ref} = 298 \text{ K}) &= h_f^0 \\ s_m^{ig}(T_{ref} = 0 \text{ K}, p_{ref} = 0,1 \text{ MPa}) &= 0 \end{aligned}$$

Where h_f^0 is the enthalpy of formation of the species.

All coefficients A-G are fluid-dependent and can be taken from public thermophysical databases (e.g. NIST Webbook [2] or the original NASA thermo data file in the \Data folder). Two sets of coefficients are configurable in Thermolib species database. The validity range of the coefficients is configurable (see variables NASA_T_min, NASA_T_mid, NASA_T_max in ChemicalMediaData.xls).

For numerical purpose, we use constant C_p for extrapolation in temperature range below NASA_T_min or above NASA_T_max.

Note the units:

h_m [J/mol]

s_m [J/(mol*K)]

g_m [J/mol]

c_{pm} [J/(mol*K)]

R [J/(mol*K)]

In some sources of the NASA coefficients the left hand side of the equations is not written relative to the absolute gas constant R . You have to keep that in mind and adjust the parameters if necessary. For any gas component, the coefficients A-G of the NASA polynomials have to be given for two temperature ranges:

Temperature Range	NASA coefficients
NASA_T_min ... NASA_T_mid	NASA_A2 ... NASA_G2
NASA_T_mid ... NASA_T_max	NASA_A1 ... NASA_G1

These coefficients and a user guide are available under Thermolib/Data/nasadat.html and Thermolib/Data/NasaGuide. The coefficients have to be entered into ChemicalMediaData.xls (see Chapter "4 Chemical Media Database").

3.3.1.2 Real Gas modeling using Peng-Robinson EOS

The Peng-Robinson equation of state relates the three interdependent state properties pressure p , temperature T , and molar volume v :

$$p = \frac{RT}{v - b} - \frac{a\alpha}{v^2 - 2bv - b^2}$$

where, ω is the acentric factor of the species, T_c and p_c are critical temperature and pressure of each species.

From the state properties (p, v, T), one may compute the departure function for molar enthalpy (denoted h) and molar entropy (s):

$$h_{T,p} - h_{T,p}^{ig} = RT_c \left[T_r(Z - 1) - 2.078(1 + \kappa)\sqrt{\alpha} \ln \left(Z + \frac{Z + 2.414B}{Z - 0.414B} \right) \right]$$

Where α is defined in the Peng-Robinson equation of state, T_r is the reduced temperature, p_r is the reduced pressure, Z is the compressibility factor $Z = pv/(RT)$, and

$$B = 0.0778 \frac{p_r}{T_r}$$

The critical properties T_c , p_c and ω are contained in ChemicalMediaData.xls (see chapter "4 Chemical Media Database").

3.3.1.3 Calculations for Water

Water plays an important role in process engineering and thermodynamic calculations, but ideal gas assumption has very large errors and even a Peng-Robinson EOS does not match water experimental data sufficiently enough. In Thermolib there are implemented two ways for improved water calculations:

- Steam Table correction
- IAPWS-IF97

If choosen Ideal Gas in the Model Setup block, then for the species "H2O" a steam table correction is applied to the enthalpy data. The corresponding lookup-table is located in the "SteamTable_dh_IG.mat" file in the \data subfolder of the Thermolib installation folder. This table was generated from table data from [3].

If choosen Peng-Robinson for modeling the gas phase, then this correction is not applied, as we assume Peng-Robinson EOS to give more accurate results (especially outside of the valid range of the steam table correction).

For applications where very precise steam properties are needed, Thermolib offers the simulation of water using the industrial standard 97 formulated by the International Association for Properties of Water and Steam (IAPWS-IF97). These calculations are very accurate, but require some additional calculation time (may slow down simulation). IF97 is available as separate Thermolib Add-on. If you have licensed the IF97-Modul and choose H2O-IF97 from species list, then the following calculations are based on IF97-functions:

- Enthalpy
- Entropy

- Density in gas phase (only if Ideal Gas is selected. For Peng-Robinson water is included into solving the PR-EOS with its critical values).
- Density in liquid phase.
In contraire to the standard, the implementation of liquid density in Thermolib is dependent to the pressure. Thermolib uses a linear interpolation of the density between critical pressure and saturation pressure.
- Saturation temperature and saturation pressure

3.3.2 Liquid Phase

Temperature dependency of the saturated liquid density is modeled using the following equation:

$$\rho_{f,sat}(T) = \rho_c(1 + A\tau^{1/3} + B\tau^{2/3} + C\tau + D\tau^{4/3})$$

where

$$\tau = 1 - \frac{T}{T_c}$$

If Real Gas calculation based on PR-EOS is activated in Model Setup block. The liquid density is calculated using PR-EOS with an offset to match the liquid density at a reference temperature (T_{v_ref}).

Liquid phase is modeled as compressible with a given bulk modul.

$$\rho_f(p, T) = \frac{\rho_{f,sat}}{1 - \frac{p - p_{sat}(T)}{E}}$$

For the liquid phase, a heat capacity dependent to temperature for all compounds is assumed:

$$c_{p_{m,f}} = A + BT + CT^2 + DT^3$$

The coefficients A, B, C and D have to be given in Chemical Media Database. Also limits for the validity of the polynomial can be given there. Outside of the limits heat capacity is assumed to be constant.

To ensure consistency with gas phase calculation, enthalpy of liquid is calculated based on gas enthalpy with transition to liquid enthalpy using an explicit given function for the evaporation enthalpy.

$$h_{m,f,sat,i}(T) = h_{m,g}(T) - h_{fg}(T)$$

To ensure consistency with the gas calculation also for paths in the dense state and supercritical regions, pressure dependent liquid enthalpy is linear interpolated between saturation pressure and critical pressure.

$$h_{m,f,i}(T, p) = h_{m,f,sat,i}(T) + \frac{(p - p_{sat,i})}{p_{c,i} - p_{sat,i}} (h_{m,g,i}(T, p_{c,i}) - h_{m,f,sat,i}(T))$$

In Thermolib releases before 5.1 the liquid enthalpy was modeled using a transition from gas to liquid enthalpy at a given reference temperature. This option is still available. To activate this, go to the Model Setup block and choose the corresponding option in section “Enthalpy/Entropy of liquid phase”.

The method works based on a gas enthalpy at a reference temperature (T_{ref}). This means, that enthalpy for fluid is calculated using:

$$h_{m,f} = h_{m,g}(T_{ref}) - h_{fg}(T_{ref}) + \int_{T_{ref}}^T c_{p,f} dT$$

Where $h_{fg}(T_{ref})$ is the enthalpy of evaporation at the reference temperature. Same consistency rule applies for calculation of entropy, where:

$$s_{m,f} = s_{m,g}(T_{ref}) - s_{fg}(T_{ref}) + \int_{T_{ref}}^T \frac{c_{p,m,f}}{T} dT$$

is used. The calculation of Gibbs function is based on the entropy and enthalpy as:

$$g_{m,f} = h_{m,f} - s_{m,f} T$$

Note, that there are some blocks, which can only handle liquids (e.g. pump) and some, which can only handle two phase mixtures or gases, but not pure liquids (e. g. Adiabatic Tank). This is stated in the corresponding block documentation.

All coefficients for heat capacity, density, evaporation enthalpy and evaporation entropy as well as the reference temperature (T_{ref}) are given in the Chemical Media Database (see Chapter 4. Chemical Media Database).

3.4 VAPOR-LIQUID EQUILIBRIUM

Thermolib can handle gas and liquid phase. Both phases are assumed to be in thermodynamic equilibrium. It means that, pressure and temperature in both phases are the same.

To calculate the concentrations of the compounds in the phases in phase equilibrium the vapor pressure curves and Raoult's rule are used.

The Antoine equation is used to calculate the saturation pressure (p_{sat}) as a function of the saturation temperature T_{sat} and vice versa.

The Antoine equation reads:

$$\log_{10}(p_{sat}) = 1 - \left(\frac{B}{T_{sat} + C} \right)$$

In which, p_{sat} has the dimension bar and T_{sat} is in Kelvin. The coefficients A , B , C have to be given in Chemical Media Database (see Chapter 4. Chemical Media Database).

If you would like to simulate a defined species always as liquid, you just have to set constant A accordingly, that the compound keeps liquid for the complete temperature range you use in the simulation, or you can simply use the values for a gas, that never condense in validation range (e. g. H_2).

For calculating vapor liquid equilibrium of a mixture, Raoult's rule is used, that is a special case of ideal solution. For liquid phase, we make two assumptions:

1. The fugacity of pure liquid at T and p is equal to the fugacity of saturated liquid or vapor at the same T :

$$f_i^f = f_i^{sat}$$

2. Pure saturated vapor behaves as an ideal gas:

$$f_i^{sat} = p_i^{sat}$$

combining these two equations we get:

$$f_i^{sat} = p_i^{sat}$$

For the vapor phase, we assume that pure gases behave as ideal gas at T and p , so that:

$$f_i^g = p$$

We use also the ideal solution assumption, in which the fugacity of component i in a mixture is expressed as the product of the mole fraction of i and the fugacity of pure i in the same phase as the mixture and at the temperature and pressure of the mixture. So that for each species i we have:

$$x_i f_i^f = y_i f_i^g$$

Here x_i is the mole fraction of species i within the liquid phase and y_i the mole fraction of species i in gas phase.

With combination of these equations, we would have:

For each species i :

$$x_i p_i^{sat} = y_i p$$

As the concentrations sum to 1 in each phase we have

for liquid phase: $\sum x_i = 1$

for gas phase: $\sum y_i = 1$

For each species i : $z_i = x_i L + y_i V$

Here L and V are the moles of liquid and vapor in the flow.

At a given temperature and pressure, in a mixture with n components, we have $2+2n$ equations and $2+2n$ unknowns (x_i 's and y_i 's and L and V). For more as two components, these equations become nonlinear and are solved by the TpVaporFraction block.

3.5 MODELING OF MIXTURES

Thermolib assumes ideal liquid and gas mixtures.

Therefore the properties of mixtures are calculated from the properties of the pure substances by simple summation:

Enthalpy of a mixture:

$$H = \sum_i (n_{i,g} h_{m,i,g} + n_{i,l} h_{m,i,l})$$

Entropy of a mixture:

$$S = \sum_i (n_{i,g} s_{m,i,g} + n_{i,l} s_{m,i,l}) - R \sum_i (n_{i,g} \ln \psi_{i,g})$$

Summation is taken over all species.

Excess enthalpies or entropies are not calculated.

3.6 CHEMICAL EQUILIBRIUM

One key component of Thermolib is the equilibrium reactor. It calculates the chemical equilibrium for a given initial composition and temperature at a given pressure.

The thermodynamic equilibrium is defined by the minimum of the Gibbs free energy G .

For a given reaction:



the chemical equilibrium condition can be written as

$$\ln(k) = -\frac{\Delta G}{RT}$$

where :

$$k = \left(\frac{\psi_C^c \cdot \psi_D^d}{\psi_A^a \cdot \psi_B^b} \right) \cdot \left(\frac{p}{p_0} \right)^{c+d-a-b}$$

and ψ_j is the mole fraction of species j , ΔG is the difference of Gibbs free energy between products and reactants and R is the universal gas constant.

It is assumed that all products and reactants are in gaseous phase. All concentrations used, are the overall concentrations of the compounds.

3.7 SUPERCRITICAL STATES

If temperature is greater than critical temperature, it is assumed that there is only vapor phase. Therefore the vapor fraction on the flow bus will be 1. Saturation pressure has no meaning in this state and is simple extrapolation of the Antoine formula beyond the critical temperature.

3.8 LIMITATIONS

Thermolib is a software aimed for system level modeling for real time simulations. Therefore Thermolib software is limited by the following:

- no detailed geometries are modeled
- no gravity nor potential energy are modeled
- no kinetic energy is modeled
- no solid state is modeled
- no interactions between species like azeotrope, solubility (except explicit chemical reaction modeling)
- no supersonic flows
- no pressure waves / shock waves

4 CHEMICAL MEDIA DATABASE

The database of the Thermophysical properties is called "Chemical Media Database" in Thermolib. This chapter explains, how you can work with it.

4.1 WHAT ARE CHEMICAL MEDIA DATA?

Chemical media are the chemical compounds that are available in Thermolib simulations. Some of the default species available are: methane (CH_4), water (H_2O), hydrogen gas (H_2), oxygen gas (O_2), carbon dioxide (CO_2), among others. To be able to simulate the physical changes that each of these substances goes through when flowing through different thermodynamic components, Thermolib requires a set of thermo-physical properties that describe each species. These are referred to as Chemical Media Data.

For instance, consider the simulation of a water stream flowing through a heater. The water at the source will be of a certain temperature and pressure; however, once it enters the heater, the temperature and pressure amongst other chemical properties of the water will change according to the amount of heat provided. Using thermo-physical properties that describe the substance "water", Thermolib can calculate the changes in temperature and pressure of the water at the output of the heater.

Chemical Media Data are stored in a mat file that Thermolib loads into the workspace whenever a simulation starts; this way the information is available to the Thermolib blocks that require it.

The name of the mat-file is user-defined. Thermolib is delivered with a default chemical media data file ChemicalMediaData.mat. For more information on how to create your own chemical media data files using MATLAB or MS Excel, see below.

4.2 UNDERSTANDING THE CHEMICAL MEDIA DATABASE

For each medium available in Thermolib, e.g. water (H_2O), there is a set of thermo-physical properties that describe it. Some of these properties are: molar mass, liquid density, heat capacity, critical temperature, critical pressure, among many others.

The media available to a model are stored in the MATLAB workspace variable "Species" as a cell array:

```
Species = { 'CH4'      'H2O'      'CO'      'CO2'      'N2'      'O2' }
```

Each property is also stored in a MATLAB workspace variable in array form. For instance, the temperature at the critical points are stored in the variable "th_T_c":

```
th_T_c = [ 190.6000  647.3000  132.9000  304.2000  33.2000  
          126.2000  154.6000 ]
```

To find the critical temperature of water (H_2O), first note that water is the second entry in the variable `Species`, so its critical temperature is the second element of `th_T_c`. Blocks in a Thermolib model that need this value can retrieve it from the MATLAB workspace.

This database structure of the species is stored in a MATLAB structure, `SMediaData`, to facilitate the addition of new species or new properties. A Thermolib model automatically loads the structure `SMediaData`, retrieves the species required for the current simulation and creates the necessary variables (such as `Species` and `th_T_c`) in the MATLAB workspace. All of this is transparent to the user. The next section explains the chemical media data structure.

4.3 CHEMICAL MEDIA DATA IN MATLAB WORKSPACE

The chemical media data are stored in a mat file in the form of a structure `SMediaData`. The default chemical media data file delivered with Thermolib is called `ChemicalMediaData.mat`.

If you enter at the MATLAB command line:

```
load ChemicalMediaData
```

the structure `SMediaData` will be loaded into the base workspace where you can examine it. `SMediaData` has two fields, `SGlobal` and `SSpecies`.

`SSpecies` is a structure array where each entry in the array describes one of the available species. The first two entries under `SMediaData.SSpecies` are shown below:

```
>> SMediaData.SSpecies(1)
```

```
ans =
```

```
    Species: 'CH4'  
    formula: NaN  
  full_name: 'methane'  
  NASA_A1_g: 1.6355  
  NASA_B1_g: 0.0101  
  NASA_C1_g: -3.3692e-006
```

```
...
```

```
    Cp_liq_A: 60  
    Cp_liq_B: 0  
    Cp_liq_C: 0  
    Cp_liq_D: 0
```

```
Cp_liq_T_min: 200
Cp_liq_T_max: 6000
```

Each entry in `SMediaData.SSpecies` is identified by the name of the medium in the field "Species". The rest are the thermo-physical properties that describe the medium; here `th_T_c` is shown among other properties.

Constants

In addition to properties for each species, correct thermodynamic simulations require several constants that are independent of any particular species. These are also stored in the structure `SMediaData` under the field `SGlobal`:

```
SMediaData.SGlobal =
    k: 1.3800e-023
    N_A: 6.0220e+023
    R: 8.3144
    e: 1.6022e-019
```

Elements

For combustion it is necessary, in which stoichiometric way elements react to which product. The field `SElements` specifies the available elements and the combustion product, when used in `LambdaBurner`.

You can adjust the Chemical Media Database by adding new properties or new species. One feature of Thermolib is that the chemical media data can be viewed and adjusted in Microsoft Excel. Thermolib offers a tool where it automatically imports the chemical media data from Excel into a structure `SMediaData` into MATLAB workspace. You can find more on this in section 4.4. If you do not have Microsoft Excel available, you can adjust the `SMediaData` structure in MATLAB directly.

Adding a New Species

Start by loading the structure `SMediaData` into the Matlab workspace:

```
load ChemicalMediaData
```

To add a new compound:

```
SMediaData.SSpecies(end+1).Species = 'NEW_COMPOUND'
```

This creates a new entry at the end of the array `SMediaData.SSpecies` for the new compound. Now you have to set the rest of the properties. For instance to set the critical temperature:

```
SMediaData.SSpecies(end).th_T_c = 700
```

For a complete list of properties and their meanings, please refer to the Chemical Media Excel File (\data\ChemicalMediaData.xls) in your Thermolib installation directory.

Once the changes are complete, save the structure SMediaData to a mat file of your choice:

```
save MyChemicalMediaData.mat SMediaData
```

To make the new species available to your model, use the file MyChemicalMediaData.mat in Model Setup (see help for "Model Setup" block).

Adding a New Property

You might find that in your simulation you need a thermo-physical property that is not available by default in SMediaData. If this property is specific to a species, add it under SMediaData.SSpecies. To add the property *property_y* to the species *MY_COMPOUND*:

Find the index of *MY_COMPOUND* in the array SMediaData.SSpecies:

```
index = find(strcmp({SMediaData.SSpecies.Species}, 'MY_COMPOUND'));  
SMediaData.SSpecies(index).property_y = 1234
```

Note that this will create a field *property_y* for all the other species and its value will be empty. If you do not wish to set it for the other species, we recommend that you check that this property is not used elsewhere.

To make the new species available to your model, first save the adjusted SMediaData in a mat file of your choice:

```
save MyChemicalMediaData.mat SMediaData
```

Then use that file in Model Setup (see help for "Model Setup" block).

Note that all fields from SMediaData.SSpecies will be exported to single workspace variables with the prefix "th_". So for your new entry "property_y" a new workspace variable "th_property_xy" will be created, which contains the vector of all values of this property for all Species selected in the Model Setup block of the current model.

To Add a New Constant

To add a new constant value (independent of a particular species) and make it available to all your Thermolib models, simply add the value under SMediaData.SGlobal:

```
SMediaData.SGlobal.new_constant = 1234
```

To make the new species available to your model, first save the adjusted SMediaData in a mat file of your choice:

```
save MyChemicalMediaData.mat SMediaData
```

Then use that file in Model Setup (see help for "Model Setup" block).

Note that all fields from `SMediaData.SGlobal` will be exported to single workspace variables with the prefix "th_const_". So our new constant will also generate a workspace variable "th_const_new_constant".

An easier and more user friendly way to view and adjust the `ChemicalMediaData` is to use a Microsoft Excel sheet, if you have Microsoft Excel available.

4.4 CHEMICAL MEDIA DATA IN MS EXCEL

Chemical Media Data is also stored in the form of an MS Excel database. To open the default Chemical Media Data Excel sheet, go to the Thermolib directory and then under the directory Data, open the file `ChemicalMediaData.xls`. A clipped view of the database is shown in Figure 1. The actual table is bigger and includes more species and more properties than this clipped view.

In the first row under the title "Chemical Media Data", the species are listed, each taking a separate column. Here we see the first three entries: CH_4 , H_2O and CO . The chemical and thermodynamic properties are listed in the dark blue column. The names appearing in this column are the variable names used in MATLAB and Simulink by Thermolib. On the left of the properties column, a description of the property and its unit are given. For instance, according to the table, the molar mass of carbon monoxide (CO) is 28.010 kg/kmol, this value will be stored in MATLAB workspace under the variable name "th_M". Also from the table, you will find that the liquid density of methane (CH_4) is 422620 g/m³.

Chemical Media Data					
		Species	CH4	H2O	CO
		formula			
		full_name	methane	water	
Chemical media data					
Molar Mass	[kg/kmol]	M	16,043	18,015	28,010
Liquid density	[g/m³]	rho_liq	422620,0	1,00E+06	7,89E+05
Antonie equation_A	[-]	Antoine_A	3,9895	5,40221	3,54
Antonie equation_B	[-]	Antoine_B	443,028	1838,675	99,395
Antonie equation_C	[-]	Antoine_C	-0,49	-31,737	7,726
Reference temperaure for evap. properties	[K]	T_fg_ref	111,6	298	81,6
Evaporation enthalpy	[Joule/mole]	hm_fg	8183	43961,4	6042
Evaporation entropy	[Joule/mole-K]	sm_fg	73,32437276	147,5214765	74,04411765
Critical Temperatur	[K]	T_c	190,600	647,300	132,900
Critical Pressure	[Pa]	p_c	4,60E+06	2,21E+07	3,50E+06
Critical Volume	[m³/mol]	vm_c	9,90E-05	5,60E-05	9,31E-05
Acentric Factor	[-]	omega	0,288	0,344	0,049
Red. De-Broglie-Wavel.	[-]	lambda_R	999,000	999,000	999,000
Critical Real-Factor	[-]	Zc	0,288	0,229	0,295
Linear Molecule	[-]	linear	1	0	1
Dipole Moment	[Debye]	mue	0,0	1,8	0,1
Boiling Temperature	[K]	Ts	111,7	373,2	81,7
Inner Rotation	[-]	c_mir	999,0	-3,0	999,0
Number of C atoms	[-]	C_atoms	1	0	1
Number of H atoms	[-]	H_atoms	4	2	0
Number of O atoms	[-]	O_atoms	0	1	1
Number of N atoms	[-]	N_atoms	0	0	0

Figure 1: Detail of "ChemData" sheet in Chemical Media Database ChemicalMediaData.xls

In a second sheet in ChemicalMediaData.xls named "Constants" a set of nature constants are listed (Figure 2):

Constants		Value
Boltzmann-Constant [Nm/K]	k	1,380E-23
Avogardo-Constant [1/mol]	N A	6,022E+23
Universal Gas Constant [J/mol K]	R	8,314
Elementary electron charge [AS]	e	1,60E-19
--		
--		
--		
--		
--		

Figure 2: Constants in Chemical Media Database

It is similar in form to the Chemical Media Data sheet. Here, the dark blue column contains the names of the constants. The white column on the right contains the values for each constant whereas the left column contains a description and the unit.

Changes to the Chemical Media Data can be done directly in the Excel sheet.

Note that these constants are not directly used. They have to be imported to a .mat file to be used in a Simulink model using the "th_GetChemicalMediaData" command.

To Add a New Species

To add a new species, simply insert a new column next to the last listed Species and fill in the rest of the properties:

Chemical Media				
Species	n-Butane	i-Pentane	R12	NEW_COMPOUND
formula	C4H10	C5H12	CCl2F2	add formula here
full_name		2-METHYLBUTANE	Dichlorodifluoromethane	add name here
NASA_A1_g	9,4453583400E+00	1,2327787000E+01	1,0708248000E+01	add property here
NASA_B1_g	2,5785807300E-02	3,0613087000E-02	2,3232186000E-03	add property here
NASA_C1_g	-9,2361912200E-06	-9,8415785000E-06	-9,0073223000E-07	add property here
NASA_D1_g	1,4863275500E-09	1,3919776000E-09	1,5261702000E-10	add property here
NASA_E1_g	-8,8789715800E-14	-7,0337345000E-14	-9,4434958000E-15	add property here
NASA_F1_g	-2,0138216500E+04	-2,5037492000E+04	-6,3102602000E+04	add property here
NASA_G1_g	-2,6347007600E+01	-4,1133494000E+01	-2,6622876500E+01	add property here
NASA_A2_g	6,1474680600E+00	1,0832882000E+00	3,8134966000E+00	add property here
NASA_B2_g	1,5594738900E-04	4,4571076000E-02	2,0036835000E-02	add property here
NASA_C2_g	9,6791351700E-05	8,2389934000E-06	-9,8986693000E-06	add property here
NASA_D2_g	-1,2548391000E-07	-3,5258047000E-08	-8,7995353000E-09	add property here
NASA_E2_g	4,9781655500E-11	1,5785762000E-11	7,1218552000E-12	add property here
NASA_F2_g	-1,7599440200E+04	-2,0807535000E+04	-6,1253551000E+04	add property here
NASA_G2_g	-1,0940987900E+00	2,1795155000E+01	8,9909711000E+00	add property here

Figure 3: Adding a new Species to the database

Save the adjusted Excel sheet under a new name: `MyChemicalMediaData.xls`. To export this adjusted media data to MATLAB and make it available to your model, use the command "th_GetChemicalMediaData" described below.

To Add a New Property

To add a new property to the species, simply add a new entry in the dark blue column and the corresponding values for each of the species:

Chemical Media Data				
Species	CH4	H2O	CO	
formula				
full_name	methane	water		
Zc	0,288	0,229	0,295	
linear	1	0	1	
mue	0,0	1,8	0,1	
Ts	111,7	373,2	81,7	
c_mir	999,0	-3,0	999,0	
C_atoms	1	0	1	
H_atoms	4	2	0	
O_atoms	0	1	1	
N_atoms	0	0	0	
F_atoms	0	0	0	
Cl_atoms	0	0	0	
New Property	New property value	New property value	New property value	

Figure 4: Adding a new property to the database

Save the adjusted Excel sheet under a new name: `MyChemicalMediaData.xls`. To export this adjusted media data to MATLAB and make it available to your model, refer to the sections below.

To Add a New Constant

To add a new constant, go to the sheet "Constants" and insert the new constant in a new row as such:

Chemical Media Data		Value
k		1,380E-23
N_A		6,022E+23
R		8,314
e		1,60E-19
NEW_CONSTANT		1234

Figure 5: Adding a new constant to the database

Save the adjusted Excel sheet under a new name: `MyChemicalMediaData.xls`. To export this adjusted media data to MATLAB workspace and make it available to your model, refer to the sections below.

To Add a New Element

To add a new element, two things are necessary. First, go to sheet “Elements” and insert a new column with the desired element. Configure, if the element should be used as fuel in the component “LambdaBurner”. If this is true, the combustion product and combustion product coefficient per atom need to be specified.

Elements			
element	C	H	NEWELEMENT
name	carbon	hydrogen	new_element
allowed_in_fuel	1	1	0
combustion_product	CO2	H2O	-
combustion_product_coefficient_per_atom	1	0,5	-

Figure 6: Adding a new element to the database

As second step, go to sheet “ChemData” and add a new row for the number of “NEWELEMENT” atoms in the existing species.

2			Chemical Media Data				
3	Yes		Species		CO	CO2	H2
4	No		comment				
5	No		formula				
6	No		full_name				hydrogen
7			species_type				
56	Yes	Number of S atoms [-]	S_atoms		0	0	0
57	Yes	Number of Ar atoms [-]	Ar_atoms		0	0	0
58	Yes	Number of NEWELEMENT atoms [-]	NEWELEMENT_atoms		0	0	0
59	Yes	Triple-point temperature [K]	T_triple		68,16	216,592	13,957

Figure 7: Adding new element to species database

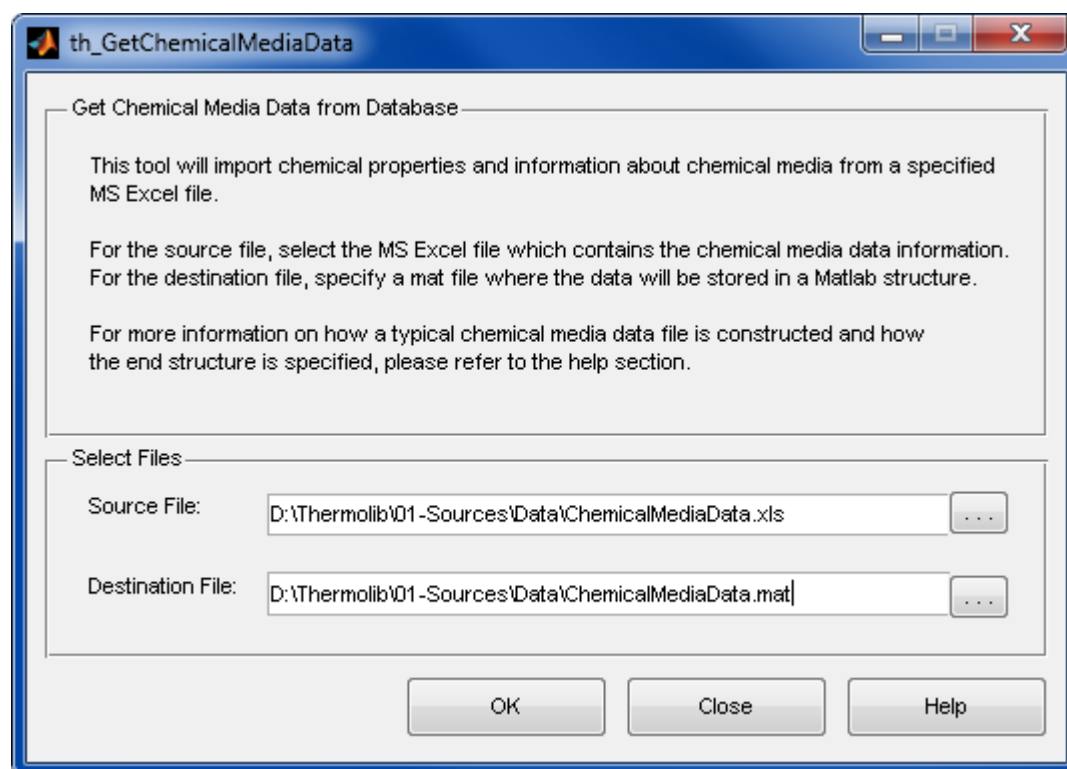
Import Chemical Media Data from Excel

Once you have saved the new Chemical Media Data in an MS Excel file, Thermolib can read the file and import the new data into a SMediaData structure.

To do this, go to the Matlab command line and type:

```
th_GetChemicalMediaData
```

This will open the following dialog:



Source File: select the MS Excel file where the new chemical media data is stored.

Destination File: Enter a name for a mat file where the imported SMediaData structure will be stored.

Once you hit the OK button, Thermolib will read the constants, all of the Species and their properties that are defined in the Excel sheet specified under Source File. It will create a structure SMediaData and store it in the mat file specified under Destination File.

To ensure that no loss of data happens, we recommend that the user do not overwrite the default files (ChemicalMediaData.xls and ChemicalMediaData.mat) delivered with Thermolib.

To make the new species available to your model, use the newly created mat file (Destination File) in Model Setup.

4.5 WHERE DO I GET THE DATA?

Thermolib uses only well documented thermophysical properties. They can be looked up in public databases like

- NIST webbook: <http://webbook.nist.gov/chemistry/>
- Korea Thermophysical Properties Data base <http://www.therc.org/kdb/>
- Burcat's Thermodynamic Data at <ftp://ftp.technion.ac.il/pub/supported/aetdd/thermodynamics/> or its mirror <http://garfield.chem.elte.hu/Burcat/burcat.html>

For compounds, which are not contained in the public databases, you can use several (paid) data services or use your own experimental data.

As a starting point for experimental data use the heat capacity function dependend on temperature. Fit the experimental data with the NASA polynomial approach above (for gases) or with the polynomial heat capacity for liquid phase.

You may then decide to use this compound only in one of the phases and adjust the parameters of the vapor pressure curve accordingly.

If you are not interested in simulating chemical reactions with your species, then the enthalpy of formation (offset in enthalpy) need not to be adjusted.

For complex synthetic fluids of unknown composition you may not have data for molar mass available. For this case we recommend to use this as pure substance only and only use mass sources and display mass flow instead of molar flows. Note that you have to calculate your molar properties (heat capacity, enthalpy etc.) with respect to this molar mass to end with the (mass) specific properties, which are more likely known.

You can look up which property values are used by the individual States-and-Property-Blocks in the document [ChemicalMediaData_Dependency_StatesAndProperties.pdf](#)

4.6 HOW DO THE DATA COME INTO MY SIMULINK MODEL?

The Chemical Media Data are stored in the structure SMediaData, which is loaded into the Matlab base workspace as soon a model is loaded or started. The question remains how does the model retrieve the information it needs from this structure.

In Model Setup, the user defines the mat-file where SMediaData is stored, and also a list of Species that will be available to the current Model. The Thermolib model will then use this list of Species to extract the data it needs from SMediaData and store it in variables in the base workspace. The example below illustrates how this works.

Example:

A Thermolib model requires the block Model Setup. Double clicking the Model Setup block opens the following dialog:

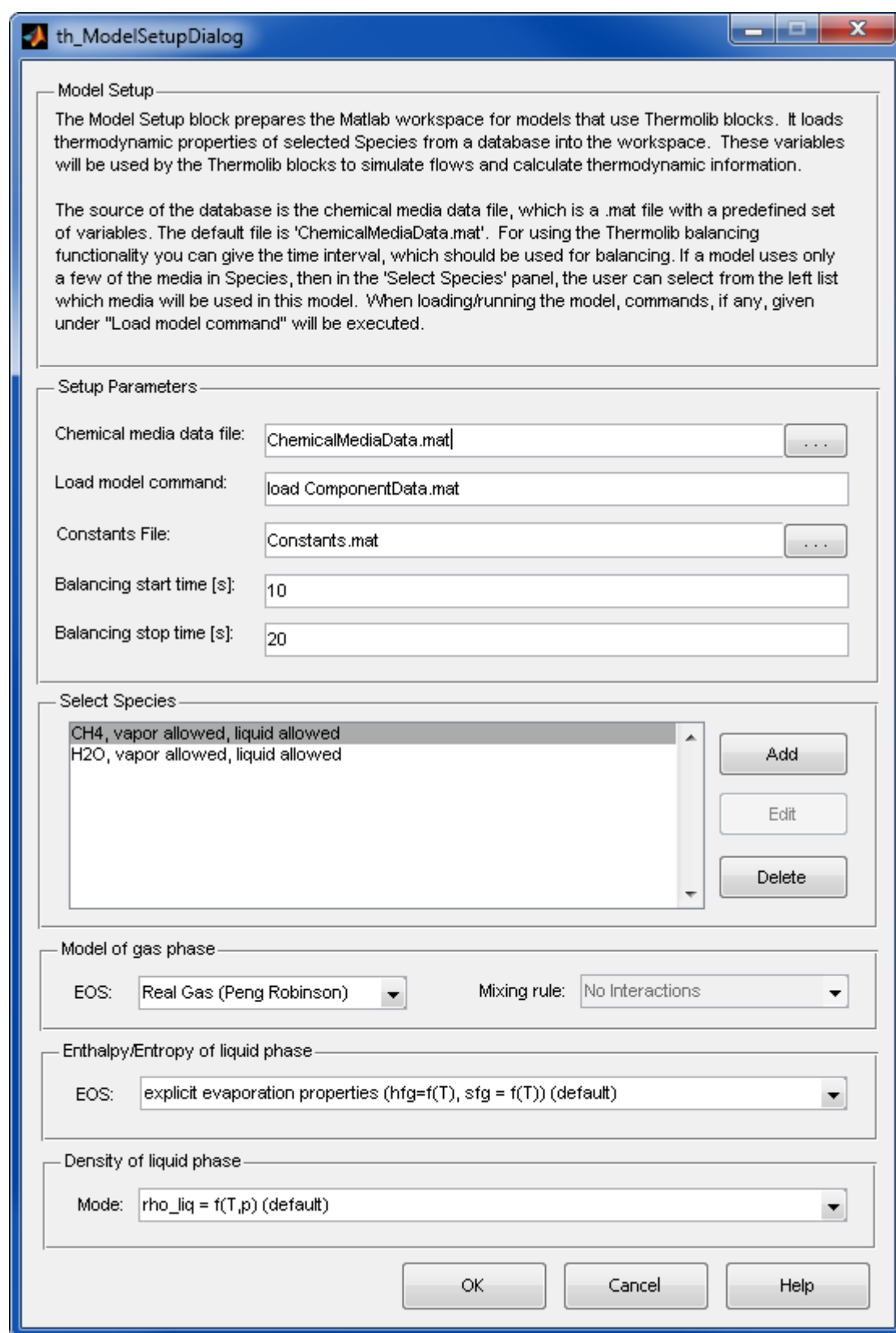


Figure 8: Model Setup dialog

The defined Chemical Media Data file ChemicalMediaData.mat has the structure SMediaData which includes information over 19 different Species.

The user indicated in Model Setup that this model will only use methane (CH_4) and water (H_2O).

When the user clicks ok, Thermolib will extract the data only for methane and water and save it in the base workspace variables in the following manner:

```
Species = {'CH4' 'H2O' }    Species tells the model which species are available  
th_M = [16.043 18.015]      th_M indicates what is the molar mass of each species  
th_rho_liq = [422620 1000000]    th_rho_liq indicates what is the liquid density
```

Note that the name of the variable corresponds to the name of the field under SMediaData.SSpecies with a "th_" prefix. Also note that since the compound water is the second element in Species, it's molar mass (or any other property) is stored in the second element of the property's vector.

This whole process is transparent to the user. Once the Model Setup block is set properly, Thermolib will take care of creating the necessary variables.

5 MODELING PRESSURE FEEDBACK

Modelling of mass flow rate based on pressure information in flow network is a feature of Thermolib, but it is suitable for more experienced users. It should be used if mass flow rate is unknown and should be determined using pressure conditions.

5.1 WHEN DO I NEED PRESSURE FEEDBACK?

Suppose the following hydraulic network:

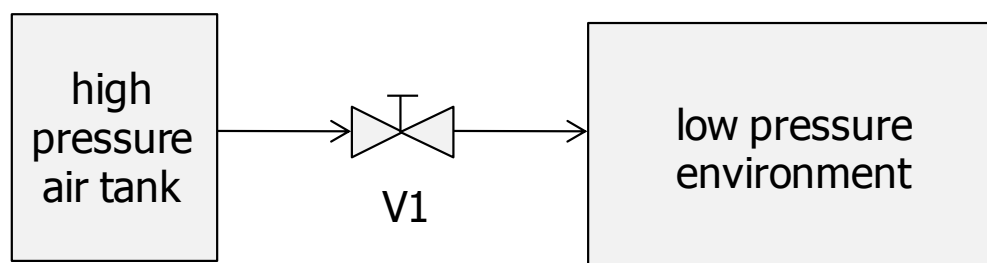


Figure 9: Simple air flow network

In this process the amount of air flowing from the high pressure tank to the low pressure environment is unknown and ruled by the valve position. If the valve is closed, then no flow will go through the valve.

Suppose we model the three elements in the figure with single blocks from Thermolib, then obviously there have to be a way, how the valve position (or it's effects) have to be feed back to the high pressure tank. This is because the outgoing flow of the high pressure tank has a mass flow assigned and the block "Tank" has to decide how much mass will flow.

In such cases, where

- there is an effect of valve positions or downstream components on the flow at the beginning of the network, or
- you do not know the mass flow through you process, but you want to model this as a result of pressures at the boundaries

you need to model your system with pressure feedback.

5.2 HOW TO CREATE MODELS WITH PRESSURE FEEDBACK?

To create models with pressure feedback, you can use the components from "Hydraulic with pressure feedback" in Thermolib. Other blocks are the Tanks in the "Tanks" section of Thermolib.

In our example we will use an "Liquid Tank" with the "constant pressure" option to model the high pressure tank. The valve is modeled by the "Valve" block. The outlet to constant pressure is modeled by a "Downstream" block. This block serves as boundary condition for the pressure equations. The outlet pressure is here entered to the mask.

The blocks are connected in the forward direction as you would do without pressure feedback. Then the PFout port of the Downstream block is connected to the PFin port of the Valve and the PFout port of the Valve is feed back to the PFin port of the tank. Note that you have to chose the "m.out = f(PF)" option in the tank to see the PFin port.

If you simulate this model with different valve positions, you can see how the mass flow will be adjusted to match the boundary condition at the outlet (see example \Examples\PressureFeedback\DemoFlowSimple.mdl).

Sometimes you want to simulate networks, where the flow is divided into two branches and they will come together after passing some elements.

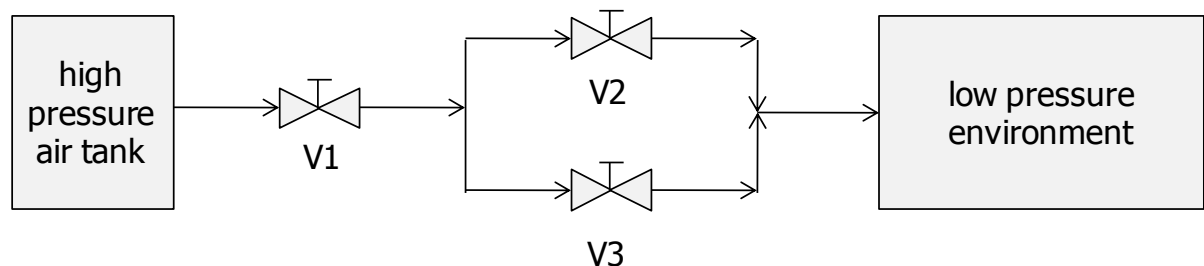


Figure 10: Pressure network with two branches

For the point, where the flow splits into two branches, use a "Splitter" block to model this situation. The point, where the branches join again can be simulated using a simple mixer block.

5.3 HOW THE PRESSURE EQUATIONS ARE SOLVED BY THERMOLIB?

Pressure information in the flow network is gathered using the PFin/PFout ports of the blocks. Most blocks do not change this signals, but feed them simply back to their predecessor in the flow network.

For solving the pressure equations, the key elements are iteration algorithms contained in the following blocks:

- Tanks
- Splitter

- Circulation Pump and Compressor

The iteration in the Tank adapts the outgoing mass flow to the pressure feedback information it gets via the PFin port. Therefore you find some parameters for the pressure feedback iteration in the Tank masks. There is:

- Maximum mass flow
- Initial mass flow
- Relaxation factor

The maximal mass flow limits the iteration algorithm at the upper end. The initial mass flow serves as starting point at $t=0$. The relaxation factor controls how aggressive the iteration changes the mass flow. If you increase this factor the iteration is faster, but may have overshoots. If you decrease this factor the iteration is slower, but with less overshoot.

The tank block feeds a current candidate for the mass flow through the network. Then each block in the network calculates the corresponding pressure drop and reduced the pressure in the flow until reaching another tank or a downstream block. In both blocks the pressure should reach a certain value, when it comes into this block:

- In Downstream block a downstream pressure is configured in mask.
- In a tank the current tank pressure should be reached.

The difference of this pressure, which it should be, and the incoming pressure (which is function of the mass flow the upstream tank tried) is then feed back through the network. The upstream tank then knows that he has to increase the mass flow to reach a lower pressure at the end of the network (because of higher pressure drop) or to decrease mass flow to reach a higher pressure at the end of the network.

The splitter has a different task from the tank. It does not influence the overall mass flow, but only changes the partitioning of the incoming mass flow to the two outlets. This is the free variable. In the case the two branches will come together in a mixer, the splitter tries to modify the splitting such, that the pressures of the two branches will be the same at the inlet of the mixer. In case the two branches will end in different Downstream blocks or tanks, the splitter tries to manage the flow such that it solves the corresponding downstream pressure condition. It will also create a corresponding feedback to its predecessor to tell if the overall mass flow has to be adapted.

Additionally to the pressure drop information the PF signals also include a flag, which signals if downstream any mass flow is accepted. If you place a valve there, which is completely closed, then the mass flow from upstream is immediately stopped. Use the block "Valve" from "Hydraulic with Pressure Feedback" for such cases.

If you want to model flow circles like they occur often in cooling circuits and you want to model them with pressure feedback, then you can use a "Circulation Pump". This block acts as a boundary condition like the "Downstream" block and iterates the outgoing mass flow like a Tank. This avoids the necessity for a Downstream block in hydraulic circuits.

Unfortunately there is no possibility to do the iteration with evaluating the complete pressure network in one simulation time step (except the user would draw an iteration subsystem around its complete system, which we find not appropriate). Therefore the iteration takes place in time domain. This means it will take some time steps, until the pressure equations are solved. Keep this in mind if you interpret the simulation results. Note, that iterations are necessary, because the user defined compressor maps and pump characteristics cannot be solved algebraic.

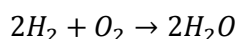
To get a better understanding of how complex pressure feedback networks can be modeled we highly recommend to have a look to the examples provided in `\Examples\PressureFeedback` in your Thermolib installation folder.

6 THERMODYNAMIC BALANCING

This chapter explains how you can get thermodynamic balancing information after a simulation run of a Thermolib model.

6.1 WHY DO YOU NEED MODEL BALANCING?

In a typical thermodynamic model, flows are regularly split and merged, and compounds inside the flows react with each other to create new ones or replace older compounds. But throughout all these processes, the net total amount of the basic elements (H, C, O, N, F and Cl) cannot change inside a closed system. To illustrate through a simple example, consider a reactor where hydrogen gas and oxygen gas react to produce water.



If this system receives 20 moles of H-atoms and 10 moles of O-atoms in total, then the output must contain also 20 moles of H-atoms and 10 moles of O-atoms in total, even if in a different form – in this case they exit the reactor as water molecules instead of hydrogen and oxygen gas molecules.

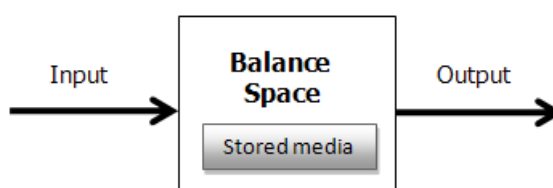
When modeling a system as simple as this, one can easily keep track if the model is correct by keeping track of the net input and output of hydrogen and oxygen atoms; however, as the model becomes more complex with more reactions, pumps, valves and various other components, the task of keeping track that the net total of elements is correct becomes tedious.

6.2 MODEL BALANCING BASICS

Model-balancing aims to ensure that in a complex model the net total of all basic elements remains correct: the mass of hydrogen (also carbon, oxygen, nitrogen, fluoride and chloride) inside a defined balance space remains the same, regardless of all the reactions, splitting and merging taking place. Additionally, thermodynamic balancing gives you an overview of where the main sinks and sources of energy and mass are, and where important conversions take place.

Balance space

A balance space is a part of the modeled system that is completely defined by its borders and the energy and mass flows crossing these borders. In a balance space, the net balance of basic elements, as well as the net balance of energy, remains zero.



The balancing takes place over a defined time period between $t = t_1$ and $t = t_2$. This period is referred to as the *balancing period* and is defined in Model Setup. We start with the element hydrogen and define the following:

\dot{m}_{H-in} : The total mass flow of hydrogen entering the balance space [g/s].

\dot{m}_{H-out} : The total mass flow of hydrogen exiting the balance space [g/s].

$m_{H-stored@t_1}$: The total mass of hydrogen stored in the balance space at $t = t_1$ [g].

$m_{H-stored@t_2}$: The total mass of hydrogen stored in the balance space at $t = t_2$ [g].

$$\int_{t_1}^{t_2} \dot{m}_{H-in} dt + m_{H-stored@t_1} = \int_{t_1}^{t_2} \dot{m}_{H-out} dt + m_{H-stored@t_2}$$

Essentially, the above formula says that within the balancing time period, the total mass of hydrogen that entered the balance space must be equal to the total mass of hydrogen that exited that balance space plus the change in hydrogen mass that remained within the balance space.

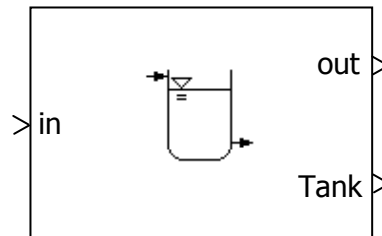
Similarly, the above is repeated for the five other basic elements: Carbon (C), Oxygen (O), Nitrogen (N), Fluoride (F) and Chloride (Cl):

$$\begin{aligned}\int_{t_1}^{t_2} \dot{m}_{C-in} dt + m_{C-stored@t_1} &= \int_{t_1}^{t_2} \dot{m}_{C-out} dt + m_{C-stored@t_2} \\ \int_{t_1}^{t_2} \dot{m}_{O-in} dt + m_{O-stored@t_1} &= \int_{t_1}^{t_2} \dot{m}_{O-out} dt + m_{O-stored@t_2} \\ \int_{t_1}^{t_2} \dot{m}_{N-in} dt + m_{N-stored@t_1} &= \int_{t_1}^{t_2} \dot{m}_{N-out} dt + m_{N-stored@t_2} \\ \int_{t_1}^{t_2} \dot{m}_{F-in} dt + m_{F-stored@t_1} &= \int_{t_1}^{t_2} \dot{m}_{F-out} dt + m_{F-stored@t_2} \\ \int_{t_1}^{t_2} \dot{m}_{Cl-in} dt + m_{Cl-stored@t_1} &= \int_{t_1}^{t_2} \dot{m}_{Cl-out} dt + m_{Cl-stored@t_2}\end{aligned}$$

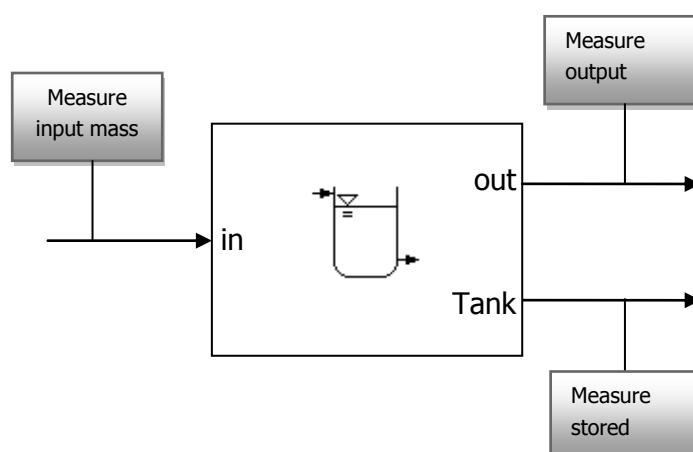
This balancing is done automatically in Thermolib using the provided balance blocks. The remainder of this chapter will explain that in detail.

6.3 THE BALANCE BLOCKS CONCEPT

To illustrate the balance block, we will consider that our balance space is a simplified liquid tank. It has one input, one output and a storage area where excess inflow is stored. The storage is indicated by a second output "Tank" where the current state of the tank is given.



To keep track of the mass of elements entering the tank, exiting it and those that remain within, we need a measurement device to track the elements at the relevant inputs and outputs, in this case the "in" input port, the "out" output port and the "Tank" output port.



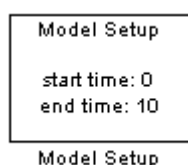
The balance blocks in Thermolib are the equivalent of these “measurement” blocks. In essence, define your balance space: what are the inputs, the outputs and the storage ports, then attach a balance block to these ports. During the simulation, the balance blocks will track the needed mass flows without any further attention from the user. At the end of the simulation, you can run the balancing command `th_ExportBalance` that will gather information from all these measurement blocks, perform the balancing and save the results in a MATLAB structure that can be easily exported to MS Excel or an ASCII file. Sections XXX and XXX explain this in further detail.

6.4 MODEL BALANCING IN THERMOLIB - HOW TO

Defining the Balancing Period

The balancing is checked within a defined time period. For example, a user can decide to check if the mass balance of the elements is correct between the start of the simulation up till 10 seconds of simulation time. If the net mass during the first 10 seconds comes out as zero, he/she may decide that the model is correct.

The balancing time period is defined in the “Model Setup” block. The user enters the start time (in seconds) and the end time (also in seconds).



Model Setup

The Model Setup block prepares the Matlab workspace for models that use Thermolib blocks. It loads thermodynamic properties of selected Species from a database into the workspace. These variables will be used by the Thermolib blocks to simulate flows and calculate thermodynamic information.

The source of the database is the chemical media data file, which is a .mat file with a predefined set of variables. The default file is 'ChemicalMediaData.mat'. For using the Thermolib balancing functionality you can give the time interval, which should be used for balancing. If a model uses only a few of the media in Species, then in the 'Select Species' panel, the user can select from the left list which media will be used in this model. When loading/running the model, commands, if any, given under "Load model command" will be executed.

Setup Parameters

Chemical media data file:

Load model command:

Constants File:

Balancing start time [s]:

Balancing stop time [s]:

Select Species

CH4, vapor allowed, liquid allowed
H2O, vapor allowed, liquid allowed

Add
Edit
Delete

Model of gas phase

EOS: Mixing rule:

Enthalpy/Entropy of liquid phase

EOS:

Density of liquid phase

Mode:

OK Cancel Help

What this means is, during the simulation between the start time and the end time, the balance blocks actively measure the mass flow at their respective ports. Before the start

time and after the end time, the balance blocks are inactive and do not store any information. At the end of the simulation, the measurements collected are only those measured between the start time and the end time.

How to Balance a Model

This tutorial will show you how to balance a Thermolib Model.

1. Start by opening the example DemoFlowSimple by typing DemoFlowSimple at the Matlab command line.
2. Set the simulation stop time to 100 seconds.
3. Open the model setup block and set the start time to zero and end time to 100. In this case we will balance the model for the entire runtime.
4. Run the model
5. Go to the MATLAB workspace. You will see that a structure SModelBalance has been created.
6. Call the command line function th_ExportBalance (see below).

The structure SModelBalance has a field SGlobal where information about the simulation is stored: the name of the model, the simulation date and time, and which Species were used in this particular run.

Note: Thermolib models always write the balance data to the structure SModelBalance. So if you simulate a model and then simulate it a second time, the SModelBalance created from the first run will be over-written. If you wish to do the balancing at a later time, we recommend you save SModelBalance to a mat file so that you can access it anytime.

th_ExportBalance

To do the balancing, Thermolib offers the following command line function:

```
SModelBalance_out =  
th_ExportBalance(SModelBalance,xls_file,ascii_file)
```

where

SModelBalance: structure created by the model after simulation.

xls_file: Microsoft Excel file to which the balancing results are exported

ascii_file: Text file to which the balancing results are exported

SModelBalance_out: structure returned by the balance function that includes all of the balancing results.

Note: if you do not wish to export to an Excel file or a text file, set the corresponding entry in the function call to NaN or empty ([]).

Example:

After running the example DemoFlowSimple, we balance the model using the following command:

```
SModelBalance_out = th_ExportBalance (SModelBalance,
'BalanceResults.xls', 'BalanceResults.txt')
```

If you do not have Microsoft Excel installed, simply leave the entry for the xls_file empty:

```
SModelBalance_out = th_ExportBalance (SModelBalance, [],
'BalanceResults.txt')
```

This command will balance the model and export the results to the files BalanceResults.xls and BalanceResults.txt in your working directory.

The output structure SModelBalance_out contains the results of the balancing.

6.5 BALANCE RESULTS IN EXCEL

If you open the Excel balance file created from the last example, you will see that two sheets have been created: SMass and SEnergy. As the names indicate, SMass contains the mass balance and SEnergy contains the energy balance.

The Balance Sheet

An example of what the mass balance sheet looks like is shown below:

Balance Space Name	Port Name	Port Direction	CH4 [mol]	H2O [mol]	H [mol]	C [mol]	O [mol]	N [mol]	F [mol]	Cl [mol]
TotalBalance			0	-4,4E-11	-8,7E-11	0	-4,4E-11	0	0	0
DemoFlowSimple/Downstream										
	Flow in	in	0	3323,945	6647,89	0	3323,945	0	0	0
	Flow out	out	0	3323,945	6647,89	0	3323,945	0	0	0
Balance			0	0	0	0	0	0	0	0
DemoFlowSimple/Liquid Tank										
	tank in	in	0	0	0	0	0	0	0	0
	tank out	out	0	3323,945	6647,89	0	3323,945	0	0	0
	tank content	stored_start	0	55509,3	111018,6	0	55509,3	0	0	0
	tank content	stored_end	0	52185,35	104370,7	0	52185,35	0	0	0
Balance			0	-4,2E-11	-8,5E-11	0	-4,2E-11	0	0	0
DemoFlowSimple/Pressure Valve 1										
	Flow in	in	0	3323,945	6647,89	0	3323,945	0	0	0
	Flow out	out	0	3323,945	6647,89	0	3323,945	0	0	0
Balance			0	0	0	0	0	0	0	0

Figure 11: Mass balance in Excel

Each balance space has its own entry under the column Balance Space Name. In this example, the balance spaces are Thermolib blocks. For instance, the second block listed is the Liquid Tank (1). Under the block name, each of the block's ports along with the

direction is shown (2). In the case of the liquid tank, you can see the input to the tank, the output, the stored content at the start of the balancing period and the stored content at the end.

Next to that we see how much mass of each species passed through the port. We see that there was no input of water to the tank, indicated by the zero. The tank however output 3323.95 moles of water during the balancing period. At the beginning, it contained 55509.3 moles of water but by the end of the simulation there were only 52185.35 moles left. The balance of the block itself is shown in the row marked "Balance" where we see it adds up to almost zero. The reason it is not exactly zero is due to inaccuracies that result from integration. The balance of the whole model is shown in the top row in red.

Finally on the right side of the sheet, you can see the balancing of basic elements (3). A quick glance at the top row in red shows that the balance of the elements is zero (or very close to it in the case of hydrogen and oxygen). This indicates that the model balancing is correct. No mass of hydrogen or oxygen was lost during the simulation.

If you do not have Excel, you can view the results in text-form by opening the file BalanceResults.txt:

```

Model name: DemoFlowSimple
Simulation time: 22-Jul-2009 10:58:43

```

Balance Space Name	Port Name	Port Direction	CH4 [mol]	H2O [mol]	H [mol]
TotalBalance			0.000	-0.000	-0.000
DemoFlowSimple/downstream					
	Flow in	in	0.000	3323.945	6647.8
	Flow out	out	0.000	3323.945	6647.8
Balance			0.000	0.000	0.000
DemoFlowSimple/Liquid Tank					
	tank in	in	0.000	0.000	0.000
	tank out	out	0.000	3323.945	6647.8
	tank content	stored_start	0.000	55509.298	111018
	tank content	stored_end	0.000	52185.353	104370
Balance			0.000	-0.000	-0.000
DemoFlowSimple/Pressure Valve 1					
	Flow in	in	0.000	3323.945	6647.8
	Flow out	out	0.000	3323.945	6647.8
Balance			0.000	0.000	0.000

Figure 12: Mass balance in ASCII

6.6 USING THE BALANCE BLOCKS

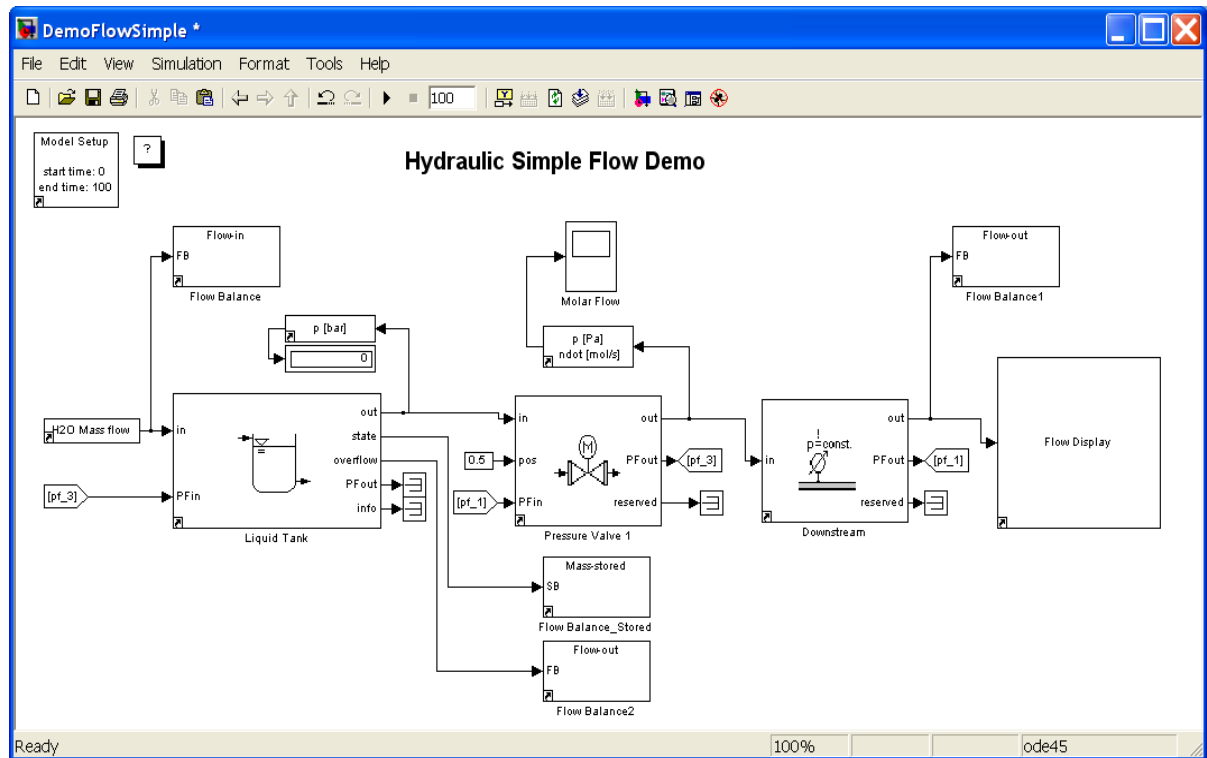
Thermolib has four basic blocks used for balancing:

- Flow Balance: tracks the mass and energy content of flows (Flow Buses).
- Flow Balance Stored: tracks the mass content of stored species (State Buses).
- Energy Balance: tracks the energy content of power signals.
- Energy Balance Stored: tracks the energy content of stored energy.

Connect these Thermolib balance blocks to the inputs and outputs of your defined balance space or system to balance it. T

Track follow steps:

3. Start by opening the Thermolib model Hydraulic Simple Flow Demo by typing DemoFlowSimple at the Matlab command line. This opens the following model:



4. In the Model Setup block, set the balancing start time to 0 and the balancing end time to 100.
5. Now attach Thermolib balance blocks to the input, output and storage ports of our balancing space: in this example, the balance space is the entire model, so the input to the space is the input to the tank. Connect a flow balance block. The output of the balance space is the output of the Downstream block, so connect a flow balance block there too. This is not the only output, because if the liquid tank overflows, mass flow will come from the “overflow” output port of the tank. So connect a flow balance block there. Finally the system stores mass inside the tank; to include that in the balancing, connect a stored balance block to the Tank’s output “state”.

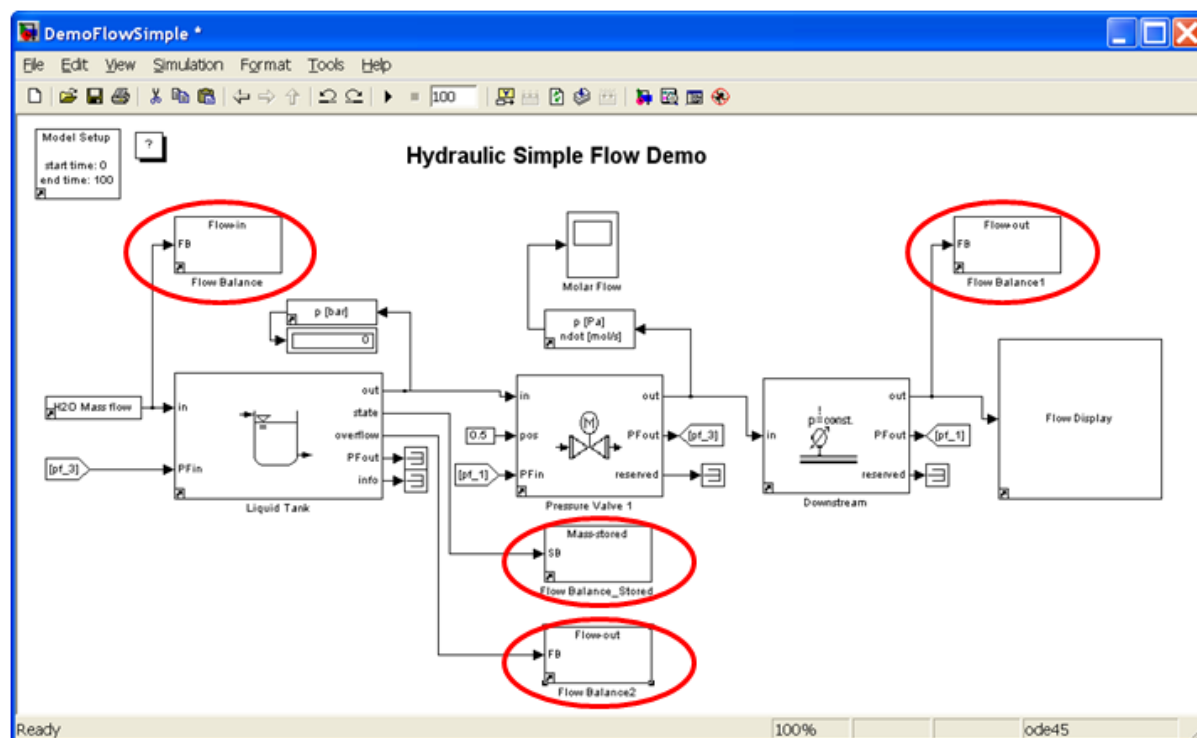
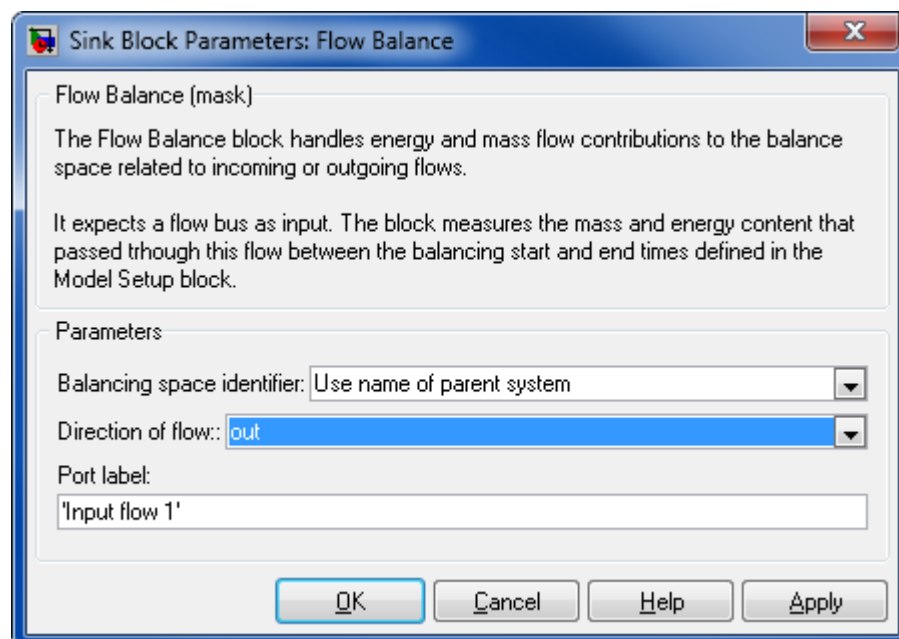
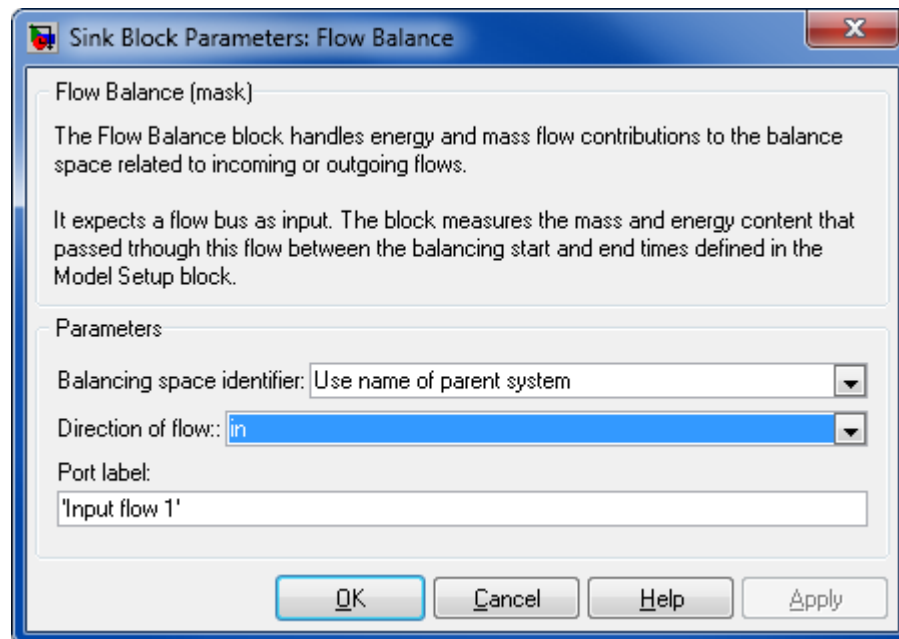
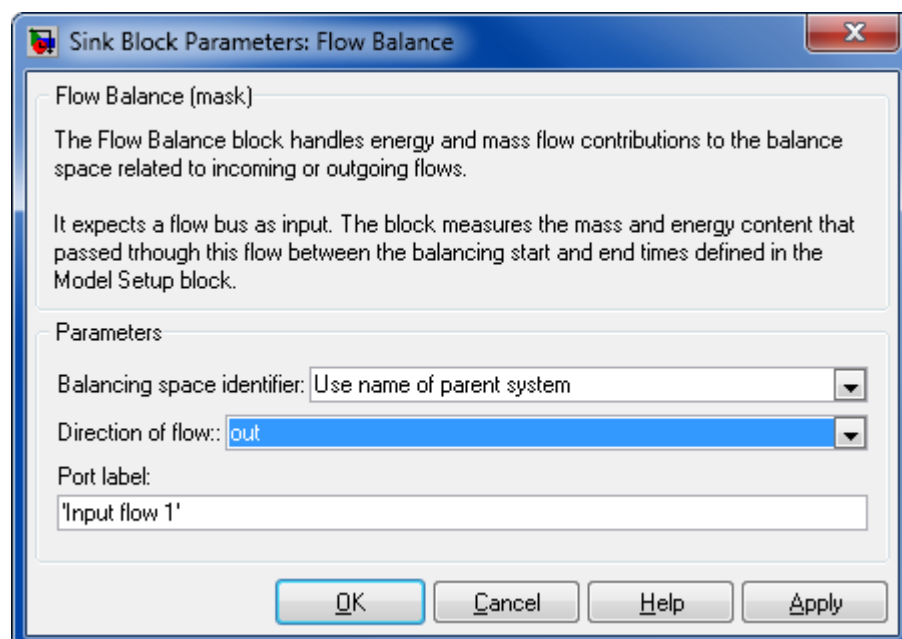
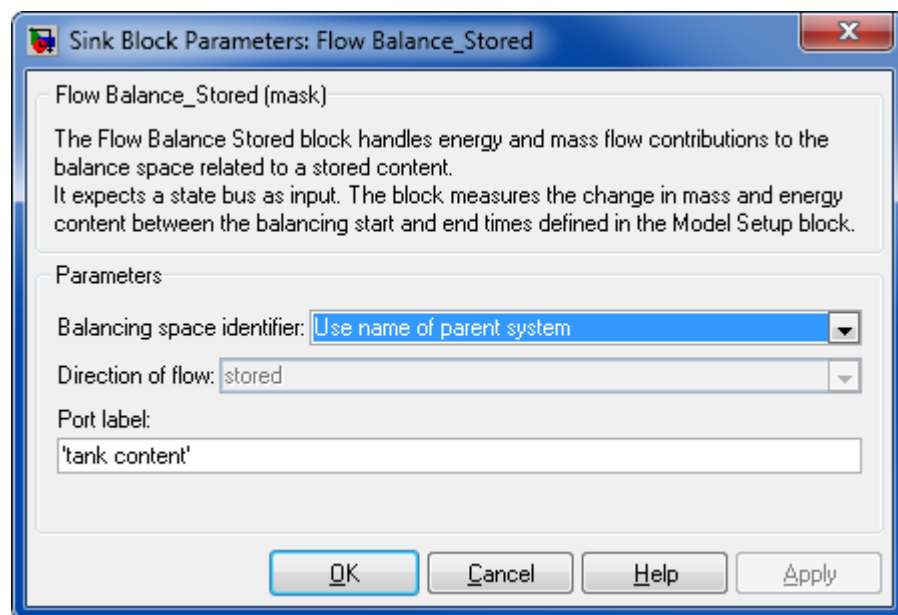


Figure 13: A user defined balance space using balancing blocks

6. Now that the balance blocks are connected, we need to adjust their parameters. First, we have to identify or define our balance space. Here two options are available, either we used the name of the parent system (in this case the model itself) or we enter a user-defined name. In the case of the latter, the name should be exactly the same (spelling- and case-wise) for all the of four blocks. For this tutorial, we will use a user-defined name: "Hydraulic Flow Balance Space". Double click on each of these blocks to fill the fields in the mask, to name the signals, to make the final balancing results more readable.
 - For balancing space identifier, select "Enter a user-defined name". A field for the balance space name appears below it. There, enter 'Hydraulic Flow Balance Space' with the apostrophe signs as shown.
 - Next we have to define the direction of flow. For the stored mass block, the direction is always "stored". For the flow balance blocks, define the direction as "in" for the inputs to the balance space, and "out" for the outputs of the balance space. Finally for each port, we have to define the label. This label is later used as an identifier for the port. Enter the port labels are shown in the dialogs below.





7. Run the model: all information about flow mass or energy in every port is saved now in a MATLAB variable "SModelBalance".
8. You can balance the Model with the `th_ExportBalance` command in the Matlab command window:

```
output = th_ExportBalance (SModelBalance, xls_file,  
                           ascii_file)
```

- saves the results of balancing in "output" and exports those in to Excel file "xls_file" and to ASCII file "ascii_file"
- output: Matlab structure with flowing information of every port and the balancing result too.
- xls_file: the full path of the folder, where the Excel file must be saved, inclusive the file name, if you give only the name of Excel file, this file will be created in the same folder as you actually are arranged
- ascii_file: the full path of the folder, where the Ascii file must be saved, inclusive the file name
- You can see the same result as in Excel file with the same design in the Ascii file, if you don't have MS Excel.

The next sections will give you more explanation on how to interpret the results saved to the files.

Example:

```
SModelBalance_out =  
th_ExportBalance(SModelBalance, 'Balance.xls', 'Balance.txt')  
  
SModelBalance_out =  
    SBalanceSpace: [1x4 struct]  
    SGlobal: [1x1 struct]
```

The Excel file "Balance.xls" is created and has two sheets: one for mass balancing and one for energy.

In the SMass sheet a column is created for each species defined in the Model Setup block. All elements are listed as columns as well. You can see the name of balance system (1) and the names of user defined port names (2) with their direction in the left half as row description.

You can track how many from each species and elements are flow in or out or stored in every port. The total balance is marked in red at the top.

1 2 3

Model name: DemoFlowSimple
Simulation time: 29.7.09 14:10

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

Balance Space Name	Port Name	Port Direction	CH4 [mol]	H2O [mol]	CO [mol]	CO2 [mol]	H2 [mol]	N2 [mol]	O2 [mol]	isooctane [mol]	Methanol [mol]	H [mol]	C [mol]	O [mol]	N [mol]	F [mol]	Cl [mol]
TotalBalance			0	-8,5E-11	0	0	0	0	0	0	0	-1,70985E-10	0	-8,549E-11	0	0	0
Hydraulic Flow Balance Space																	
	Input water flow	in	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Output water flow	out	0	3323,945	0	0	0	0	0	0	0	6647,890368	0	3323,9452	0	0	0
	Overflow	out	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Stored water	stored_start	0	55509,3	0	0	0	0	0	0	0	111018,5956	0	55509,298	0	0	0
	Stored water	stored_end	0	52185,35	0	0	0	0	0	0	0	104370,7052	0	52185,353	0	0	0
Balance			0	-4,2E-11	0	0	0	0	0	0	0	-8,4583E-11	0	-4,229E-11	0	0	0
DemoFlowSimple/Downstream																	
	Flow in	in	0	3323,945	0	0	0	0	0	0	0	6647,890368	0	3323,9452	0	0	0
	Flow out	out	0	3323,945	0	0	0	0	0	0	0	6647,890368	0	3323,9452	0	0	0
Balance			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
DemoFlowSimple/Liquid Tank																	
	tank in	in	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	tank out	out	0	3323,945	0	0	0	0	0	0	0	6647,890368	0	3323,9452	0	0	0
	overflow	out	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	tank content	stored_start	0	55509,3	0	0	0	0	0	0	0	111018,5956	0	55509,298	0	0	0
	tank content	stored_end	0	52185,35	0	0	0	0	0	0	0	104370,7052	0	52185,353	0	0	0
Balance			0	-4,2E-11	0	0	0	0	0	0	0	-8,4583E-11	0	-4,229E-11	0	0	0
DemoFlowSimple/Pressure Valve 1																	
	Flow in	in	0	3323,945	0	0	0	0	0	0	0	6647,890368	0	3323,9452	0	0	0
	Flow out	out	0	3323,945	0	0	0	0	0	0	0	6647,890368	0	3323,9452	0	0	0
Balance			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

In the SEnergy sheet you dont see any direction, because the direction is visible on the basis of sign.

1 2 3

Model name: DemoFlowSimple
Simulation time: 29.7.09 14:10

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

Balance Space Name	Port Name	Port Direction	[J]
TotalBalance			1899872188
Hydraulic Flow Balance Space			
	Input water flow	-	0
	Output water flow	-	949936094
	Overflow	-	0
Balance			949936094
DemoFlowSimple/Downstream			
	Flow in	-	-949936094
	Flow out	-	949936094
Balance			0
DemoFlowSimple/Liquid Tank			
	tank in	-	0
	tank out	-	949958397
	overflow	-	0
Balance			949958397
DemoFlowSimple/Pressure Valve 1			
	Flow in	-	-949958397
	Flow out	-	949936094
Balance			-22303,5141

7 USING THERMOLIB MATLAB FUNCTIONS

For several purposes you may want to use thermodynamic state calculations also from MATLAB command line instead always using Simulink. Therefore Thermolib contains MATLAB functions for some of blocks in "Thermodynamic States And Properties" subgroup.

How this functionality can be used is described in the following chapter.

7.1 CREATING SPECIES DIAGRAMS

The most obvious use of MATLAB functions for thermodynamic calculations is to plot meaningful diagrams, which is difficult task in Simulink.

The following code creates a T-s diagram for water:

```
% plot T-S diagram for pure water
load ChemicalMediaData
T = 200:2000;
n = length(T);
ndot=ones(1,n);
p = 1e5 * ones(1,n);
psi=ones(1,n);
[State,err] = th_TpState('ndot',ones(1,n),'T',T,'p',p,'psi',psi,...
                        'Species',{'H2O'}, 'MediaData',SMediaData)
plot([State.Sdot],T)
title('T-s Diagram for water')
xlabel('molar entropy [J/K*mol]')
ylabel('temperature [K]')
```

The first command loads the thermophysical properties into MATLAB workspace. On your workspace you will find a variable SMediaData, which contains these data. The following lines define vectors of temperature, amount of substance (ndot), composition (psi), pressure (p).

Note that each column of these vectors represents one thermodynamic state. As we want to calculate the properties of pure water the composition psi has only one row. It contains one row per species contained.

The th_TpState state command is then used to calculate the complete thermodynamic states for the given properties. For each column of the input vectors an element of the

struct array `State` is created. The struct array `"State"` has a field `"Sdot"` which gives the total Entropy of the flow. We extract all entropies as a vector and plot this.

8 BLOCK REFERENCE

This chapter describes the blocks provided by Thermolib.

8.1 BALANCE

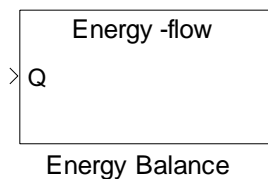
8.1.1 Energy Balance

Energy balance calculation of heat or power exchange.

Library

Balance

Description



The Energy Balance block handles energy contributions to the balance system related to an incoming or outgoing heat exchange or power signal crossing the balance system border.

It expects one power signal, which flows in or out of the balance space and saves this between the balancing start and end times as defined in Model Setup block. The gathered information will be saved in variable "SModelBalance" in MATLAB base workspace.

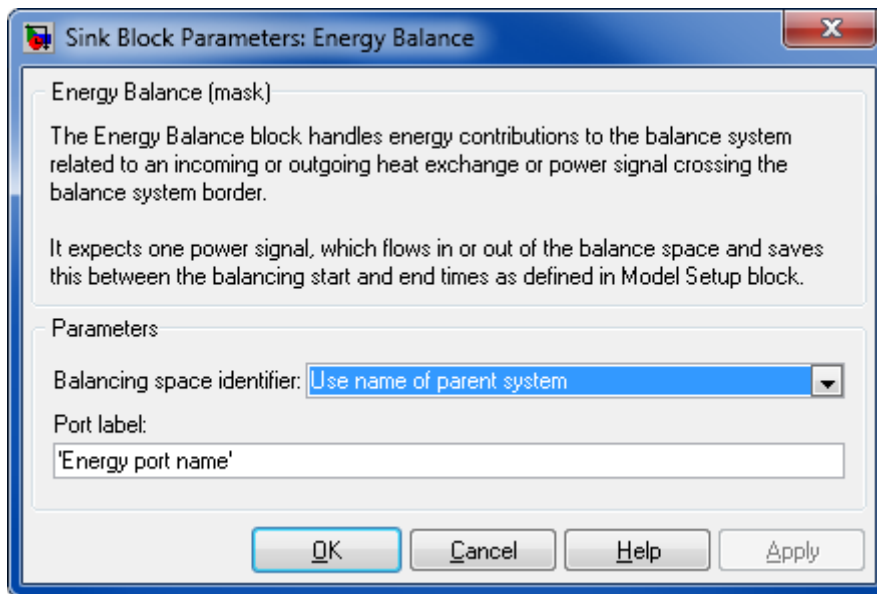
Note: A balance is generated by calling the function `th_ExportBalance` from the MATLAB command window. Refer to chapter "Thermodynamic Balancing".

Data Type Support

Inputs

\dot{Q} /P	power signal	scalar[W]
--------------	--------------	-----------

Parameters and Dialog Box



Balancing space identifier

The name by which the balance space is identified. This name must be equal (exact spelling and case) in all balance blocks part of the same balance space; you can either use the name of parent system or enter a user-defined name.

Port label

The name of the port or signal where the balance block is connected. This label identifies a flow within a balance space and is used in resulting balance tables to indicate this “flow”.

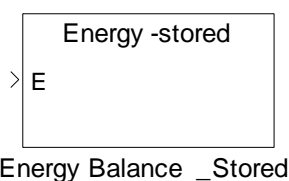
8.1.2 Energy Balance_Stored

Energy balance calculation of stored energy.

Library

Balance

Description



The Energy Balance_Stored block handles energy contributions to the balance system related to stored energy, as in a thermal mass.

It expects one energy signal, which reflects the energy stored in the system and saves this at the start and end times of the balancing interval defined in the Model Setup block. It saves gathered information in variable "SModelBalance" in MATLAB base workspace.

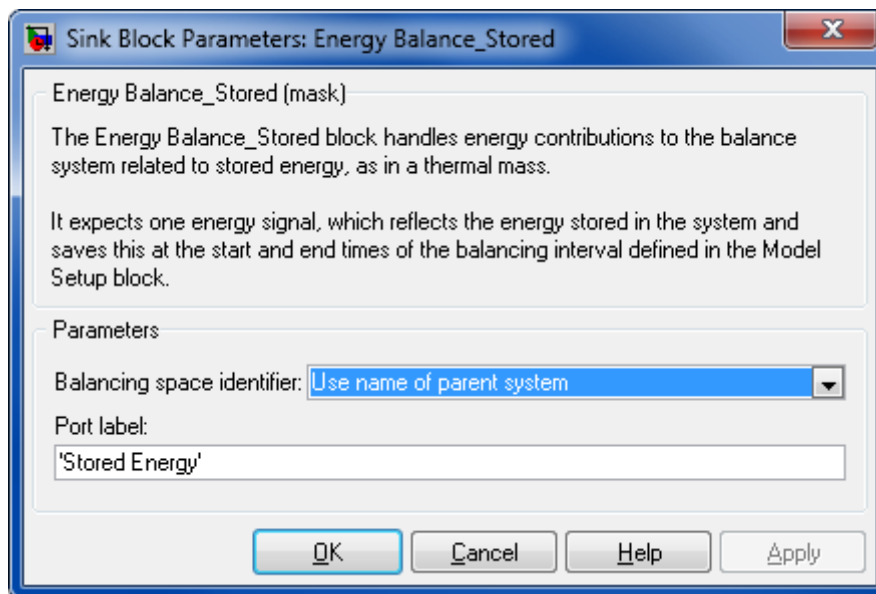
Note: A balance is generated by calling the function `th_ExportBalance` from the MATLAB command window. Refer to chapter "Thermodynamic Balancing".

Data Type Support

Inputs

Input	Description	Units
E	Energy signal	Scalar[Ws]

Parameters and Dialog Box



Balancing space identifier

The name by which the balance space is identified. This name must be equal (exact spelling and case) in all balance blocks part of the same balance space; you can either use the name of parent system or enter a user-defined name.

Port label

The name of the port or signal where the balance block is connected. This label identifies an energy storage area within a balance space and is used in the resulting balance tables to indicate this "storage".

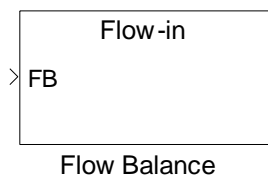
8.1.3 Flow Balance

Energy and Mass balance calculation of a flow

Library

Balance

Description



The Flow Balance block handles energy and mass flow contributions to the balance space related to incoming or outgoing flows.

It expects a flow bus as input. The block measures the mass and energy content that passed through this flow between the balancing start and end times defined in the Model Setup block. The block then saves the balance measurements into the variable SModelBalance in the MATLAB base workspace.

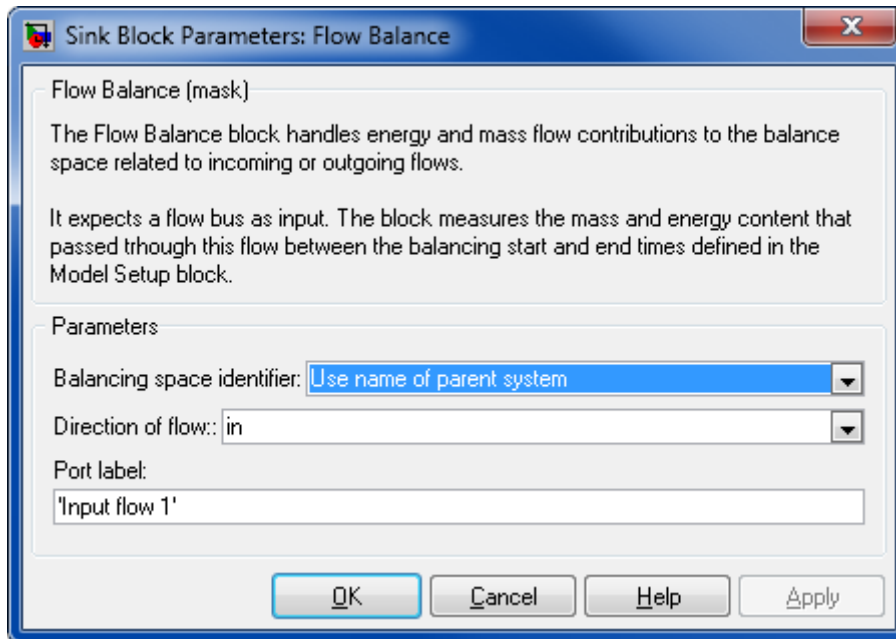
Note: A balance is generated by calling the function `th_ExportBalance` from the MATLAB command window. Refer to chapter "Thermodynamic Balancing".

Data Type Support

Inputs

FB	Flow which is entering or leaving the balance space	flow bus
----	---	----------

Parameters and Dialog Box

**Balancing space identifier**

The name by which the balance space is identified. This name must be equal (exact spelling and case) in all balance blocks part of the same balance space; you can either use the name of parent system or enter a user-defined name.

Direction of flow

Set this to "in" if the attached flow is entering your balance space. Set it to "out" if it is leaving your balance space.

Port label

The name of the port or signal where the balance block is connected. This label identifies a flow within a balance space and is used in the resulting balance tables to indicate this "flow".

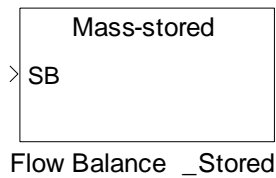
8.1.4 Flow Balance_Stored

Mass and Energy balance calculation for a stored content

Library

Balance

Description



The Flow Balance_Stored block handles energy and mass flow contributions to the balance space related to a stored content.

It expects a state bus as input. The block measures the change in mass and energy content between the balancing start and end times defined in the Model Setup block. The block then saves the balance measurements into the variable SModelBalance in the MATLAB base workspace.

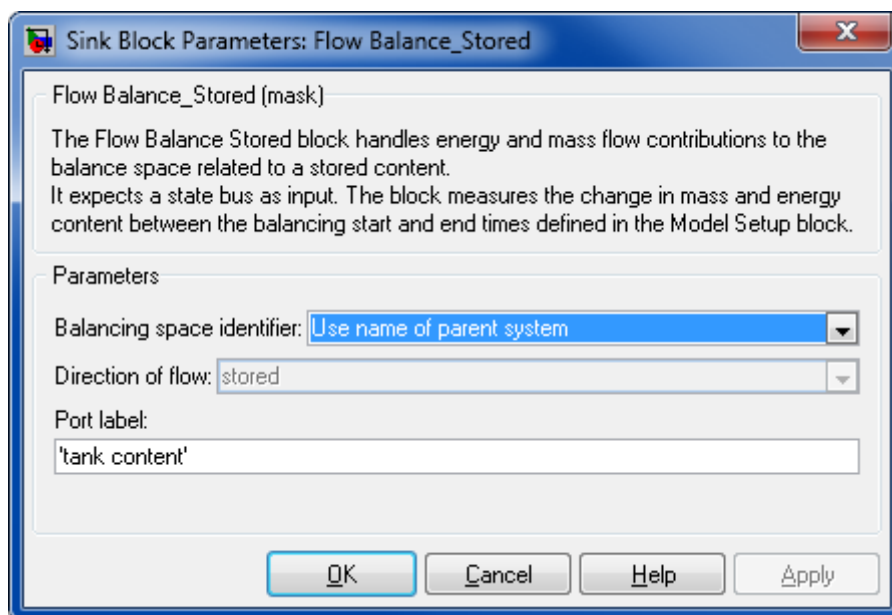
Note: A balance is generated by calling the function `th_ExportBalance` from the MATLAB command window. Refer to chapter "Thermodynamic Balancing".

Data Type Support

Inputs

SB	state	state bus
----	-------	-----------

Parameters and Dialog Box



Balancing space identifier

The name by which the balance space is identified. This name must be equal (exact spelling and case) in all balance blocks part of the same balance space; you can either use the name of parent system or enter a user-defined name.

Direction of flow

This is set automatically to "stored" and cannot be changed by the user.

Port label

The name of the port or signal where the balance block is connected. This label identifies a storage area within a balance space and is used in the resulting balance tables to indicate this "storage".

8.2 COMPONENTS

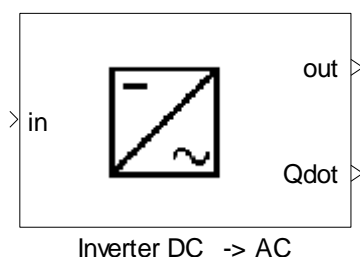
8.2.1 Electric

8.2.1.1 Inverter DC -> AC

Electric current inverter.

Library

Components/Stacks

Description

The Inverter DC-AC block inverts electric currents from DC to AC, DC to DC or AC to DC and alternates the voltage if required.

The inverter efficiency is either a constant or a function of the rate of utilization to the ratio of actual power to maximum power.

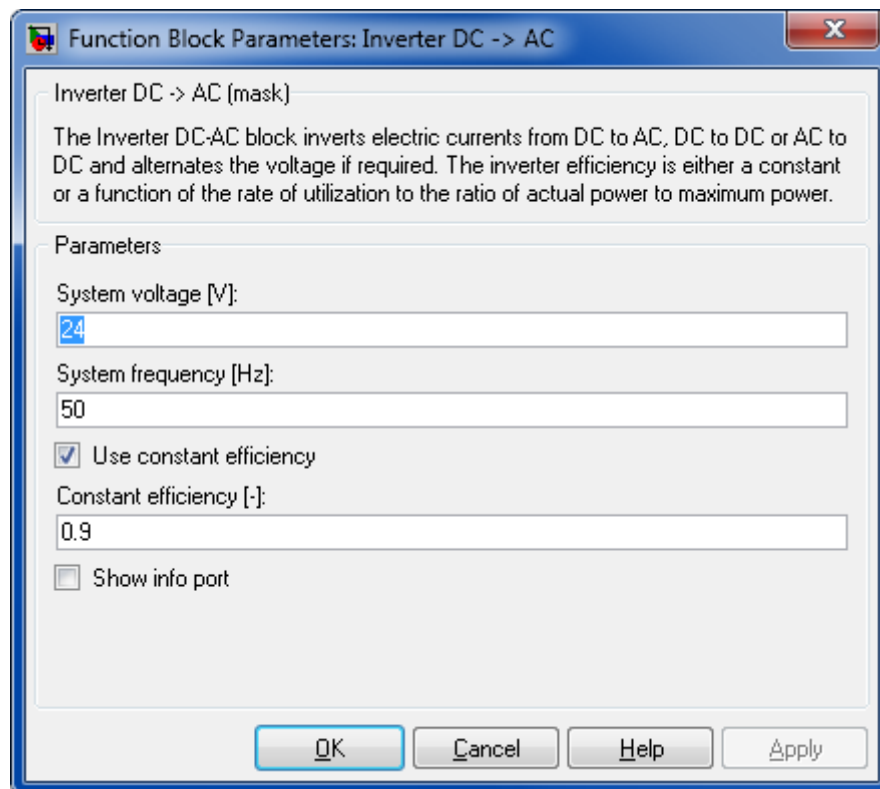
Data Type Support

Inputs

in	Direct current	Power vector
----	----------------	--------------

Outputs

out	Alternating current	Power vector
Qdot	Heat loss due to inefficiency	Scalar [W]
info	N.A. (Optional)	N.A.

Parameters and Dialog Box**System voltage [V]**

System voltage.

System Frequency [Hz]

System Frequency.

Use Constant Efficiency

Uses constant efficiency if selected.

Constant Efficiency

Constant efficiency.

Show info port

Show or hide info port.

Characteristics

Direct Feedthrough	Yes
--------------------	-----

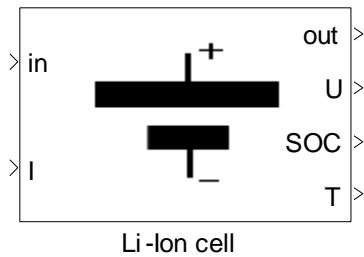
8.2.1.2 Li-Ion cell

Simulates a Lithium-Ion Battery

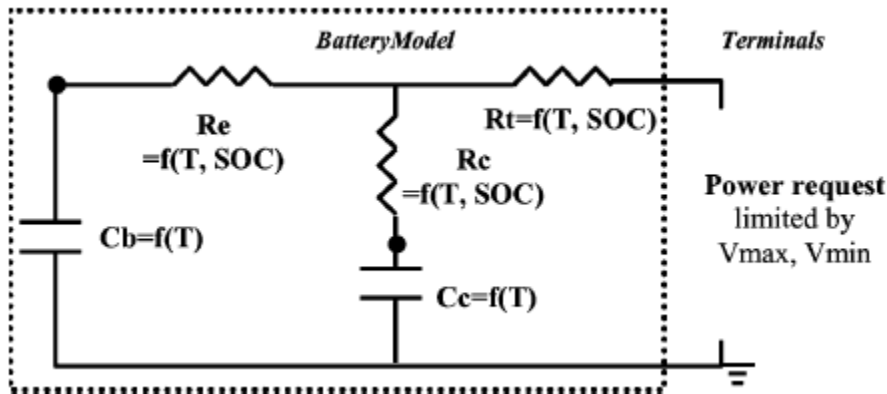
Library

Components/Electric

Description



This model simulates a high power Lithium-Ion cell and is based on the RC. The schematic of circuit that has been used in the model is:



This model simulates the battery by solving the following state space equations:

$$\begin{pmatrix} \dot{V}_{cb} \\ \dot{V}_{cc} \end{pmatrix} = \begin{pmatrix} -\frac{1}{C_b(R_e + R_c)} & \frac{1}{C_b(R_e + R_c)} \\ \frac{1}{C_c(R_e + R_c)} & -\frac{1}{C_c(R_e + R_c)} \end{pmatrix} \begin{pmatrix} V_{cb} \\ V_{cc} \end{pmatrix} + \begin{pmatrix} -\frac{R_c}{C_b(R_e + R_c)} \\ \frac{1}{C_c} + \frac{R_c}{C_b(R_e + R_c)} \end{pmatrix} (I_s)$$

$$V_0 = \frac{R_c V_{cb} + R_e V_{cc}}{R_e + R_c} + \left(R_t - \frac{R_c R_e}{R_e + R_c} \right) I_s$$

Indexes b, c and e are referred in the figure above and V_0 is referred to terminal voltage.

The resistances and capacitance are dependent to temperature and state of charge (SOC) as shown in above figure.

It is assumed, that the whole battery has a uniform temperature and is associated with a thermal mass (lumped parameter model).

The temperature of the battery can be controlled using a media flow, which passes from input port in to the output port out. The current temperature of the thermal mass can be read at the output port T. Heat exchange with the environment is neglected.

The heat exchange between flow and battery thermal mass is calculated based on a NTU method, where the thermal mass is assumed to be an infinite body, in relation to mass flow. That means a constant temperature along the heat exchange area is assumed for the thermal mass. With this assumption, the effectiveness ε (the actual heat transfer rate divided to the maximum possible heat transfer) is calculated as:

$$\varepsilon = \left(\frac{\dot{Q}}{\dot{Q}_{max}} \right) = 1 - \exp^{-N}$$

$$N = \left(\frac{KA}{\dot{m}c_p} \right)$$

KA is heat transfer rate between the flow and thermal mass [W/K]. This is the product of heat transfer coefficient and effective heat exchange area between the flows. c_p is the specific heat capacity of fluid.

Data Type Support

Inputs

in	incoming flow of cooling media	Flow bus
I	direct current drawn from battery	Scalar [A]

Outputs

out	outgoing flow of cooling media	Flow bus
U	terminal voltage	Scalar [V]
SOC	state of charge	Scalar [-]
T	temperature of battery	Scalar [K]

Parameters and Dialog Box

Function Block Parameters: Li-Ion cell

Li-Ion cell (mask)

This model simulate a high power Lithium-Ion battery. It is based on the RC model of Saft.

Parameters

Initial state of charge [%]
100

Row index input values, Temperature [K]
[0 20 40]

Column index input values, SOC []
[0 0.3 0.6 1]

Re: Lookup Table[ohm] = f (T,SOC)
[1.1 1.1 1.1 1.1;1.1 1.1 1.1 1.1;1.1 1.1 1.1 1.1]*1e-3

Rt: Lookup Table[ohm] = f (T,SOC)
[1.2 1.2 1.2 1.2;1.2 1.2 1.2 1.2;1.2 1.2 1.2 1.2]*1e-3

Rc: Lookup Table[ohm] = f (T,SOC)
[0.4 0.4 0.4 0.4;0.4 0.4 0.4 0.4;0.4 0.4 0.4 0.4]*1e-3

Cb: Lookup Table[farad] = f (T)
[82 82 82]*1e3

Cc: Lookup Table[farad] = f (T)
[4.074 4.074 4.074]*1e3

Open circuit voltage: Lookup Table[volt] = f (T,SOC)
[1. 1. 1. 1.;1. 1. 1. 1.;1. 1. 1. 1.]*50

Maximum Ah [Ampere-Hour]
12

Coulombic efficiency when charging []
1

Heat transfer coefficient [W/K]
30

Initial Temperature [K]
273.15+20

Heat Capacity of Battery[J/K]
20

OK Cancel Help Apply

Initial state of charge [%]

Initial state of charge.

Row index input values, temperature [K]

Row index input values, temperature. Give here the row vector (1 x n) of temperatures for the points given in temperature dependent functions of resistance and capacities below.

Column index input values, SOC

Column index input values, SOC. Give here the row vector (1 x m) of SOC values for the points in SOC dependent function of resistance and capacities below.

Re: Lookup table [Ohm] = f(T,SOC)

Give here the matrix (n x m) of Re values.

Rt: Lookup table [Ohm] = f(T,SOC)

Give here the matrix (n x m) of Rt values.

Rc: Lookup table [Ohm] = f(T,SOC)

Give here the matrix (n x m) of Rc values.

Cb: Lookup table [F] = f(T)

Give here the capacity as a row vector (1 x n).

Cc: Lookup table [F] = f(T)

Give here the capacity as a row vector (1 x n).

Open circuit voltage: Lookup table [V] = f(T,SOC)

Open circuit voltage as a matrix (n x m).

Maximum charge capacity [Ah]

Maximum charge capacity of the battery.

Columbic efficiency when charging [-]

Columbic efficiency when charging.

Heat transfer coefficient [W/K]

Heat transfer coefficient between flow and thermal mass of battery.

Initial temperature [K]

Initial temperature of battery.

Heat capacity of battery[J/K]

Heat capacity of battery (its thermal mass).

Characteristics

Direct Feedthrough I => U in => out	Yes
I => T I => out in => T in => U	No

References

- Johnson, V.H., Battery performance models in ADVISOR. Journal of Power Sources 110 (2002) 321–329
- J. P. Holman, Heat Transfer, 9th edition, 2002.
- V. Johnson, M. Zolot, A. Pesaran, Development and validation of a temperature-dependent resistance/capacitance battery model for ADVISOR, in: Proceedings of the 18th Electric Vehicle Symposium, Berlin, Germany, October 2001.

8.2.2 Heat and Mass Transfer

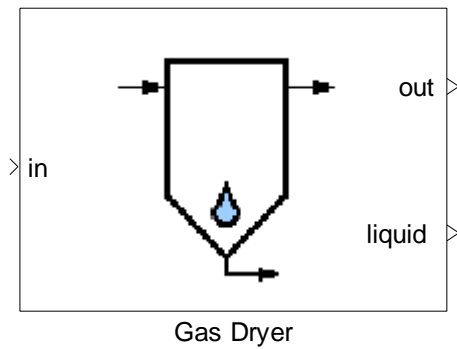
8.2.2.1 Gas Dryer

Reduction of condensed liquid in a gas flow.

Library

Components/Heat and Mass Transfer

Description



The Gas Dryer block reduces the liquid portion of a flow in an adiabatic process. The retention rate can be adjusted by the factor “retention efficiency”, which is a function of the flow rate:

$$m_{LiquidWaterSeparated} = m_{LiquidWaterEntered} \times \eta,$$

where η denotes the retention efficiency. This relation is given in terms of a lookup table. The higher the retention rate, the more the liquid is separated.

Data Type Support

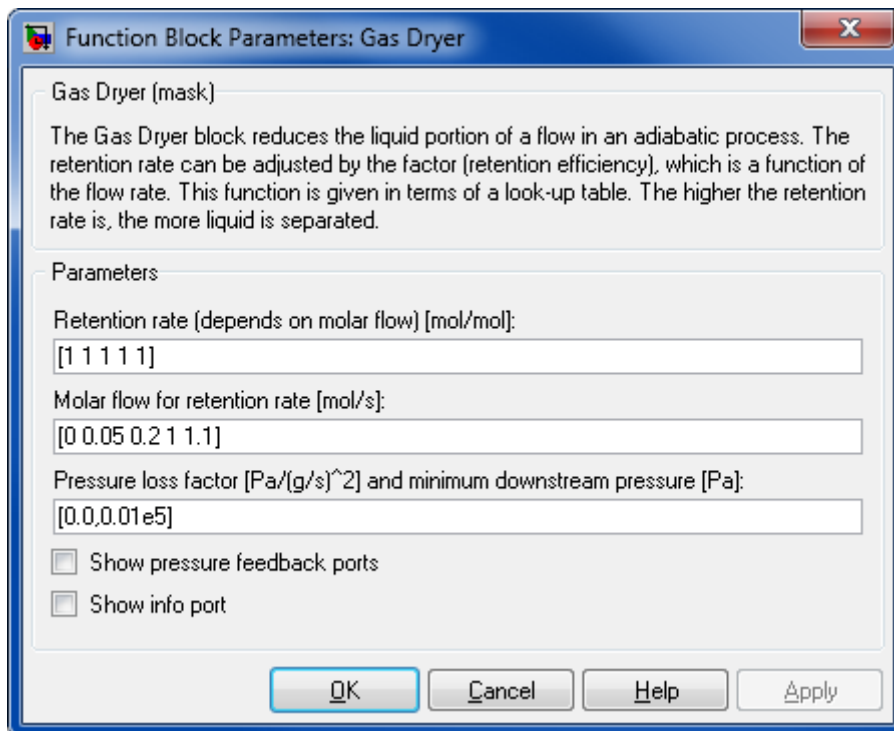
Inputs

in	Incoming flow of wet gas	Flow bus
PFin	Pressure feedback for output (Optional)	4-doubles vector

Outputs

out	Outgoing flow of dried gas	Flow bus
liquid	Outgoing flow of remaining liquid	Flow bus
PFout	Pressure feedback for input (Optional)	4-doubles vector
info	N.A. (Optional)	N.A.

Parameters and Dialog Box



Retention rate (depends on molar flow) [mol/mol]

Y-values of the lookup table describing the retention efficiency as a function of the molar flow.

Molar flow for retention rate [mol/s]

X-values of the lookup table describing the retention efficiency as a function of the molar flow.

Pressure loss factor [Pa/(g/s)²] and minimum downstream pressure [Pa]

The factor k and the pressure p are given as a vector $[k, p]$. The pressure loss is proportional to the mass flow rate squared $\Delta p = k\dot{m}^2$.

Show pressure feedback ports

Show or hide pressure feedback ports.

Show info port

Show or hide info port.

Characteristics

Direct Feedthrough	Yes
--------------------	-----

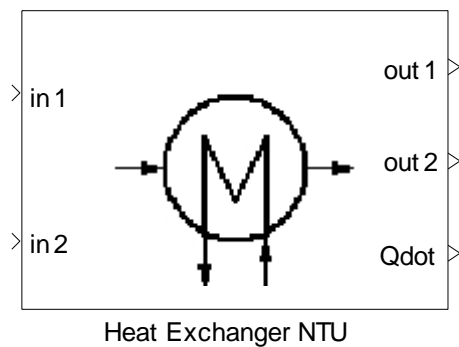
8.2.2.2 Heat Exchanger NTU

Heat exchange between two fluid flows based on NTU (number of transfer unit) method.

Library

Components/Heat and Mass Transfer

Description



The Heat Exchanger block calculates the change of state of two media caused by indirect heat exchange.

Three types of exchange are supported:

- counter flow,
- parallel flow, and
- cross flow.

The NTU-Method (number of transfer units), used to model the heat exchanger, calculates the maximum possible heat transfer between two flows. The actual heat transfer rate can be determined by calculating the effectiveness ε , which is the actual heat transfer divided by the maximum possible heat transfer.

$$\varepsilon = \left(\frac{\dot{Q}}{\dot{Q}_{\max}} \right)$$

Parallel flow:

$$\varepsilon = \left(\frac{1 - \exp(-N(1 + C))}{1 + C} \right)$$

Counter flow:

$$\varepsilon = \left(\frac{1 - \exp(-N(1 - C))}{1 - C \times \exp(-N(1 - C))} \right)$$

Cross flow:

$$\varepsilon = 1 - \exp\left(\frac{\exp(-C \times N^{0.78}) - 1}{C \times N^{-0.22}}\right)$$

Where:

$$N = \frac{UA}{\dot{C}_{min}}$$

$$C = \frac{\dot{C}_{min}}{\dot{C}_{max}}$$

$$\dot{C}_{min} = \min(\dot{m}_1 c_{p1}, \dot{m}_2 c_{p2})$$

$$\dot{C}_{max} = \max(\dot{m}_1 c_{p1}, \dot{m}_2 c_{p2})$$

According to the above equations, we have:

$$Q_{max} = \dot{C}_{min}(T_{hi} - T_{ci})$$

UA is heat transfer rate between the flows [W/K]. It represents the heat transfer between flow and wall as well as the heat conduction in the wall. This is the product of heat transfer coefficient and effective heat exchange area between the flows.

Heat transfer rate coefficient can be given in block parameter dialog or given as external input signal. To use an external input signal, use the checkbox from block parameter dialog. It is assumed, that this heat transfer rate is constant over the area of the heat exchanger or it represents a mean of the heat exchange rate.

Internal representation and thermal masses

To approximate the dynamic thermal behavior of the block, the heat exchanger is assumed to have a thermal mass (heat capacity [J/K]), which is supplied by the user in the mask. Each of the two flows entering the heat exchanger exchanges heat with its own thermal mass, which is half of the total thermal mass defined in the mask. For more information, review the thermal mass block. The two thermal masses are not interacting, but they have a term representing the heat exchange with environment.

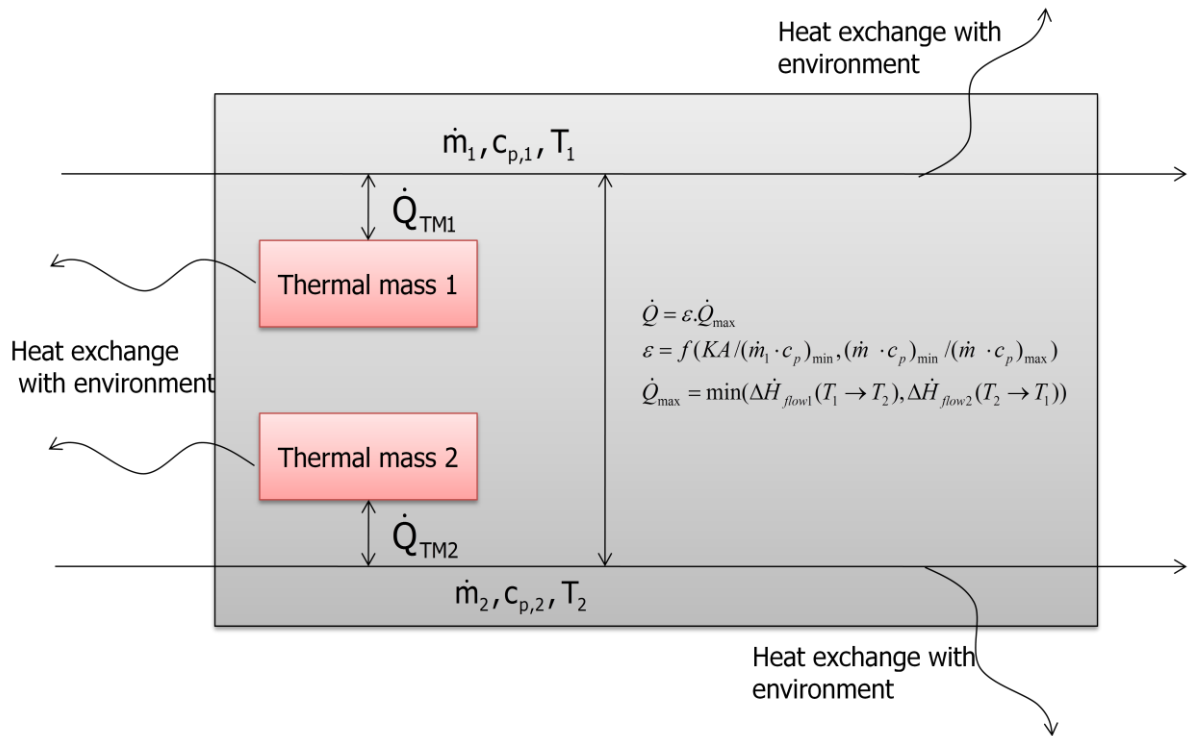


Figure 14: Principle of internal representation of heat exchanger

This internal representation involves several heat flows.

The heat exchange with environment is divided in four parts:

- both thermal masses (for flow 1 and flow 2) exchange heat with environment,
- both output flows exchange heat with environment.

The effective area for heat exchange with environment is equally divided over these four parts. The heat transfer coefficient to environment and environment temperature is assumed to be the same for all of them.

The heat exchange coefficient between the flows and its thermal masses is a mask parameter. To use the heat exchanger without thermal mass, this heat exchange coefficient can be set to zero. (Note, that defining the thermal mass itself to zero will lead to a division by zero.)

The heat exchange between a thermal mass and the environment is modeled by the simple relation:

$$\dot{Q}_{env} = K_{env} A_{env} (T_{TM} - T_{env})$$

For the heat exchange between the flows and the thermal masses a NTU approach is used (see help for Thermal Mass block). The heat exchange between the two outlet flows and the environment is modeled by the simple relation:

$$Q_{1,env}(flow1 \rightarrow env) = K_{env}A_{env}(T_1 - T_{env})$$

However to avoid temperature of the flow "crossing" the environment temperature, which is not physically, this heat exchange is limited by the enthalpy difference of the flow at T_1 and T_{env} .

$$\dot{Q}_{1,env} = \text{sign}(\dot{Q}'_{env}) \times \min(|\dot{Q}'_{1,env}|, |\dot{H}_1(T_1) - \dot{H}_1(T_{env})|)$$

How to choose the parameters for my heat exchanger

The heat exchange rate is mainly influenced by the effective heat exchange area and the flow characteristics of the flows on both sides of the wall.

E. g. for a liquid/liquid heat exchanger for water with turbulent flows on both sides a typical range for the heat exchange coefficient is 2000 to 6000 W/(m²K). If we assume a plate heat exchanger with 20 plates of 20x50 cm each and think of this as the effective heat exchange area (20*0.2*0.5 m²=2 m²). So as the heat exchange rate you can write in the parameter mask:

$$2000 \frac{\text{W}}{\text{m}^2\text{K}} \times 2\text{m}^2 = 4000 \frac{\text{W}}{\text{K}}$$

For gas/liquid heat exchangers the gas side is the limiting factor for heat exchange. Typical heat exchange coefficients are in the range of 10...100 W/(m²K). The heat exchange coefficient strongly varies with the turbulence characteristics on the gas side. So if you blow air with a fan through your heat exchanger this is much higher than if you have a passive cooling and the air flow is only induced by gravity.

For a detailed description of heat transfer see classical textbooks (see references below)

To estimate the thermal mass of your heat exchanger you can estimate the overall mass of the heat exchanger and multiply it with the specific heat capacity of the main material.

Suppose you have a heat exchanger from the example above, which has a mass of about 20 kg of steel you can estimate your thermal mass (heat capacity) to be

$$C_{TM} = 20\text{kg} \times 460 \frac{\text{J}}{\text{kgK}} = 9200 \frac{\text{J}}{\text{K}}$$

Data Type Support

Inputs

in1	input flow 1	Flow bus
in2	input flow 2	Flow bus
PFin1	Pressure feedback for output 1 (Optional)	4-doubles vector

PFin2	Pressure feedback for output 2 (Optional)	4-doubles vector
UA	Overall heat exchange coefficient (Optional)	Scalar [W/K]

Outputs

out1	output flow 1	Flow bus
out2	output flow 2	Flow bus
Qdot	Heat exchange with environment	Scalar [W]
PFout1	Pressure feedback for input 1	4-doubles vector
PFout2	Pressure feedback for input 2	4-doubles vector
	Info bus with signals (optional)	
info	Q_stored (stored energy in thermal mass)	Bus

Parameters and Dialog Box

Function Block Parameters: Heat Exchanger NTU

Heat Exchanger NTU (mask)

The Heat Exchanger block calculates the change of state of two media caused by indirect heat exchange. Three types of exchange are supported: counter flow, parallel flow and cross flow.

The NTU-Method (number of transfer unit), used to model the heat exchanger, calculates the maximum possible heat transfer between the two flows. The actual heat transfer rate can be determined by calculating the effectiveness, which is the actual heat transfer divided by the maximum possible heat transfer.

To model the thermal dynamics the input flows are passed through two thermal mass elements.

Parameters

Flow direction: Counter Flow

☐ Overall heat exchange coefficient from external

Coefficient of overall heat transfer rate [W/K]:
1e3

Initial temperature [K]:
273+25

Total surface to environment [m²):
1

Environment temperature [K]:
273+25

Coefficient for calculating heat exchange with the environment [W / m² K]:
0

Pressure loss factor [Pa/(g/s)²] and minimum downstream pressure [Pa] for first flow:
[0,0.1e5]

Pressure loss factor [Pa/(g/s)²] and minimum downstream pressure [Pa] for second flow:
[0,0.1e5]

Thermal mass of the heat exchanger [Joules/K]:
1*4.18e3

Coefficient of heat transfer rate from flow 1 to thermal mass 1 [W/K]
0

Coefficient of heat transfer rate from flow 2 to thermal mass 2 [W/K]
0

☐ Show pressure feedback ports

☐ Show info port

OK Cancel Help Apply

Flow Direction

Selects the flow direction:

- Parallel flow
- Counter flow
- Cross flow

The selection of the flow type affects the output temperature as described in the reference below.

Overall Heat Exchange Coefficient from external

Use this checkbox to enable an additional input port for the overall heat exchange coefficient. If this is checked, then the parameter for heat exchange coefficient will be ignored and the input is used. If this is not checked, the parameter for heat exchange coefficient from the block parameter dialog will be used.

Coefficient of Overall Heat Transfer Rate [W/K]

The coefficient of overall heat transfer rate that characterizes the local heat per temperature difference between the flows.

Initial temperature

The initial temperature is used for initialization of the two thermal masses.

Total Surface to environment [m²]

The external surface of the component for calculation of the heat exchange with environment. This surface is equally distributed over the four heat exchange parts for environment (two input thermal masses and the output temperatures).

Environment Temperature [K]

Environment temperature.

Coefficient of Environmental Heat Transfer [W / (m² K)]

The heat rate coefficient for calculating the heat exchange with environment.

Pressure loss factor [Pa/(g/s)²] and minimum downstream pressure [Pa] for first flow

The factor k and the pressure p are given as a vector $[k, p]$. The pressure loss is proportional to the mass flow rate squared $\Delta p = k\dot{m}^2$.

Pressure loss factor [Pa/(g/s)²] and minimum downstream pressure [Pa] for second flow

The factor k and the pressure p are given as a vector $[k, p]$. The pressure loss is proportional to the mass flow rate squared $\Delta p = k\dot{m}^2$.

Thermal mass of the heat exchanger [Joules/K]

Heat capacity of the heat exchanger. Thermal mass is assumed to be divided into two thermal masses of same size (see above).

Coefficient of heat transfer rate from flow 1 to thermal mass [W/K]

Heat Exchange rate between first flow and first thermal mass.

Coefficient of heat transfer rate from flow 2 to thermal mass [W/K]

Heat Exchange rate between second flow and second thermal mass.

Show pressure feedback ports

Show or hide pressure feedback ports.

Show info port

Show or hide info port.

Characteristics

Direct Feedthrough	Yes
--------------------	-----

References

- Incropera, et al. Fundamentals of Heat Transfer. 6th ed. John Wiley & Sons, 2007
- Nellis, Klein. Heat Transfer. Cambridge University Press, 2009.
- J. P. Holman, Heat Transfer, 9th edition, 2002.

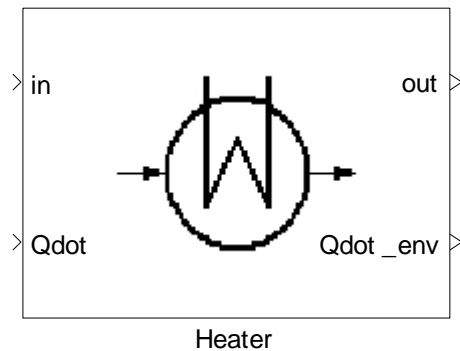
8.2.2.3 Heater

Heats up a fluid.

Library

Components/Heat and Mass Transfer

Description



The Heater block heats up or cool down the incoming fluid by adding or subtracting a given heat flux to the medium's enthalpy. Additionally, a heat exchange with the surrounding environment is taken into account.

In addition to the heated outgoing flow, this block outputs the heat exchange with the environment.

Data Type Support

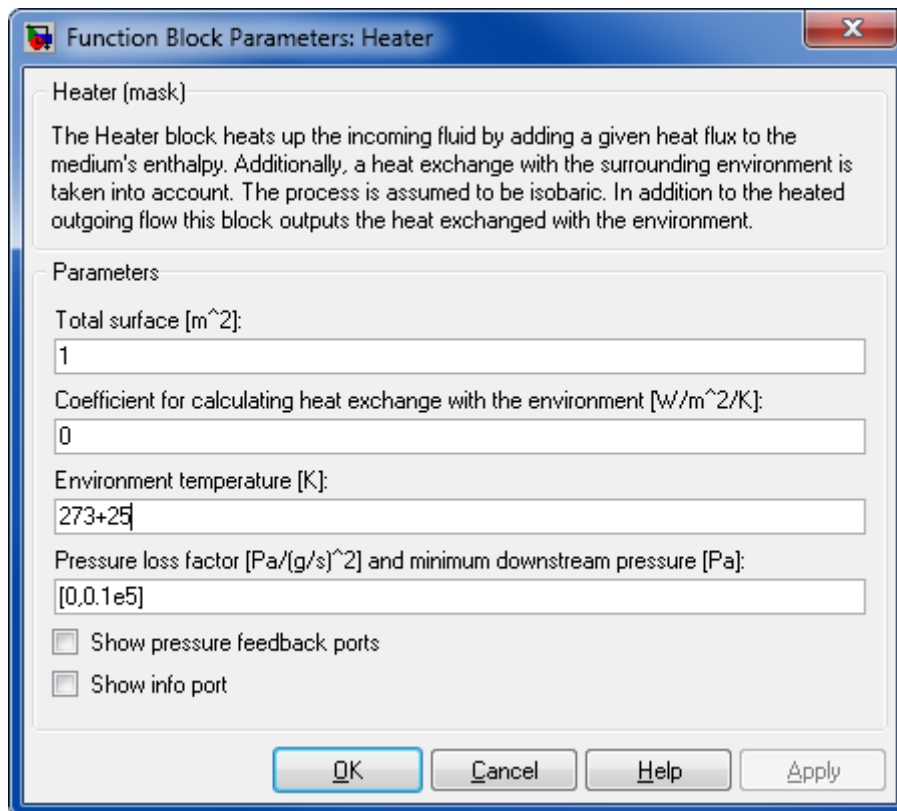
Inputs

in	Incoming flow	Flow bus
Qdot	Input heat flux	Scalar [W]
PFin	Pressure feedback for output (Optional)	4-doubles vector

Outputs

out	Outgoing flow	Flow bus
Qdot_env	Heat loss	Scalar [W]
PFout	Pressure feedback for input (Optional)	4-doubles vector
info	N.A. (Optional)	N.A.

Parameters and Dialog Box



Total surface [m²]

The external surface at which the heat exchange with the environment takes place. It is assumed that the temperature is constant at the surface.

Coefficient for calculating heat exchange with the environment [W/m²/K]

Coefficient for calculating heat exchange with the environment. This is used in combination with the environment temperature and the total surface area defined to calculate the total heat exchange with the environment.

Environment temperature [K]

Environment temperature.

Pressure loss factor [Pa/(g/s)²] and minimum downstream pressure [Pa]

The factor k and the pressure p are given as a vector $[k, p]$. The pressure loss is proportional to the mass flow rate squared $\Delta p = k \dot{m}^2$.

Show pressure feedback ports

Show or hide pressure feedback ports.

Show info port

Show or hide info port.

Characteristics

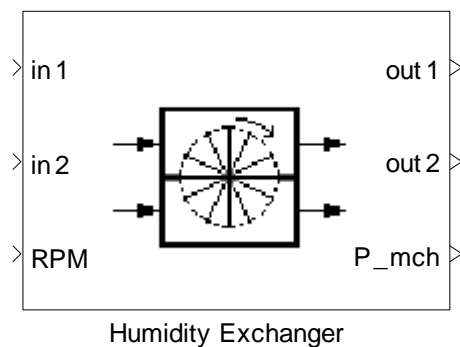
Direct Feedthrough	Yes
--------------------	-----

8.2.2.4 Humidity Exchanger

Humidity and heat exchange between two fluids.

Library

Components/Heat and Mass Transfer

Description

The Humidity Exchanger block calculates the change of state of two media caused by heat and humidity exchange.

It is a phenomenological simulation. Parts of the two incoming flows are mixed, separated, and put back into the original flows. This means that the expected behavior of a humidity exchanger is only achieved when feeding it with wet air flows.

In principle, all species might pass over to the other flow. The extent to which the flows are mixed is tuned by the parameter "Humidity efficiency as a function of RPM [-]". The relationship between humidity efficiency and RPM is given by a lookup table.

The block is adiabatic, but heat exchange between the flows is modeled by an integrated heat exchanger.

The required power is calculated as:

$$P_{mch} = f_{power} \times (\dot{m}_1 + \dot{m}_2)$$

with units P_{mch} [W], f_{power} [W/(g/s)] and sum of \dot{m} [g/s] and Power Factor is given as a parameter.

Data Type Support

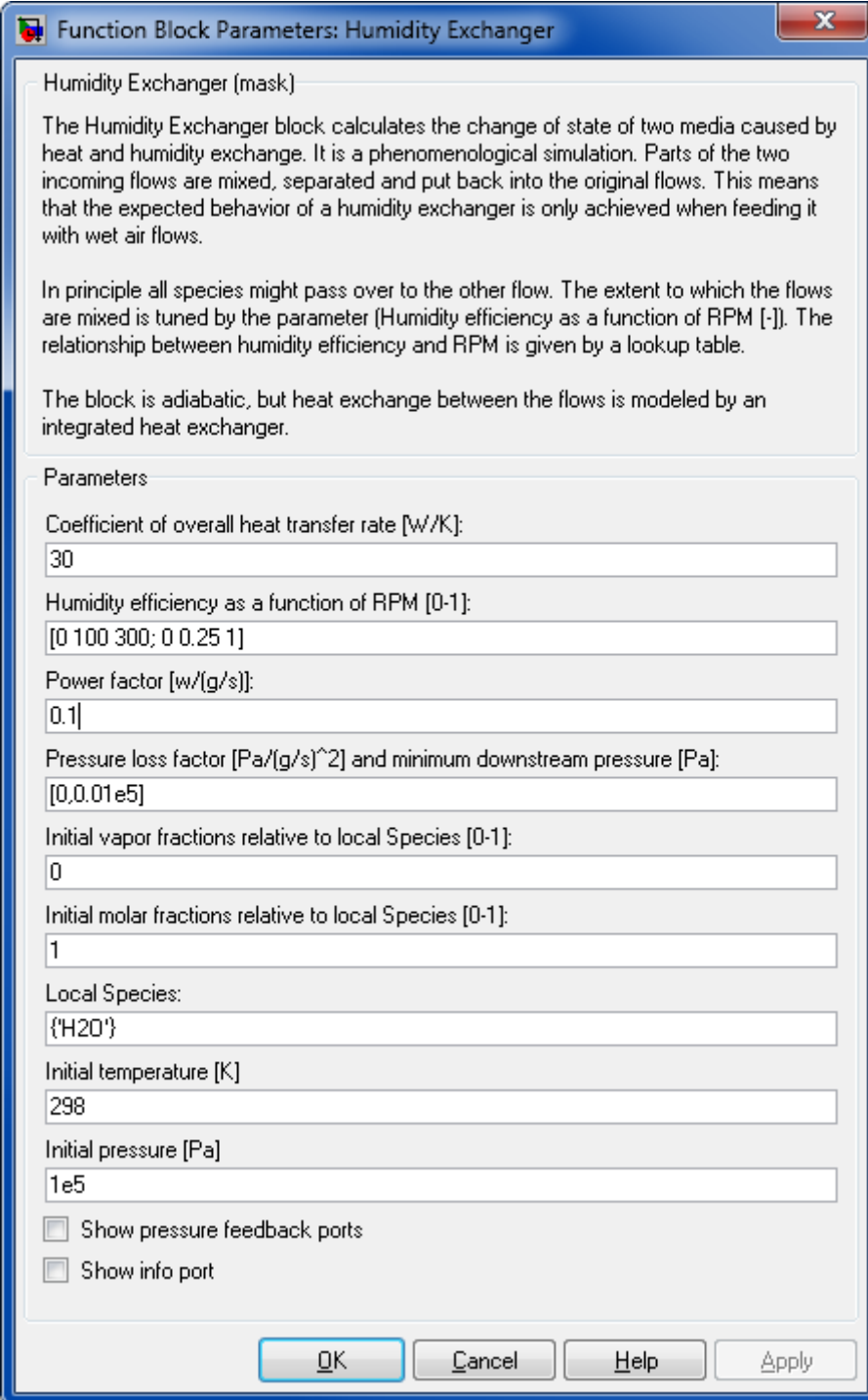
Inputs

in1	First incoming flow	Flow bus
in2	Second incoming flow	Flow bus
RPM	Rotations per minute	Scalar
PFin1	Pressure feedback for output1 (Optional)	4-doubles vector
PFin2	Pressure feedback for output2 (Optional)	4-doubles vector

Outputs

out1	Outgoing flow corresponding to in1	Flow bus
out2	Outgoing flow corresponding to in2	Flow bus
P_mch	Mechanical power consumption	Scalar [W]
PFout1	Pressure feedback for input1 (Optional)	4-doubles vector
PFout2	Pressure feedback for input2 (Optional)	4-doubles vector
info	N.A. (Optional)	N.A.

Parameters and Dialog Box



Function Block Parameters: Humidity Exchanger

Humidity Exchanger (mask)

The Humidity Exchanger block calculates the change of state of two media caused by heat and humidity exchange. It is a phenomenological simulation. Parts of the two incoming flows are mixed, separated and put back into the original flows. This means that the expected behavior of a humidity exchanger is only achieved when feeding it with wet air flows.

In principle all species might pass over to the other flow. The extent to which the flows are mixed is tuned by the parameter (Humidity efficiency as a function of RPM [-]). The relationship between humidity efficiency and RPM is given by a lookup table.

The block is adiabatic, but heat exchange between the flows is modeled by an integrated heat exchanger.

Parameters

Coefficient of overall heat transfer rate [W/K]:
30

Humidity efficiency as a function of RPM [0-1]:
[0 100 300; 0 0.25 1]

Power factor [w/(g/s)]:
0.1

Pressure loss factor [Pa/(g/s)^2] and minimum downstream pressure [Pa]:
[0,0.01e5]

Initial vapor fractions relative to local Species [0-1]:
0

Initial molar fractions relative to local Species [0-1]:
1

Local Species:
{H2O}

Initial temperature [K]:
298

Initial pressure [Pa]:
1e5

☐ Show pressure feedback ports

☐ Show info port

OK Cancel Help Apply

Coefficient of overall heat transfer rate [W/K]

The coefficient of overall heat transfer rate that characterizes the local heat per temperature difference between the flows.

Humidity efficiency as a function of RPM

A matrix that defines the efficiency of humidity exchange as a function of RPM. The first row of the matrix provides RPM; the second row contains the corresponding efficiency values belonging to the range [0,1].

Power factor [W/(g/s)]

The power factor used to calculate the power consumed (f_{power}).

Pressure loss factor [Pa/(g/s)²] and minimum downstream pressure [Pa]

The factor k and the pressure p are given as a vector $[k, p]$. The pressure loss is proportional to the mass flow rate $\Delta p = k\dot{m}^2$.

Show pressure feedback ports

Show or hide pressure feedback ports.

Show info port

Show or hide info port.

Characteristics

Direct Feedthrough	Yes
--------------------	-----

References

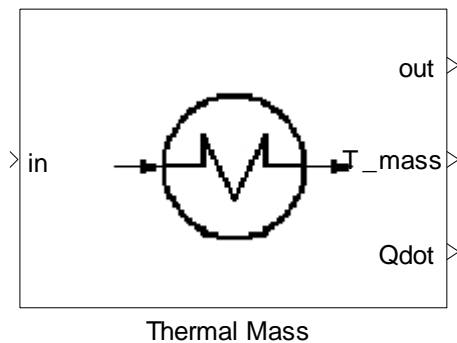
- J. P. Holman, Heat Transfer, 9th edition, 2002.

8.2.2.5 Thermal Mass

Library

Components/Heat and Mass Transfer

Description



This block represents a thermal mass body that exchanges heat with a medium flow. If the flow is of higher temperature than the thermal mass, then the mass gains heat and the flow loses. If the flow is cooler, then it gains heat from the thermal mass.

The heat exchange between flow and thermal mass is calculated based on a NTU method, where the thermal mass is assumed to be an infinite body with constant temperature along the heat exchange area.

With this assumption, the effectiveness ε (the actual heat transfer rate divided to the maximum possible heat transfer) is calculated as:

$$\varepsilon = \left(\frac{Q}{Q_{max}} \right) = 1 - \exp(-N)$$

$$N = \left(\frac{KA}{\dot{m}c_p} \right)$$

KA is heat transfer rate between the flow and thermal mass [W/K]. This is the product of heat transfer coefficient and effective heat exchange area between the flows. c_p is the heat capacity of fluid.

When flowing through the thermal mass, a flow can experience pressure loss which is defined in the block's mass.

The flow passes from input port 'in' and out the output port 'out'. The current temperature of the thermal mass can be read at the output port 'T_mass'. Output port Qdot gives the heat exchanged with the environment.

This block helps model the dynamic behavior of a system more correctly.

Data Type Support

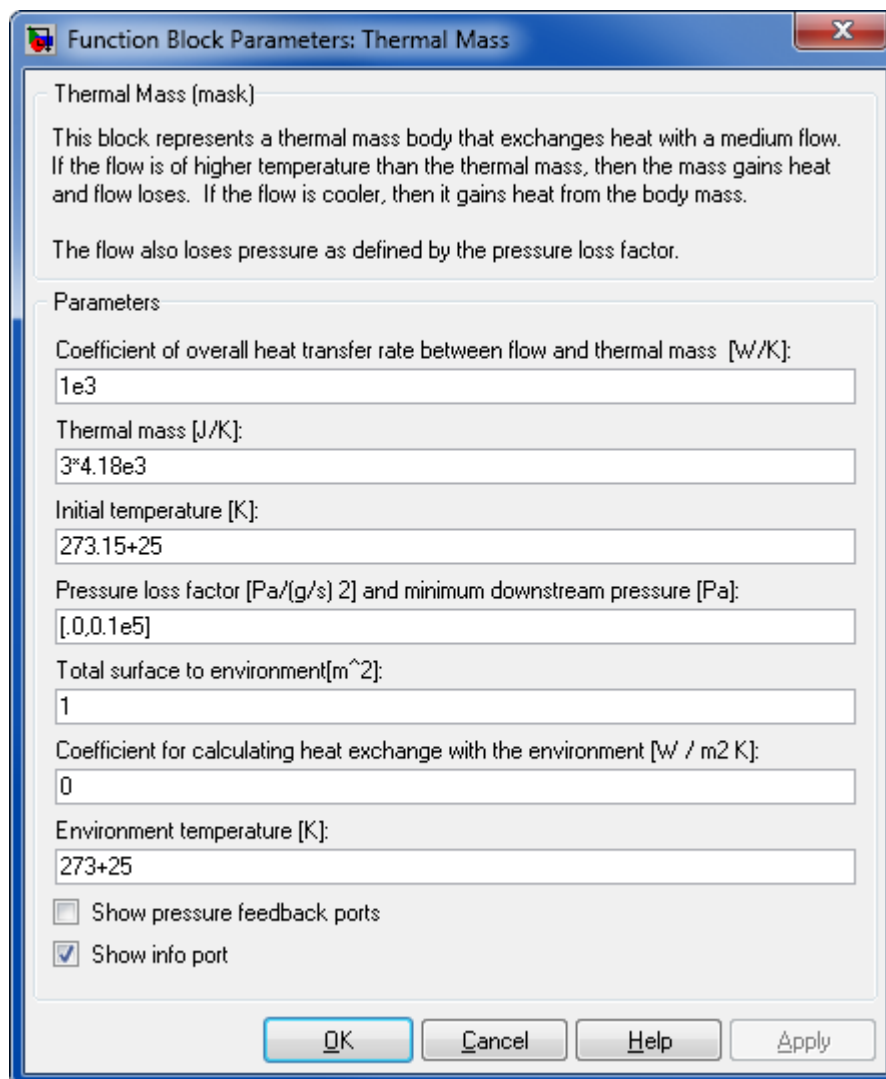
Inputs

in	incoming flow	Flow bus
PFin	Pressure feedback (Optional)	4-doubles vector

Outputs

out	Outgoing flow	Flow bus
T_mass	Temperature of the thermal mass	Scalar [K]
Qdot	Heat exchange	Scalar [W]
PFout	Pressure feedback (Optional)	4-doubles vector
info	N.A. (Optional)	N.A.

Parameters and Dialog Box



Function Block Parameters: Thermal Mass

Thermal Mass (mask)

This block represents a thermal mass body that exchanges heat with a medium flow. If the flow is of higher temperature than the thermal mass, then the mass gains heat and flow loses. If the flow is cooler, then it gains heat from the body mass.

The flow also loses pressure as defined by the pressure loss factor.

Parameters

Coefficient of overall heat transfer rate between flow and thermal mass [W/K]:
1e3

Thermal mass [J/K]:
3*4.18e3

Initial temperature [K]:
273.15+25

Pressure loss factor [Pa/(g/s)²] and minimum downstream pressure [Pa]:
[0,0.1e5]

Total surface to environment[m²]:
1

Coefficient for calculating heat exchange with the environment [W / m² K]:
0

Environment temperature [K]:
273+25

☐ Show pressure feedback ports

☒ Show info port

OK Cancel Help Apply

Coefficient of overall heat transfer rate between flow and thermal mass [W/K]

This tunes the heat exchange between flow and thermal mass. Note, that it represents area and heat transfer coefficient (" $U \cdot A$ ").

Heat capacity of the thermal mass [Joules/K]

The heat capacity of the thermal mass used to calculate the temperature change of thermal mass during heat exchange with the flow.

Initial temperature of the thermal mass [K]

The temperature of the thermal mass at the start of the simulation.

Pressure loss factor [$\text{Pa}/(\text{g/s})^2$] and minimum downstream pressure [Pa]

The factor k and the pressure p are given as a vector $[k, p]$. The pressure loss is proportional to the mass flow rate $\Delta p = k \dot{m}^2$.

Total surface in contact with the environment [m^2]

The total surface area through which heat exchange with the environment takes place.

Coefficient for calculating heat exchange with the environment [$\text{W}/\text{m}^2\text{K}$]

Coefficient used in combination with the total surface area and the environment temperature to calculate the heat exchanged with the environment.

Environment temperature [K]

The temperature of the environment in which the thermal mass is located.

Show pressure feedback ports

Show or hide pressure feedback ports.

Show info port

Show or hide info port.

Characteristics

Direct Feedthrough	Yes
--------------------	-----

8.2.3 Hydraulics

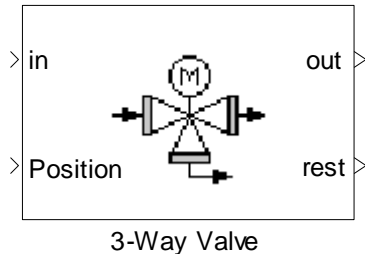
8.2.3.1 3-Way Valve

Splitting up a fluid flow into two parts.

Library

Components/Hydraulic

Description



The 3-Way Valve splits-up an incoming fluid flow into two parts.

The given position value determines the partitioning. A valve position of 1 means that the valve is wide open and all flow is directed towards the first output, whereas a value of 0 represents a closed valve, leading all flow towards the second output.

$$\begin{aligned}\dot{m}_{out1} &= pos \dot{m}_{in} \\ \dot{m}_{out2} &= (1 - pos) \dot{m}_{in}\end{aligned}$$

The process is adiabatic.

From first law of thermodynamics:

$$\dot{m}_{in} h_{in} = \dot{m}_{out1} h_{out1} + \dot{m}_{out2} h_{out2}$$

The pressure loss is independent of the valve position.

If at the start of the simulation the flow goes entirely through one output and the other output is empty, then there is a problem of defining the content of the empty output. The molar flow is set to zero, but the molar and vapor fractions have to be defined. The user gives a cell array of which species are used and the initial molar and vapor fractions. These are used by the block to define the initial content of an output port in case it is empty.

Data Type Support

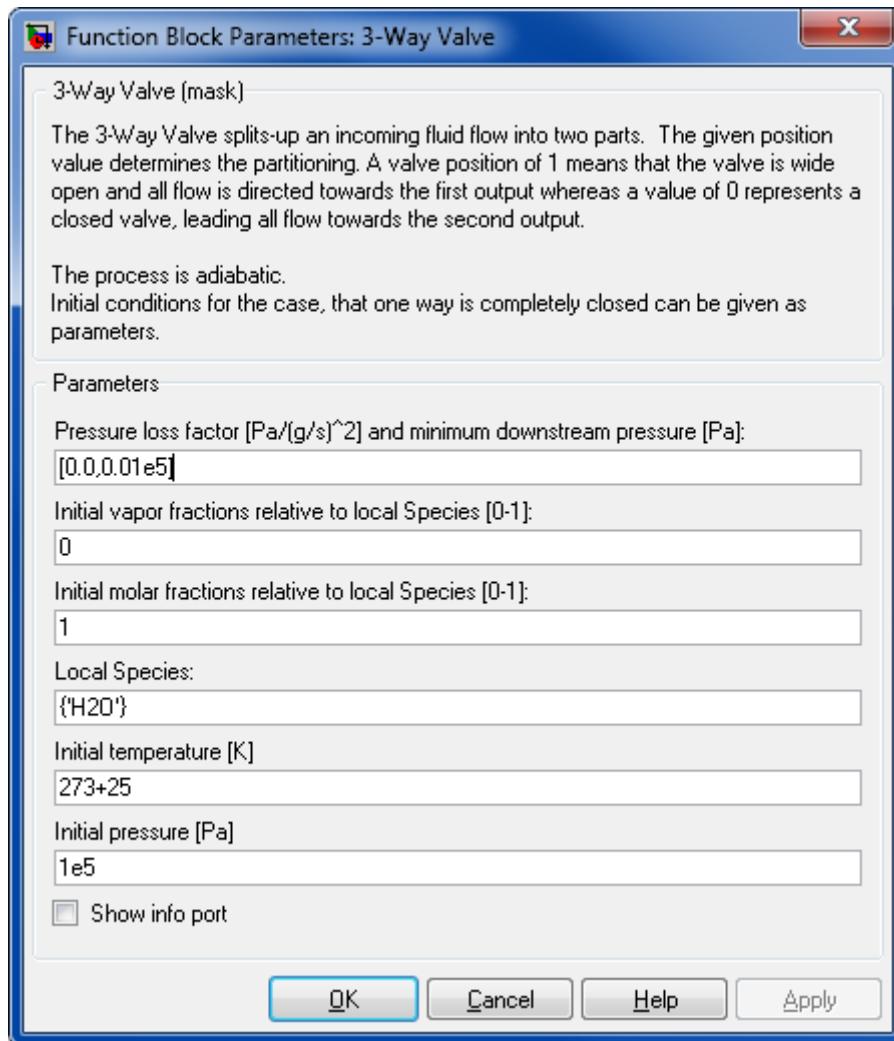
Inputs

in	Incoming fluid	Flow bus
Position	Valve position	Scalar [0-1]

Outputs

out	First portion of outgoing fluid	Flow bus
rest	Rest of outgoing fluid	Flow bus
info	N.A. (Optional)	N.A.

Parameters and Dialog Box



Pressure loss factor [Pa/(g/s)²] and minimum downstream pressure [Pa]

The factor k and the pressure p are given as a vector $[k, p]$. The pressure loss is proportional to the mass flow rate squared $\Delta p = k\dot{m}^2$.

Initial vapor fractions relative to local species [0-1]

The vapor fractions of the flow going through the valve at the start of the simulation

Initial molar fractions relative to local species [0-1]

The molar fractions of the flow going through the valve at the start of the simulation

Local species

A cell array of the media that flows through the valve. This is used in combination with the initial molar and vapor fractions to define the content of the valve at the beginning of the simulation.

Initial temperature [K]

Initial temperature of the outlet flows. This is hold until the first flow is going to the outlet.

Initial pressure [Pa]

Initial pressure at the outlet flows. This is hold until the first flow is going to the outlet.

Show info port

Show or hide info port.

Characteristics

Direct Feed through	Yes
---------------------	-----

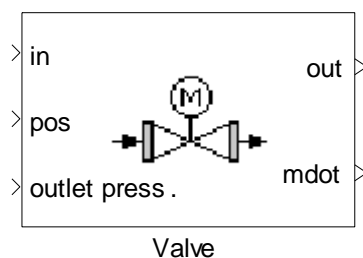
8.2.3.2 Valve

Valve with four operation modes

Library

Components/Hydraulic

Description



The valve block can model four types of valves, three passive valves and one active valve.

The passive valve simply applies a pressure drop to the flow stream which is passing through it; the mass flow rate remains unchanged.

The active valve determines the mass flow rate so that the flow matches the desired outlet pressure of the valve.

The valve characteristics are given as lookup tables with given pressure loss values for specific flow rates in the block mask.

If the position is 1, then the valve is closed, otherwise it is open.

When the valve is closed, no mass stream flows out and the temperature, pressure, vapor and molar fractions are held at the last valid values. The valve also indicates to the block receiving pressure feedback from it that it is closed by setting the third element in the vector PFout to zero.

Mode 1: passive (calculates pressure loss based on mass flow)

The Pressure Valve block reduces the pressure in the incoming flow depending on the valve position.

$$\begin{aligned}\dot{H}_{out} &= \dot{H}_{in} \\ \dot{m}_{out} &= \dot{m}_{in} \\ p_{out} &= p_{in} - k(pos) \dot{m}\end{aligned}$$

p_{out} is limited with p_{min} given in mask

k is the pressure loss coefficient as a function of valve position.

Mode 2: passive (calculates pressure loss based on volume flow)

The same relations as in mode 1 are applied except that the pressure loss is calculated as following:

$$p_{out} = p_{in} - k(pos) \left(\frac{\dot{m}}{\rho} \right)^2$$

Mode 3: passive (calculates pressure loss based on Kv-method)

The K_v -Valve block throttles the pressure of a fluid. K_v is the volume flow [m³/h] with 1 bar pressure loss for water at normal state. The value of K_v dependent to valve position is given in block mask. The pressure drop in a valve according to the current running flow is given by

$$\Delta p[bar] = \rho \left[\frac{kg}{m^3} \right] / 1000 \left[\frac{kg}{m^3} \right] \left(\frac{Q \left[\frac{m^3}{h} \right]}{K_v \left[\frac{m^3}{h} \right]} \right)^2$$

Where, Q denotes the current flow [m³/h] and ρ the density of the flow [kg/m³].

Mode 4: active (block determines mass flow)

In this model the valve flow rate is calculated based on the desired output pressure. The same relations and lookup table as in mode 3 are used. The only difference is that the mass flow rate (mdot) is calculated and input value of mass flow rate is ignored.

Data Type Support

Inputs (Modes 1, 2, 3)

in	Incoming flow	Flow bus
position	Valve position	Scalar [0-1]
PFin	Pressure feedback for output (Optional)	4-doubles vector

Outputs (Modes 1, 2, 3)

out	Outgoing flow	Flow bus
dp	Pressure loss (Optional)	Scalar [Pa]
PFout	Pressure feedback for input (Optional)	4-doubles vector
info	N.A. (Optional)	N.A.

Inputs (Mode 4)

in	Incoming flow	Flow bus
position	Valve position	Scalar [0-1]
outlet press.	Outlet pressure value	Scalar [Pa]

Outputs (Mode4)

out	Outgoing flow	Flow bus
mdot	Mass flow	Scalar [g/s]
dp	Pressure loss (Optional)	Scalar [Pa]
info	N.A. (Optional)	N.A.

Parameters and Dialog Box

Valve type:

Four valve operating modes;

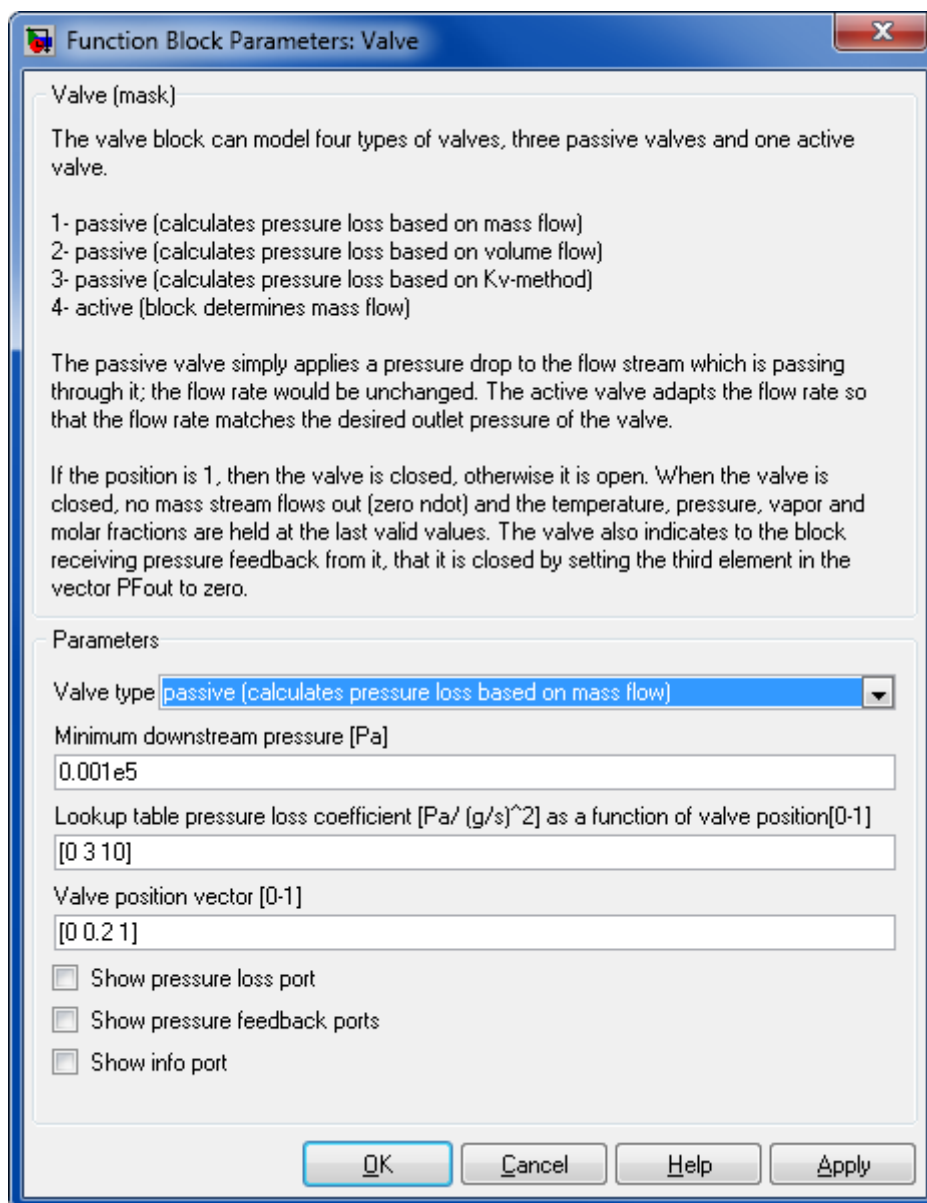
Mode 1: passive (calculates pressure loss based on mass flow)

Mode 2: passive (calculates pressure loss based on volume flow)

Mode 3: passive (calculates pressure loss based on Kv-method)

Mode 4: active (block determines mass flow)

Mode 1: passive (calculates pressure loss based on mass flow)



Minimum downstream pressure [Pa]

Minimum downstream pressure.

Lookup table pressure loss coefficient $[\text{Pa}/(\text{g/s})^2]$ as a function of valve position [0-1]

A vector of pressure loss values for unit mass flow for different valve positions.

Valve position vector [0-1]

Valve position 0 (open) to 1 (closed).

Show pressure loss ports

Show or hide pressure loss port.

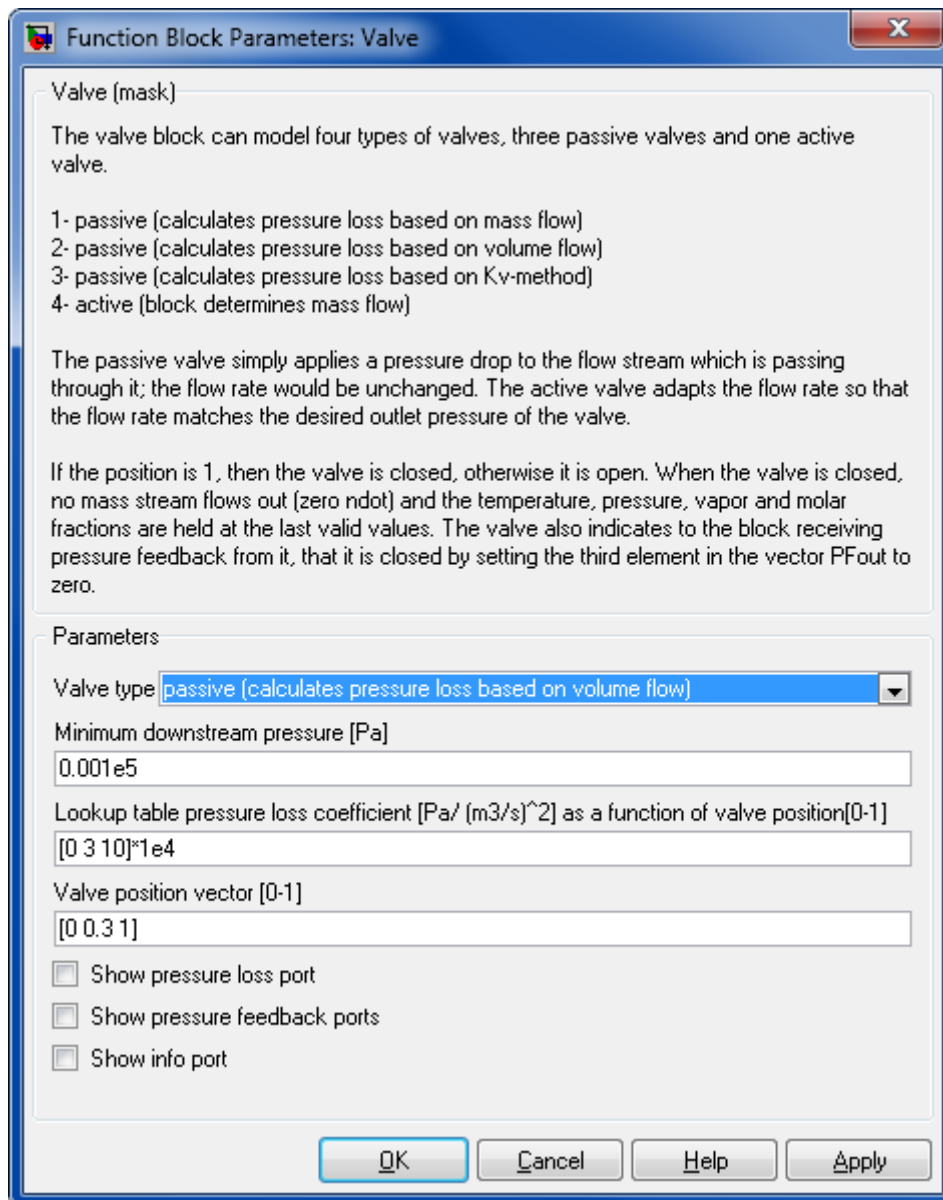
Show pressure feedback ports

Show or hide pressure feedback ports.

Show info port

Show or hide info port.

Mode 2: passive (calculates pressure loss based on volume flow)



Minimum downstream pressure [Pa]

Minimum downstream pressure

Lookup table pressure loss coefficient [Pa/(m³/s)²] as a function of valve position [0-1]

A vector of pressure loss values for unit volume flow for different valve positions.

Valve position vector [0-1]

Valve position 0 (open) to 1 (closed).

Show pressure loss ports

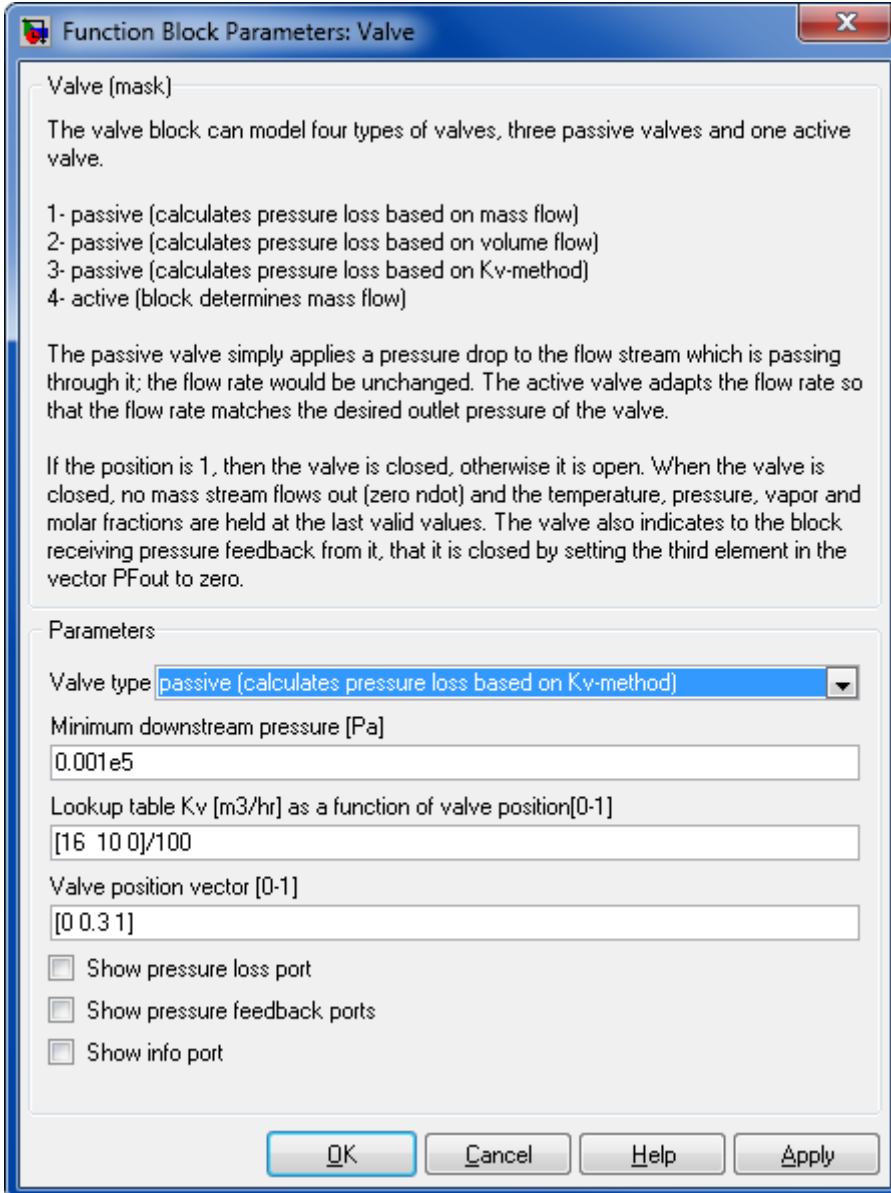
Show or hide pressure loss port.

Show pressure feedback ports

Show or hide pressure feedback ports.

Show info port

Show or hide info port.

Mode 3: passive (calculates pressure loss based on Kv-method)

The dialog box titled "Function Block Parameters: Valve" contains the following sections:

- Valve (mask)**
 - The valve block can model four types of valves, three passive valves and one active valve.
 - 1- passive (calculates pressure loss based on mass flow)
 - 2- passive (calculates pressure loss based on volume flow)
 - 3- passive (calculates pressure loss based on Kv-method)
 - 4- active (block determines mass flow)
- The passive valve simply applies a pressure drop to the flow stream which is passing through it; the flow rate would be unchanged. The active valve adapts the flow rate so that the flow rate matches the desired outlet pressure of the valve.

If the position is 1, then the valve is closed, otherwise it is open. When the valve is closed, no mass stream flows out (zero \dot{m}) and the temperature, pressure, vapor and molar fractions are held at the last valid values. The valve also indicates to the block receiving pressure feedback from it, that it is closed by setting the third element in the vector PF_{out} to zero.
- Parameters**
 - Valve type: **passive (calculates pressure loss based on Kv-method)** (selected in a dropdown menu)
 - Minimum downstream pressure [Pa]: **0.001e5**
 - Lookup table Kv [m³/hr] as a function of valve position[0-1]: **[16 10 0]/100**
 - Valve position vector [0-1]: **[0 0.3 1]**
 - ☐ Show pressure loss port
 - ☐ Show pressure feedback ports
 - ☐ Show info port

Buttons at the bottom: **OK**, **Cancel**, **Help**, **Apply**.

Minimum downstream pressure [Pa]

Minimum downstream pressure.

Lookup table Kv [m³/hr]

The volume flow [m³/hr] lookup table values with a pressure loss of 1 bar for different valve positions.

Valve position vector [0-1]

Valve position 0 (open) to 1 (closed).

Show pressure loss ports

Show or hide pressure loss port.

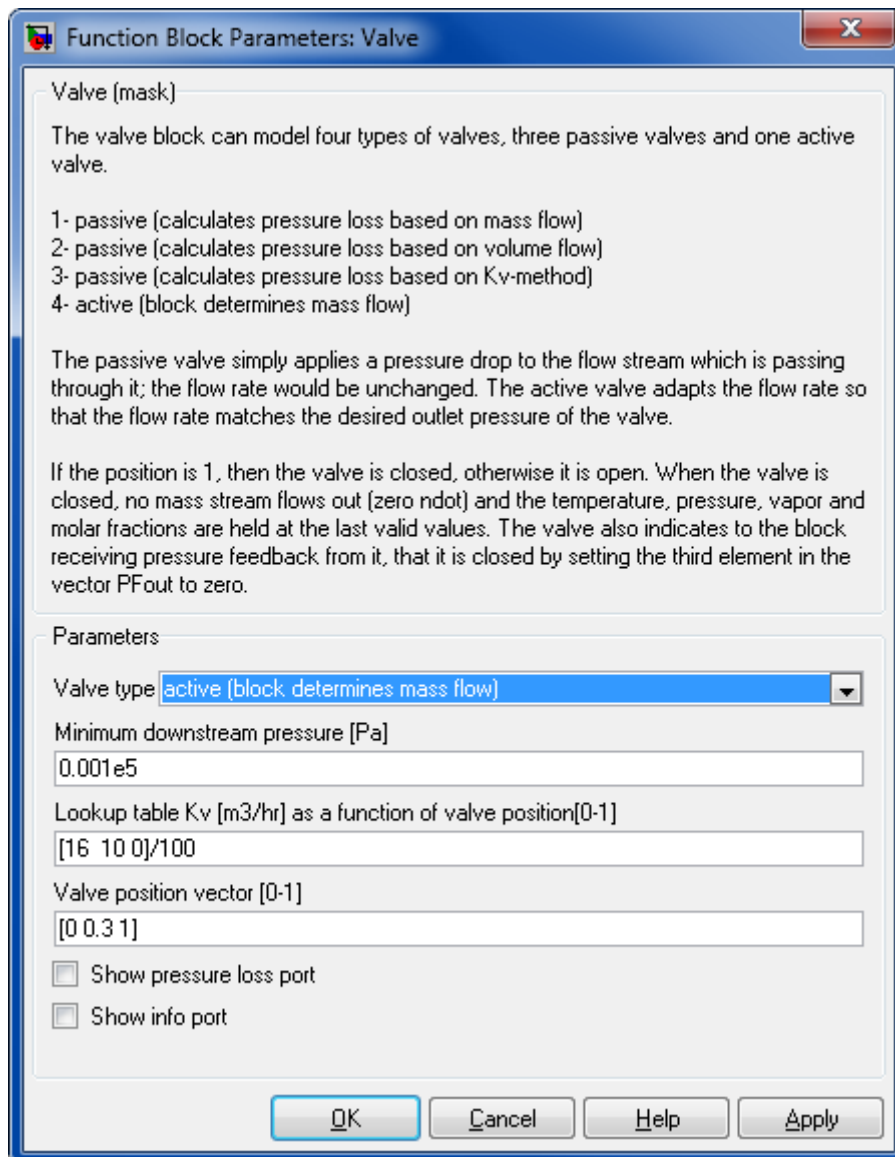
Show pressure feedback ports

Show or hide pressure feedback ports.

Show info port

Show or hide info port.

Mode 4: active valve (calculates mass flow)



Minimum downstream pressure [Pa]

Minimum downstream pressure.

Lookup table Kv [m³/hr]

The volume flow lookup table values with a pressure loss of 1 bar for different valve positions.

Valve position vector [0-1]

Valve position 0 (open) to 1 (closed).

Show pressure loss ports

Show or hide pressure loss port.

Show info port

Show or hide info port.

Characteristics

Direct Feedthrough	Yes
--------------------	-----

References

- VDI-Wärmeatlas, Berechnungsblätter für den Wärmeübergang, Abschnitt Lc, sechste, erweiterte Auflage, 1991

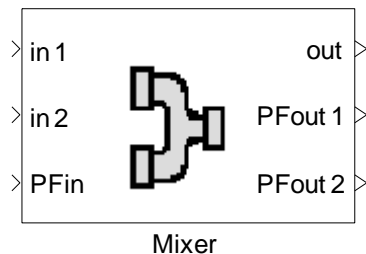
8.2.3.3 Mixer

Mixer with two operation modes.

Library

Components/Hydraulic

Description



The Mixer block calculates temperature, composition and pressure after an adiabatic mixing of two fluids. It takes phase changes into account, but no chemical reactions are considered. The output enthalpy is the sum of the input enthalpies.

Mixer block can model two modes of mixers:

Mode 1: Mixer

Simply mixes two fluids with or without phase change. Output mass flow is the sum of input mass flows.

First law of thermodynamics for Mixer:

$$\dot{m}_1 h_1 + \dot{m}_2 h_2 = \dot{m}_{out} h_{out}$$

The pressure of the resulting flow is calculated in two ways, which can be chosen in mask. The first one is the minimum of both pressure and second one is a molar weighted average of incoming pressure.

$$p_{out} = \frac{\dot{n}_1 p_{in1} + \dot{n}_2 p_{in2}}{\dot{n}_1 + \dot{n}_2} - \Delta p$$

After that, the pressure loss, with the pressure loss factor k, is considered as follows:

$$\Delta p = k \dot{m}^2$$

Mode 2: Ratio Mixer

Ratio Mixer that calculates temperature, composition and pressure after an adiabatic mixing of flow 1 (in1) with a second flow given at (in2) in a defined molar ratio, which is given at input port "ratio".

The required flow of media 2 is calculated and given at output "in2". The calculation of the outgoing flow state is the same as the simple mixer.

$$ratio = \frac{\dot{n}_2}{\dot{n}_1}$$

$$\dot{m}_{out} = \dot{m}_1 + \dot{m}_2 \times ratio \times \frac{M_2}{M_1}$$

The pressure of the resulting flow is calculated as molar weighted pressure of incoming pressure:

$$p_{out} = \frac{\dot{n}_1 p_{in1} + \dot{n}_2 p_{in2}}{\dot{n}_1 + \dot{n}_2} - \Delta p$$

or calculated as the minimum of pressures:

$$p_{out} = \min(p_{in1}, p_{in2}) - \Delta p$$

Where for the pressure loss Δp with the pressure loss factor k it holds:

$$\Delta p = k \dot{m}^2$$

Data Type Support

Inputs (Mode 1)

in1	First incoming flow	Flow bus
in2	Second incoming flow	Flow bus
PFin	Pressure feedback for output (Optional)	4-doubles vector

Outputs (Mode 1)

out	Outgoing flow mixed from in1 and in2	Flow bus
PFout1	Pressure feedback for input1 (Optional)	4-doubles vector
PFout2	Pressure feedback for input2 (Optional)	4-doubles vector
info	N.A. (Optional)	N.A.

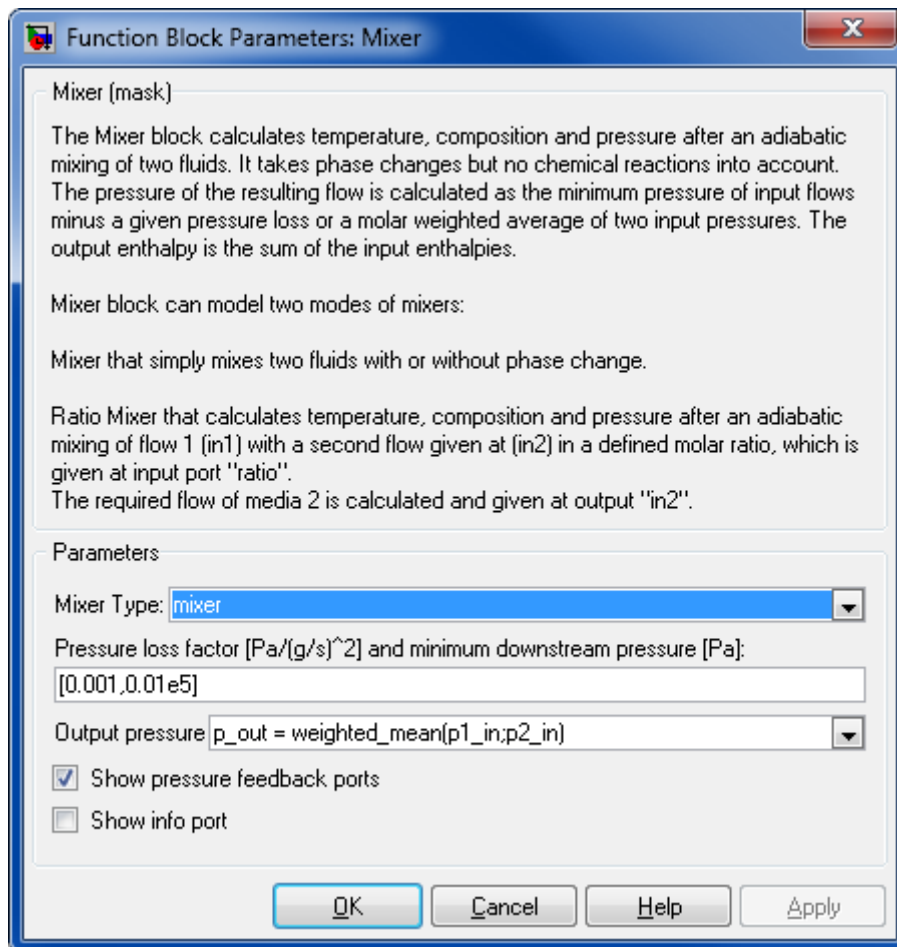
Inputs (Mode 2)

in1	First incoming flow	Flow bus
in2	Second incoming flow	Flow bus
ratio	mixing molar ratio (ndot2/ndot1)	Scalar
PFin	Pressure feedback for output (Optional)	4-doubles vector

Outputs (Mode 2)

out	Outgoing flow mixed from in1 and in2	Flow bus
in2	Second flow entered	Flow bus
PFout1	Pressure feedback for input1 (Optional)	4-doubles vector
PFout2	Pressure feedback for input2 (Optional)	4-doubles vector
info	N.A. (Optional)	N.A.

Parameters and Dialog Box***Mode 1: Mixer***



Pressure loss factor [Pa/(g/s)²] and minimum downstream pressure [Pa]

The factor k and the pressure p are given as a vector $[k, p]$. The pressure loss is proportional to the mass flow rate squared $\Delta p = k \dot{m}^2$.

Output pressure

Method of calculating of output pressure. Minimum or molar weighted average could be chosen.

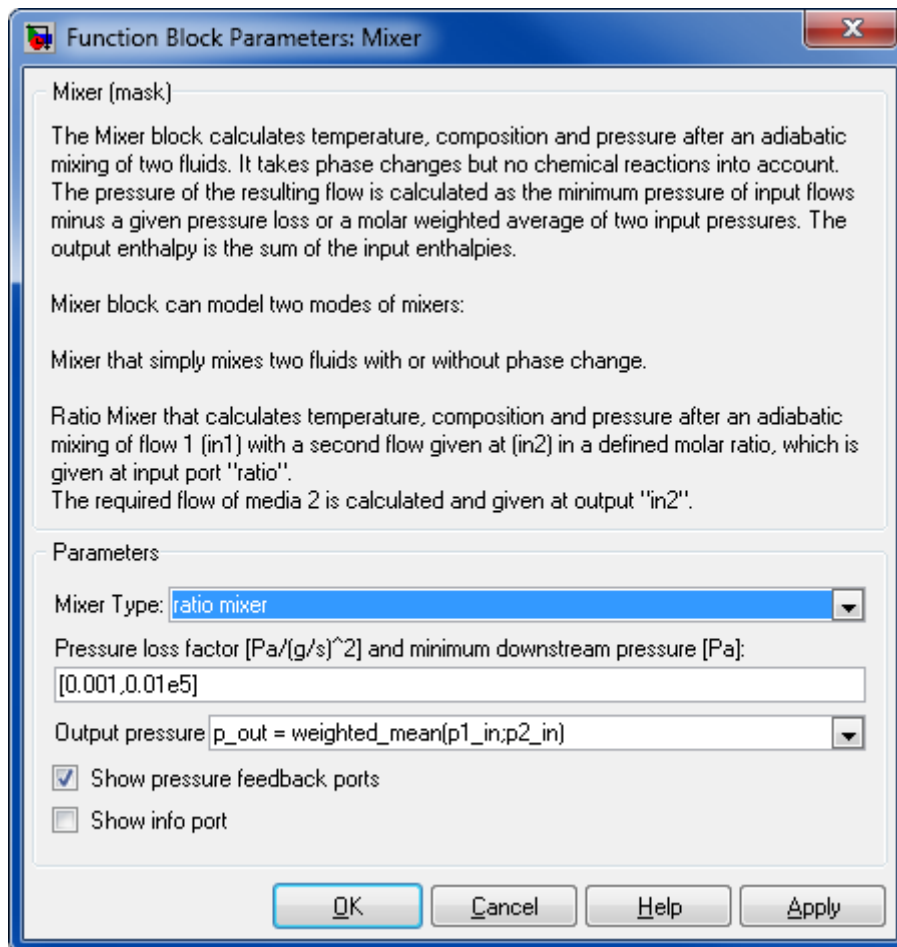
Show pressure feedback ports

Show or hide pressure feedback ports.

Show info port

Show or hide info port.

Mode 2: Ratio Mixer



Pressure loss factor [$\text{Pa}/(\text{g/s})^2$] and minimum downstream pressure [Pa]

The factor k and the pressure p are given as a vector $[k, p]$. The pressure loss is proportional to the mass flow rate squared $\Delta p = k \dot{m}^2$.

Output pressure

Method of calculating of output pressure. Minimum or molar weighted average could be chosen.

Show pressure feedback ports

Show or hide pressure feedback ports.

Show info port

Show or hide info port.

Characteristics

Direct Feedthrough	Yes
--------------------	-----

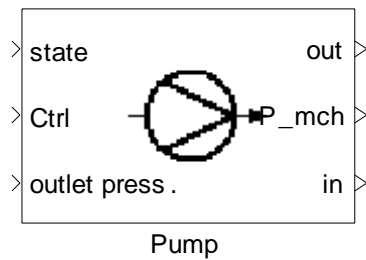
8.2.3.4 Pump

Pump model with two operating modes.

Library

Components/Hydraulic

Description



The pump block increases the pressure of an incoming liquid flow to a given target pressure. It determines the thermodynamic state of the outgoing flow along with the required mechanical power consumption of the pump. This block should be used with liquid or incompressible flows only. The pump is modeled based on the isentropic efficiency and the general relations given below:

From the first law of thermodynamics for pump:

$$\dot{m} \left(h_{in} + \frac{1}{2} v_{in}^2 \right) = \dot{m} \left(h_{out} + \frac{1}{2} v_{out}^2 \right) + \dot{W}$$

Simplifying

$$\dot{m}(h_{out} - h_{in}) = (p_{out} - p_{in})/\rho$$

Where p_{out} and p_{in} are pressure of outlet and inlet of pump. The work is calculated by:

$$\frac{\dot{W}}{\dot{m}} = \frac{h_{out} - h_{in}}{\eta} = \frac{(p_{out} - p_{in})}{\eta \rho}$$

Where η is the pump efficiency. The required electric power P_{el} is calculated as:

$$P_{mch} = \frac{\dot{m} \Delta p}{\rho \eta}$$

Where ρ denotes the density. The output enthalpy is equal to the sum of input enthalpy and electric power:

$$\dot{H}_{out} = \dot{H}_{in} + P_{mch}$$

A characteristic time is used to delay the output mass flow. For this purpose, we use a first-order transfer function with a time constant. The corresponding Transfer function block implements the transfer function

$$\frac{y(s)}{u(s)} = \frac{1}{\tau s + 1}$$

Where u denotes the control signal for mass flow rate, y the output delayed signal, and τ the characteristic time.

Calculation method:

There are two options available for the user to define the pump flow rate and isentropic efficiency.

Mode 1: Calculate mass flow due to pressure difference

If this option is selected, the pump flow rate is defined as a function of its pressure difference, and should be given as lookup table vectors in block mask for the nominal pump RPM.

For convenience the mass flow and pressure rise are expressed in terms of volumetric flow rate and head.

The pump volumetric flow rate is given as a function of its head. The head is calculated from the desired pressure rise:

$$Q = f(Head)$$

$$Q = \frac{\dot{m}}{\rho}$$

$$Head = \frac{\Delta p}{\rho g}$$

The pump isentropic efficiency is determined as a function of its volumetric flow rate.

$$\eta = f(Q)$$

The values for these lookup tables are given in the pump block mask as 2-D vectors.

For other RPMs, pump similarity laws are simply used to determine the pump flow rate.

$$\frac{Q_1}{Q_2} = \frac{RPM_1}{RPM_2}$$

$$\frac{\Delta p_1}{\Delta p_2} = \left(\frac{RPM_1}{RPM_2} \right)^2$$

Then:

$$\frac{Q}{RPM} = f\left(\frac{Head}{RPM^2}\right)$$

Mode 2: mass flow from input value

If this option is selected, the realized mass flow rate is determined by multiplication of a given maximum mass flow rate and a given control value and the mass flow rate from the given media is ignored.

In this mode the isentropic efficiency would be simply a given constant value in the block mask.

Note: Usually this block is used in combination with a Reservoir source. While the Reservoir provides the thermodynamic state of the flow, the Pump block calculates the mass flow and the rise in pressure.
A warning is generated if the input flow contains gaseous species. The Compressor block should be used in this case.

Data Type Support**Inputs (Mode 1)**

state	Incoming fluid state	Flow bus
RPM	Pump rotational speed	Scalar [RPM]
outlet press.	Target pressure at the output	Scalar [Pa]

Outputs (Mode 1)

out	Outgoing flow	Flow bus
P_mch	Mechanical power consumption	Scalar [W]
in	Calculated flow input to the pump	Flow bus
mdot	Mass flow rate	Scalar [g/s]
info	N.A. (Optional)	N.A.

Inputs (Mode 2)

state	Incoming fluid state	Flow bus
ctrl	Fraction of maximum mass flow	Scalar [0-1]
outlet press.	Target pressure at the output	Scalar [Pa]

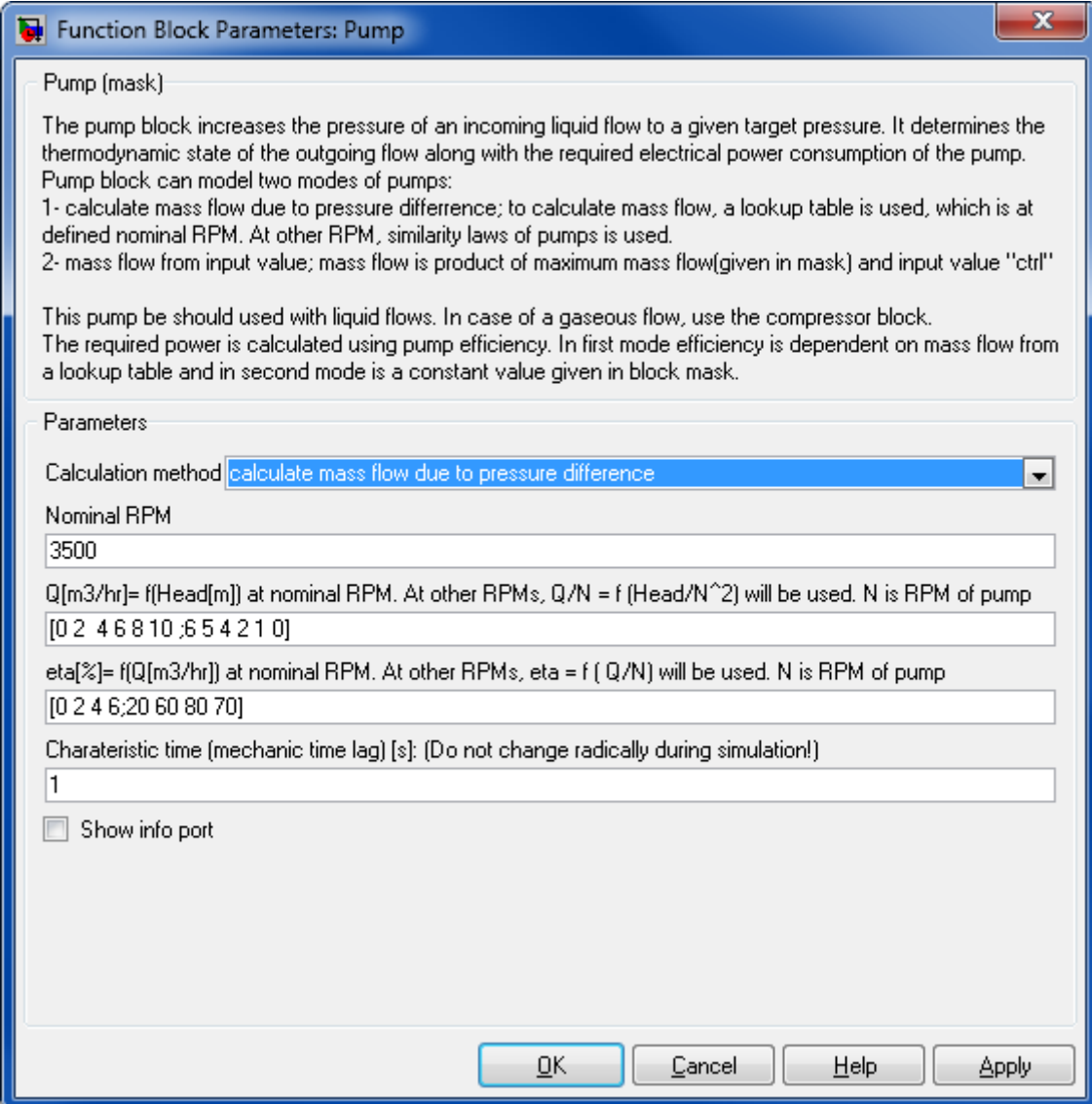
Outputs (Mode 2)

out	Outgoing flow	Flow bus
P_mch	Mechanical power consumption	Scalar [W]
in	Calculated flow input to the pump	Flow bus
info	N.A. (Optional)	N.A.

Parameters and Dialog Box**Calculation Method**

The block mask and input fields change according to the chosen calculation method.

Mode 1 (calculate mass flow due to pressure difference)



Function Block Parameters: Pump

Pump (mask)

The pump block increases the pressure of an incoming liquid flow to a given target pressure. It determines the thermodynamic state of the outgoing flow along with the required electrical power consumption of the pump. Pump block can model two modes of pumps:

- 1- calculate mass flow due to pressure difference; to calculate mass flow, a lookup table is used, which is at defined nominal RPM. At other RPM, similarity laws of pumps is used.
- 2- mass flow from input value; mass flow is product of maximum mass flow(given in mask) and input value "ctrl"

This pump be should used with liquid flows. In case of a gaseous flow, use the compressor block. The required power is calculated using pump efficiency. In first mode efficiency is dependent on mass flow from a lookup table and in second mode is a constant value given in block mask.

Parameters

Calculation method **calculate mass flow due to pressure difference**

Nominal RPM
3500

$Q[m^3/hr] = f(Head[m])$ at nominal RPM. At other RPMs, $Q/N = f(Head/N^2)$ will be used. N is RPM of pump
[0 2 4 6 8 10 ;6 5 4 2 1 0]

$\eta[\%] = f(Q[m^3/hr])$ at nominal RPM. At other RPMs, $\eta = f(Q/N)$ will be used. N is RPM of pump
[0 2 4 6;20 60 80 70]

Charateristic time (mechanic time lag) [s]: (Do not change radically during simulation!)

1

☐ Show info port

OK Cancel Help Apply

Nominal RPM [RPM]

Pump nominal working RPM, the characteristics in lookup tables are given at this speed.

$Q[m^3/hr] = f(Head[m])$ at nominal RPM. At other RPMs, $Q/N = f(Head/N^2)$ will be used. N is RPM of pump.

Volumetric flow as a function of head lookup table values

$\eta[\%] = f(Q[m^3/hr])$ at nominal RPM. At other RPMs, $\eta = f(Q/N)$ will be used. N is RPM of pump

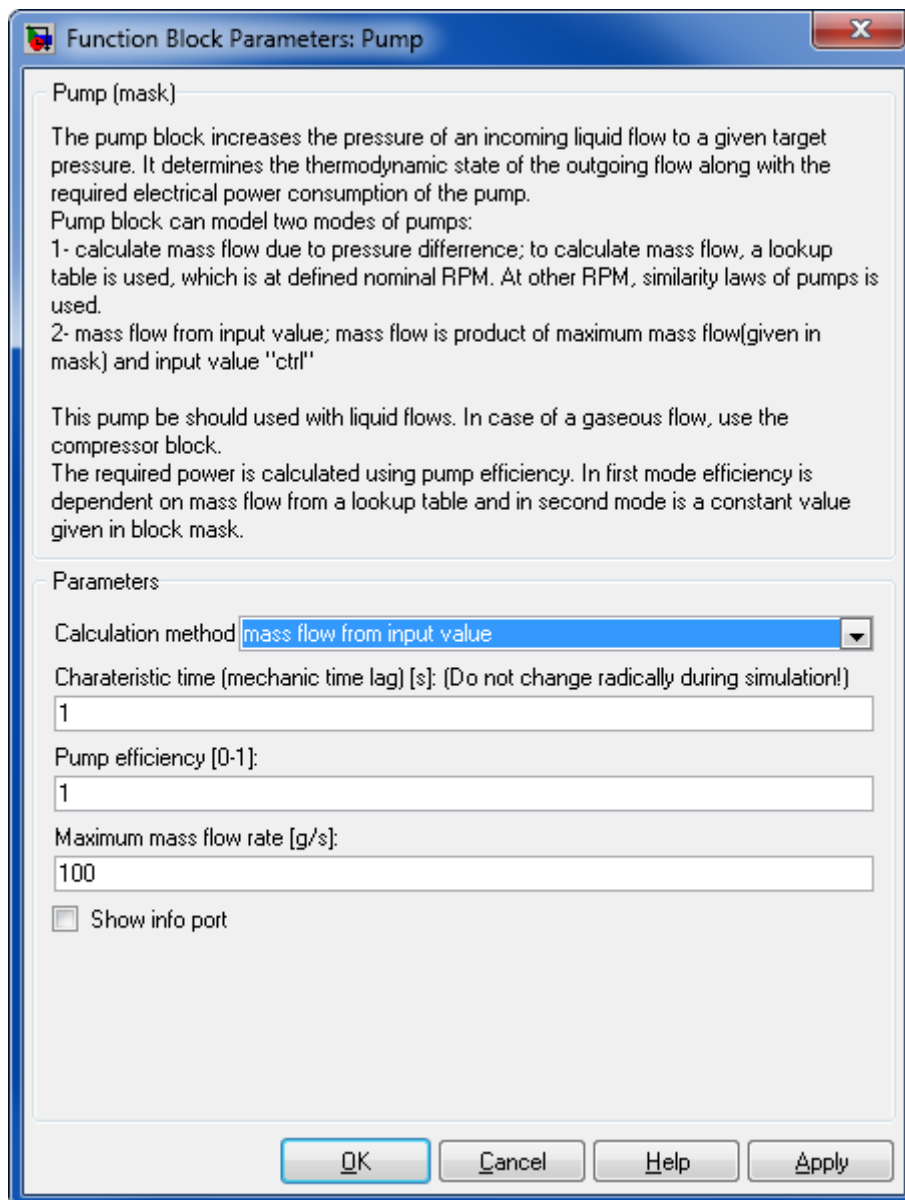
Isentropic efficiency as a function of volumetric flow lookup table values

Characteristic Time (mechanic time lag) [s]

Characteristic time

Show info port

Show or hide info port.

Mode 2 (mass flow from input value)**Characteristic Time (mechanic time lag) [s]**

Characteristic time

Pump Efficiency [0-1]

Pump efficiency in the range [0,1]

Max mass flow rate [g/s]

Maximum mass flow rate

Show info port

Show or hide info port.

Characteristics

Direct Feedthrough	Yes
--------------------	-----

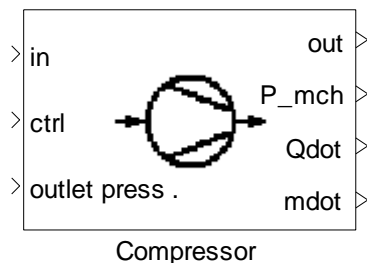
8.2.3.5 Compressor

Compressor model with four operating modes

Library

Components/Hydraulic

Description



The Compressor block increases the pressure of an incoming flow to a given outlet pressure or a given mechanical power. There are four operating modes available.

Mode 1: isentropic

The isentropic compressor model determines the thermodynamic state of the outgoing flow along with the compressor's required mechanical power consumption at a given isentropic efficiency. The efficiency is retrieved from a lookup table as a function of the mass flow.

The realized output mass flow rate is determined by multiplication of a given maximum mass flow rate and a given control value. The mass flow rate from the input media is ignored.

$$\dot{m} = \dot{m}_{max} \times ctrl$$

From the first law of thermodynamics for compressor

$$P_{mch} = \dot{m}(h_{out} - h_{in})$$

and isentropic efficiency

$$\eta_s = \frac{h_{out,s} - h_{in}}{h_{out} - h_{in}}$$

the mechanical power can be calculated as

$$P_{mch} = \frac{\dot{m}(h_{out,s} - h_{in})}{\eta_s}$$

A characteristic time is used to delay the mass flow. For this purpose, we use a first-order transfer function with a time constant. The corresponding Transfer Function block implements the transfer function:

$$\frac{y(s)}{u(s)} = \frac{1}{\tau s + 1}$$

Where, u denotes the control signal for mass flow rate, y the delayed control signal, and τ the characteristic time.

Mode 2: polytropic

In this mode, a polytropic relation ($p v^n = \text{constant}$) between the input and output states is used to determine the thermodynamic state of the output and the required mechanical power.

$$P_{mch} = \frac{n(p_2 v_2 - p_1 v_1)}{1 - n}$$

$$\frac{T_2}{T_1} = \left(\frac{p_2}{p_1}\right)^{\frac{(n-1)}{n}}$$

Where P_{mch} denotes required electric power per mass flow rate, p pressure and v specific volume. Index 1 indicates the input to the compressor, index 2 indicates the output.

Energy conservation is used to calculate required heat exchange with environment and is given as Qdot at the output.

The realized output mass flow rate is determined by multiplication of a given maximum mass flow rate and a given control value. The mass flow rate from the input media is ignored.

$$\dot{m}_{out} = \dot{m}_{max} \times ctrl$$

A characteristic time is used to delay the mass flow. For this purpose, we use a first-order transfer function with a time constant. The corresponding Transfer Function block implements the transfer function:

$$\frac{y(s)}{u(s)} = \frac{1}{\tau s + 1}$$

Where, u denotes the control signal for mass flow rate, y the delayed control signal, and τ the characteristic time.

Mode 3: mechanic

The mechanic Compressor model increases the pressure of an incoming flow to a higher pressure. It uses a given amount of mechanical power to determine the thermodynamic state of the outgoing flow and the produced heat rate during the process. A polytropic relation ($p v^n = \text{constant}$) between the input and output states is used to determine the thermodynamic state of the output and the required mechanical power.

$$P_{mch} = \frac{n(p_2 v_2 - p_1 v_1)}{1 - n}$$

$$\frac{T_2}{T_1} = \left(\frac{p_2}{p_1}\right)^{\frac{(n-1)}{n}}$$

Where P_{mch} denotes required mechanical power per mass flow rate, p pressure and v specific volume. Index 1 indicates the input to the compressor, index 2 indicates the output. Energy conservation is used to calculate the required heat exchange with environment and is given as \dot{Q} at the output.

Mode 4: active (block determines mass flow)

The active compressor increases the pressure of an incoming flow to a given outlet pressure. It calculates the mass flow rate and mechanical power consumption depending on the isentropic efficiency and determines the outgoing flow state.

Compressor mass flow rate is a function of given rotational speed (RPM) and pressure ratio (PR) due to lookup table in mask.

The values of the mass flow rate and isentropic efficiency are retrieved from a compressor map as given in the block mask. The map is described here with a vector of RPMs, and the corresponding pressure ratio, mass flow and isentropic efficiency values along each RPM line. The values are corrected to a reference point given.

The pressure ratio values are given as a matrix in which a row introduces an RPM line and columns are the values of pressure ratio for some selected points on that line. Mass flow rate and isentropic efficiency values are given for this specific points in consequent matrices that similarly, in each matrix a row introduces an RPM line and columns are the values of mass flow or isentropic efficiency for points on that RPM line; Note that in a matrix, the number of columns for all rows should be the same. I.e. having an equal number of points on each RPM, if not, you can add zeros to the end of rows to match a constant number of columns. These values are not used, instead the model will extrapolate based on the non zero values in this area.

Because of different values and ranges of pressure ratios on each RPM line, the map data entered in the mask is regenerated and interpolated for a fixed set of pressure ratio grid points to allow a single lookup table formation for all RPMs in the whole range of pressure ratios.

These grid points are also given in the block mask.

Mass flow rate and efficiency outside of surge and choke lines of compressor, are assumed constant. This causes the interpolation method can be used in all range of compressor performance. Mass flow rate outside of choke and surge lines, is placed by minimum and maximum value of lookup table, respectively.

Also, efficiency outside of choke and surge lines, is placed by minimum and maximum value of lookup table, respectively.

Data Type Support

Inputs (Mode 1)

in	Incoming flow	Flow bus
ctrl	Fraction of maximum mass flow	Scalar [0-1]
outlet press.	Target pressure at the output	Scalar [Pa]

Outputs (Mode 1)

out	Outgoing flow	Flow bus
P_mch	Mechanical power consumption	Scalar [W]
mdot	Mass flow rate	Scalar [g/s]
info	N.A. (Optional)	N.A.

Inputs (Mode 2)

in	Incoming flow	Flow bus
ctrl	Fraction of maximum mass flow	Scalar [0-1]
outlet press.	Target pressure at the output	Scalar [Pa]

Outputs (Mode 2)

out	Outgoing flow	Flow bus
P_mch	Mechanical power consumption	Scalar [W]
Qdot	Heat exchange	Scalar [W]
mdot	Mass flow rate	Scalar [g/s]
info	N.A. (Optional)	N.A.

Inputs (Mode 3)

in	Incoming flow	Flow bus
P_mch	Input mechanical power	Scalar [W]
PFin	Pressure feedback for output (Optional)	4-doubles vector

Outputs (Mode 3)

out	Outgoing flow	Flow bus
Qdot	Heat exchange with environment	Scalar [W]
PFout	Pressure feedback for input (Optional)	4-doubles vector
info	N.A. (Optional)	N.A.

Inputs (Mode 4)

in	Incoming flow	Flow bus
RPM	Rotational speed	Scalar [RPM]
outlet press.	Target pressure at the output	Scalar [Pa]

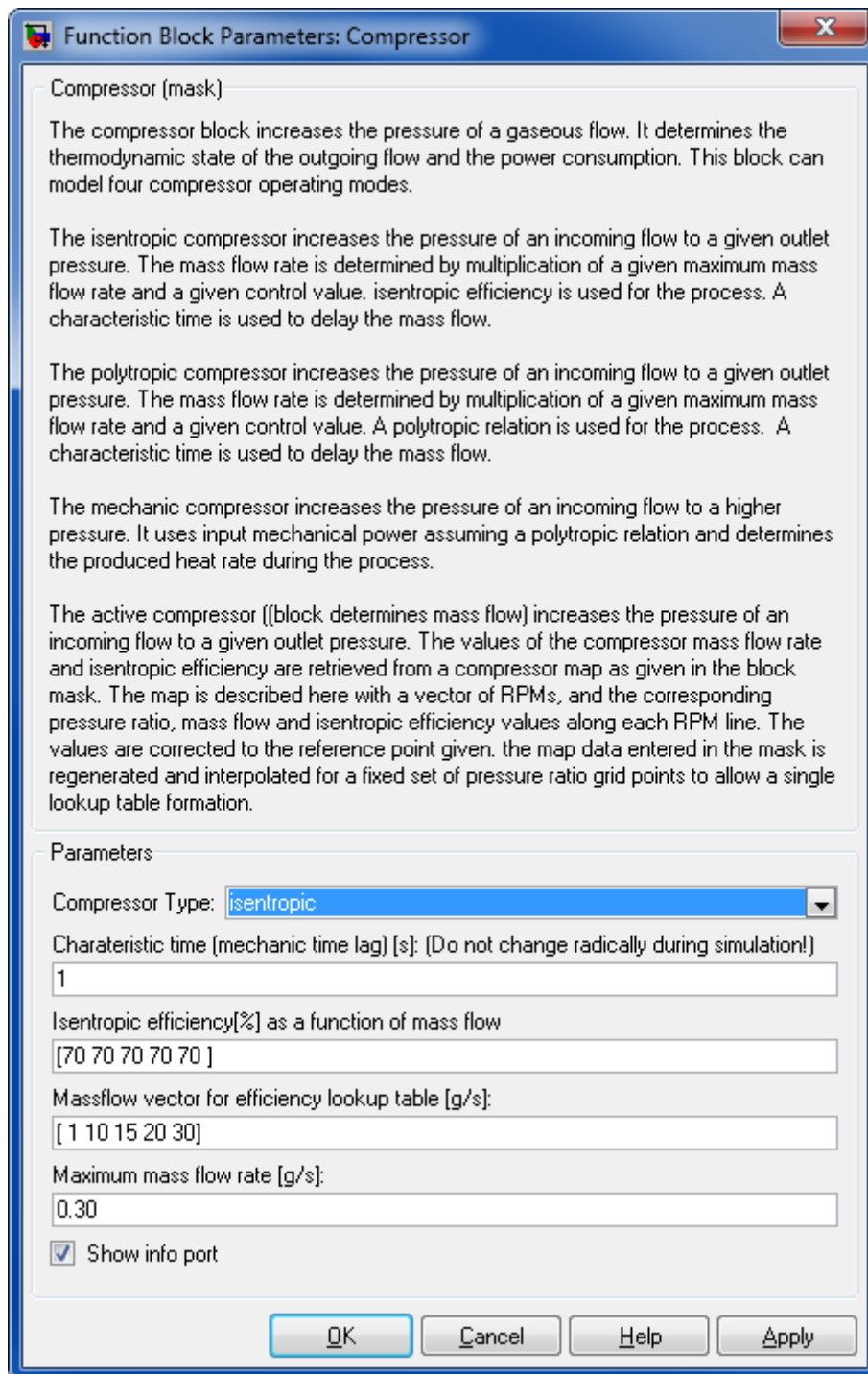
Outputs (Mode 4)

out	Outgoing flow	Flow bus
P_mch	Mechanical power consumption	Scalar [W]
mdot	Mass flow rate	Scalar [g/s]
info	N.A. (Optional)	N.A.

Parameters and Dialog Box**Compressor Type:**

Four compressor modes

Mode 1:isentropic



Characteristic time (mechanic time lag) [s]

Characteristic time.

Isentropic efficiency as a function of mass flow

Efficiency lookup table first row: the isentropic efficiency as a function of mass flow

Mass-flow vector for efficiency lookup table [g/s]

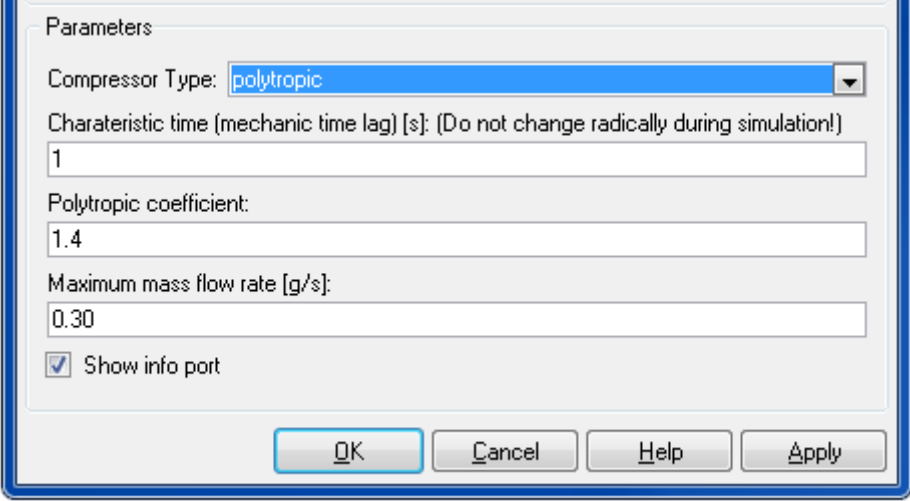
Efficiency lookup table second row: the mass flow vector for the isentropic efficiency lookup table.

Maximum Mass Flow Rate [g/s]

Maximum mass flow rate.

Show info port

Show or hide info port.

Mode 2: polytropic

Parameters

Compressor Type: **polytropic**

Characteristic time (mechanic time lag) [s]: (Do not change radically during simulation!)

1

Polytropic coefficient:

1.4

Maximum mass flow rate [g/s]:

0.30

☒ Show info port

OK Cancel Help Apply

Characteristic time (mechanic time lag) [s]

Characteristic time.

Polytropic Coefficient

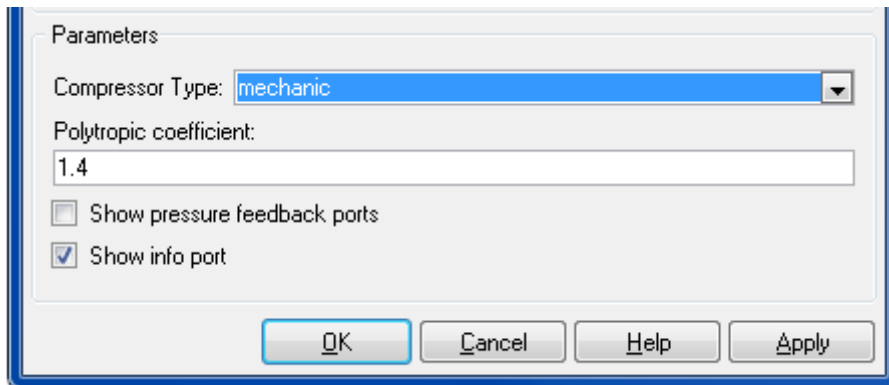
Exponent for polytropic change of state.

Maximum Mass Flow Rate [g/s]

Maximum mass flow rate.

Show info port

Show or hide info port.

Mode 3: mechanic**Polytropic Coefficient**

Exponent for polytropic change of state

Show pressure feedback ports

Show or hide pressure feedback ports.

Show info port

Show or hide info port.

Mode 4: active (*Calculates mass flow*)

Parameters

Compressor Type: **active (block determines mass flow)**

Reference temperature [K] and pressure [Pa]
[350 1e5]

Vector of compressor map RPMs [RPM]
[5e4 1e5 1.3e5 1.6e5]

Matrix of pressure ratio(PR) values at each RPM [-]
[1.25 1.2 1.18 1.15 1 ; 1.85 1.75 1.55 0 0; 2.7 2.69 2.05 0 0; 3.7 3.5 2.8 0 0]

Matrix of mass flow values at each RPM and PR [g/s]
[0.08 0.1 0.11; 0.08 0.18 0.255 0 0; 0.14 0.26 0.35 0 0; 0.22 0.34 0.39 0 0]*455/60

Matrix of efficiency values at each RPM and PR [%]
[70 70 70 0 0; 70 70 70 0 0; 70 70 70 0 0; 70 70 70 0 0]

Vector of pressure ratio lookup table grid points
1:0.05:4

☒ Show info port

OK Cancel Help Apply

Reference temperature [K] and pressure [Pa]

The reference point for lookup map correction

Vector of compressor map RPMs [RPM]

RPM values for RPM lines of the map

Matrix of pressure ratio(PR) values at each RPM [-]

Pressure ratio values for selected points on each RPM line

Matrix of mass flow values at each RPM and PR [g/s]

Mass flow values for points with the specific pressure ratios selected on each RPM line

Matrix of efficiency values at each RPM and PR [%]

Isentropic efficiency values for points with the specific pressure ratios selected on each RPM line

Vector of pressure ratio lookup table grid points

Fixed set of break points for lookup table pressure ratios in whole range

Show info port

Show or hide info port.

Characteristics

Direct Feedthrough	Yes
--------------------	-----

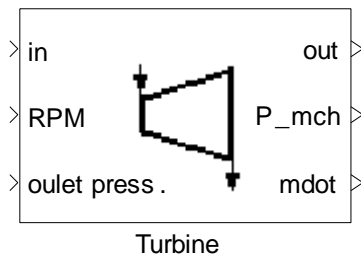
8.2.3.6 Turbine

Turbine model with two operating modes.

Library

Components/Hydraulic

Description



The Turbine block decreases the pressure of an incoming flow to a given outlet pressure. It determines the thermodynamic state of the outgoing flow along with the produced mechanical power at given isentropic efficiency. Turbine is adiabatic and should be used with gaseous flows. This block can model two operating modes.

Mode 1: passive (mass flow remains unchanged)

The thermodynamic state of the outgoing flow along with the produced mechanical power is determined at given isentropic efficiency. The isentropic efficiency is retrieved from a lookup table as a function of the mass flow and is defined as ratio of actual enthalpy difference to enthalpy difference for isentropic change of state with the same pressure drop.

First law of thermodynamics for turbine:

$$P_{mch} = \dot{m}(h_{in} - h_{out})$$

$$\eta_s = \frac{\Delta h}{\Delta h_s}$$

$$P_{mch} = \dot{m} \times \eta_s \times \Delta h_s$$

Subscript *s* is for isentropic state change.

Mode 2: active (block determines mass flow)

The active turbine calculates the mass flow rate and mechanical power consumption depending on the isentropic efficiency and determines the outgoing flow state.

The values of the turbine mass flow rate are functions of given rotational speed (RPM) and given pressure ratio (PR).

The values of the mass flow rate and isentropic efficiency are retrieved from a turbine map as given in the block mask. The map is described here with a vector of RPMs, and the corresponding pressure ratio, mass flow and isentropic efficiency values along each RPM line. The values should be given at a reference point.

The pressure ratio values are given as a matrix in which a row introduces an RPM line and columns are the values of pressure ratio for some selected points on that line. Mass flow rate and isentropic efficiency values are given for this specific points in consequent matrices that similarly, in each matrix a row introduces an RPM line and columns are the values of mass flow or isentropic efficiency for points on that RPM line; Note that in a matrix, the number of columns for all rows should be the same. I.e. having an equal number of points on each RPM line, if not, you can add zeros to the end of rows to match a constant number of columns. The values of pressure ratio in lookup table have to begin with value 1 (one).

Because of different values and ranges of pressure ratios on each RPM line, the map data entered in the mask is regenerated and interpolated for a fixed set of pressure ratio grid points to allow a single lookup table formation for all RPMs in the whole range of pressure ratios.

These grid points are also given in the block mask.

Data Type Support

Inputs (Mode 1)

in	Incoming flow	Flow bus
outlet press.	Target pressure at the output	Scalar [Pa]

Outputs (Mode 1)

out	Outgoing flow	Flow bus
P_mch	Mechanical power consumption	Scalar [W]
mdot	Mass flow rate	Scalar [g/s]
info	N.A. (Optional)	N.A.

Inputs (Mode 2)

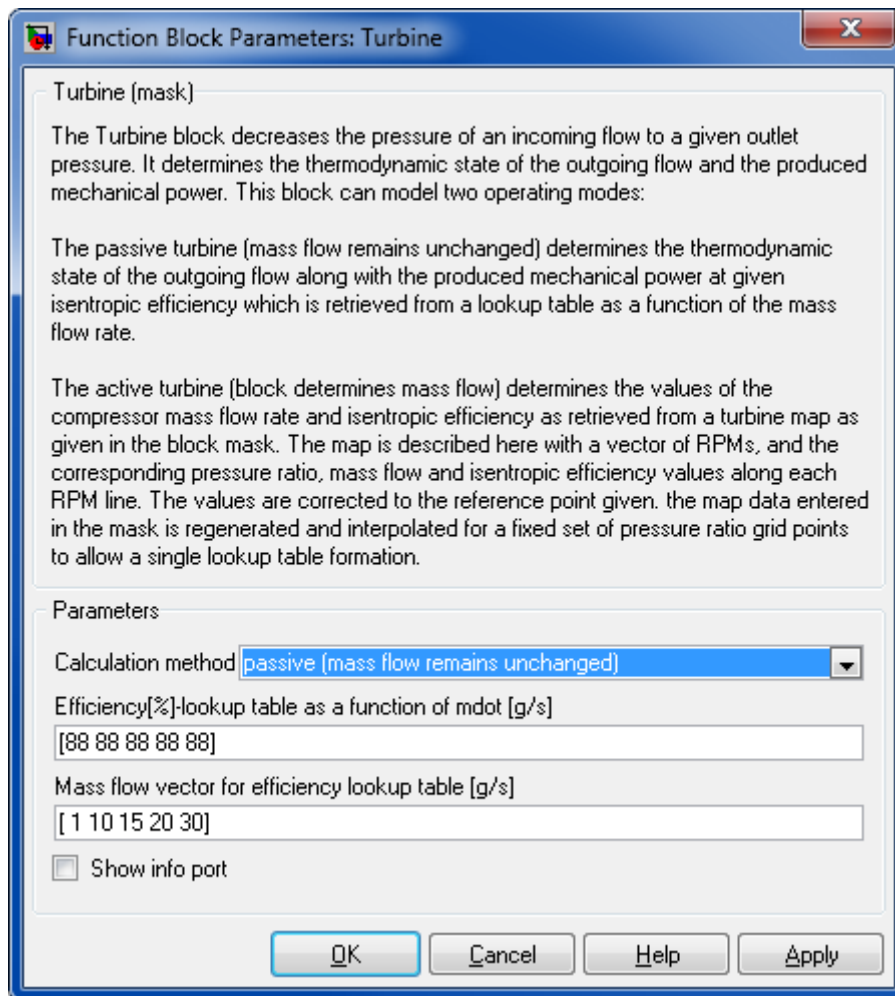
in	Incoming flow	Flow bus
RPM	Rotational speed	Scalar [RPM]
outlet press.	Target pressure at the output	Scalar [Pa]

Outputs (Mode 2)

out	Outgoing flow	Flow bus
P_mch	Mechanical power consumption	Scalar [W]
mdot	Mass flow rate	Scalar [g/s]
info	N.A. (Optional)	N.A.

Parameters and Dialog Box

Mode 1: passive (mass flow remains unchanged)



Efficiency [%]-lookup table as a function of mdot [g/s]

Efficiency lookup table first row: the isentropic efficiency as a function of mass flow

Mass flow vector for efficiency lookup table [g/s]

Efficiency lookup table second row: the mass flow vector for the isentropic efficiency lookup table.

Show info port

Show or hide info port.

Mode 2: active (block determines mass flow)

Parameters

Calculation method **active (block determines mass flow)**

Reference temperature [K] and pressure [Pa]
[298 1e5]

Vector of turbine map RPMs [RPM]
[10000 20000 30000]

Matrix of pressure ratio (PR) values at each RPM [-]
[1 1.25 2 2.5 3 ; 1 1.25 2 2.5 3 ; 1 1.25 2 2.5 3]

Matrix of mass flow values at each RPM and PR [g/s]
[0 12 18.5 20 20.5; 0 15 22.5 23.5 23.5; 0 18 23 26.5 26.7]

Matrix of efficiency values at each RPM and PR [%]
[70 70 70 70 70; 70 70 70 70 70; 70 70 70 70 70]

Vector of pressure ratio lookup table grid points
1:0.1:3

☐ Show info port

OK Cancel Help Apply

Reference temperature [K] and pressure [Pa]

The reference point for lookup map correction

Vector of turbine map RPMs [RPM]

RPM values for RPM lines of the map

Matrix of pressure ratio (PR) values at each RPM [-]

Pressure ratio values for selected points on each RPM line

Matrix of mass flow values at each RPM and PR [g/s]

Mass flow values for points with the specific pressure ratios selected on each RPM line

Matrix of efficiency values at each RPM and PR [%]

Isentropic efficiency values for points with the specific pressure ratios selected on each RPM line

Vector of pressure ratio lookup table grid points

Fixed set of break points for lookup table pressure ratios in whole range

Show info port

Show or hide info port.

Characteristics

Direct Feedthrough	Yes
--------------------	-----

8.2.4 Tanks

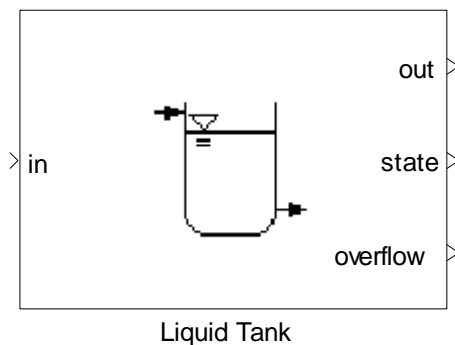
8.2.4.1 Liquid Tank

A tank for purely liquid mixtures

Library

Components/Tanks

Description



This block simulates a tank for liquid mixtures. The tank can be modeled either as a constant pressure tank or a constant volume tank. It is assumed, that

- The tank is adiabatic.
- The mechanical work exchange with environment in case of constant pressure is neglected.

The initial tank mass, the initial temperature and the initial molar fractions of the tank are defined in the mask.

Constant Pressure Liquid Tank

In a constant pressure tank, the pressure of the tank is held constant at the value specified in the mask. This mode can be used e.g. to simulate typical open water tanks, at environment pressure.

The output molar flow is defined via the molar flow output method which can be one of three options:

$$\dot{m}_{out} = \dot{m}_{in}$$

\dot{m}_{out} given as external input

$\dot{m}_{out} = f(PF)$ The mass flow is determined using the pressure feedback mechanisms of Thermolib.

If option 1 is chosen, then the output mass flow is always equal to the input mass flow. The second input port has no meaning then.

If option 2 is chosen, then the output mass flow can be given at the second input port (in [g/s]).

If the mass of the constant-pressure tank falls below the minimum tank mass, there would be no output molar flow, regardless of the output method selected. If the mass of the constant-pressure tank rises above the maximum tank mass, then the output molar flow is calculated according to the selected output method, and the rest is output through the output port 'Overflow'.

Constant Volume Liquid Tank

In a constant volume tank, the volume of the tank is assumed constant and the tank is always full. The output molar flow is always equal to the input molar flow. This mode can be used e. g. to simulate closed water tanks, which are completely filled with water.

Data Type Support

Inputs

in	Incoming flow	Flow bus
	Port describing the output mass flow. If the output method selected is $\dot{m}_{out} = \text{external}$, then the output mass flow will be equal to the value input at this port. If the output method selected is $\dot{m}_{out} = \dot{m}_{in}$, then the output mass flow is equal to the input mass flow. If $\dot{m}_{out} = f(PF)$ is selected, the output massflow rate is calculated via pressure feedback.	[g/s]/-
\dot{m}_{out}/PF_{in}		

Outputs

out	Outgoing flow	Flow bus
state	State of the tank: describes the mass content, temperature, pressure and composition of the tank.	State bus
Overflow	Overflowing flow from the tank. Is used only if the tank model is constant pressure	Scalar [g/s]
PFout	Pressure feedback for inlet (Optional)	4-doubles vector
info	Error information (Optional)	N.A.

Parameters and Dialog Box

th_LiquidTank

Description

This block simulates a tank for purely fluid mixtures. The tank can be modeled either as a constant pressure tank or a constant volume. In a constant pressure tank, the output molar flow is calculated dependent on the molar flow output method. In case the tank overflows, the overflow goes out the third output. In a constant volume tank, the output molar flow is always equal to the input molar flow.

Parameters

Tank model type:

Constant volume

Molar flow output method:

m.out = m.in

Initial tank mass [g]:

1e6

Initial temperature [K]:

300

Pressure [Pa]:

2e5

Minimum tank mass [g]:

10

Maximum tank mass [g]:

400

Max. mass flow with PF [g/s]

0

Initial mass flow with PF [g/s]

0

Relaxation factor for PF [-]

0

Initial Liquid Fractions

CO molar fraction [-]:

0

H2O molar fraction [-]:

1

CO2 molar fraction [-]:

0

H2 molar fraction [-]:

0

N2 molar fraction [-]:

0

O2 molar fraction [-]:

0

☐ Show info port

☐ Show PF-port for inlet

OK

Cancel

Help

Tank model type

The type of the tank's model. Two options are available, a constant pressure tank or a constant volume tank. For the latter choice, the parameters initial pressure, minimum tank mass and maximum tank mass are disabled.

Molar flow output method

This parameter determines how the output mass flow is calculated. Three choices are available: $m.out = m.in$, the output mass flow is equal to the input mass flow; $m.out = external$, the output mass flow is equal to the value input at the second port; $m.out = f(PF)$, the output mass flow is a function of the pressure value given at the second input port.

Initial tank mass [g]

This is the initial mass of the tank.

Initial temperature [K]

The initial temperature of the tank. The temperature of the tank will change as the input flow changes during the simulation.

Pressure [Pa]

The initial pressure of the tank. The pressure of the tank will change as the input flow changes during the simulation. This parameter is available only a constant-pressure tank model is used.

Minimum tank mass [g]

This is the minimum mass of the tank in grams. If the tank's mass falls below this value, then the output mass flow is reduced to zero regardless of the selected molar flow output method. This parameter is available only a constant-pressure tank model is used.

Maximum tank mass [g]

This is the maximum mass of the tank in grams. If the tank's mass reaches this value, then the output mass flow is equal to the input mass flow regardless of the selected molar flow output method. This parameter is available only a constant-pressure tank model is used.

Maximum mass flow with PF [g/s]

Maximum allowable mass flow rate if pressure feedback mechanism is used.

Initial mass flow with PF [g/s]

Initial mass flow rate, if pressure feedback mechanism is used.

Relaxation factor for PF [-]

Relaxation factor used by pressure feedback mechanism.

Initial liquid fractions

The initial composition of media in the tank. This composition changes as the input flow changes during the simulation.

Show PF-port for inlet

Show or hide pressure feedback ports.

Show info port

Show or hide info port.

Characteristics

Direct Feedthrough	Yes
--------------------	-----

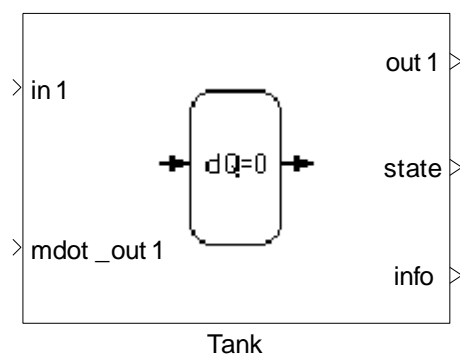
8.2.4.2 Tank

Tank with three operation modes for vapor and vapor-liquid phases.

Library

Components/Tanks

Description



The tank block can model three types of tanks:

- Generic
- Adiabatic

- Isothermal

This block simulates a general non-adiabatic tank in generic mode. In this mode, heat exchange can be defined via explicit input "Qdot". Adiabatic mode simulates an adiabatic tank where there is no heat exchange to the environment. In isothermal mode, block simulates an isothermal tank. So there will be heat exchange to the environment to maintain the initial temperature. The heat exchange with the environment is calculated and given at output "Qdot".

The initial values for composition inside the tank, pressure and temperature are defined in the block mask. During the simulation, these values will change according to the input and output flows.

The following assumptions are made in all types:

- The tank has a fixed volume. This means no work is done to the environment.
- The concentrations and temperature in the tank are equally distributed. The output flow always has the temperature, pressure and concentrations of the tank volume.
- The pressure in the tank cannot fall below a certain value. The simulation will stop if the lower limit is not satisfied.
- Input flow is always accepted by the tank.

The tank can model its content to be vapor or in vapor-liquid equilibrium. For liquid media, please use the Liquid Tank block.

The number of input and output flows can be freely configured to be 0, 1 or 2 (input and output separately). The corresponding input ports appear automatically after closing the parameter mask.

For output flows the following properties can be configured:

- output flow determination method (external or via pressure feedback)
If this option is set to "external" then an input "mdot_out1/2" appears, where the mass flow in [g/s] can be given. If set to "f(PF)", the ports are labeled "PFin" and have to be connected to the next block downstream connected to the corresponding out-port. The mass flow is then determined using the pressure feedback mechanisms of Thermolib.
- Position at tank (top / bottom)
Using this setting the phase (and composition) of the outflow is determined. If there is (at least some) liquid in the tank, then at the bottom position will be output liquid and has the liquid-phase composition. If there is (at least some) gas in the tank, then at the top-position output flow will be gas and has gas-phase composition.

The initial condition in the tank can be specified via a separate dialog. To set the initial condition click on the "Edit ..." button in the "Initial Condition" area in the lower part of the parameter mask.

The initial condition dialog enables you to specify pure or mixture conditions. For the pure conditions the number of phases can be chosen and for the 2-phase case the set of variables to specify the initial condition can be chosen ("temperature and vapor fraction" or "pressure and vapor fraction"). For specification of mixtures you give temperature and pressure as independent variables.

In the lower part the composition of the mixture is configured.

For the molar quantities of the species n_i stored in the tank the ODE

$$\frac{dn_i}{dt} = \dot{n}_{i,in} - \dot{n}_{i,out}$$

is implemented. To calculate the complete thermodynamic state (especially T and p) for the tank content the equation

$$\frac{dU}{dt} = \dot{H}_{in} - \dot{H}_{out} + \dot{Q}$$

and the calculated density from the mass balance is used in an iteration algorithm.

In adiabatic mode, the above equation can be simplified to the following form

$$\frac{dU}{dt} = \dot{H}_{in} - \dot{H}_{out}$$

In isothermal mode, for determination of the pressure an iteration algorithm is used to satisfy the density constraint given by the material balance and the constant temperature. The isothermal tank should be used to simulate slow process with good heat transfer with environment. That means tanks with negligible temperature variation. This may also hold for tanks which are operated under temperature control. A bottle of hydrogen, from which slowly hydrogen is taken out and which has nearly environment temperature is such an example.

Data Type Support

Inputs

in	Incoming flow	Flow bus
mdot_out/PFin	Port describing the output mass flow. If the output method selected is m.out = external, then the output mass flow will be equal to the value	[g/s]/-

	input at this port. If m.out = f(PF) is selected, then), the output massflow rate is calculated via pressure feedback	
Qdot	Heat exchange rate. (Only in generic mode)	[W]

Outputs

Qdot	Heat exchange with environment, necessary to keep temperature constant. (Only in isothermal mode)	[W]
out	Outgoing flow	Flow bus
state	State of the tank: describes the mass content, temperature, pressure and composition of the tank.	State bus
PFout1	Pressure feedback for input1 (Optional)	4-doubles vector
PFout2	Pressure feedback for input2 (Optional)	4-doubles vector
Info	Error information (Optional)	N.A.

Parameters and Dialog Box

th_Tank

Description

This block simulates an adiabatic tank in which no heat exchange with the environment takes place. The initial media concentrations, pressure and temperature are defined below as parameters. During the simulation, these values will change according to the input and output flows.

This tank can be used with purely gaseous media or a mixture of gas/liquid. For purely liquid media, please use the Liquid Tank block.

Parameters

Tank mode:

☐ Generic Tank
☒ Adiabatic Tank
☐ Isothermal Tank

Number of inlet flows: 1

☐ show PFout-ports for inlets

Outlets:

☒ enable outlet 1: top m.out1: m.out = external
☐ enable outlet 2: bottom m.out2: m.out = external

Tank volume [m³]: 0.4

Minimum tank pressure [Pa]: 1e5

Max. mass flow with PF [g/s]: 1000

Initial mass flow with PF [g/s]: 1

Relaxation factor for PF [-]: 0.3

Initial Condition

psi(H2O)=1
T = 273+410K; p = 350e3Pa

Edit ...

☒ Show info port

OK Cancel Help

Tank mode

Specify the tank mode (generic, adiabatic and isothermal)

Number of inlet flows

The number of inlet flows that are feed into the tank. Input ports are automatically added / removed as necessary after leaving the dialog.

Show PFout-ports for inlet

Show or hide pressure feedback ports of inlet.

Outlets

Specifies whether outputs ports are enabled or disabled. For each output port there is the choice if it is at the top or at the bottom of the tank.

If position is chosen to "top", then output contains gas phase composition as soon as gas phase is existent (otherwise liquid phase composition). If position is "bottom" the corresponding output will contain liquid phase composition if there is liquid phase in the tank or gas phase if there is no liquid phase.

Via a combo box user can choose the method to determine the mass flow at the corresponding output. If chosen " \dot{m} =external" the mass flow can be given as external input signal [g/s]. If chosen " \dot{m} =f(PF)" the mass flow is calculated using pressure feedback.

Tank volume [m³]

The volume of the tank.

Minimum tank pressure [Pa]

The minimum allowed pressure of the tank. When the tank pressure reaches this value, the molar flow of the output is zero.

Maximum mass flow with PF [g/s]

Maximum allowable mass flow rate if pressure feedback mechanism is used.

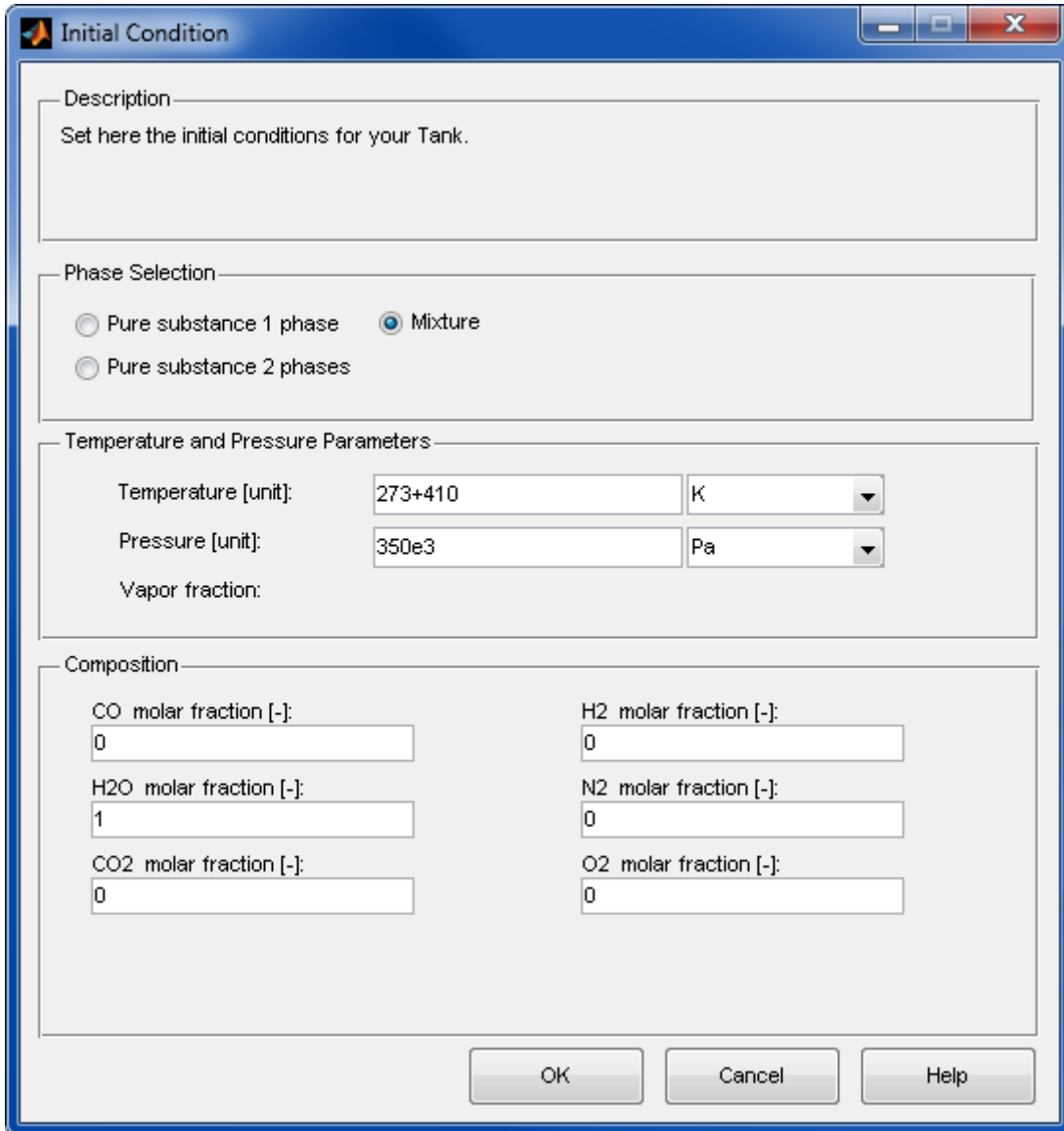
Initial mass flow with PF [g/s]

Initial mass flow rate, if pressure feedback mechanism is used.

Relaxation factor for PF [-]

Relaxation factor used by pressure feedback mechanism.

Initial Condition



The dialog box is titled "Initial Condition" and contains four main sections:

- Description:** A text area with the instruction "Set here the initial conditions for your Tank."
- Phase Selection:** Three radio buttons: "Pure substance 1 phase", "Mixture" (selected), and "Pure substance 2 phases".
- Temperature and Pressure Parameters:** Fields for "Temperature [unit]" (273+410, K) and "Pressure [unit]" (350e3, Pa). A "Vapor fraction:" label is present but has no input field.
- Composition:** Six input fields for molar fractions: CO (0), H2O (1), CO2 (0), H2 (0), N2 (0), and O2 (0).

At the bottom are "OK", "Cancel", and "Help" buttons.

Phase Selection

Selection whether the initial condition is pure substance composition with one phase or two phases, or if it is a mixture composition.

Temperature and pressure parameters

If pure with two phases is selected, the dialog enables the vapor fraction field and radio-buttons to choose temperature or pressure as the variable additional to vapor fraction. Otherwise only temperature and pressure fields are enabled to specify the initial condition.

Composition

The initial composition of media in the tank. The composition changes as the input flow changes during the simulation using the mass balance above.

Show info port

Show or hide info port.

Characteristics

Direct Feedthrough	No
--------------------	----

8.2.5 Hydraulics with Pressure Feedback

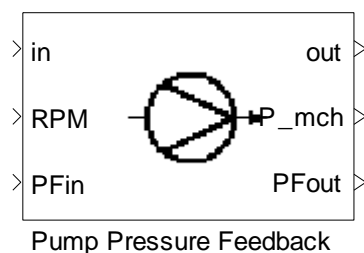
8.2.5.1 Pump Pressure Feedback

Pump model with pressure feedback in two modes

Library

Components/Hydraulic with Pressure Feedback

Description



The pump with pressure feedback can operate in two modes:

Mode1: feed pump

The feed Pump block raises the flows pressure by a certain amount, depending on the pumps RPM (revolutions per minute) and the incoming mass flow (mdot). The relation between the pressure rise, RPM and mdot is described in a lookup table given in the blocks mask. The pressure feedback information is simply transferred unchanged and doesn't affect the pump flow rate.

Mode2: Circulation pump

The circulation pump serves as a pump in closed flow circuits, which the user wants to calculate with pressure feedback. The relation between the pressure rise, RPM and mdot is described in a lookup table given in the blocks mask.

Pressure-feedback information at PFin port is used by the Pump block, which adjusts the amount of mass flow passing through so that the difference between the reference pressure (P_{ref}) and the current pressure (P_{is}) reaches zero. In other words, the Pump block corrects the overall output mass flow in such a way, that input pressure reaches the pressure reference defined in the mask. Using this block you can avoid using a downstream block in a closed circuit. The maximum mass flow rate that the Circulation Pump can generate is defined by the user in the mask.

For both modes, In addition to the outgoing flow, electric power consumption is calculated.

The required electric power P_{mch} is calculated as:

$$P_{mch} = \frac{\dot{m}\Delta p}{\rho\eta}$$

Where, ρ denotes the density.

The output enthalpy is calculated as:

$$\dot{H}_{out} = \dot{H}_{in} + P_{mch}$$

This block should be used with liquid or in incompressible flows only. In case of a gaseous input flow, the output port "info" produces a warning.

Data Type Support

Inputs (Modes 1, 2)

in	Incoming flow	Flow bus
RPM	Rounds per minute	Scalar
PFin	Pressure feedback (Optional in Mode 1)	4-doubles vector

Outputs (Modes 1, 2)

out	Outgoing flow	Flow bus
P_mch	Electric power	Scalar [W]
PFout	Pressure feedback (Optional in Mode 1)	4-doubles vector
info	Error information (Optional in Mode 1)	N.A.

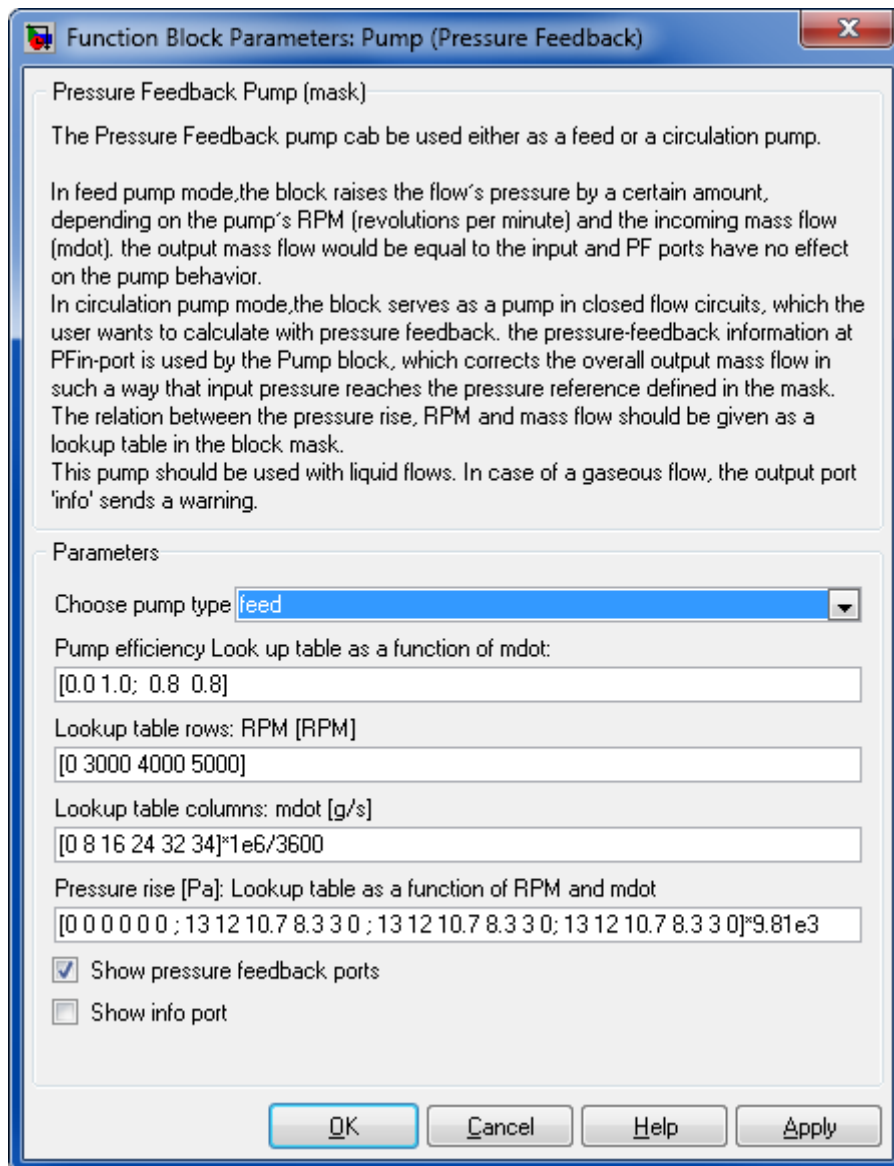
Parameters and Dialog Box

Choose pump type:

The two pump modes that can be selected:

- Mode1: feed pump
- Mode2: Circulation pump

Mode 1: feed



Pump Efficiency Look up table as a function of mdot:

The pump efficiency values for different mass flow rates

Lookup table rows: RPM [RPM]

A vector containing the RPM, which is used in the lookup table for pressure rise

Lookup Table columns: mdot [g/s]

A vector containing the mass flow, which is used in the lookup table for pressure rise

Pressure Rise [Pa]: Lookup table as a function of RPM and mdot

A matrix containing the pressure rise, Each row corresponds to a value of the RPM vector given, each column corresponds to a value of mdot vector given.

Show pressure feedback ports

Show or hide pressure feedback ports.

Show info port

Show or hide info port.

Mode 2: circulation

Parameters

Choose pump type: **circulation**

Pump efficiency Look up table as a function of mdot:
[0.0 1.0; 0.8 0.8]

Lookup table rows: RPM [RPM]
[0 3000 4000 5000]

Lookup table columns: mdot [g/s]
[0 8 16 24 32 34]*1e6/3600

Pressure rise [Pa]: Lookup table as a function of RPM and mdot
[0 0 0 0 0 0 ; 13 12 10.7 8.3 3 0 ; 13 12 10.7 8.3 3 0; 13 12 10.7 8.3 3 0]*9.81e3

Maximum mass flow [g/s]
34e6/3600

Reference pressure [Pa]
2e5

Relaxation factor []
10

☐ Show info port

OK Cancel Help Apply

Pump Efficiency Look up table as a function of mdot:

The pump efficiency values for different mass flow rates.

Lookup table rows: RPM [RPM]

A vector containing the RPM, which is used in the lookup table for pressure rise.

Lookup Table columns: mdot [g/s]

A vector containing the mass flow, which is used in the lookup table for pressure rise.

Pressure Rise [Pa]: Lookup table as a function of RPM and mdot

A matrix containing the pressure rise, each row corresponds to a value of the RPM vector, each column corresponds to a value of mdot vector.

Maximum mass Flow [g/s] (only in mode 2)

Maximum mass flow rate of pump.

Reference Pressure [Pa] (only in mode 2)

Desired pressure at input.

Relaxation factor (only in mode 2)

A factor that is used to iterate pressure, Default value is given in mask, the higher this value, the more aggressive the iteration behaves. This may lead to instable iterations. The lower the value the less aggressive the iteration behaves. This results in slower convergence.

Show info port

Show or hide info port.

Characteristics

Direct Feedthrough	Yes
--------------------	-----

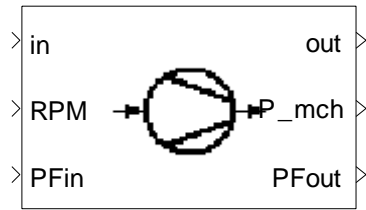
8.2.5.2 Compressor Pressure Feedback

Compressor model with pressure feedback in two modes.

Library

Components/Hydraulic with Pressure Feedback

Description



Compressor Pressure Feedback

The block model raises the input flow's pressure by a certain amount, depending on the compressor's RPM (revolutions per minute) and mass flow rate. The relation between the pressure rise, RPM and mass flow rate is described in a lookup table given in the block's mask at normal condition. In other conditions, the values of lookup table are corrected due to equations:

$$RPM_{corrected} = \frac{RPM}{\sqrt{\frac{T}{T_{ref}}}}$$

$$\dot{m}_{corrected} = \frac{\dot{m} \sqrt{\frac{T}{T_{ref}}} \left(\frac{R_{gas}}{R_{air}} \right)}{\frac{p_{in}}{p_{ref}}}$$

In addition to the outgoing flow, the compressor outputs its mechanical power consumption that is calculated with help of an adjustable isentropic efficiency lookup table defined in the block's mask.

$$\eta_s = \frac{\Delta \dot{h}_s}{\Delta \dot{h}}$$

The compressor is adiabatic and should be used only with gaseous flows.

In case of **circulation compressor** pressure-feedback information at PFin port is used by the compressor block to adjust the amount of mass flow passing through it so that the difference between the reference pressure p_{ref} and the current pressure p_{is} reaches zero. In other words, the compressor block corrects the overall output mass flow in such a way that input pressure reaches the pressure reference defined in the mask. Using this block you can avoid using a downstream block in a closed circuit. The maximum mass flow rate that the circulation compressor can generate is defined by the user in the mask.

In case of **feed compressor**, outgoing mass flow is the same as ingoing mass flow.

Data Type Support

Inputs (Modes 1, 2)

in	Incoming flow	Flow bus
RPM	Rotational speed	Scalar [RPM]
PFin	Pressure feedback in (Optional in Mode 1)	4-doubles vector

Outputs (Modes 1, 2)

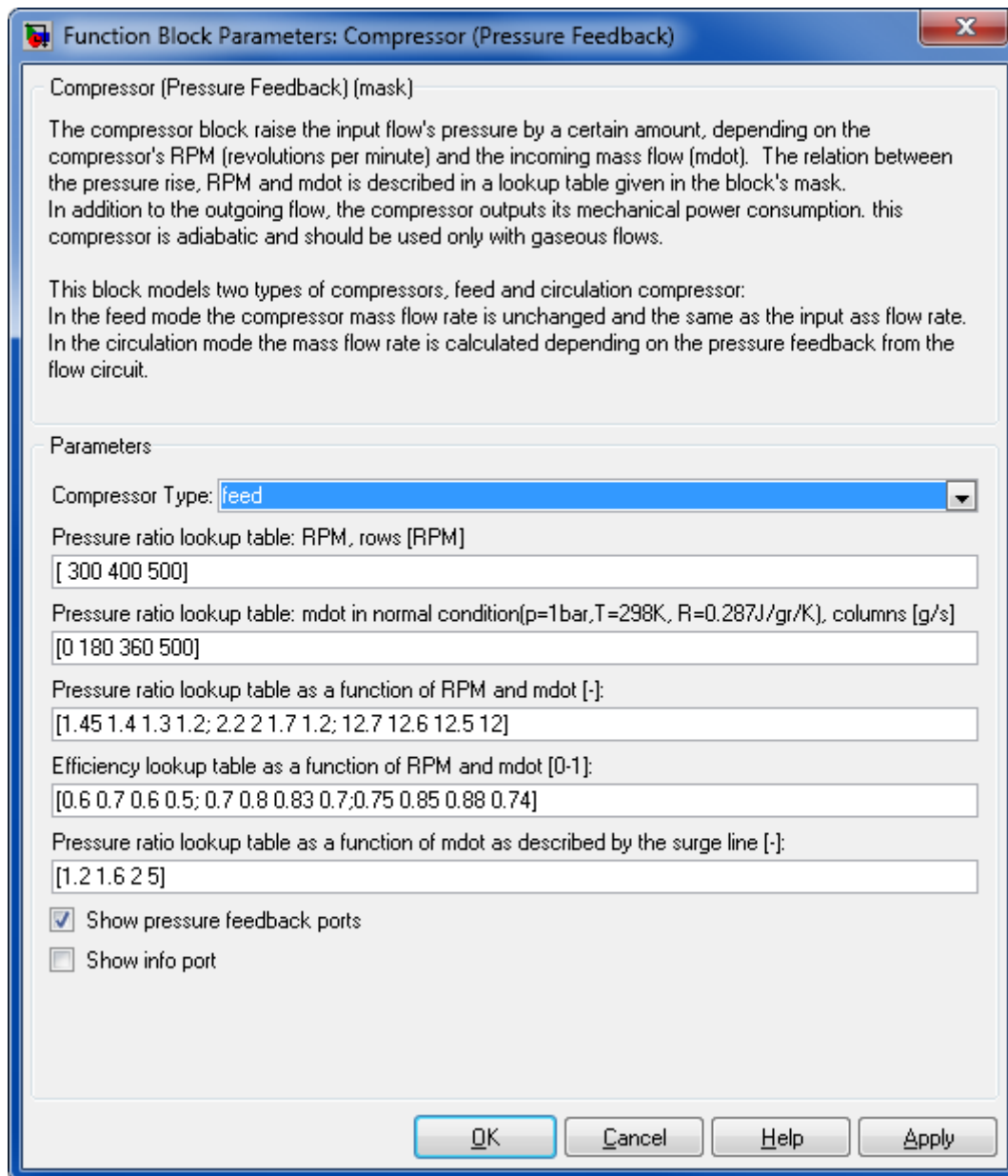
out	Outgoing flow	Flow bus
P_mch	Mechanical power consumption	Scalar [W]
PFout	Pressure feedback out (Optional in Mode 1)	4-doubles vector
Info	N.A. (Optional)	N.A.

Parameters and Dialog Box

Compressor Type:

Two compressor modes,

Mode 1: feed



Pressure ratio lookup table: RPM, rows [RPM]

The RPM axis of the pressure ratio lookup table. This vector will correspond to the rows in the pressure ratio lookup table.

Pressure ratio lookup table: mdot (in normal condition), columns [g/s]

The mdot axis of the pressure ratio lookup table. This vector will correspond to the columns in the pressure ratio lookup table.

Pressure ratio lookup table as a function of RPM and mdot [-]

The pressure ratio lookup table as a function of RPM and \dot{m} .

Efficiency lookup table as a function of RPM and \dot{m} [0-1]

The compressor's efficiency in the form of a lookup table as a function of RPM and \dot{m} .

Pressure ratio lookup table as a function of \dot{m} as described by the surge line [-]

Pressure ratio lookup table as a function of \dot{m} as described by the surge line.

Show pressure feedback ports

Show or hide pressure feedback ports.

Show info port

Show or hide info port.

Mode 2: circulation

Function Block Parameters: Compressor (Pressure Feedback)

Compressor (Pressure Feedback) (mask)

The compressor block raise the input flow's pressure by a certain amount, depending on the compressor's RPM (revolutions per minute) and the incoming mass flow (mdot). The relation between the pressure rise, RPM and mdot is described in a lookup table given in the block's mask. In addition to the outgoing flow, the compressor outputs its mechanical power consumption. this compressor is adiabatic and should be used only with gaseous flows.

This block models two types of compressors, feed and circulation compressor:
 In the feed mode the compressor mass flow rate is unchanged and the same as the input ass flow rate.
 In the circulation mode the mass flow rate is calculated depending on the pressure feedback from the flow circuit.

Parameters

Compressor Type: **circulation**

Pressure ratio lookup table: RPM, rows [RPM]
 [300 400 500]

Pressure ratio lookup table: mdot in normal condition($p=1\text{ bar}$, $T=298\text{ K}$, $R=0.287\text{ J/gr/K}$), columns [g/s]
 [0 180 360 500]

Pressure ratio lookup table as a function of RPM and mdot [-]:
 [1.45 1.4 1.3 1.2; 2.2 2 1.7 1.2; 12.7 12.6 12.5 12]

Efficiency lookup table as a function of RPM and mdot [0-1]:
 [0.6 0.7 0.6 0.5; 0.7 0.8 0.83 0.7; 0.75 0.85 0.88 0.74]

Pressure ratio lookup table as a function of mdot as described by the surge line [-]:
 [1.2 1.6 2 5]

Maximum mass flow [g/s]
 18000

Reference pressure [Pa]
 1.8e5

Relaxation factor []
 0.1

☐ Show info port

OK Cancel Help Apply

Pressure ratio lookup table: RPM, rows [RPM]

The RPM axis of the pressure ratio lookup table. This vector will correspond to the rows in the pressure ratio lookup table.

Pressure ratio lookup table: mdot (in normal condition), columns [g/s]

The \dot{m} axis of the pressure ratio lookup table. This vector will correspond to the columns in the pressure ratio lookup table.

Pressure ratio lookup table as a function of RPM and \dot{m} [-]

The pressure ratio lookup table as a function of RPM and \dot{m} .

Efficiency lookup table as a function of RPM and \dot{m} [0-1]

The compressor's efficiency in the form of a lookup table as a function of RPM and \dot{m} .

Pressure ratio lookup table as a function of \dot{m} as described by the surge line [-]

Pressure ratio lookup table as a function of \dot{m} as described by the surge line.

Maximum mass flow [g/s]

Maximum mass flow rate of the compressor.

Reference pressure [Pa]

Desired pressure at input

Relaxation factor [-]

A factor that is used to iterate pressure, default value is given in mask, the higher this value, the more aggressive the iteration behaves. This may lead to instable iterations. The lower the value the less aggressive the iteration behaves. This results in slower convergence.

Show info port

Show or hide info port.

Characteristics

Direct Feedthrough	Yes
--------------------	-----

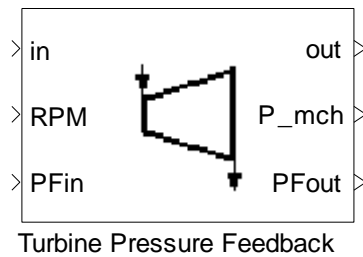
8.2.5.3 Turbine Pressure Feedback

A turbine to produce mechanical power

Library

Components / Hydraulic with Pressure Feedback

Description



The Turbine Pressure Feedback block decreases the flow pressure by a certain amount, depending on the turbine's RPM (revolutions per minute) and the incoming mass flow (\dot{m}). The relation between the pressure ratio, RPM and \dot{m} is described in a lookup table given in the block's mask. In addition to the outgoing flow, the turbine outputs the produced mechanical power that is calculated with help of an adjustable isentropic efficiency lookup table defined in the block's mask. Isentropic efficiency is defined as ratio of enthalpy difference for isentropic change of state to actual enthalpy difference with the same pressure drop.

$$\eta_s = \frac{\Delta \dot{h}}{\Delta \dot{h}_s}$$

This turbine is adiabatic and should be used with gaseous flows.

Data Type Support

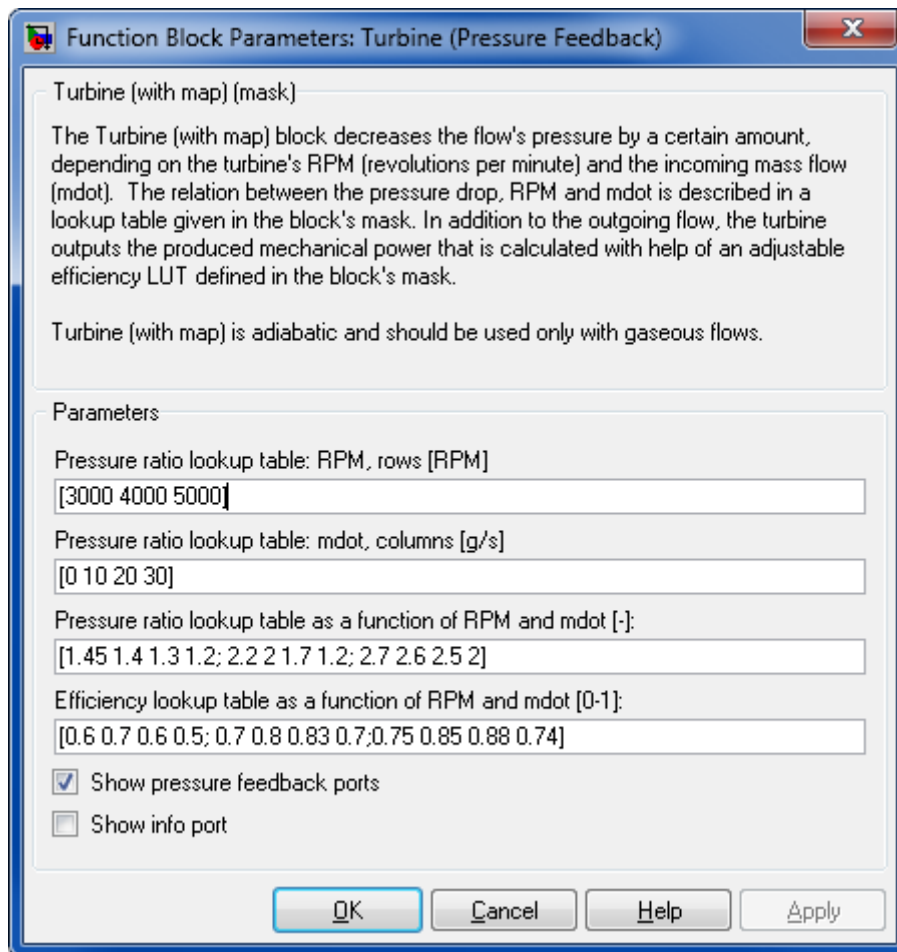
Inputs

in	Incoming flow	Flow bus
RPM	Rounds per minute	Scalar [RPM]
PFin	Pressure feedback in (Optional)	Scalar [Pa]

Outputs

out	Outgoing flow	Flow bus
P_mch	Mechanical power consumption	Scalar [W]
PFout	Pressure feedback out (Optional)	Scalar [Pa]
info	N.A. (Optional)	N.A.

Parameters and Dialog Box



Pressure ratio lookup table: RPM, rows [RPM]

The RPM axis of the pressure ratio lookup table. This vector will correspond to the rows in the pressure ratio lookup table.

Pressure ratio lookup table: mdot, columns [g/s]

The mdot axis of the pressure ratio lookup table. This vector will correspond to the columns in the pressure ratio lookup table.

Pressure ratio lookup table as a function of RPM and mdot [-]

The pressure ratio lookup table as a function of RPM and mdot.

Efficiency lookup table as a function of RPM and mdot [0-1]

The compressor's efficiency in the form of a lookup table as a function of RPM and mdot.

Show pressure feedback ports

Show or hide pressure feedback ports.

Show info port

Show or hide info port.

Characteristics

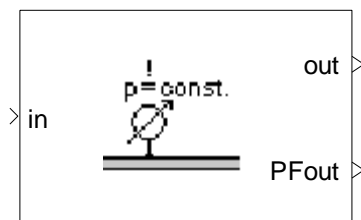
Direct Feedthrough	Yes
--------------------	-----

8.2.5.4 Downstream

Downstream pressure control.

Library

Components/Hydraulic with Pressure Feedback

Description

Downstream

The Downstream block is used in a hydraulic system to indicate what the pressure at the end of a flow should be. The block itself does not maintain the pressure but rather acts as a pressure reference point for the system. It simply passes the mass stream through as it is, and outputs pressure feedback information on what the pressure at this point in the flow should be (Pref), and what it is right now (Pis).

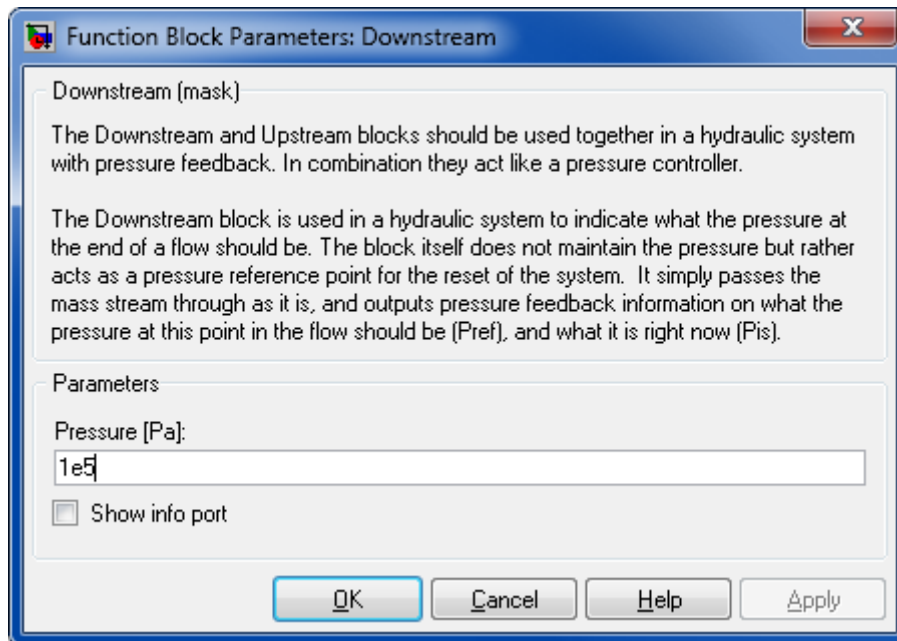
Data Type Support**Inputs**

in	Incoming flow	Flow bus
----	---------------	----------

Outputs

out	Outgoing flow	Flow bus
PFout	Pressure feedback for input	4-doubles vector
info	N.A. (Optional)	N.A.

Parameters and Dialog Box



Pressure

Desired pressure at position of this block.

Show info port

Show or hide info port.

Characteristics

Direct Feedthrough	Yes
--------------------	-----

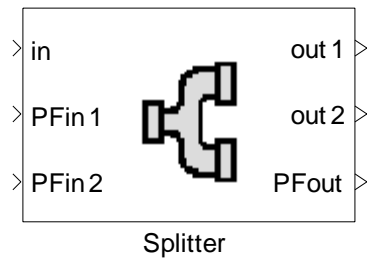
8.2.5.5 Splitter

Flow splitting

Library

Components/Hydraulics with Pressure Feedback

Description



The Splitter block divides an incoming flow into two, based on the two pressure feedbacks at the input. Each input (PFin) indicates what the pressure at a reference point is and what it should be. The splitter aims to divide the incoming flow in such a way that for each PFin, the difference between the desired pressure and the actual pressure approaches zero.

If the flows out of the splitter are mixed again, then the splitter aims to equalize the pressures of the two flows entering the mixer. On the other hand, if they are not re-mixed, then the splitter divides the flow such that the pressure specified at the downstream elements is reached.

If one of the output ways is closed initially, or initial flow is 0 at inlet, then the initial conditions are hold until the first time a flow ($\dot{n} > 0$) comes in or the outlet is open for at least one sample. If an outlet is closed downstream during simulation the output at this time will be hold until flow resumes.

Data Type Support

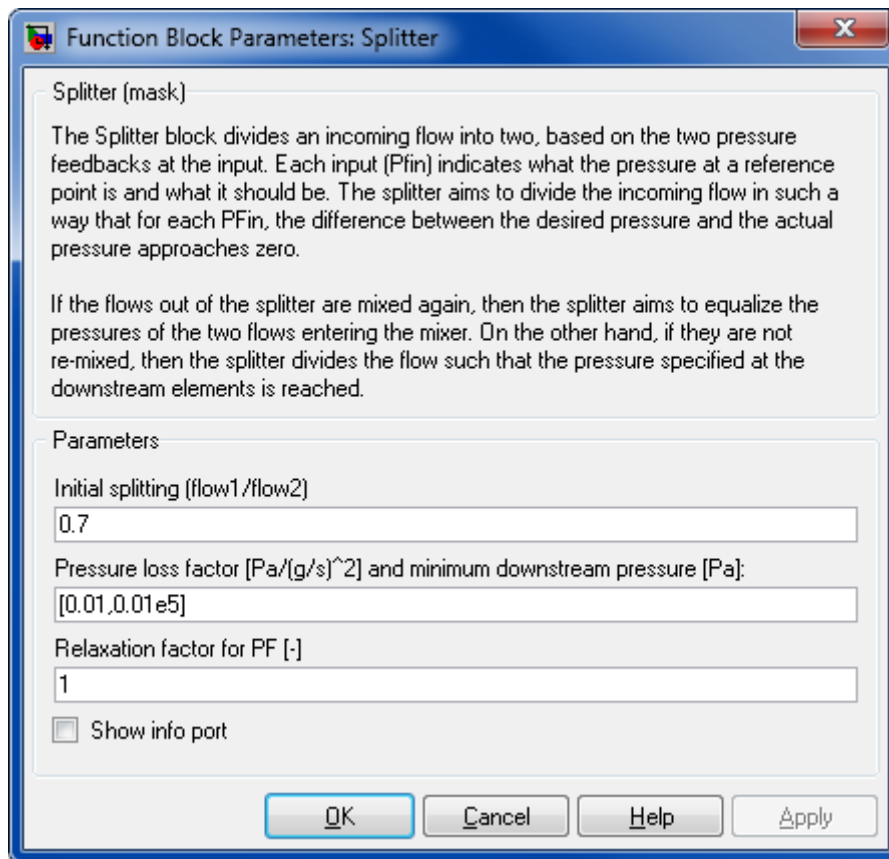
Inputs

in	Incoming flow to be split	Flow bus
PFin1	Pressure feedback for output	4-doubles vector
PFin2	Pressure feedback for rest	4-doubles vector

Outputs

out1	First part of outgoing flow	Flow bus
out2	Rest of outgoing flow	Flow bus
PFout	Pressure feedback for input	4-doubles vector
info	N.A. (Optional)	N.A.

Parameters and Dialog Box



Initial splitting

Initial splitting of the flow (0...1). 0 means all to output 1.

Pressure loss factor $[\text{Pa}/(\text{g/s})^2]$ and minimum downstream pressure $[\text{Pa}]$

The factor k and the pressure p are given as a vector $[k, p]$. The pressure loss is proportional to the mass flow rate squared $\Delta p = k\dot{m}^2$.

Relaxation factor

A factor that is used to iterate pressure. Default value is given in mask, the higher this value, the more aggressive the iteration behaves. This may lead to instable iterations. The lower the value the less aggressive the iteration behaves. This results in slower convergence.

Show info port

Show or hide info port.

Characteristics

Direct Feedthrough	Yes
--------------------	-----

8.2.6 Reactors

8.2.6.1 C8H18-Autothermal Reformer

The C8H18-Autothermal Reformer block is a link to the Equilibrium Reactor block.

8.2.6.2 CH4-Autothermal Reformer

The CH4-Autothermal Reformer block is a link to the Equilibrium Reactor block.

8.2.6.3 CH4-Burner

The CH4-Burner Reformer block is a link to the Equilibrium Reactor block.

8.2.6.4 CH4-Reformer

The CH4-Reformer block is a link to the Equilibrium Reactor block.

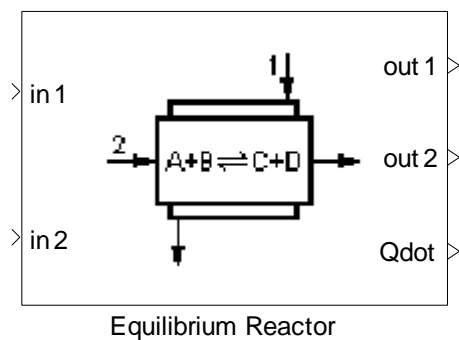
8.2.6.5 Equilibrium Reactor

Perform an equilibrium reaction.

Library

Components/Reactors

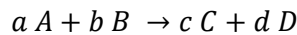
Description



The Equilibrium Reactor block computes the outgoing flow bus (FB) after chemical reactions, a heat exchange with a second fluid and the environment and a pressure loss. All properties from the outgoing flow are consistent. In particular, they are results of a coupled computation.

The chemical reactions are defined in the block's dialog. It is assumed that the reactions take place simultaneously. The reactions take place only in the second flow which is given at input port 2.

For a given reaction:



the chemical equilibrium condition can be written as

$$\ln(k) = -\frac{\Delta G}{RT}$$

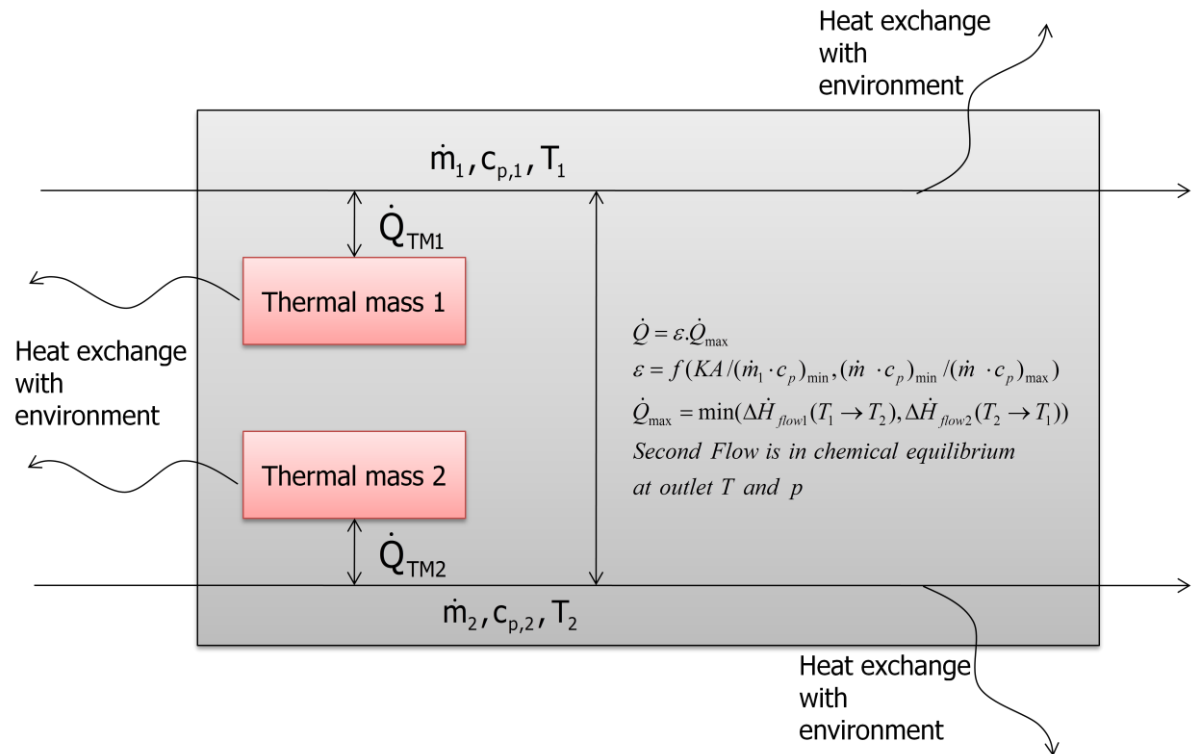
where:

$$k = \left(\frac{\psi_C^c \cdot \psi_D^d}{\psi_A^a \cdot \psi_B^b} \right) \cdot \left(\frac{p}{p_0} \right)^{c+d-a-b}$$

and ψ_j is the mole fraction of species j , ΔG is the difference of Gibbs free energy between products and reactants and R is the universal gas constant.

Via the "Add" button in the parameter dialog you can add an arbitrary number of reactions. Note however that with the number of reactions (and especially with the size of the stoichiometric coefficients a, b, c, d) the computational load increase dramatically. There might arise computational inaccuracies at the solution of the equation system for large equation systems. The solution of the equation system is done iteratively.

To approximate the dynamic thermal behavior of the block, the reactor is assumed to have a thermal mass (heat capacity), which is parameterized by the user in the mask. Each of the two flows entering the reactor exchanges heat with its own thermal mass, which is half of the total thermal mass defined in the mask. For more information, see the following figure and review the thermal mass block.



Three types of indirect heat exchange between the two flows are supported:

- counter flow,
- parallel flow, and
- cross flow.

For further details see Heat Exchanger.

The Equilibrium Reactor also models a pressure loss in both fluids.

The second outgoing flow is in chemical and thermodynamic equilibrium at outlet temperature and pressure.

Note: The reactants are input to the reactor block via the second input port, whereas the media used for heat exchange is input via the first input port. The products from the reaction(s) are output on the second output port.

Data Type Support

Inputs

in1	Incoming flow used for heat	Flow bus
-----	-----------------------------	----------

	exchange	
in2	Incoming flow of reactants	Flow bus
PFin1	Pressure feedback for output1 (Optional)	4-doubles vector
PFin2	Pressure feedback for output2 (Optional)	4-doubles vector

Outputs

out1	Outgoing flow used for heat exchange	Flow bus
out2	Outgoing flow of products	Flow bus
Qdot	Heat exchange with environment	Scalar [W]
PFout1	Pressure feedback for input1 (Optional)	4-doubles vector
PFout2	Pressure feedback for input2 (Optional)	4-doubles vector
info	N.A. (Optional)	N.A.

Parameters and Dialog Box

th_EquilibriumReactor

Equilibrium Reactor

The Equilibrium Reactor block computes the outgoing flow bus (FB) after chemical reactions, a heat exchange with a second fluid and the environment and a pressure loss. All properties from the outgoing flow are consistent. In particular, they are results of a coupled computation. It is assumed that the reactions take place simultaneously.

Heat exchange with a second media and the surrounding environment is taken into account. Three types of indirect heat exchange are supported: counter flow, parallel flow and cross flow. For further details see Heat Exchanger.

The Equilibrium Reactor also models a pressure loss in both fluids. The outgoing flow is in chemical and thermodynamical equilibrium. To achieve a more realistic model, the equilibrium reactor has a thermal mass that is defined as a parameter below.

Panel

Flow direction: Counter Flow

Reactions

2H2 + O2 <=> 2H2O

Add Edit Delete

Heat Exchange with Environment and Reactor Parameters

Coefficient of overall heat transfer rate [W/K]:
1e7

Initial temperature [K]:
300

Total surface [m^2]:
5

Environment temperature [K]:
300

Coefficient for calculating heat exchange with the environment [W/(m^2 K)]:
1

Pressure loss factor [Pa/(g/s)^2] and minimum downstream pressure [Pa] for first flow:
[.1,1e4]

Pressure loss factor [Pa/(g/s)^2] and minimum downstream pressure [Pa] for second flow:
[0.1,1e4]

Thermal mass of the reactor [JK]
2*4.180e3

Heat exchange rate for heat transfer between flow 1 and thermal mass 1 [W/K]
0

Heat exchange rate for heat transfer between flow 2 and thermal mass 2 [W/K]
0

☐ Show pressure feedback ports ☐ Show info port

OK Cancel Help

Flow direction

Selects the flow direction:

- Parallel flow,
- Counter flow, or
- Cross flow.

The selection of the flow type affects the output temperature as described in the reference below.

Reactions

List of chemical reactions that take place in the equilibrium reactor. Click on the add button to add a new reaction, the edit button to edit an existing one and the delete button to remove a reaction from the list. You can add arbitrary number of reactions.

Coefficient of overall heat transfer rate [W/K]

The coefficient of overall heat transfer rate characterizing the local heat per temperature difference between the flows.

Initial temperature

The initial temperature is used for the transfer function.

Total Surface [m²]

The external surface at which the heat exchange with the environment takes place. It is assumed that the temperature is constant at the surface.

Environment Temperature [K]

Environment temperature.

Coefficient of Environmental Heat Transfer [W/(m²K)]

The coefficient for calculating heat exchange with the environment.

Pressure loss factor [Pa/(g/s)²] and minimum downstream pressure [Pa] for first flow

The factor k and the pressure p are given as a vector $[k, p]$. The pressure loss is proportional to the squared mass flow rate $\Delta p = k\dot{m}^2$.

Pressure loss factor [Pa/(g/s)²] and minimum downstream pressure [Pa] for second flow

The factor k and the pressure p are given as a vector $[k, p]$. The pressure loss is proportional to the squared mass flow rate $\Delta p = k\dot{m}^2$.

Thermal mass of the equilibrium reactor [Joules/K]

The thermal mass (heat capacity) of the equilibrium reactor.

Coefficient of heat transfer rate from flow 1 to thermal mass [W/K]

Heat Exchange rate between first flow and first thermal mass.

Coefficient of heat transfer rate from flow 2 to thermal mass [W/K]

Heat Exchange rate between second flow and second thermal mass.

Show pressure feedback ports

Show or hide pressure feedback ports.

Show info port

Show or hide info port.

Characteristics

Direct Feedthrough	Yes
--------------------	-----

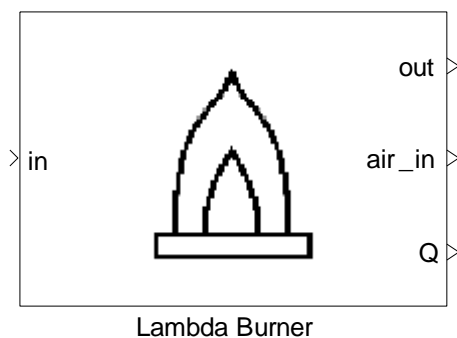
8.2.6.6 Lambda Burner

Performs a lambda-driven combustion.

Library

Components/Reactors

Description



The Lambda Burner block simulates a combustion reaction and computes the air flow needed for a given air balance. It takes into account pressure loss and heat exchange with the environment.

The fuel for the lambda burner is selected in the block's mask. Not all available species can be used as fuel: if an inappropriate fuel is selected then the block's dialog will return an error message. At this time, only one fuel can be selected at a time.

The fuel is fed to the burner via the input port. There is no need to explicitly supply the burner with air. It assumes the amount of needed air is available and outputs that amount at output port 'air_in'.

Air is assumed to consist of 21% O₂ and 79% N₂.

From the mask parameter Lambda, the block determines the amount of O₂ not required for the combustion. For instance, a value of 1 means that all O₂ is needed; a value of 1.04 means that 4% of the O₂ is not needed for the reaction.

If you want to use this block, make sure, that you select O₂ and N₂ in the Model Setup block as species to work with.

Data Type Support

Inputs

in	Incoming combustion fuel (H ₂ , CH ₄)	Flow bus
PFin	Pressure feedback for output (Optional)	4-doubles vector

Outputs

out	Exhaust gas flow	Flow bus
air_in	Estimated air flow required for combustion	
Q	Heat loss	Scalar [W]
PFout	Pressure feedback for input (Optional)	4-doubles vector
info	N.A. (Optional)	N.A.

Parameters and Dialog Box

th_LambdaBurner

Description

The Lambda Burner block simulates a combustion reaction and computes the air flow needed for a given air balance. It takes into account pressure loss and heat exchange with the environment.

The mask parameter Lambda determines the amount of O2 not required for the combustion. A value of 1 means, all O2 is needed; a value of 1.04 means, that there is 4% O2 not needed for the reaction, for instance.

Air is assumed to consist of 21% O2 and 79% N2.

Select Fuel

CO

CO + 0.5O2 <=> CO2

Add fuel

Remove fuel

Parameters

Lambda

123

Air temperature [K]:

300

Air pressure [Pa]:

123654

Pressure loss factor [Pa/(g/s)²] and minimum downstream pressure [Pa]:

[.001,1e4]

Total surface [m²]

1

Environment temperature [K]:

500

Coefficient for calculating heat exchange with the environment [W/(m² K)]:

0

☐ Show pressure feedback ports

☐ Show info port

OK Cancel Help

Select fuel

The fuel used by the Lambda Burner. A drop down list shows the available fuels for the Lambda Burner though not all species are appropriate fuels. Select a fuel and press Add fuel to include it. In this version, only one fuel at a time is supported.

Lambda

The excess air ratio.

Air temperature

Air temperature.

Air pressure

Air pressure.

Pressure loss factor [Pa/(g/s)²] and minimum downstream pressure [Pa]

The factor k and the pressure p are given as a vector $[k, p]$. The pressure loss is proportional to the mass flow rate squared $\Delta p = k\dot{m}^2$.

Total Surface [m²]

The external surface at which the heat exchange with the environment takes place. It is assumed that the temperature is constant at the surface.

Environment Temperature [K]

The temperature of the environment in which the Burner is situated.

Coefficient of Environmental Heat Transfer [W/(m²K)]

The coefficient for calculating heat exchange with the environment.

Show pressure feedback ports

Show or hide pressure feedback ports.

Show info port

Show or hide info port.

Characteristics

Direct Feedthrough	Yes
--------------------	-----

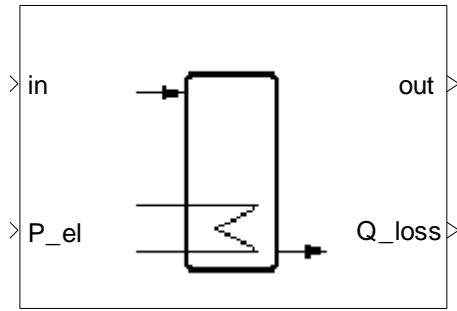
8.2.6.7 Preferential Oxidation Reactor

Perform a preferential oxidation reaction.

Library

Components/Reactors

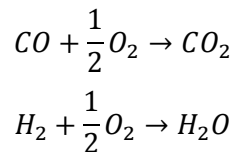
Description



Preferential Oxidation Reactor

The Preferential Oxidation Reactor block calculates the composition and temperature in a preferential oxidation reactor considering external heat flow and by using a lookup table for calculating the composition in the reactor.

Two chemical reactions take place:



The reaction is controlled by Selectivity (S), which is defined as the ratio of oxygen consumption for the CO oxidation over total oxygen consumption, and Efficiency (η), which is the conversion rate for the CO reaction.

Both variables depend on the temperature and are given as lookup tables.

The conversion rate for H_2 is given as $\eta^* (1-S)$, i.e., for $S=1$, no H_2 reaction takes place.

Usually PROX reactors are equipped with electrical heating to control the temperature of the reaction. The block implemented adds the "P_el" as heat to the input flow, before calculating the reactions.

Data Type Support

Inputs

in	Incoming flow	Flow bus
P_el	Electric power	Scalar [W]

PFin	Pressure feedback (Optional)	4-doubles vector
------	------------------------------	------------------

Outputs

out	Outgoing flow	Flow bus
Q_loss	Heat loss	Scalar [W]
PFout	Pressure feedback (Optional)	4-doubles vector
info	N.A. (Optional)	N.A.

Parameters and Dialog Box

Function Block Parameters: Preferential Oxidation Reactor

Preferential Oxidation Reactor (mask)

The Preferential Oxidation Reactor block calculates the composition and temperature in a preferential oxidation reactor according to a continuous stirred tank reactor model considering external heat flow and by using a look-up table for calculating the composition in the reactor.

Two chemical reactions take place:
 $\text{CO} + 1/2 \text{O}_2 \rightarrow \text{CO}_2$
 $\text{H}_2 + 1/2 \text{O}_2 \rightarrow \text{H}_2\text{O}$

The reaction is controlled by Selectivity (S), defined as the ratio of oxygen consumption for the CO oxidation over total oxygen consumption, and Efficiency (eta) being the conversion rate for the CO reaction. Both variables depend on the temperature and are given as lookup tables.

The conversion rate for H2 is given as $\text{eta} * (1-S)$, i.e., for $S=1$ no H2 reaction takes place.

Parameters

Total surface [m^2]:

Coefficient for calculating heat exchange with the environment [$\text{W}/\text{m}^2 \text{ /K}$]:

Environment temperature [K]:

Efficiency [-] (function of temperature):

Selectivity [-] (function of temperature):

Temperature lookup [K]:

Pressure loss factor [$\text{Pa}/(\text{g/s})^2$] and minimum downstream pressure [Pa]:

reaction matrix local

species local

☐ Show pressure feedback ports

☐ Show info port

Total surface [m^2]

The external surface at which the heat exchange with the environment takes place. It is assumed that the temperature is constant at the surface.

Coefficient of environmental heat transfer [W/(m²K)]

The coefficient for calculating heat exchange with the environment.

Environment temperature [K]

Environment temperature.

Efficiency[-] (function of temperature)

The vector of Efficiencies.

Selectivity[-] (function of temperature)

The vector used for Selectivity.

Temperature lookup table [K]

The vector of Temperatures for defining the lookup tables for efficiency and selectivity.

Pressure loss factor [Pa/(g/s)²] and minimum downstream pressure [Pa]

The factor k and the pressure p are given as a vector $[k, p]$. The pressure loss is proportional to the mass flow rate square $\Delta p = k\dot{m}^2$.

Reaction matrix local

Stoichiometric coefficients for the chemical reaction taking place (one row per reaction, one column per species). The order of coefficients corresponds to the order of the species given in "Species local" parameter.

Species local

1 by s cell array of species names. The order of this cell array gives the meaning of the reaction matrix local.

Show pressure feedback ports

Show or hide pressure feedback ports.

Show info port

Show or hide info port.

Characteristics

Direct Feedthrough	Yes
--------------------	-----

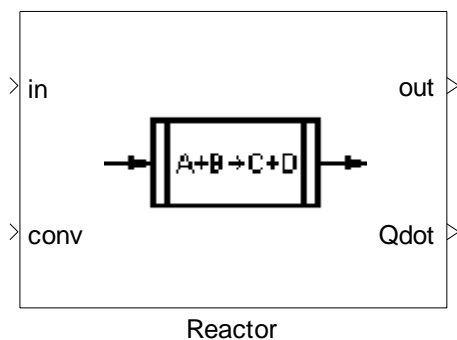
8.2.6.8 Reactor

Performs a chemical reaction

Library

Components/Reactors

Description



The Reactor block computes the outgoing flow bus (FB) after one reaction, a heat exchange with the environment and a pressure loss. All properties from the outgoing flow are consistent. In particular, they are results of a coupled computation. Heat exchange with the surrounding environment is taken into account. In general, the outgoing flow is not in chemical equilibrium as the Reactor performs a chemical reaction depending on a rate of reaction. This rate is given as an input parameter such that it can be modified during the simulation.

First law of thermodynamics for a chemical reaction:

$$\sum_R \dot{m}_{in} h_{in} = \sum_p \dot{m}_{out} h_{out} + \dot{Q}$$

The chemical reaction for the reactor is entered in the block's mask by clicking the Edit button. In this version, only one reaction can take place in the reactor block. For equilibrium reactions or heat exchange with a second fluid, use the Equilibrium Reactor.

Steps:

- Calculating Pressure loss
- Calculating conversion
- H-p-State with Heat Exchange (calculating all flow bus parameter and Qdot)

Note: The Reactor tries to realize the given rate of reaction. If no material is left, the reaction stops.

Data Type Support

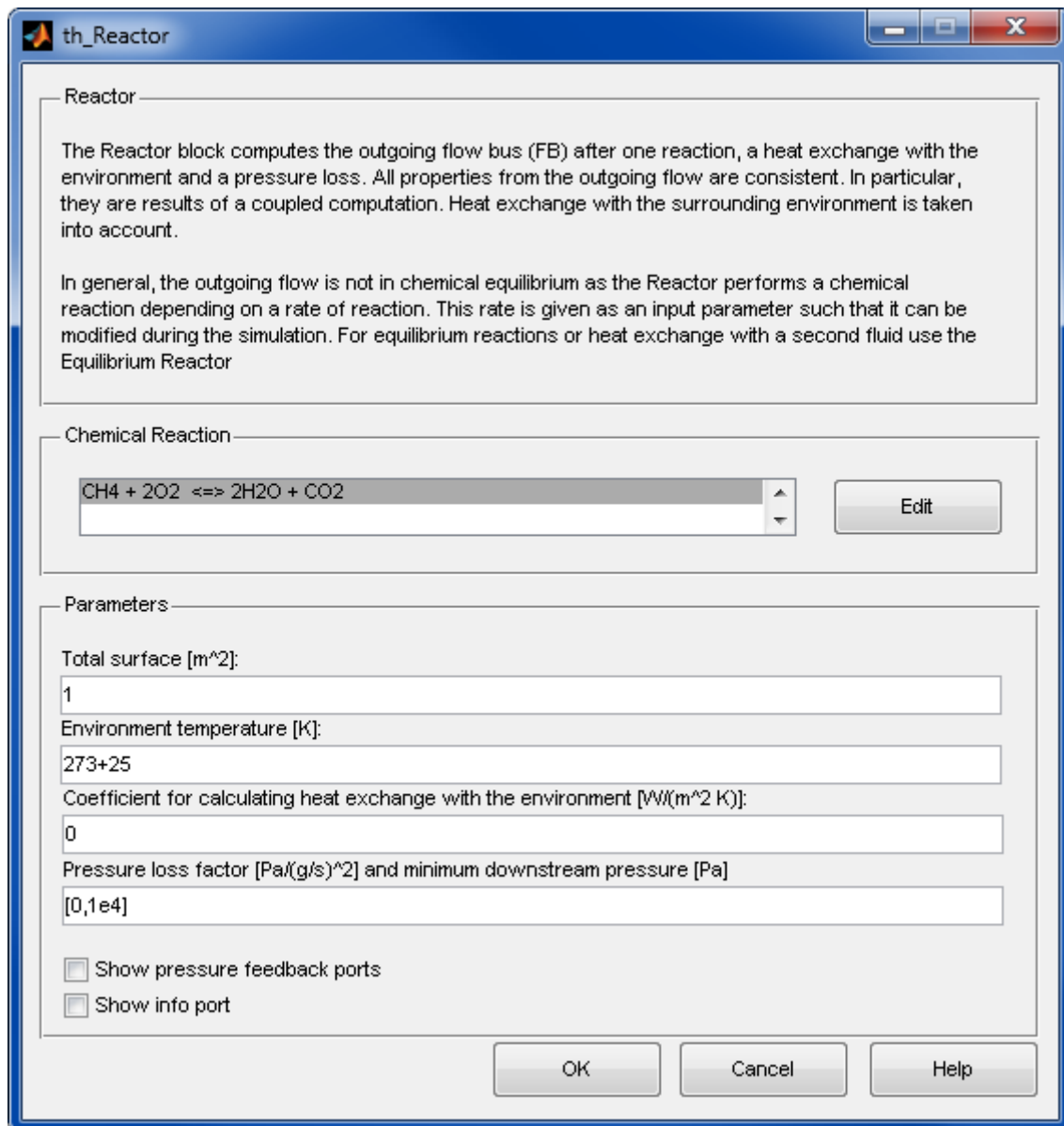
Inputs

in	Incoming flow of reactants	Flow bus
conv	Fraction to convert	Scalar [0-1]
PFin	Pressure feedback for output (Optional)	4-doubles vector

Outputs

out	Outgoing flow of products	Flow bus
Qdot	Heat loss	Scalar [W]
PFout	Pressure feedback for input (Optional)	4-doubles vector
info	N.A. (Optional)	N.A.

Parameters and Dialog Box



Chemical reaction

The chemical reaction that takes place in the reactor.

Total surface [m²]

The external surface at which the heat exchange with the environment takes place. It is assumed that the temperature is constant at the surface.

Environment temperature [K]

Environment temperature.

Coefficient of environmental heat transfer [W/(m² K)]

The coefficient for calculating heat exchange with the environment.

Pressure loss factor [Pa/(g/s)²] and minimum downstream pressure [Pa]

The factor k and the pressure p are given as a vector $[k, p]$. The pressure loss is proportional to the mass flow rate square $\Delta p = k\dot{m}^2$.

Show pressure feedback ports

Show or hide pressure feedback ports.

Show info port

Show or hide info port.

Characteristics

Direct Feedthrough	Yes
--------------------	-----

8.2.6.9 Shift Reactor

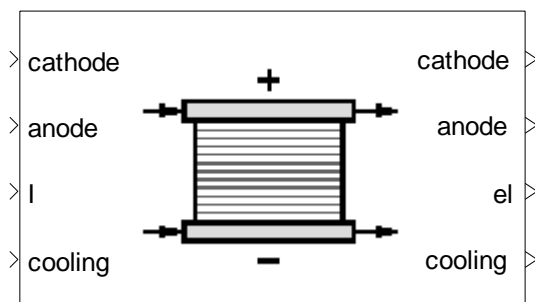
The Shift Reactor block is a link to the Equilibrium Reactor block.

8.2.7 Stacks**8.2.7.1 Current-Controlled Stack with Cooling**

Simulates a generic fuel cell stack controlled by an electric current.

Library

Components/Stacks

Description

Current -Controlled Stack with Cooling

The Current-Controlled Stack block simulates a fuel cell controlled by an electric current. The amount of power generated by the current-controlled stack is controlled by means of an electric current fed through the third input port 'I'. The stack receives a hydrogen flow at the anode, and a flow of oxygen or air at the cathode.

As expected, the higher the amount of hydrogen supplied, the more power the stack produces, assuming there is enough oxygen available for it to react. The amount of hydrogen converted in the stack is given as a parameter, "Conversion Rate". The conversion rate is a value between 0 and 1.

Increasing the conversion rate reduces the amount of hydrogen needed at the input anode at the expense of increasing the electric current needed.

The amount of hydrogen supplied to the stack at the anode, as well as the conversion rate of hydrogen specified in the stack's mask, determine the electric current needed.

$$I[A] = \frac{2 \times 96480 \times \text{ConversionRate} \times \text{HydrogenFlow}}{N_{\text{cells}} \times 22.4 \times 3600 \times 100}$$

The hydrogen flow is measured in normal liters per hour [l/hr] and N_{cells} refers to the number of cells in the stack.

In the current-controlled stack, the user has the option of adjusting the current the stack uses and hence it can fall below the required amount calculated above. In this case, the stack will then not produce the maximum power it theoretically can. If the user supplies more current than the stack needs, this will not increase the power the stack supplies.

The fuel cell's voltage, and consequently the stack's power, is determined using a lookup table supplied through the stack's mask. The user can choose the variables which the voltage-lookup is dependent on:

- Current density $f(I/A)$,
- Current density and pressure $f(I/A \text{ } p)$,
- Current density, pressure and temperature $f(I/A \text{ } p \text{ } T)$.

For the different options the expected dimensions are as follows:

Option	expected dimensions	Example
$U = f(I/A)$	cell voltage: n-by-1	KL_Sta_U1(:,1)
	current density: n-by-1	KL_Sta_J
$U = f(I/A, P)$	cell voltage: n-by-m	KL_Sta_U1
	current density: n-by-1	KL_Sta_J
	H ₂ partial pressures: 1-by-m	KL_Sta_p

$U = f(I/A, P, T)$	cell voltage: n-by-m-by-t	cat(3,KL_Sta_U1, KL_Sta_U2, KL_Sta_U3)
	current density: n-by-1	L_Sta_J
	H ₂ partial pressures: 1-by-m	KL_Sta_p
	temperatures: 1-by-t	[KL_Sta_T1, KL_Sta_T2, KL_Sta_T3]

To plot the polarization curves use commands like:

```
% plots for the temperature 1 for the first partial pressure of
hydrogen...
```

```
plot(KL_Sta_J, KL_Sta_U1(:,1))
```

```
% plots for all 3 partial pressures for temperature 1
```

```
plot(KL_Sta_J, KL_Sta_U1)
```

The demo data KL_Sta* are contained in the "ComponentData.mat" file and might be changed accordingly. The user may also load its own performance data from a different file. This can be changed in the Model Setup.

The stack is assumed to have a thermal mass that exchanges heat with the incoming flows. The thermal mass of the stack is defined in the block's mask. For more information, review the thermal mass block.

Cooling: the stack also receives an input flow that acts as a cooling medium for the stack. If the user does not wish to use a cooling medium, then the input port can be grounded by using a source block with the molar flow set to zero. Alternatively, the Current-Controlled Stack block can be used.

Data Type Support

Inputs

cathode	Incoming flow at the cathode	Flow bus
anode	Incoming flow at the anode	Flow bus
I	Controlling electric current	Scalar [A]

cooling	Incoming flow that cools the stack	Flow bus
PFinCath	Input pressure feedback for cathode (Optional)	4-doubles vector
PFinAn	Input pressure feedback for anode (Optional)	4-doubles vector

Outputs

cathode	Outgoing flow at the cathode	Flow bus
anode	Outgoing flow at the anode	Flow bus
el	Output power	Scalar [W]
cooling	The outgoing cooling flow	Flow bus
PFoutCath	Output pressure feedback for cathode (Optional)	4-doubles vector
PFoutAn	Output pressure feedback for anode (Optional)	4-doubles vector
info	N.A. (Optional)	N.A.

Parameters and Dialog Box

Function Block Parameters: Current-Controlled Stack with Cooling

Current-Controlled Stack with Cooling (mask)

The Current-Controlled Stack block simulates a fuel cell controlled by an electric current. The amount of power generated by the current-controlled stack is controlled by means of an electric current fed through the third input port. The stack receives a hydrogen flow at the anode, and a flow of oxygen or air at the cathode.

In the current-controlled stack, the user has the option of adjusting the current the stack uses and hence it can fall below the required amount. In this case, the stack will then not produce the maximum power it theoretically can. If the user supplies more current than the stack needs, this will not increase the power the stack supplies.

The fuel cell's voltage, and consequently the stack's power, is determined using a lookup table supplied through the stack's mask. The user can choose the variables upon which the voltage-lookup is dependent on: current density $f(I/A)$, current density and pressure $f(I/A, p)$, and current density, pressure and temperature $f(I/A, p, T)$.

Parameters

Number of cells [-]:
30

Maximum hydrogen conversion rate:
0.90

Heat transfer coefficient [W/K]:
350

Initial temperature of stack [K]:
323

Active membrane area (total) [m²]:
0.5

Thermal mass of stack [Joule/K]:
2*4.18e3

Choose dimension of lookup table: $f(I/A, P)$

Lookup table current density data [A/m²]:
[1:3]

Lookup table partial hydrogen pressure data [Pa]:
[1:3]

Lookup table cell voltage data [V]:
[1 1 1; 2 2 2; 3 3 3]

☐ Show pressure feedback ports

☐ Show info port

OK Cancel Help Apply

Number of cells

Number of cells in the stack.

Maximum hydrogen conversion rate

Maximum hydrogen conversion rate.

Heat transfer coefficient [W/K]

Coefficient for calculating heat transfer with the cooling medium.

Initial temperature of the stack [K]

The initial temperature of the stack.

Active membrane area [m²]

Active Membrane Area.

Thermal mass of the stack [Joule/K]

Thermal mass of the stack.

Dimension of lookup table

Choice of which data is the cell voltage dependent on. Three options are available: the cell voltage is dependent on the current density $f(I/A)$, the cell voltage is dependent on the current density and the partial hydrogen pressure $f(I/A \text{ } p)$, or the cell voltage is dependent on the current density, pressure and temperature $f(I/A \text{ } p \text{ } T)$.

Lookup table current density data [A/m²]

Characteristic current density data.

Lookup table partial hydrogen pressure data [Pa]

Characteristic partial hydrogen pressure data.

Lookup table temperature data [K]

Characteristic temperature data.

Lookup table cell voltage data [V]

Characteristic cell voltage data.

Show pressure feedback ports

Show or hide pressure feedback ports.

Show info port

Show or hide info port.

Characteristics

Direct Feedthrough	Yes
--------------------	-----

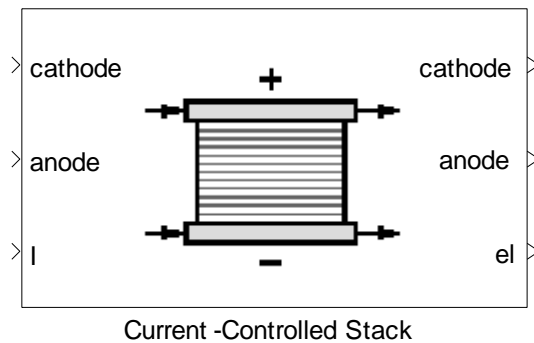
8.2.7.2 Current-Controlled Stack

Simulates a generic fuel cell stack controlled by an electric current.

Library

Components/Stacks

Description



The block is a "Current Controlled Stack with Cooling" where the cooling media ports are disabled. For details refer to help of block "Current Controlled Stack with Cooling".

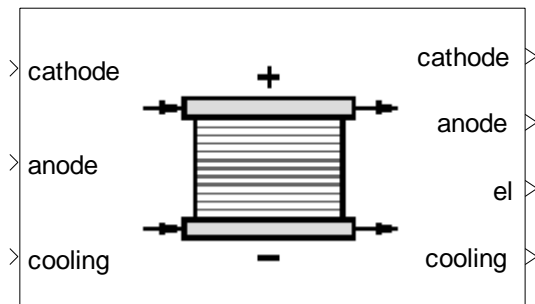
8.2.7.3 Fixed Conversion Rate Stack with Cooling

Simulates a generic fuel cell stack that converts hydrogen at a fixed rate

Library

Components/Stacks

Description



Fixed Conversion Rate Stack with Cooling

The Fixed Conversion Rate Stack block simulates a fuel cell stack that converts hydrogen at a fixed rate. It receives a hydrogen flow at the anode, and a flow of oxygen or air at the cathode.

As expected, the higher the amount of hydrogen supplied, the more power the stack produces, assuming there is enough oxygen available for it to react. The amount of hydrogen converted in the stack is given as a parameter 'Conversion Rate'. The conversion rate is a value between 0 and 1. Increasing the conversion rate reduces the amount of hydrogen needed at the input anode at the expense of increasing the electric current needed.

The amount of hydrogen supplied to the stack at the anode, as well as the conversion rate of hydrogen specified in the stack's mask, determine the electric current needed.

$$I[A] = \frac{2 \times 96480 \times \text{ConversionRate} \times \text{HydrogenFlow}}{N_{\text{cells}} \times 22.4 \times 3600 \times 100}$$

The hydrogen flow is measured in normal liters per hour [l/hr] and N_{cells} refers to the number of cells in the stack.

In the fixed conversion-rate stack, the electric current used is fixed and will always be equal to the value calculated above. If the user wishes to adjust the current which the stack uses, please consider to use the "Current-controlled Stack" block.

The fuel cell's voltage, and consequently the stack's power, is determined using a lookup table supplied through the stack's mask. The user can choose the variables which the voltage-lookup is dependent on:

- Current density $f(I/A)$,
- Current density and pressure $f(I/A \text{ p})$,
- Current density, pressure and temperature $f(I/A \text{ p T})$.

For the different options the expected dimensions are as follows:

Option	expected dimensions	Example
--------	---------------------	---------

$U = f(I/A)$	cell voltage: n-by-1	KL_Sta_U1(:,1)
	current density: n-by-1	KL_Sta_J
$U = f(I/A, P)$	cell voltage: n-by-m	KL_Sta_U1
	current density: n-by-1	KL_Sta_J
	H ₂ partial pressures: 1-by-m	KL_Sta_p
$U = f(I/A, P, T)$	cell voltage: n-by-m-by-t	cat(3, KL_Sta_U1, KL_Sta_U2, KL_Sta_U3)
	current density: n-by-1	L_Sta_J
	H ₂ partial pressures: 1-by-m	KL_Sta_p
	temperatures: 1-by-t	[KL_Sta_T1, KL_Sta_T2, KL_Sta_T3]

To plot the polarization curves use commands like:

```
% plots for the temperature 1 for the first partial pressure of
hydrogen...
```

```
plot(KL_Sta_J, KL_Sta_U1(:,1))
```

```
% plots for all 3 partial pressures for temperature 1
```

```
plot(KL_Sta_J, KL_Sta_U1)
```

The demo data `KL_Sta*` are contained in the "ComponentData.mat" file and might be changed accordingly. The user may also load its own performance data from a different file. This can be changed in the Model Setup.

The stack is assumed to have a thermal mass that exchanges heat with the incoming flows. The thermal mass of the stack is defined in the block's mask. For more information, please review to the thermal mass block.

Cooling: the stack also receives a input flow that acts as a cooling medium for the stack. If the user does not wish to use a cooling medium, then the input port can be grounded by using a source block with the molar flow set to zero. Alternatively, the Fixed Conversion Rate Stack block can be used.

Data Type Support

Inputs

cathode	Incoming flow at the cathode	Flow bus
anode	Incoming flow at the anode	Flow bus
cooling	Incoming flow that cools the stack	Flow bus
PFinCath	Input pressure feedback for cathode (Optional)	4-doubles vector
PFinAn	Input pressure feedback for anode (Optional)	4-doubles vector

Outputs

cathode	Outgoing flow at the cathode	Flow bus
anode	Outgoing flow at the anode	Flow bus
el	Output power	Scalar [W]
cooling	The outgoing cooling flow	Flow bus
PFoutCath	Output pressure feedback for cathode (Optional)	4-doubles vector
PFoutAn	Output pressure feedback for anode (Optional)	4-doubles vector
info	N.A. (Optional)	N.A.

Parameters and Dialog Box

Function Block Parameters: Fixed Conversion Rate Stack with Cooli...

Fixed Conversion Rate Stack with Cooling (mask) (link)

The Fixed Conversion Rate Stack block simulates a fuel cell stack that converts hydrogen at a fixed rate. It receives a hydrogen flow at the anode, and a flow of oxygen or air at the cathode.

As expected, the higher the amount of hydrogen supplied, the more power the stack produces, assuming there is enough oxygen available for it to react. The amount of hydrogen converted in the stack is given as a parameter 'Conversion Rate'. The conversion rate is a value between 0 and 1. Increasing the conversion rate reduces the amount of hydrogen needed at the input anode at the expense of increasing the electric current needed.

The fuel cell's voltage, and consequently the stack's power, is determined using a lookup table supplied through the stack's mask. The user can choose the variables upon which the voltage-lookup is dependent on: current density $f(I/A)$, current density and pressure $f(I/A, p)$, and current density, pressure and temperature $f(I/A, p, T)$.

Parameters

Number of cells [-]:
30

Maximum hydrogen conversion rate:
0.8

Heat transfer coefficient [W/K]:
350

Initial temperature of stack [K]:
323

Active membrane area (total) [m^2]:
0.5

Thermal mass of stack [Joule/K]:
2*4.18e3

Choose dimension of lookup table: $f(I/A, P)$

Lookup table current density data [A/m^2]:
[1:3]

Lookup table partial hydrogen pressure data [Pa]:
[1:3]

Lookup table cell voltage data [V]:
[1 1 1; 2 2 2; 3 3 3]

☐ Show pressure feedback ports

☐ Show info port

OK Cancel Help Apply

Number of cells

Number of cells in the stack.

Maximum hydrogen conversion rate

The maximum hydrogen conversion rate.

Heat transfer coefficient [W/K]

Coefficient for calculating heat transfer with the cooling medium.

Initial temperature of the stack [K]

The initial temperature of the stack.

Active membrane area [m²]

The total active Membrane Area.

Thermal mass of the stack [Joule/K]

Thermal mass of the stack.

Dimension of lookup table

Choice of which data is the cell voltage dependent on. Three options are available: the cell voltage is dependent on the current density $f(I/A)$, the cell voltage is dependent on the current density and the partial hydrogen pressure $f(I/A \text{ } p)$, or the cell voltage is dependent on the current density, pressure and temperature $f(I/A \text{ } p \text{ } T)$.

Lookup table current density data [A/m²]

Characteristic current density data.

Lookup table partial hydrogen pressure data [Pa]

Characteristic partial hydrogen pressure data.

Lookup table temperature data [K]

Characteristic temperature data.

Lookup table cell voltage data [V]

Characteristic cell voltage data.

Show pressure feedback ports

Show or hide pressure feedback ports.

Show info port

Show or hide info port.

Characteristics

Direct Feedthrough	Yes
--------------------	-----

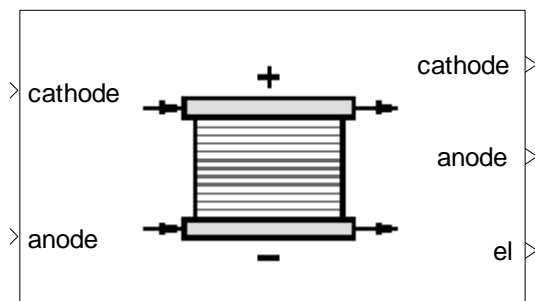
8.2.7.4 Fixed Conversion Rate Stack

Simulates a generic fuel cell stack that converts hydrogen at a fixed rate

Library

Components/Stacks

Description



Fixed Conversion Rate Stack

The block is a "Fixed Conversion Stack with Cooling" where the cooling media ports are disabled. For details refer to help of block "Fixed Conversion Stack with Cooling".

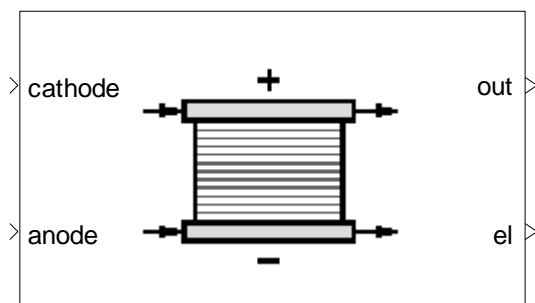
8.2.7.5 SOFC Stack

Simulates a solid oxide fuel cell stack.

Library

Components/Stacks

Description



SOFC Stack

The SOFC Stack block simulates a solid oxide fuel cell stack. Hydrogen is produced via defined reactions (chemical equilibrium is assumed). The block outputs a heat flow and electrical power. It receives a fuel flow at the anode, and a flow of oxygen or air at the cathode. This block is a combination of a chemical reactor and a stack.

At the cathode, the oxygen molecules from the air are split into oxygen ions with the addition of four electrons. The oxygen ions are conducted through the electrolyte and combine with hydrogen at the anode, releasing four electrons. The electrons travel an external circuit providing electric power and producing by-product heat.

Anode Reaction: $2 \text{H}_2 + 2 \text{O}^{2-} \Rightarrow 2 \text{H}_2\text{O} + 4 \text{e}^-$

Cathode Reaction: $\text{O}_2 + 4 \text{e}^- \Rightarrow 2 \text{O}^{2-}$

Overall Cell Reaction: $2 \text{H}_2 + \text{O}_2 \Rightarrow 2 \text{H}_2\text{O}$

The chemical equilibrium of the reformation reactions (as defined in mask) as well as the heat production of the hydrogen conversion is solved simultaneously.

The main purpose of this block is to calculate a stack performance when an internal reformation takes place. Note, that the chemical reactions defined in the mask should not contain the reaction $2\text{H}_2 + \text{O}_2 \Rightarrow 2\text{H}_2\text{O}$. This reaction is added intern automatically.

Produced electrical power depends on the hydrogen production and the given polarization curve given in parameter mask. The stack is assumed to be adiabatic. In this stack, the electric current used is fixed and will always be equal to the value calculated.

The fuel cell's voltage, and consequently the stack's power, is determined using a lookup table supplied through the stack's mask. The user can choose the variables which the voltage-lookup is dependent on:

- Current density $f(I/A)$,
- Current density and pressure $f(I/A \text{ } p)$,
- Current density, pressure and temperature $f(I/A \text{ } p \text{ } T)$.

For the different options the expected dimensions are as follows:

Option	expected dimensions	Example
$U = f(I/A)$	cell voltage: n-by-1	KL_Sta_U1(:,1)
	current density: n-by-1	KL_Sta_J
$U = f(I/A, P)$	cell voltage: n-by-m	KL_Sta_U1
	current density: n-by-1	KL_Sta_J

	H ₂ partial pressures: 1-by-m	KL_Sta_p
U = f(I/A, P, T)	cell voltage: n-by-m-by-t	cat(3, KL_Sta_U1, KL_Sta_U2, KL_Sta_U3)
	current density: n-by-1	L_Sta_J
	H ₂ partial pressures: 1-by-m	KL_Sta_p
	temperatures: 1-by-t	[KL_Sta_T1, KL_Sta_T2, KL_Sta_T3]

To plot the polarization curves use commands like:

```
% plots for the temperature 1 for the first partial pressure of
hydrogen...
```

```
plot(KL_Sta_J, KL_Sta_U1(:,1))
```

```
% plots for all 3 partial pressures for temperature 1
```

```
plot(KL_Sta_J, KL_Sta_U1)
```

The demo data KL_Sta* are contained in the "ComponentData.mat" file and might be changed accordingly. The user may also load its own performance data from a different file. This can be changed in the Model Setup.

Data Type Support

Inputs

cathode	Incoming flow at the cathode	Flow bus
anode	Incoming flow at the anode	Flow bus
PFinCath	Input pressure feedback for cathode (Optional)	4-doubles vector
PFinAn	Input pressure feedback for anode (Optional)	4-doubles vector

Outputs

out	The stack's outgoing flow	Flow bus
el	Output power	Scalar [W]
PFoutCath	Output pressure feedback for cathode (Optional)	4-doubles vector
PFoutAn	Output pressure feedback for anode (Optional)	4-doubles vector
info	N.A. (Optional)	N.A.

Parameters and Dialog Box

SOFC Stack

The SOFC Stack block simulates a solid oxide fuel cell stack. Hydrogen is produced via defined reactions (chemical equilibrium is assumed). The block outputs produced heat and electrical power. It receives a fuel flow at the anode, and a flow of oxygen or air at the cathode.

In this stack, the electric current used is fixed and will always be equal to the value calculated. The fuel cell's voltage, and consequently the stack's power, is determined using a lookup table supplied through the stack's mask. The user can choose the variables upon which the voltage-lookup is dependent on: current density $f(I/A)$, current density and pressure $f(I/A, p)$, or current density, pressure and temperature $f(I/A, p, T)$.

Reactions

CH4 + H2O <=> CO + 3H2

H2O + CO <=> CO2 + H2

Add

Edit

Delete

Stack Parameters

Number of cells [-]

100

Maximum hydrogen conversion rate:

0.9

Initial temperature of stack [K]:

323

Active cell area of one cell [m^2]:

0.2*0.2

Polarization Curves

Choose cell voltage dependency

f(I/A)

Cell voltage lookup table [V]:

KL_Sta_U1(:,1)

Current density data [A/m^2]:

KL_Sta_J

Partial hydrogen pressure data [Pa]:

KL_Sta_p

Temperature data [K]:

[KL_Sta_T1 KL_Sta_T2 KL_Sta_T3]

☐ Show pressure feedback ports
 ☐ Show info port

OK

Cancel

Help

Reactions

List of chemical reactions take place in the stack. Use the Add, Edit and Delete buttons to adjust the chemical reactions. Note that the reaction $2\text{H}_2 + \text{O}_2 \Rightarrow 2\text{H}_2\text{O}$ is handled internally and must not be added to this list.

Number of cells [-]

Number of cells in the stack.

Maximum hydrogen conversion rate

The maximum hydrogen conversion rate.

Initial temperature of the stack [K]

The initial temperature of the stack.

Active membrane area [m²]

The total active Membrane Area.

Choose cell voltage dependency

Choice of which data is the cell voltage dependent on. Three options are available: the cell voltage is dependent on the current density $f(I/A)$, the cell voltage is dependent on the current density and the partial hydrogen pressure $f(I/A, p)$, or the cell voltage is dependent on the current density, pressure and temperature $f(I/A, p, T)$.

Cell voltage lookup table [V]

Characteristic cell voltage data.

Current density data [A/m²]

Characteristic current density data.

Partial hydrogen pressure data [Pa]

Characteristic partial hydrogen pressure data.

Temperature data [K]

Characteristic temperature data.

Show pressure feedback ports

Show or hide pressure feedback ports.

Show info port

Show or hide info port.

Characteristics

Direct Feedthrough	Yes
--------------------	-----

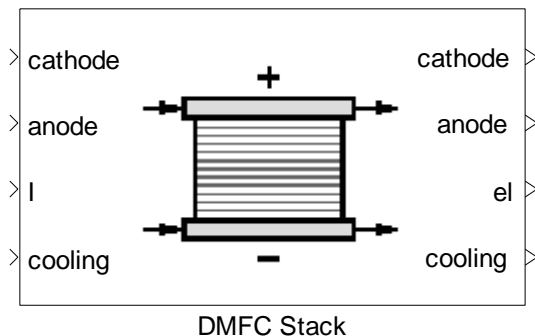
8.2.7.6 DMFC Stack

Simulates a direct methanol fuel cell stack.

Library

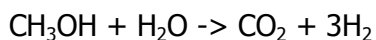
Components/Stacks

Description



This block simulates a direct methanol fuel cell stack controlled by an electric current. The stack receives a flow of oxygen or air at the cathode, and a water-methanol mixture at the anode.

The methanol conversion rate is dependent on the temperature. This is calculated via a lookup table defined in the mask. The chemical reaction for methanol conversion is:



The fuel cell's voltage, and consequently the stack's power, is determined using a lookup table supplied through the stack's mask. The user can choose the variables which the voltage-lookup is dependent on:

- Current density $f(I/A)$,
- Current density and pressure $f(I/A \text{ } p)$,
- Current density, pressure and temperature $f(I/A \text{ } p \text{ } T)$.

The stack is assumed to be adiabatic and have a thermal mass that exchanges heat with the incoming flows. The thermal mass of the stack is defined in the block's mask.

Cooling: the stack also receives an input flow that acts as a cooling medium for the stack. If the user does not wish to use a cooling medium, then the input port can be grounded by using a source block with the molar flow set to zero.

Data Type Support

Inputs

cathode	Incoming flow at the cathode	Flow bus
anode	Incoming flow at the anode	Flow bus
I	Controlling electric current	Scalar [A]
cooling	Incoming flow that cools the stack	Flow bus
PFinCath	Input pressure feedback for cathode (Optional)	4-doubles vector
PFinAn	Input pressure feedback for anode (Optional)	4-doubles vector

Outputs

cathode	Outgoing flow at the cathode	Flow bus
anode	Outgoing flow at the anode	Flow bus
el	Output power	Scalar [W]
cooling	The outgoing cooling flow	Flow bus
PFoutCath	Output pressure feedback for cathode (Optional)	4-doubles vector
PFoutAn	Output pressure feedback for anode (Optional)	4-doubles vector
info	N.A. (Optional)	N.A.

Parameters and Dialog Box

Function Block Parameters: DMFC Stack

DMFC Stack (mask) (link)

This block simulates a direct methanol fuel cell stack controlled by an electric current. The stack receives a flow of oxygen or air at the cathode, and a methanol mixture at the anode.

The methanol conversion rate is dependent on the temperature. This is calculated via a lookup table defined in the mask. The chemical reaction for methanol conversion is:

$$\text{CH}_3\text{OH} + \text{H}_2\text{O} \rightarrow \text{CO}_2 + 3\text{H}_2$$

The fuel cell's voltage, and consequently the stack's power, is determined using a lookup table supplied through the stack's mask. The user can choose the variables upon which the voltage-lookup is dependent on: current density $f(I/A)$, current density and pressure $f(I/A, p)$, and current density, pressure and temperature $f(I/A, p, T)$.

Parameters

Number of cells [-]:
30

Maximum hydrogen conversion rate:
0.8

Heat transfer coefficient for heat exchange with cooling media [W/K]:
350

Initial temperature of stack [K]:
323

Active cell area of one cell [m^2]:
0.05*0.05

Methanol conversion rate lookup table [T,rate]:
[50+273 120+273 ;0.5 0.8]

Thermal mass of the stack [J/K]:
2*4.18e3

Choose dimension of lookup table: $f(I/A)$

Lookup table current density data [A/m^2]:
KL_Sta_J

Lookup table cell voltage data [V]:
KL_Sta_U1(:,1)

☐ Show pressure feedback ports

☐ Show info port

OK Cancel Help Apply

Number of cells

Number of cells in the stack.

Maximum hydrogen conversion rate

The maximum hydrogen conversion rate.

Heat transfer coefficient for heat exchange with cooling media [W/K]

Heat transfer coefficient for calculating the heat transfer with the cooling medium.

Initial temperature of the stack [K]

The initial temperature of the stack.

Active membrane area [m²]

The total active Membrane Area.

Methanol conversion rate lookup table [T; rate]

A lookup table to determine the methanol conversion rate depending on the stack's temperature. The temperature is the first row in the lookup table and the conversion rate is the second row.

Thermal mass of the stack [J/K] Thermal mass of the stack.**Choose dimension of lookup table**

Choice of which data is the cell voltage dependent on. Three options are available: the cell voltage is dependent on the current density $f(I/A)$, the cell voltage is dependent on the current density and the partial hydrogen pressure $f(I/A \text{ } p)$, or the cell voltage is dependent on the current density, pressure and temperature $f(I/A \text{ } p \text{ } T)$.

Lookup table current density data [A/m²]

Characteristic current density data.

Lookup table partial hydrogen pressure data [Pa]

Characteristic partial hydrogen pressure data.

Lookup table temperature data [K]

Characteristic temperature data.

Lookup table cell voltage data [V]

Characteristic cell voltage data.

Show pressure feedback ports

Show or hide pressure feedback ports.

Show info port

Show or hide info port.

Characteristics

Direct Feedthrough	Yes
--------------------	-----

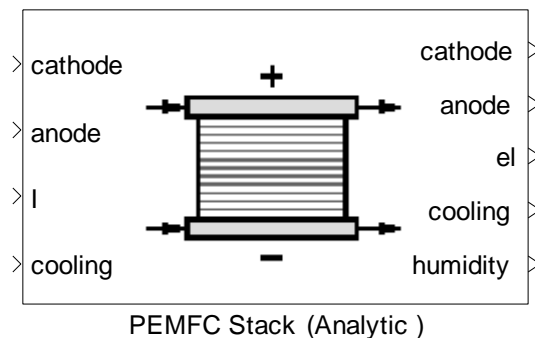
8.2.7.7 PEMFC Stack Analytic

Simulates a PEM fuel cell stack

Library

Components/Stacks

Description



The PEMFC Stack is a stack, where the polarization curve is determined using fixed relationships, which are specialized to simulate a low temperature PEM behavior. The relations are taken from [4].

The stack is assumed to have a thermal mass at uniform temperature, which exchanges heat with the flows going through the stack. The thermal mass of the stack is defined in the block's mask. The outgoing flows at cathode and anode side have the same temperature, which is the "stack temperature".

There is no heat loss to environment and stack temperature is calculated using energy balance as follows:

$$\left(m \frac{dU}{dt}\right)_{stack} = \sum \dot{H}_{in} - \sum \dot{H}_{out} - P_{el}$$

The humidity of the membrane is calculated using the mean of the relative humidity of anode and cathode inlets/outlets. The relation between this mean relative humidity and the

internally used “membrane water content” is given in a lookup table. The first row is the relative humidity, the second row the membrane water content (0=dry, 14 wet).

Cooling: the stack also receives an input flow that acts as a cooling medium for the stack. If the user does not wish to use a cooling medium, then the input port can be grounded by using a source block with the molar flow set to zero.

Data Type Support

Inputs

cathode	Incoming flow at the cathode	Flow bus
anode	Incoming flow at the anode	Flow bus
I	Controlling electric current	Scalar [A]
cooling	Incoming flow that cools the stack	Flow bus
PFinCath	Input pressure feedback for cathode (Optional)	4-doubles vector
PFinAn	Input pressure feedback for anode (Optional)	4-doubles vector

Outputs

cathode	Outgoing flow at the cathode	Flow bus
anode	Outgoing flow at the anode	Flow bus
el	Output power	Scalar [W]
cooling	The outgoing cooling flow	Flow bus
humidity	Membrane humidity	[0..14]
PFoutCath	Output pressure feedback for cathode (Optional)	4-doubles vector
PFoutAn	Output pressure feedback for anode (Optional)	4-doubles vector
info	N.A. (Optional)	N.A.

Parameters and Dialog Box

Function Block Parameters: PEMFC Stack Analytic

PEMFC Stack Analytic (mask)

The PEMFC Stack is a stack, where the polarization curve is determined using fixed relationships, which are specialized to simulate a low temperature PEM behavior. The relations are taken from literature (see manual for reference).

The stack is assumed to have a thermal mass that exchanges heat with the flows going through the stack. The thermal mass of the stack is defined in the block's mask. The outgoing flows at cathode and anode side have the same temperature, which is the stack temperature. The humidity of the membrane is calculated using the mean of the relative humidity of anode and cathode inlets/outlets. The relation between this mean relative humidity and the internally used membrane water content is given in a lookup table. The first row is the relative humidity, the second row the membrane water content (0=dry, 14 wet).

Cooling: the stack also receives a input flow that acts as a cooling medium for the stack. Heat exchange with the cooling medium can be parametrized.

Parameters

Number of cells [-]:

Initial temperature of stack [K]:

Active area of one cell [m²]:

Lookup table for voltage reduction efficiency caused by high utilization in Anode [0-1; 0-1]

Lookup table for voltage reduction efficiency caused by high utilization in Cathode [0-1; 0-1]

Membrane thickness [cm]:

Membrane water content as a function of relative humidity [-]:

Thermal mass of stack [J/K]:

Coefficient of heat transfer to cooling media [W/K]:

☐ Show pressure feedback ports

☐ Show info port

Number of cells

Number of cells in the stack.

Initial temperature of the stack [K]

The initial temperature of the stack.

Active area of one cell [m²]

The active membrane area of one cell.

Lookup table for voltage reduction efficiency caused by high utilization in Anode [0-1; 0-1]

The first row of this 2 x n matrix contains the hydrogen utilizations. The second row contains the corresponding factors, by which the voltage should be reduced, when the utilization of the first row is reached.

Lookup table for voltage reduction efficiency caused by high utilization in Cathode [0-1; 0-1]

The first row of this 2 x n matrix contains the oxygen utilizations. The second row contains the corresponding factors, by which the voltage should be reduced, when the utilization of the first row is reached.

Membrane thickness [cm]

Thickness of the membrane is used in the model for the polarization curves.

Membrane water content as a function of relative humidity [-]

This parameter is a 2 x n matrix. The first row contains a list of mean relative humidity values, while the second row contains the corresponding values of membrane water content.

Thermal mass of stack [J/K]

This is the thermal mass of the stack.

Coefficient of heat transfer to cooling media [W/K]

Overall heat transfer rate between the stack thermal mass and the cooling media.

Show pressure feedback ports

Show or hide pressure feedback ports.

Show info port

Show or hide info port.

Characteristics

Direct Feedthrough	Yes
--------------------	-----

8.3 CONTROLLERS

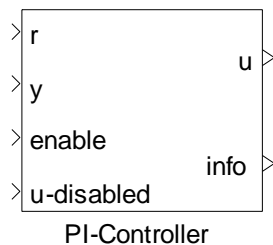
8.3.1 PI-Controller

PI Controller with Anti-Windup

Library

Controllers

Description



The PI Controller block calculates a new actuator value (u) depending on the measured value (y) and a reference set point value (r).

This is done by using the difference between measured and reference value (the error) to adjust the actuator value such that the measured value approaches the set point.

In a PI controller, the correction is realized via proportional gain (K_r) and integration of the error signal. The proportional gain handles the current error ($e = r - y$), whereas the integration part attempts to use past information to correct the signal.

The gain K_r and the integral time T_n are given by the user in the block's mask. The initial value of the output signal, as well as its minimum and maximum values, is also given in the mask.

The controller can be disabled by setting the "enable" port to zero. In this case, the output value is that of the fourth input port "u-disabled".

Data Type Support

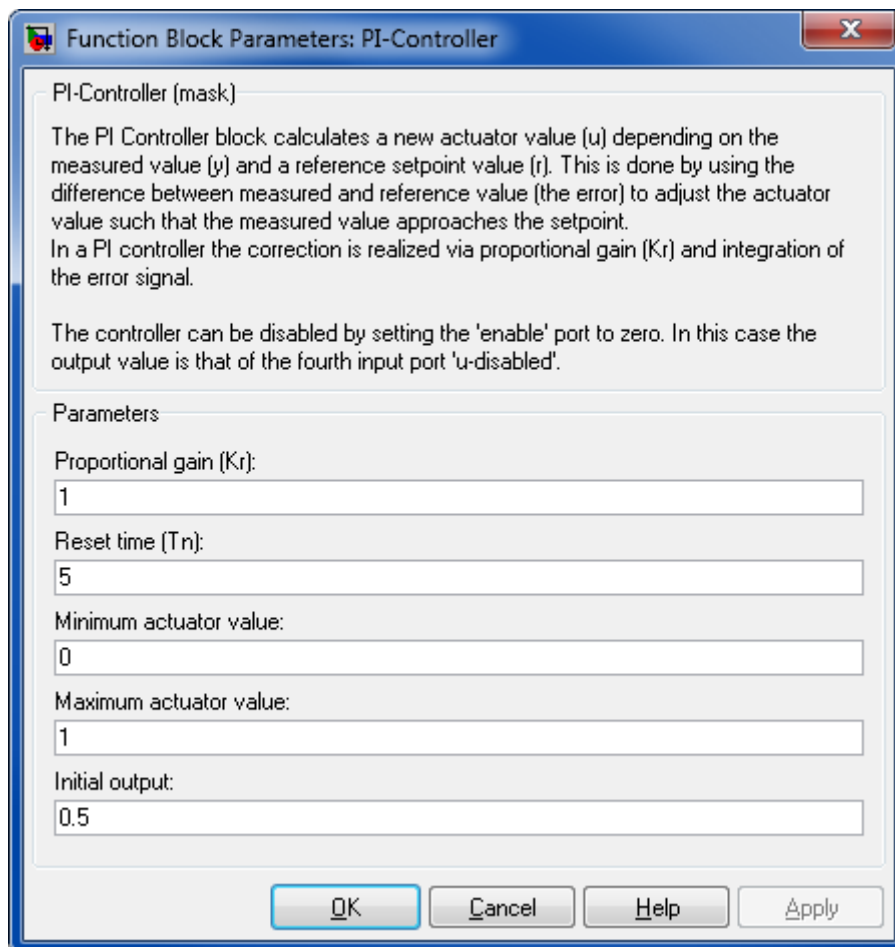
Inputs

r	Target value for signal	Scalar
y	Signal to be controlled	Scalar
enable	Enable or disable control mechanism	Scalar [0,1]
u-disabled	Output when control is disabled	Scalar

Outputs

u	Output value	Scalar
info	N.A.	N.A.

Parameters and Dialog Box



Proportional Gain (K_r)

Gain for the P-part.

Reset time (Tn)

Integral time for the I-part

Minimum Actuator Value

Minimum actuator value

Maximum Actuator Value

Maximum actuator value

Initial Output

Initial Output

Characteristics

Direct Feedthrough	No
--------------------	----

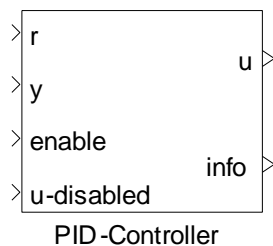
8.3.2 PID-Controller

PID Controller with Anti-Windup

Library

Controllers

Description



The PID Controller block calculates a new actuator value (u) depending on the measured value (y) and a reference set point value (r).

This is done by using the difference between measured and reference value (the error) to adjust the actuator value such that the measured value approaches the set point.

In a PID controller, the correction is realized via proportional gain (K_r), integration and derivation of the error signal. The proportional gain handles the current error, whereas the integration part attempts to use past information to correct the signal. The derivation part attempts to predict the future in correcting the error.

The gain K_r , the integral time T_n and the derivative time T_k are given by the user in the block's mask. The initial value of the output signal, as well as its minimum and maximum values is also given in the mask.

The controller can be disabled by setting the "enable" port to zero. In this case, the output value is that of the fourth input port "u-disabled".

Data Type Support

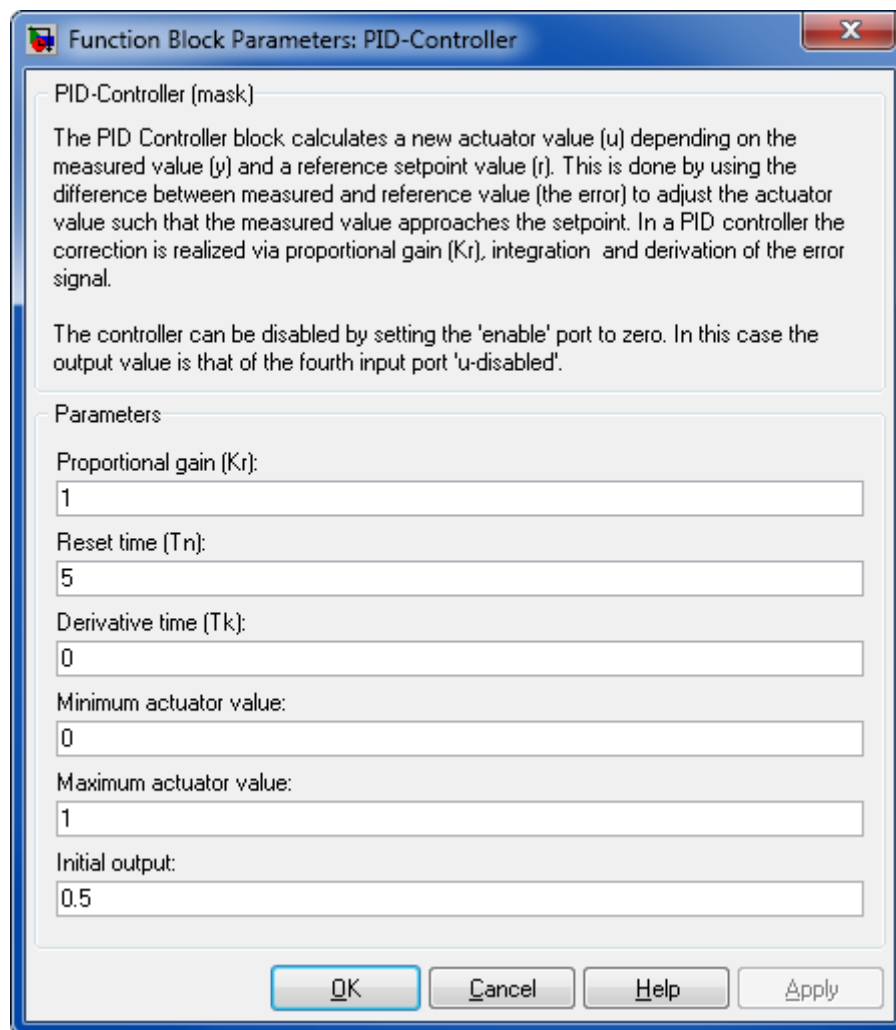
Inputs

r	Target value for signal	Scalar
y	Signal to be controlled	Scalar
enable	Enable or disable control mechanism	Scalar [0,1]
u-disabled	Output when control is disabled	Scalar

Outputs

u	Output value	Scalar
info	N.A.	N.A.

Parameters and Dialog Box

**Proportional Gain (K_r)**

Gain for the P-part.

Reset time (T_n)

Integral time for the I-part

Derivative Time (T_k)

Derivative time for the D-part

Minimum Actuator Value

Minimum actuator value

Maximum Actuator Value

Maximum actuator value

Initial Output

Initial Output

Characteristics

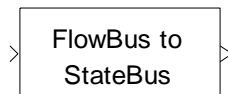
Direct Feedthrough	No
--------------------	----

8.4 MISCELLANEOUS**8.4.1 Convert FlowBus to StateBus**

Converts FlowBus to StateBus

Library

Miscellaneous

Description

Convert FlowBus to StateBus

Converts the named signals on the flow bus to corresponding names in the state bus.

The background of the block is as follows:

In several components like tanks the blocks from subsection "Thermodynamic states & properties" are used, which are designed to work with the Flow bus only (IFB and FB-ports). That means the outputs of the blocks are flows. To have a state output of the tank the output of the main state calculation block is converted using a "Convert FlowBus to StateBus".

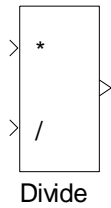
8.4.2 Divide

Division block with divide-by-zero handling

Library

Miscellaneous

Description



The Divide block performs a divide operation that detects and handles a division by zero. In such a case, the value supplied in the mask is used as output instead.

Data Type Support

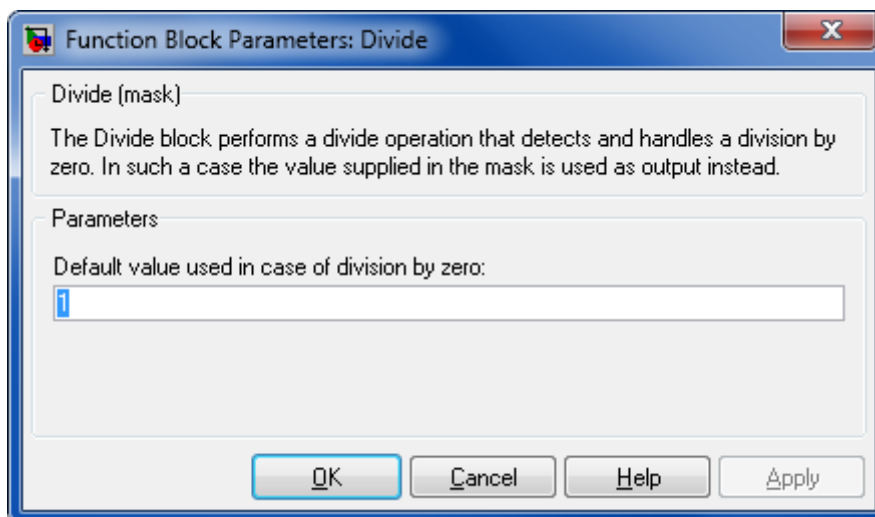
Inputs

In1	Dominator	Scalar / Vector
In2	Divisor	Scalar / Vector

Outputs

Out	Result	Scalar / Vector
-----	--------	-----------------

Parameters and Dialog Box



Default value used in case of division by zero

Default value used in case of division by zero.

Characteristics

Direct Feedthrough	Yes
--------------------	-----

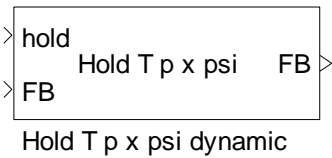
8.4.3 Hold T p x psi dynamic

Flow state memory.

Library

Miscellaneous

Description



The "Hold T p x psi dynamic" block can be used to hold the last values of temperature (T), pressure (p), vapor fraction (x) and molar fractions (psi). Initial condition is given in block-mask.

Data Type Support

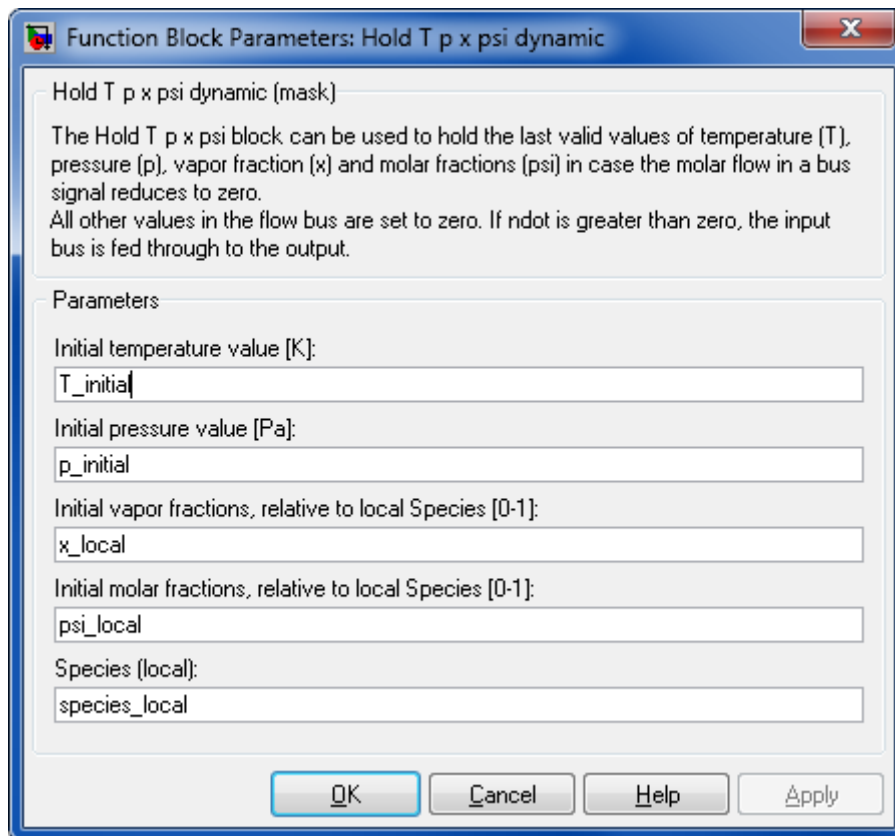
Inputs

hold	Activation signal	Scalar [0 or 1]
FB	Incoming flow (complete)	Flow bus

Outputs

FB	Outgoing flow (complete)	Flow bus
----	--------------------------	----------

Parameters and Dialog Box

**Initial temperature value [K]**

Initial temperature

Initial pressure value [Pa]

Initial pressure

Initial vapor fractions [0-1]

Initial vapor fractions relative to the parameter local Species

Initial molar fractions [0-1]

Initial molar fractions relative to the parameter local Species

Species (local)

Cell array of species used in this block for initialization

Characteristics

Direct Feedthrough	Depends
--------------------	---------

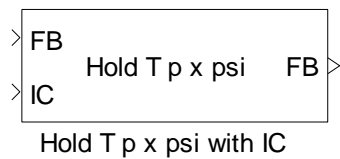
8.4.4 Hold T p x psi with IC

Flow state memory.

Library

Miscellaneous

Description



“Hold T p x psi with IC” hold the last value of the flow state (T,p,x,psi), but take the initial condition from second input port, in case the molar flow in a bus signal reduces to zero.

All other values in the flow bus are set to zero. If \dot{n} is greater than zero, the input bus is fed through to the output.

Data Type Support

Inputs

FB	Incoming flow (complete)	Flow bus
IC	Initial condition	Flow bus

Outputs

FB	Outgoing flow (complete)	Flow bus
----	--------------------------	----------

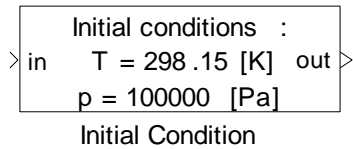
8.4.5 Initial Condition

Memory block for a Flow Bus

Library

Miscellaneous

Description



The Initial Condition block applies a one integration step delay. It works like a Simulink memory block for a complete flow bus with the initial states determined from the mask parameters. It uses the same units as the incoming flow bus.

Usually this block is used to break up algebraic loops in the model and provide initial states where demanded.

Please note, that 2-Phase flows of pure substances cannot be defined by this block.

Data Type Support

Inputs

FB	Incoming flow (complete)	Flow bus
PFin	Pressure feedback (Optional)	4-doubles vector

Outputs

FB	Outgoing flow (complete)	Flow bus
PFout	Pressure feedback for inlet (Optional)	4-doubles vector

Parameters and Dialog Box

Block Parameters: Initial Condition

Description

The Initial Condition block applies a one integration step delay. It works like a Simulink memory block for a complete flow bus with the initial states determined from the mask parameters. It uses the same units as the incoming flow bus.

Usually this block is used to break up algebraic loops in the model and provide initial states where demanded.

Parameters

Initial flow rate [mol/s]:

Temperature [unit]: C

Pressure [unit]: bar

Compounds

CO molar fraction [-]: <input type="text" value="0"/>	H2 molar fraction [-]: <input type="text" value="0"/>
H2O molar fraction [-]: <input type="text" value="0"/>	N2 molar fraction [-]: <input type="text" value="0.79"/>
CO2 molar fraction [-]: <input type="text" value="0"/>	O2 molar fraction [-]: <input type="text" value="0.21"/>

☐ Show pressure feedback ports

OK Cancel Help

Molar Flow [mol/s]

Initial molar flow

Temperature [unit]

Initial temperature

Pressure [unit]

Initial pressure

Compounds

Initial composition of the outgoing flow. This is a vector of the same length as the Species vector.

Show pressure feedback ports

Show or hide pressure feedback ports.

Characteristics

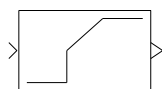
Direct Feedthrough	No
--------------------	----

8.4.6 LimitNdot

Library

Miscellaneous

Description



LimitNdot

This block is mainly a combination of a “wrap to zero” and saturation with upper limit. It uses internally the variables

TH_THERM_NDOT_MIN

TH_THERM_NDOT_MAX

from /Data/Constants.mat, which are loaded with the Thermolib workspace.

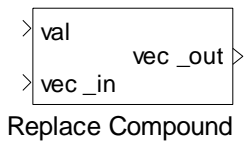
8.4.7 Replace Compound

Replaces a value within a vector at position of a given species

Library

Miscellaneous

Description



Given a vector “val” of same length like variable “Species” it replaces the vector element at position of the species specified in the block mask with the value at input vec_in.

This block is useful to change concentrations in psi vector or vapor fractions in x vector.

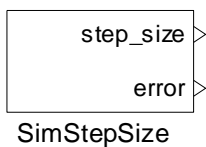
8.4.8 SimStepSize

Block provides the current simulation step size.

Library

Miscellaneous

Description



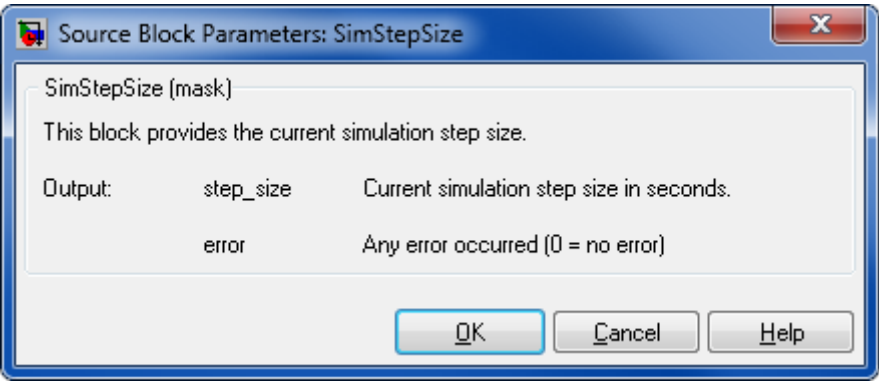
The SimStepSize block provides the current simulation step size.

Data Type Support

Outputs

step_size	Current step size of simulation	Scalar
error	Current error	Scalar

Parameters and Dialog Box



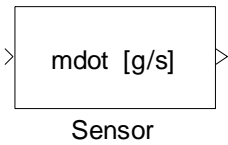
8.4.9 Sensor

Generalized measurement

Library

Miscellaneous

Description



The Sensor block extracts or computes a set of signals from a flow bus and outputs them in specified units. The possible unit conversions are described at the Unit Conversion block.

The list of extracted signals can be modified in the mask. It is possible to add, delete and edit those signals as well as to define a specific output order.

Note: The conversion table must not be empty.

Data Type Support

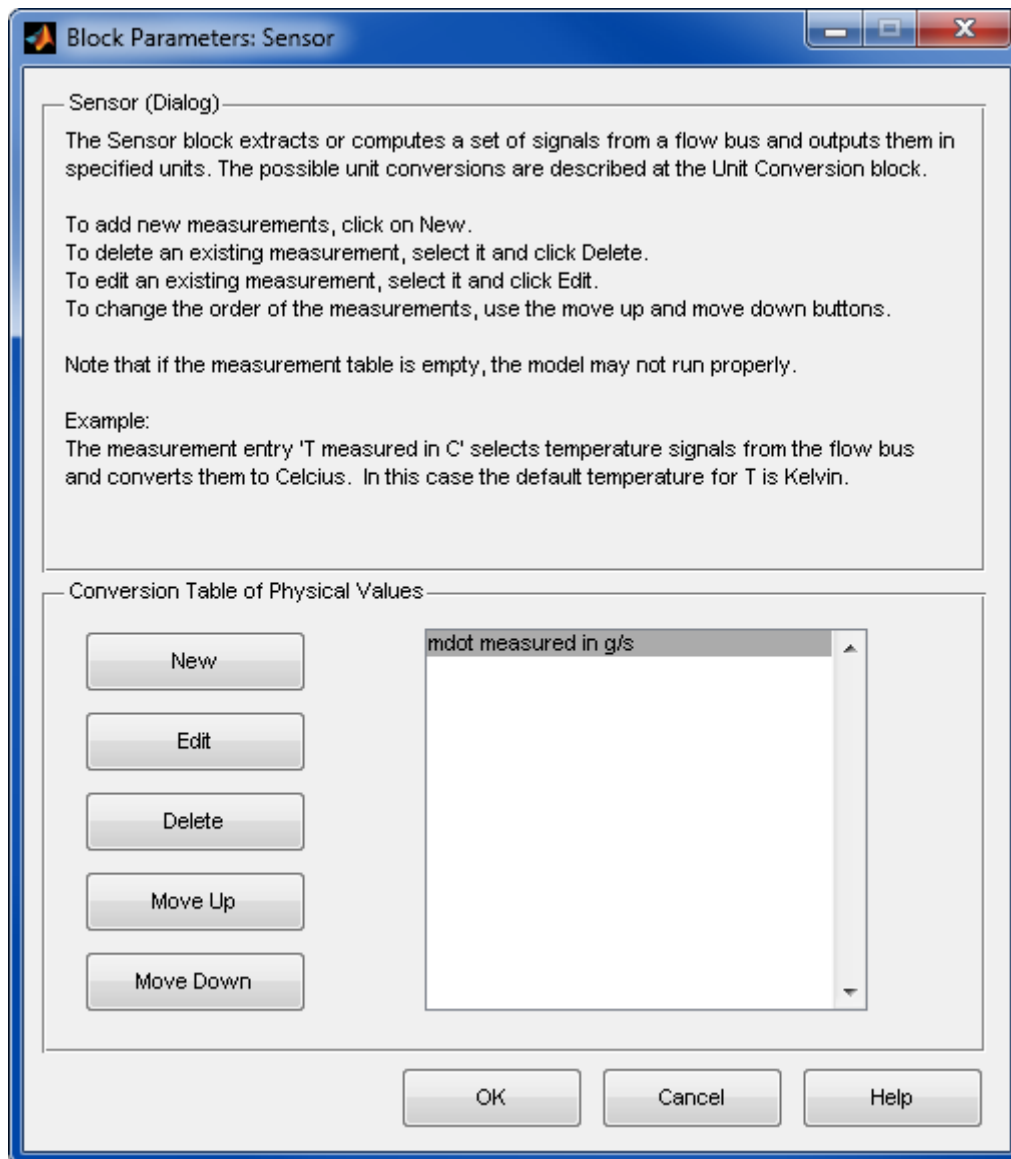
Inputs

Input port	Incoming flow bus	Flow bus
------------	-------------------	----------

Outputs

Output port	Selected signals in specified units	Scalar / vector
-------------	-------------------------------------	-----------------

Parameters and Dialog Box



Conversion Table of Physical Values

New: Create a new entry in the list.

Edit: Edit the selected entry from the list.

Delete: Delete the selected entry from the list.

Move Up: Move up the selected entry by one position.

Move Down: Move down the selected entry by one position.

Characteristics

Direct Feedthrough	Yes
--------------------	-----

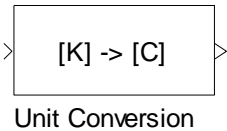
8.4.10 Unit Conversion

Unit conversion

Library

Miscellaneous

Description



The Unit Conversion block converts a signal between different compatible units. The following conversions are possible:

- $K \Leftrightarrow F \Leftrightarrow C$;
- $mW \Leftrightarrow W \Leftrightarrow kW$;
- $s \Leftrightarrow min \Leftrightarrow h$;
- $mol, kmol/s, min, h \Leftrightarrow mol, kmol/s, min, h$;
- $W s \Leftrightarrow kW h$;
- $mbar, bar, Pas \Leftrightarrow mbar, bar, Pas$;
- $g \Leftrightarrow kg$;
- $l, m^3/s, min, h \Leftrightarrow l, m^3/s, min, h$.

Data Type Support

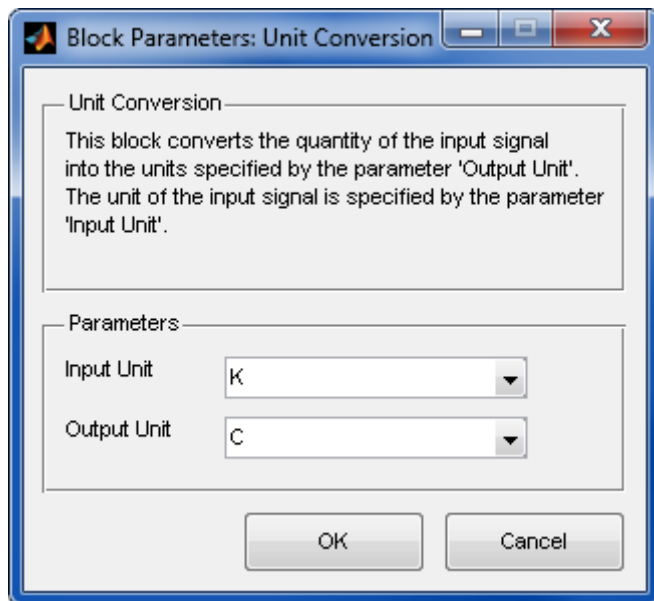
Inputs

Input port	Signal or list of signals	Scalar / Vector
------------	---------------------------	-----------------

Outputs

Output port	Signal(s) after unit conversion	Scalar / Vector
-------------	---------------------------------	-----------------

Parameters and Dialog Box



Input Unit

Unit the input signal is based on.

Output Unit

Unit the input signal is converted to.

Characteristics

Direct Feedthrough	Yes
--------------------	-----

8.5 SETUP

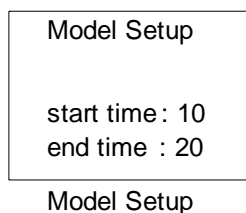
8.5.1 Model Setup

Initializes the Thermolib

Library

Properties

Description



The Model Setup block is always required for the Thermolib blocks to work correctly. It loads workspace variables needed for the simulation and prepares the generation of mass flow and energy balances.

Before the simulation starts, this block loads a set of mat files into the MATLAB workspace containing information about the used species and their chemical and thermodynamic properties, thermodynamic and numeric constants and other information required.

It is possible to change, add or remove data as needed to and from these mat files. This can be done either by using Excel, or, of course, by saving data from the MATLAB workspace directly.

The user specifies a chemical media data source, a mat file that contains compounds available to the model and their chemical and thermodynamic properties. This file is loaded when opening the model or starting a simulation. Other Thermolib blocks will also query Model Setup for this file to load the chemical information they need. To create another chemical media data file, it is recommended that you start with the ChemicalMediaData.xls file delivered with Thermolib rather than creating a xls-file from scratch. This is to ensure that the variable names confirm to the Thermolib standard and to ensure compatibility with the automatic translation script "th_GetChemicalMediaData".

Note: To create a .mat file with chemical media data from an modified ChemicalMediaData.xls, use the command

```
th_GetChemicalMediaData
```

Additionally the user can enter a load command, which is evaluated at model start time. This can be used to load data e. g. for specifying dimensions of components. The default is to load ComponentData.mat. This file contains default stack polarization curves.

In order to load further mat files or perform additional initializations, the recommended way to add that to the load command.

Note: There are certain internal variables in the MATLAB workspace required to run Thermolib using simulations appropriately. Therefore, it is not recommended to delete or modify the original mat files

ChemicalMediaData.mat, Constants.mat,
FlowBus.mat, SensorBus.mat,

or the load function discussed above in such a way that these variables are not accessible anymore.

Using the Model Setup block you can specify the Species used in this model. In the mask in the panel 'Select Species', a list shows all Species from the chemical media data file currently used in this model. The user can add species from chemical media data file by pressing 'Add'. The species to be added and whether it's allowed to be vapor and liquid during the simulation of the model (options not yet activated) can be chosen in the new upcoming dialog. If a medium is not selected as available and a block requires this medium, an error message will notify the user when they attempt to run a simulation.

Note: Some Thermolib blocks heavily depend on the existence of specific compounds. For instance, consider the Preferential Oxidation reactor in which a chemical reaction is specified $\text{CO} + 1/2\text{O}_2 \rightarrow \text{CO}_2$.

That means you will get error messages, if you do not select CO, O₂ and CO₂ as Species for this model in the Model Setup block.

The Model Setup block is also required for flow balance and energy balance generation as it provides the time span during which balance-related data is collected. Balancing is done by blocks Flow Balance and Energy Balance that are located in all Thermolib components.

Note: A balance is generated by calling the function `th_ExportBalance` from the MATLAB command window.

In the panel 'Model of gas phase' the equation of state (EOS) for the gas phase can be specified. Current options are:

- Ideal gas and
- Real gas (Peng Robinson).

Enthalpy of liquid can be modeled using two different ways:

- Using explicit evaporation properties ($h_{fg} = f(T)$)
- Using reference temperature and $h_{fg}(T_{ref})$ together with integration over $c_{p-liquid}$

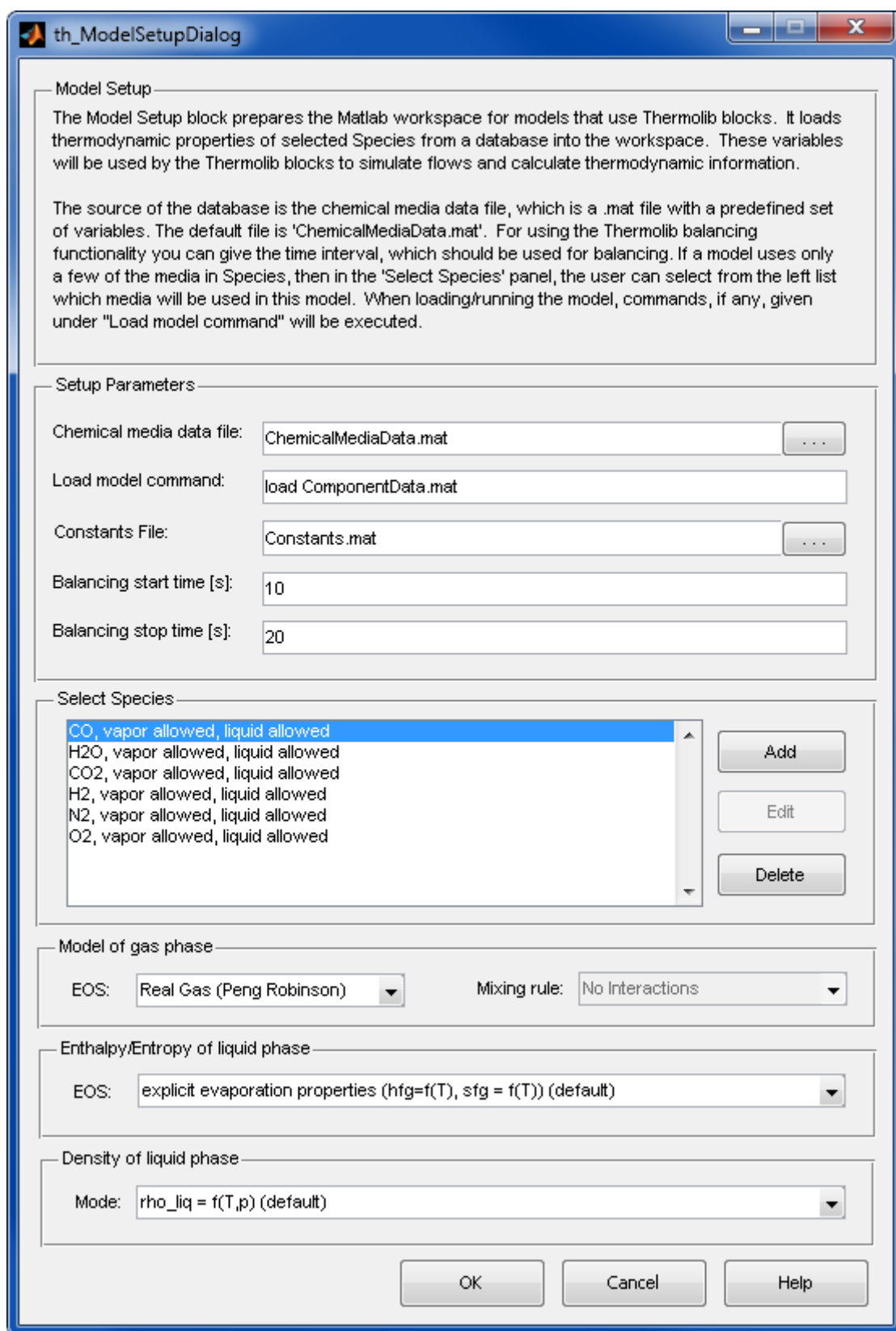
The first option is default and much more accurate. The second option is only to reproduce calculations made with older Thermolib releases and will be removed in future releases.

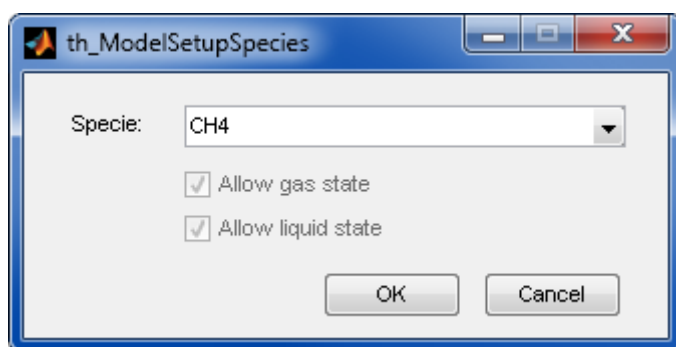
Liquid density can be modeled as

- Compressible ($\rho_{liq} = f(T, p)$) (default)
- Incompressible ($\rho_{liq} = \text{const}$)

The second option is only to reproduce results from older Thermolib results and will be removed in future releases.

Parameters and Dialog Box





Chemical media data file

A mat file which holds a database of available species and their chemical and thermodynamic data. The default file is 'ChemicalMediaData.mat'. To create a new file, it is strongly recommended to modify the original file under a different name, and ensure that the variable names confirm to the Thermolib standard. Alternatively, if the MATLAB tool box Excel-link is available, the user can use the ChemicalMediaData.xls file to adjust the data.

Load model command

Matlab statements that are executed when loading/executing the model. This can be a function call or a set of Matlab commands. The load model command will be evaluated in the MATLAB base workspace.

Balancing start time [s]

The start time of the time period during which the balance data is recorded.

Balancing end time [s]

The end time of the time period during which the balance data is recorded.

Select Species

Select species that will be available to the model. Most models do not use all the available species. Here the user has the option to add species from the chemical media data file to the model or remove them. Therefore the button 'Add' has to be pressed. In the new pop up dialog the user can select the specie to add from a dropdown menu that contains all mediums from chemical media data file.

Note: It's not allowed to run the simulation with both mediums, 'H2O' and 'H2O-IF97'. If the user tries to use both mediums, an error occurs.

Model of gas phase

Define which equation of state is used in the simulation of the model. Actual options are 'Ideal Gas' and 'Real Gas (Peng Robinson)'. For details about the equations used in the different cases, please refer to the Thermodynamic-Background chapter in this manual.

Characteristics

Direct Feedthrough	N.A.
--------------------	------

8.6 SINKS

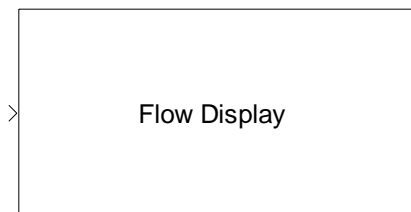
8.6.1 FlowDisplay

Show flow or state properties.

Library

Sinks

Description



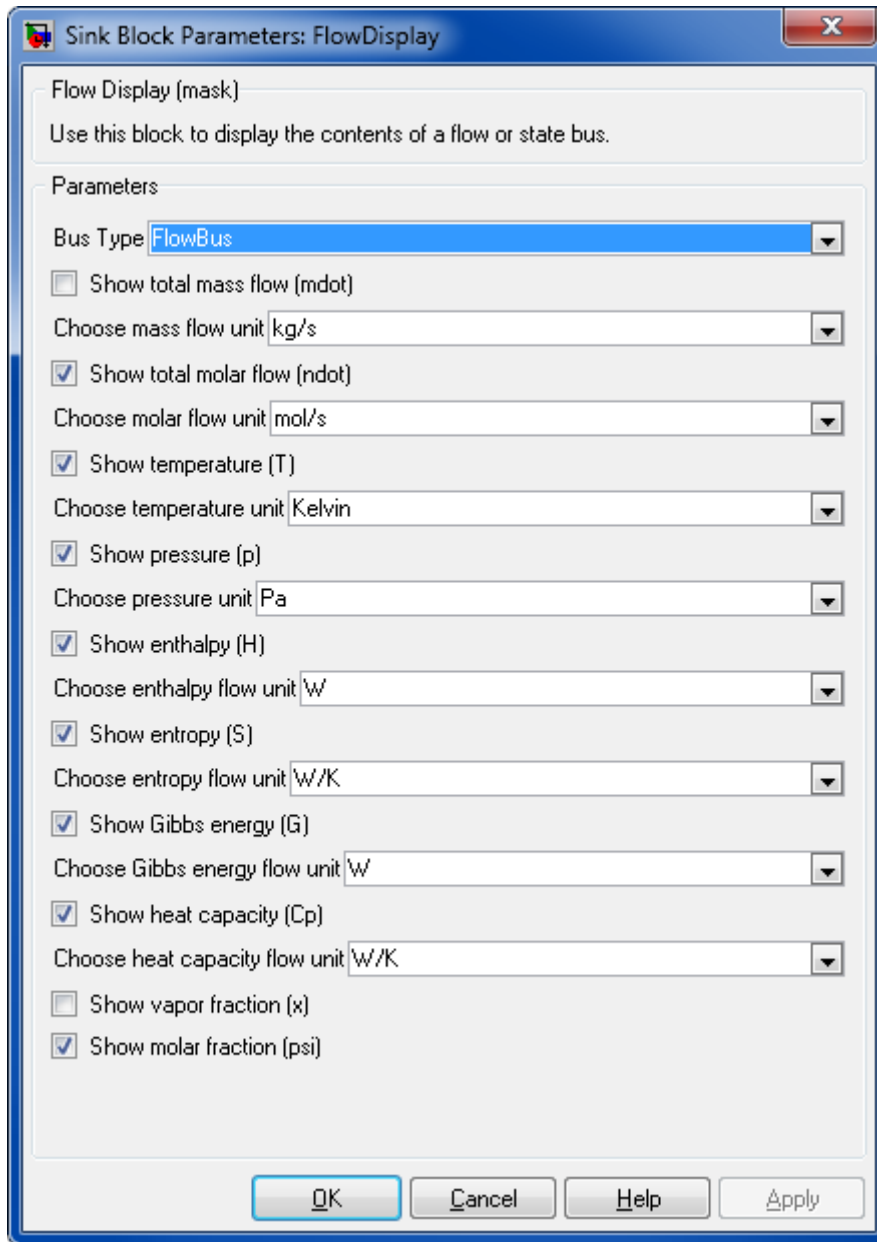
This block dynamically displays flow or state properties according to the given mask parameters. You can select which signals to show and which units should be used to display the selected values.

Data Type Support

Inputs

in	Input flow	Flow bus / State bus
----	------------	----------------------

Parameters and Dialog Box



Select Bus Type

Select the type of the incoming bus. Available options are FlowBus and StateBus. Change this option depending on the bus type you connect this block to.

Show Total Mass Flow (mdot)

Displays the total mass flow if selected

Show Total Molar Flow (ndot)

Displays the total molar flow (or total number of moles) if selected.

Show Temperature (T)

Displays temperature in chosen unit if selected

Show Pressure (p)

Displays pressure in chosen unit if selected

Show Enthalpy (H)

Displays enthalpy in chosen unit if selected

Show Entropy (S)

Displays entropy in chosen unit if selected

Show Gibbs Energy (T)

Displays Gibbs energy in chosen unit if selected

Show Heat Capacity (cp)

Displays heat capacity in chosen unit if selected

Show Vapor Mass Fraction (x)

Displays vapor mass fraction in chosen unit if selected

Show Molar fraction (psi)

Displays molar fraction in chosen unit if selected

Characteristics

Direct Feedthrough	Yes
--------------------	-----

8.6.2 Measurement Point**Library**

Sinks

Description

This block represents a measurement point in the real plant. Functionality is quite similar to the "Sensor" block from the subsection "Miscellaneous" with the following differences:

- The Measurement point can only measure ONE signal
- If enabled in the mask the signal can be saved to a workspace signal
- A tag is associated with the measurement point, which is quite common in process industry.

The block should be considered for a "drawing" of the plant process and instrumentation diagram, which is closer to the real world P&I drawings.

Data Type Support

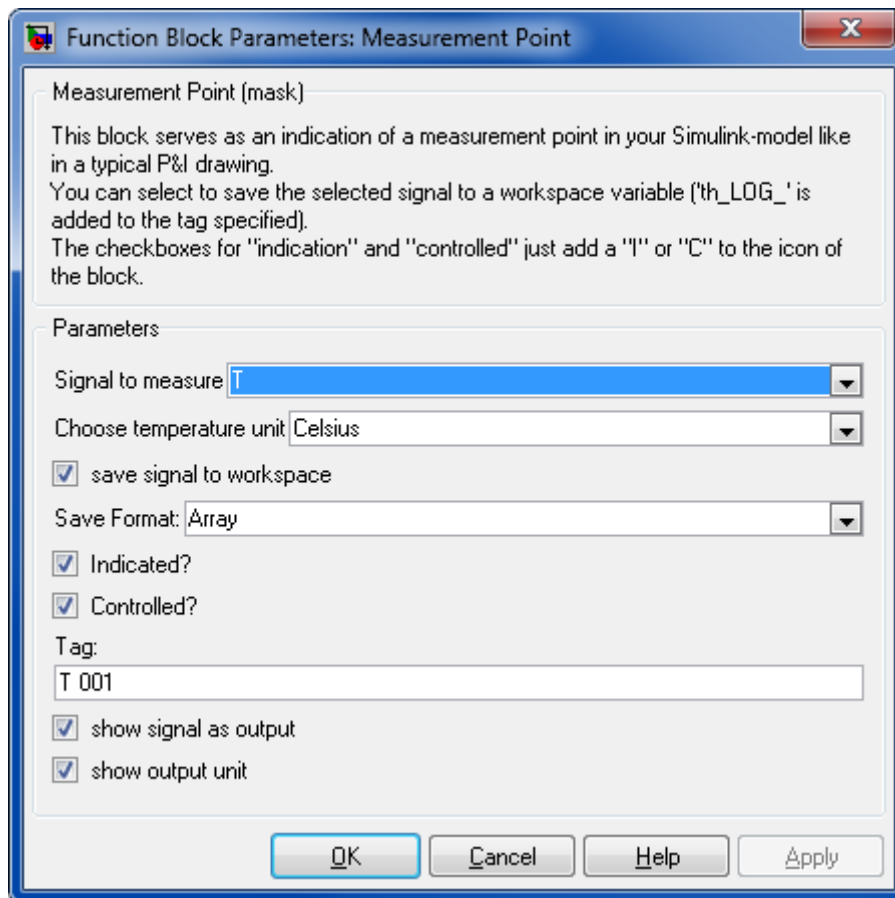
Inputs

in	Input flow	Flow bus / State bus
----	------------	----------------------

Outputs

-	scalar
---	--------

Parameters and Dialog Box



Signal to measure

Here currently only p (pressure) and T (temperature) are allowed. This is the physical signal, which is measured.

Choose pressure/temperature unit

Select the unit for the output (or saved) signal

Save signal to workspace

If checked, the signal is saved to MATLAB workspace in a variable, which is derived from the tag-name replacing all spaces by "_" and using the prefix "th_LOG". E. g. "T 001" will create a workspace variable "th_LOG_T_001".

Save format

"Array" or "Structure" or "Structure with Time". Correspond to the setting of the "Write to Workspace" block which is used internally. For details of the different formats, please check the Simulink documentation of the "Write to Workspace" block.

Indicated

If checked a "I" is added to the upper line of the icon.

Controlled

If checked a "C" is added to the upper line of the icon.

Tag

Unique name of the signal to be measured. This determines the workspace variable name used for saving the signal. The tag is displayed in the second line of the block.

Show signal as output

If enabled, the block has an output port of the selected and scaled physical signal.

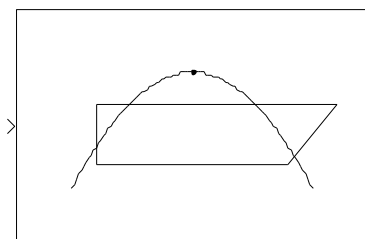
Show output unit

If enabled, the selected unit will be displayed in [] brackets in the upper line of the icon

8.6.3 State Plot

Library

Sinks

Description

State Plot

This block, when connected to a flow bus, displays dynamically the states in the selected iso-map. The selection of the Plot type determines automatically the 2-tuple of characteristics to display. This block allows monitoring of several states in an iso-map by setting its number of inputs to the desired value. Moreover, it is possible to observe the evolution of states over time by tuning the number of state samples to display. The recent samples are displayed in a dark color and oldest in a light color. The text annotation of the iso-lines can be refreshed,

when a zoom is applied, by pressing the pushbutton appearing in left-bottom corner of the figure.

Data Type Support

Inputs

in	Input flow	Flow bus
----	------------	----------

Parameters and Dialog Box

Sink Block Parameters: State Plot

State plot (mask)

This block monitors the state evolution over time. The plotted features are determined according to the selected plot type. The plot type defines the ISO-lines to put in the figure background. The plot of ISO-lines requires the specification of the present species, their concentration and the variation range of temperature and pressure. This block can be enabled or disabled by activating or deactivating the enable checkbox. The number of states to plot can be specified by choosing the number of inputs of the block. The history of state can be set by modifying the number of state samples to plot. A sampling value can be set after selecting the type of sampling.

Parameters

☒ Enable

Number of inputs
4

Plot type log(p)-h

Species in diagram
{H2O,'CO2'}

Concentrations
[1;0]

Unit of the concentrations mol/mol

Temperatures [K]
linspace(330,460,5)

Pressures [Pa]
logspace(3,7,20)

Number of state samples to plot
20

Sampling Sample time

Sample time
1

Line style specifier
-

Marker

☐ Save data

Limit data points to last
5000

Name for the saved data
StatePlotdata

Format of the saved data Structure with time

OK Cancel Help Apply

Enable

This check box, when disabled, allows saving computation by turning off the complete block. During simulation running, the call-subfunctions of the s-function are not called.

Select Number of inputs

Select the number of inputs of the block and consequently the number of states to display. The displayed states are indicated in the plot by assigning to them the corresponding number of input.

Choose the Plot Type

Choose the type of isolines to put in the background of the displayed figure and consequently the state pair to display.

Species in diagram

The specification of the species is mainly used for the background display i.e. the isolines.

Concentrations

The concentration of species is also used for the background display.

Unit of the concentrations

The unit of the concentration points out that the given concentrations should be specified in mol/mol.

Temperatures [K]

The specified range of the temperatures is used for the background display. It is used to compute and plot the $\log(p)$ -H, T-H, T-S and T-v within these specified temperatures.

Pressures [Pa]

The specified range of the pressures is used for the background display. It is used to compute and plot the $\log(p)$ -H, T-H, T-S and T-v within these specified pressures.

Number of state samples to plot

A shift from the value "1" to another integer value "n" greater than "1" of "The number of state samples to plot" makes the plot of the current accompanied by a record of "n-1" last states.

Sampling

This popup menu allows choosing the sampling mode of the State Plot block.

Decimation/Sample time

Specify the desired decimation or Sample time for the block.

Line style specifier

Specify the desired style for the lines linking the last displayed points. The lines are not displayed if the field is left empty.

Marker

Specify the desired marker for displayed points. The default setting "." is set when this field is left empty.

Save data

This checkbox, when enabled, allows saving the displayed data to the workspace.

Limit data points to last

Specify the number of sample points to save. Please specify "Inf", if all the points should be saved.

Name of the saved data

A name for the saved data can be specified. A default name "StatePlotData" is set when this field is left empty.

Format of the saved data

This popup menu allows choosing a format type for the data to save to the workspace.

8.7 SOURCES

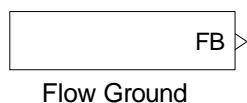
8.7.1 Flow Ground

Flow bus with zero-flow.

Library

Sources

Description



This block simulates a flow bus with $\text{ndot} = 0$.

The species, which is simulated, is the first species defined in the Model Setup block.

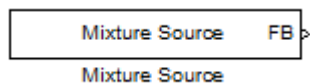
8.7.2 Mixture Source

One phase flow or reservoir of a mixture of substances.

Library

Sources

Description



This block represents a source of a one phase flow or reservoir of a mixture of substances. Its flow rate is described on a molar, volumetric or mass basis. Optionally values can be specified externally via ports, by specifying "Ext".

According to Gibbs' phase rule in the one phase region, two variables have to be specified – here the temperature and the pressure.

The composition of the mixture is determined by the respective molar fractions of the compounds.

Note: The list of substances is taken from the chemical database. In case of a reservoir, the flow rate is not defined. All values will be automatically converted from the chosen units to SI units used by the simulation model. The Mixture Source block outputs a bus signal in order to address properties by name.

Data Type Support

Inputs

T [unit]	Source Temperature (optional)	Scalar (unit)
p [unit]	Source Pressure (optional)	Scalar (unit)
flow [unit]	Source Flow (optional)	Scalar (unit)

Outputs

out	Outgoing flow	Flow bus
-----	---------------	----------

Parameters and Dialog Box

Block Parameters: Mixture Source

Description

This block represents a source of a one phase flow or reservoir of a mixture of substances. Its flow rate is described on a molar, volumetric or mass basis.

According to Gibbs phase rule in the one phase region, two variables have to be specified: here the temperature and the pressure. The composition of the mixture is determined by the respective molar fractions of the compounds.

Parameters

Type:

Flow Rate: ☐ Ext

Temperature: ☐ Ext

Pressure: ☐ Ext

Compounds

CO molar fraction [-]: <input type="text" value="0"/>	H2 molar fraction [-]: <input type="text" value="0"/>
H2O molar fraction [-]: <input type="text" value="0"/>	N2 molar fraction [-]: <input type="text" value="0.79"/>
CO2 molar fraction [-]: <input type="text" value="0"/>	O2 molar fraction [-]: <input type="text" value="0.21"/>

OK Cancel Help

Type

The flow is either a

- Mass Flow,
- Molar Flow,
- Volume Flow, or a
- Reservoir.

Flow Rate

According to the selected type, the flow rate is expressed in the following units:

- Mass flow: [g/s], [g/min], [g/h], [kg/s], [kg/min], [kg/h];
- Molar flow: [mol/s], [mol/min], [mol/h], [kmol/s], [kmol/min], [kmol/h];
- Volumetric flow: [m³/s], [m³/min], [m³/h], [l/s], [l/min], [l/h].

In case of a reservoir, the flow rate is not defined and is set internally to NaN.

Temperature and Pressure Parameters

For fluids in one phase, temperature and pressure is given in

- Temperature: [C], [K], [F];
- Pressure: [Pa], [bar], [mbar].

Compounds

The chemical composition is given by means of a list of molar fractions

- <substance> molar fraction: belongs to the interval [0,1].

The sum of the molar fractions is one. The user can give fractions larger than one; they will be normalized by the sum of all fractions.

Characteristics

Direct Feedthrough	N.A.
--------------------	------

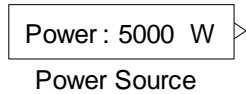
8.7.3 Power Source

Electric flow vector

Library

Sources

Description



A Power Source contains electric properties being admitted to a system or leaving a system, respectively. It models a direct current (DC) power flow if the frequency value is set to zero, otherwise an alternating current (AC) power flow.

The electric power is calculated from voltage and current by means of the formula $P = U I$.

Data Type Support

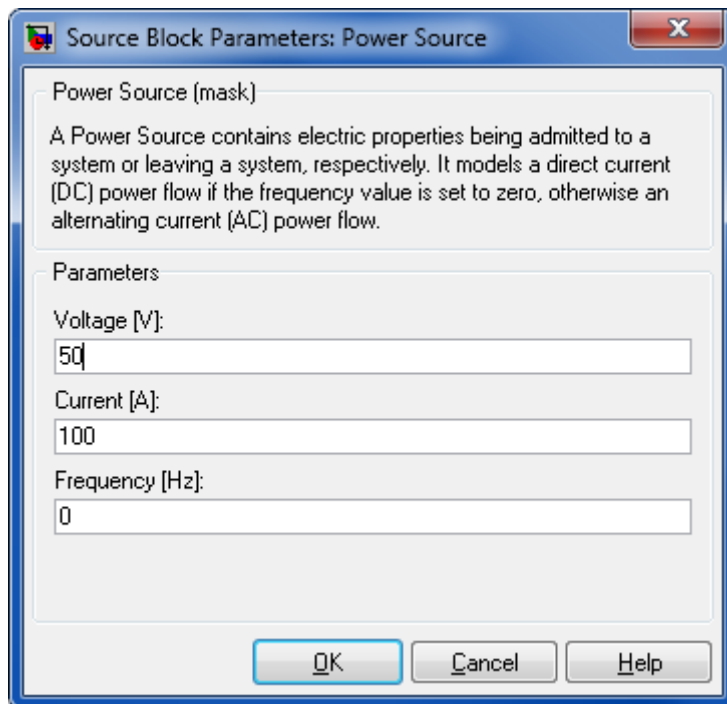
Outputs

out	Outgoing power vector	Power bus
-----	-----------------------	-----------

The Power Vector block outputs a vector of electric properties structured as follows:

Pos.	Name	Symbol	Unit
1	Electric Power	P	[W]
2	Voltage	U	[V]
3	Current	I	[A]
4	Frequency	F	[Hz]

Parameters and Dialog Box

**Voltage [V]**

The voltage of the power source

Current [A]

The electric current of the power source

Frequency [Hz]

The frequency of the power source. This determines whether direct (DC) or alternating current (AC) is used:

- DC: 0 Hz,
- AC: > 0 Hz.

Characteristics

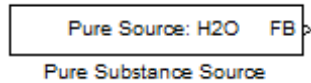
Direct Feedthrough	Yes
--------------------	-----

8.7.4 Pure Substance Source

One- or two-phase flow or reservoir

Library

Sources

Description

This block represents a source of a one- or two-phase flow or reservoir of a pure substance. Optionally values can be specified externally via ports, by specifying "Ext".

Its flow rate is described on a molar, volume or mass basis.

According to Gibbs' phase rule in a one phase region, two variables have to be specified – here, the temperature and the pressure. In contrast, in a two-phase region, only one degree of freedom is left. Thus, either the temperature or the pressure has to be given, in addition to the vapor fraction.

Note, that the volume flow units represent actual volumes, which are not normalized to standard conditions.

Note: The list of substances is taken from the chemical database. In case of a reservoir, the flow rate is not defined. At present, only the liquid and vapor as well as the mixture phases are taken into account. All values will be automatically converted from the chosen units to SI units used by the simulation model. The Pure Substance Source block outputs a bus signal in order to address properties by name.

Data Type Support**Inputs**

T [unit]	Source Temperature (optional)	Scalar (unit)
p [unit]	Source Pressure (optional)	Scalar (unit)
flow [unit]	Source Flow (optional)	Scalar (unit)
x [-]	Source Vapor Fraction (optional)	Scalar (-)

Outputs

out	Outgoing flow	Flow bus
-----	---------------	----------

Parameters and Dialog Box

Block Parameters: Pure Substance Source

Pure Substance Flow Source

This block represents a source of a one or two-phase flow or reservoir of a pure substance. Its flow rate is described on a molar, volumetric or mass basis.

According to Gibbs' phase rule in a one phase region, two variables have to be specified - here the temperature and the pressure. In contrast, in a two phase region only one degree of freedom is left. Thus either the temperature or the pressure have to be given, in addition to the vapor fraction.

Parameters

Select substance:

Type:

Flow Rate: ☐ Ext

Phase Selection

☒ 1 Phase
☐ 2 Phases

Temperature and Pressure Parameters

Temperature: ☐ Ext

Pressure: ☐ Ext

Vapor fraction: ☐ Ext

OK Cancel Help

Substance

The represented substance

Type

The flow is either a

- Mass flow,
- Molar flow,
- Volume flow, or a
- Reservoir.

Flow Rate

According to the selected Type, the flow rate is expressed in the following units:

- Mass flow: [g/s], [g/min], [g/h], [kg/s], [kg/min], [kg/h];
- Molar flow: [mol/s], [mol/min], [mol/h], [kmol/s], [kmol/min], [kmol/h];
- Volumetric flow: [m³/s], [m³/min], [m³/h], [l/s], [l/min], [l/h].

In case of a reservoir, the flow rate is not defined and is set internally to NaN.

Note, that the volume flow units represent actual volumes, which are not normalized to standard conditions.

Phase Selection

One- phase or two-phase region

Temperature and Pressure Parameters

For fluids in one phase, both the temperature and the pressure are given in

- Temperature: [C], [K], [F];
- Pressure: [Pa], [bar], [mbar].

In the two-phase region, either the equilibrium temperature or the pressure can be set, in addition to the vapor fraction

- Vapor Fraction: belonging to the interval [0,1]

which describes the fraction of vapor in the substance. A value of 0 means 100% liquid whereas a value of 1 means 100% vapor.

Characteristics

Direct Feedthrough	Yes
--------------------	-----

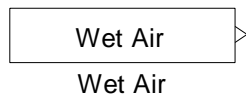
8.7.5 Wet Air

Wet or dry air vector

Library

Sources

Description



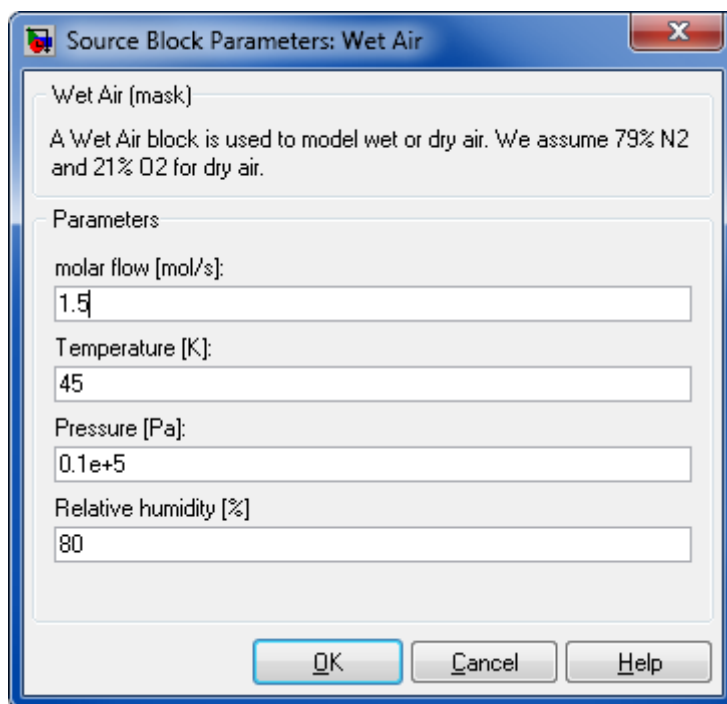
A Wet Air block is used to model wet or dry air. We assume 79% N₂ and 21% O₂ for dry air.

Data Type Support

Outputs

out	Outgoing flow	Flow bus
-----	---------------	----------

Parameters and Dialog Box



Volumetric flow (tot) [m³/h]

Total volumetric flow

Temperature [K]

Temperature

Pressure [Pa]

Pressure

Relative Humidity [-]

Humidity

Characteristics

Direct Feedthrough	Yes
--------------------	-----

8.8 THERMODYNAMIC STATES AND PROPERTIES

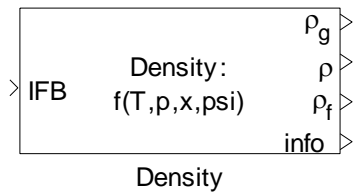
8.8.1 Density

Density calculation of a substance

Library

Thermodynamic States and Properties

Description



The Density block determines the density of a given media flow in both the gaseous and the liquid phase as well as for the overall flow in g/m^3 . From these, the overall density is computed depending on the respective vapor fractions of the species.

Data Type Support

Inputs

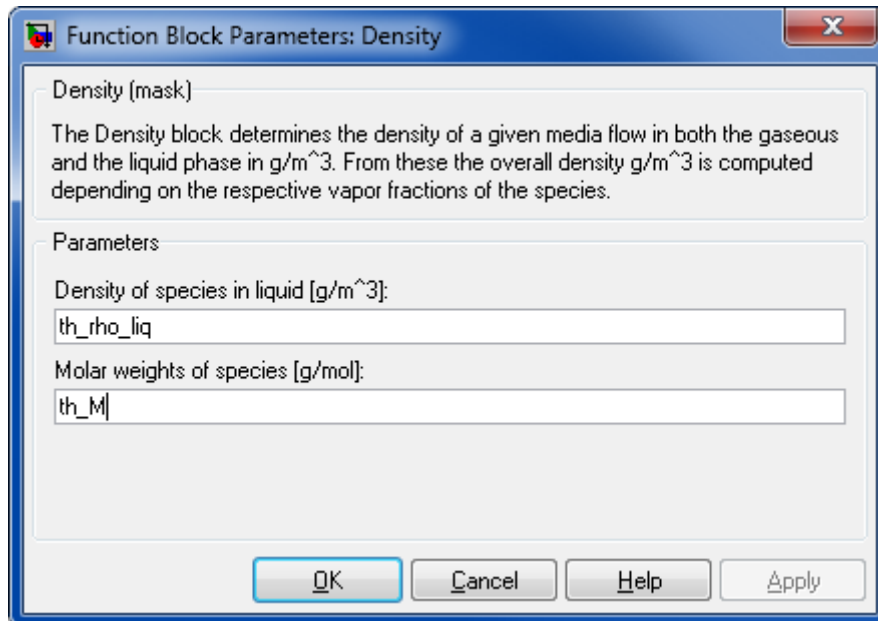
IFB	Incoming flow	Incomplete flow bus
-----	---------------	---------------------

Outputs

ρ_g	Density for the gaseous portion	Scalar $[\text{g/m}^3]$
ρ	Overall density	Scalar $[\text{g/m}^3]$

ρ_f	Density for the liquid portion	Scalar [g/m ³]
info	N.A.	N.A.

Parameters and Dialog Box



Density of Species in Liquid [g/m³]

A vector containing the species' density in the liquid phase

Molar Weight of Species [g/mol]

A row vector contains the molar weights. This is a vector of the same length as the Species vector.

Characteristics

Direct Feedthrough	Yes
--------------------	-----

You can look up which property values you will need for using this block in the document [ChemicalMediaData_Dependency_StatesAndProperties.pdf](#)

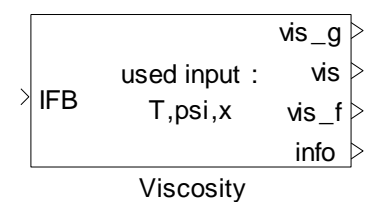
8.8.2 Viscosity

Viscosity calculation of substance

Library

Thermodynamic States and Properties

Description



The Viscosity block determines the viscosity from a given temperature and mole fractions of components. This is done with the help of an equation approximating the thermal conductivity as a function of temperature for pure substances.

for gases:

$$\text{VisG} = A + B \cdot T + C \cdot T^2 + D \cdot T^3$$
 where T in Kelvin and VisG in [Pa.s]

for liquids:

$$\ln(\text{VisL}) = A + B/T + C \cdot T + D \cdot T^2$$
 where T in K and VisL in c_p

Raoult rule is used to calculate viscosity of mixtures.

No model is used to consider two phase flow; therefore the calculation is not valid enough in case of two phase flow.

Data Type Support

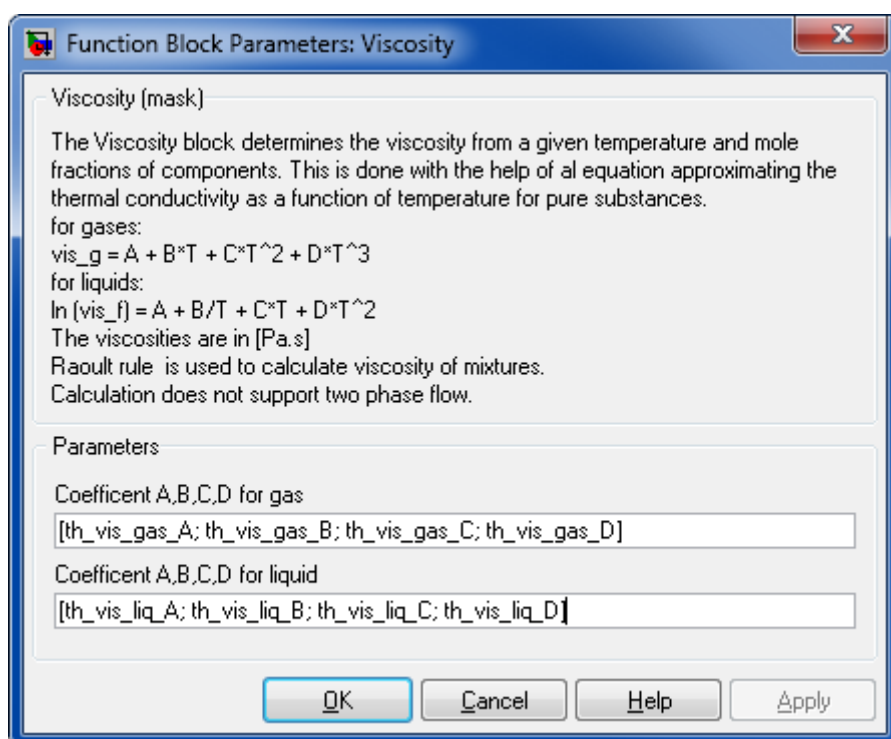
Inputs

IFB	Incoming flow	Incomplete flow bus
-----	---------------	---------------------

Outputs

vis_g	Viscosity for the gaseous phase	Scalar [Pa.s]
vis	Overall Viscosity	Scalar [Pa.s]
vis_f	Viscosity for the liquid phase	Scalar [Pa.s]
info	N.A.	N.A.

Parameters and Dialog Box



Coefficient A, B, C, D for gas

Coefficients for viscosity polynomial in gas phase.

Coefficient A, B, C, D for liquid

Coefficients for viscosity polynomial in liquid phase.

Characteristics

Direct Feedthrough	Yes
--------------------	-----

You can look up which property values you will need for using this block in the document [ChemicalMediaData_Dependency_StatesAndProperties.pdf](#)

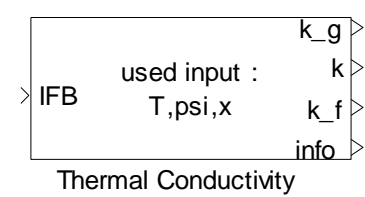
8.8.3 Thermal Conductivity

Thermal conductivity calculation

Library

Thermodynamic States and Properties

Description



The Thermal Conductivity block determines the thermal conductivity from a given temperature and mole fractions of components. This is done with the help of the polynomial equation approximating the thermal conductivity as a function of temperature for pure substances.

$k = A + B \cdot T + C \cdot T^2 + D \cdot T^3$ where T in K and K in W/m.K

Raoult rule is used to calculate thermal conductivity of mixtures.

Calculation does not support two phase flow.

No model is used to consider two phase flow; therefore the calculation is not valid enough in case of two phase flow.

Data Type Support

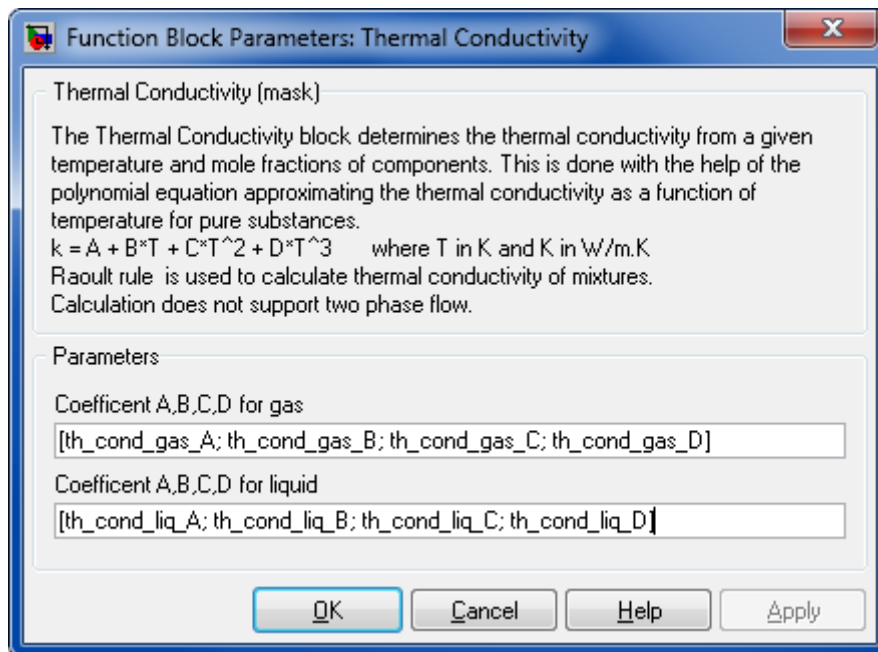
Inputs

IFB	Incoming flow	Incomplete flow bus
-----	---------------	---------------------

Outputs

k_g	Thermal conductivity for the gaseous phase	Scalar [W/(m K)]
k	Overall thermal conductivity	Scalar [W/(m K)]
k_f	Thermal conductivity for the liquid phase	Scalar [W/(m K)]
info	N.A.	N.A.

Parameters and Dialog Box



Coefficient A, B, C, D for gas

Coefficients for thermal conductivity polynomial in gas phase.

Coefficient A, B, C, D for liquid

Coefficients for thermal conductivity polynomial in liquid phase.

Characteristics

Direct Feedthrough	Yes
--------------------	-----

You can look up which property values you will need for using this block in the document [ChemicalMediaData_Dependency_StatesAndProperties.pdf](#)

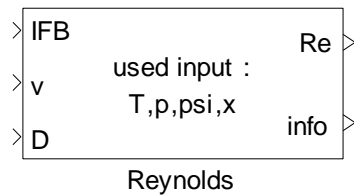
8.8.4 Reynolds

Calculation of Reynolds number

Library

Thermodynamic States and Properties

Description



The Reynolds block calculates Reynolds number of a flow.

Data Type Support

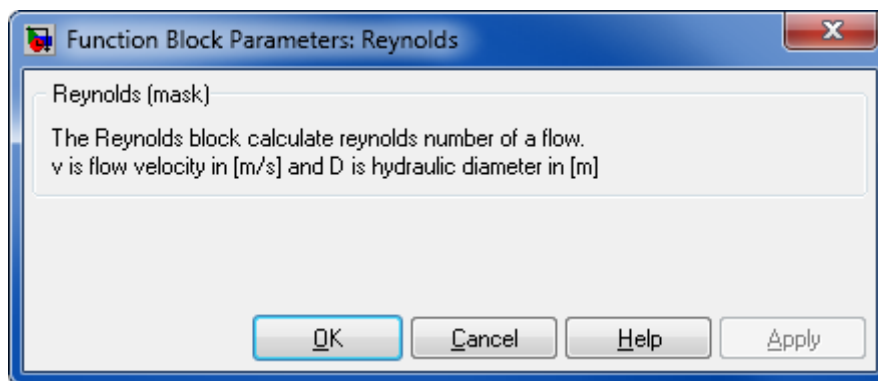
Inputs

IFB	Incoming flow	Incomplete flow bus
v	Velocity	Scalar [m/s]
D	Hydraulic diameter	Scalar [m]

Outputs

Re	Reynolds number	Scalar []
info	N.A.	N.A.

Parameters and Dialog Box



Characteristics

Direct Feedthrough	Yes
--------------------	-----

You can look up which property values you will need for using this block in the document [ChemicalMediaData_Dependency_StatesAndProperties.pdf](#)

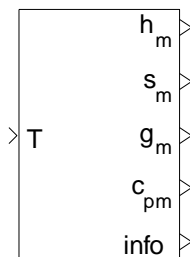
8.8.5 Gas Properties

Enthalpy, Entropy, free Gibbs Energy and Heat Capacity calculation for gases.

Library

Thermodynamic States and Properties

Description



Gas Properties

The Gas Properties block determines the

- molar enthalpy h_m ,
- molar entropy s_m ,
- molar free Gibbs energy g_m , and the
- molar heat capacity at constant pressure c_{pm}

at a given temperature.

$$\frac{c_{p,m}}{R} = A + BT + CT^2 + DT^3 + ET^4$$

$$\frac{h_m}{RT} = A + \frac{BT}{2} + \frac{CT^2}{3} + \frac{DT^3}{4} + \frac{ET^4}{5} + \frac{F}{T}$$

$$\frac{s_m}{R} = A \ln(T) + BT + \frac{CT^2}{2} + \frac{DT^3}{3} + \frac{ET^4}{4} + G$$

$$g_m = h_m - s_m T$$

All coefficients A-G are fluid-dependent and can be taken from the original NASA thermo data file. They assume that all gases are ideal gases and the validity domain is usually between $th_NASA_T_min$ and $th_NASA_T_max$ given in ChemicalMediaData.

Temperature Range	NASA coefficients
th_NASA_T_min to th_NASA_T_mid	A1 ... G1
th_NASA_T_mid to th_NASA_T_max	A ... G

The pressure dependency of entropy is not considered in this block.

Data Type Support

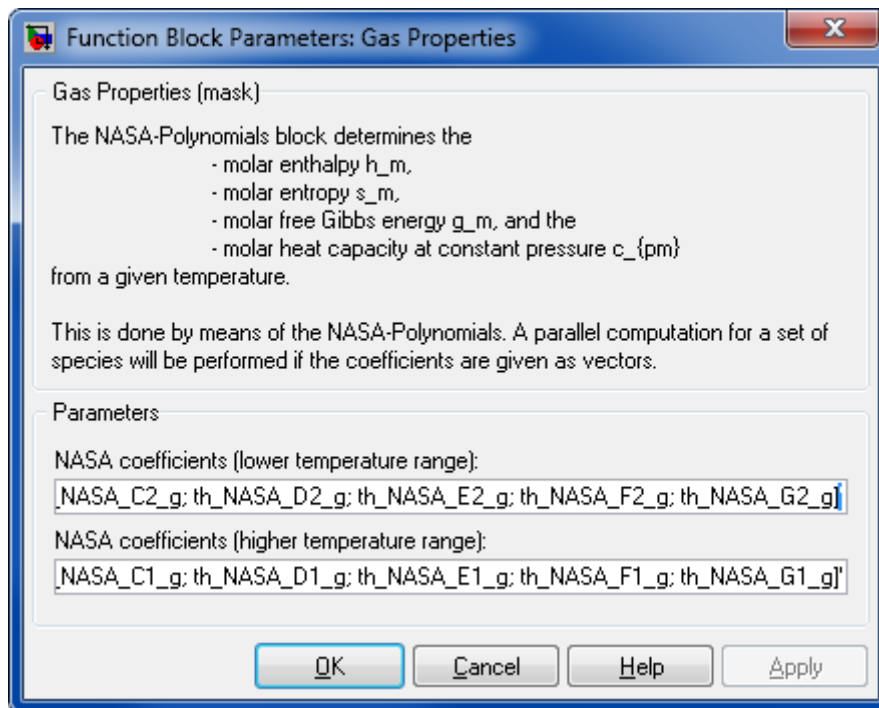
Inputs

T	Given temperature	Scalar
---	-------------------	--------

Outputs

h_m	Molar enthalpy	Vector [J/mol]
s_m	Molar entropy	Vector [J/(mol K)]
g_m	Molar free Gibbs energy	Vector [J/mol]
C_{pm}	Molar heat capacity at constant pressure	Vector [J/(mol K)]
info	N.A.	N.A.

Parameters and Dialog Box



NASA Coefficients (lower temperature range):

The coefficients used for temperatures at lower temperature range.

NASA Coefficients (higher temperature range):

The coefficients used for temperatures at higher temperature range.

Characteristics

Direct Feedthrough	Yes
--------------------	-----

You can look up which property values you will need for using this block in the document [ChemicalMediaData_Dependency_StatesAndProperties.pdf](#)

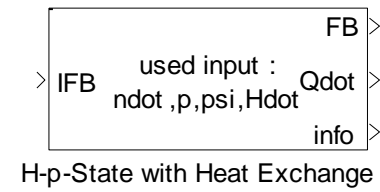
8.8.6 H-p-State with Heat Exchange

Complete flow bus using enthalpy and pressure considering heat exchange.

Library

Thermodynamic States and Properties

Description



The H-p-State with Heat Exchange block completes the outgoing flow bus (FB) using enthalpy and pressure. All properties of the outgoing flow are consistent.

Additionally, a heat exchange with the surrounding environment is taken into account. In particular, the outgoing temperature and heat exchange are results of a coupled computation.

Data Type Support

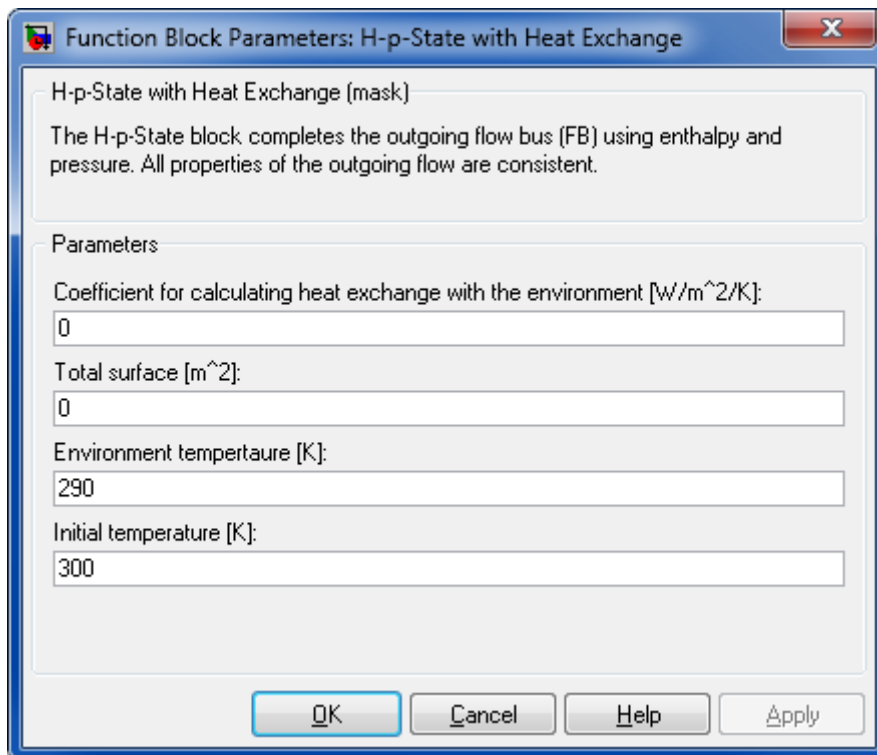
Inputs

IFB	Incoming flow	Incomplete flow bus
-----	---------------	---------------------

Outputs

FB	Outgoing flow	Complete flow bus
Qdot	Heat exchanged with environment	Scalar [W]
	Bus with internal information e. g.:	
info	<i>error_code</i> ... vector with error codes	simulink bus
	<i>T_iter</i> ... number of temperature iterations	

Parameters and Dialog Box



Coefficient of calculating heat exchange with the environment [W/(m²K)]

The coefficient for calculating heat exchange with the environment.

Total Surface [m²]

The external surface, at which the heat exchange with the environment takes place. It is assumed that the temperature is constant at the surface.

Environment temperature [K]

Environment temperature.

Initial temperature [K]

The initial value used for the computation.

Characteristics

Direct Feedthrough	Yes
--------------------	-----

You can look up which property values you will need for using this block in the document [ChemicalMediaData_Dependency_StatesAndProperties.pdf](#)

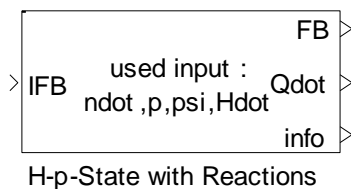
8.8.7 H-p-State with Reactions

Complete flow bus using enthalpy and pressure considering reactions and heat exchange.

Library

Thermodynamic States and Properties

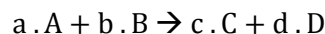
Description



The H-p-State with Reactions block completes the outgoing flow bus (FB) using enthalpy and pressure. All properties of the outgoing flow are consistent.

A reactions matrix is given the block's mask where each row holds the coefficients of a separate reaction. It is assumed that all species are in gaseous form and all the reactions take place simultaneously.

For a given reaction:



the chemical equilibrium condition can be written as:

$$\ln(K) = -\frac{\Delta G}{RT}$$

Where:

$$K = \left(\frac{\psi_C^c \cdot \psi_D^d}{\psi_A^a \cdot \psi_B^b} \right) \cdot \left(\frac{p}{p_0} \right)^{c+d-a-b}$$

and ψ_j is the mole fraction of species j , ΔG is the difference of Gibbs free energy between products and reactants and R is the universal gas constant.

Heat exchange with the surrounding environment is taken into account. In particular, the outgoing temperature and heat exchange are results of a coupled computation. It is assumed that the reactions take place simultaneously.

The outgoing flow is in chemical and thermodynamic equilibrium.

Data Type Support

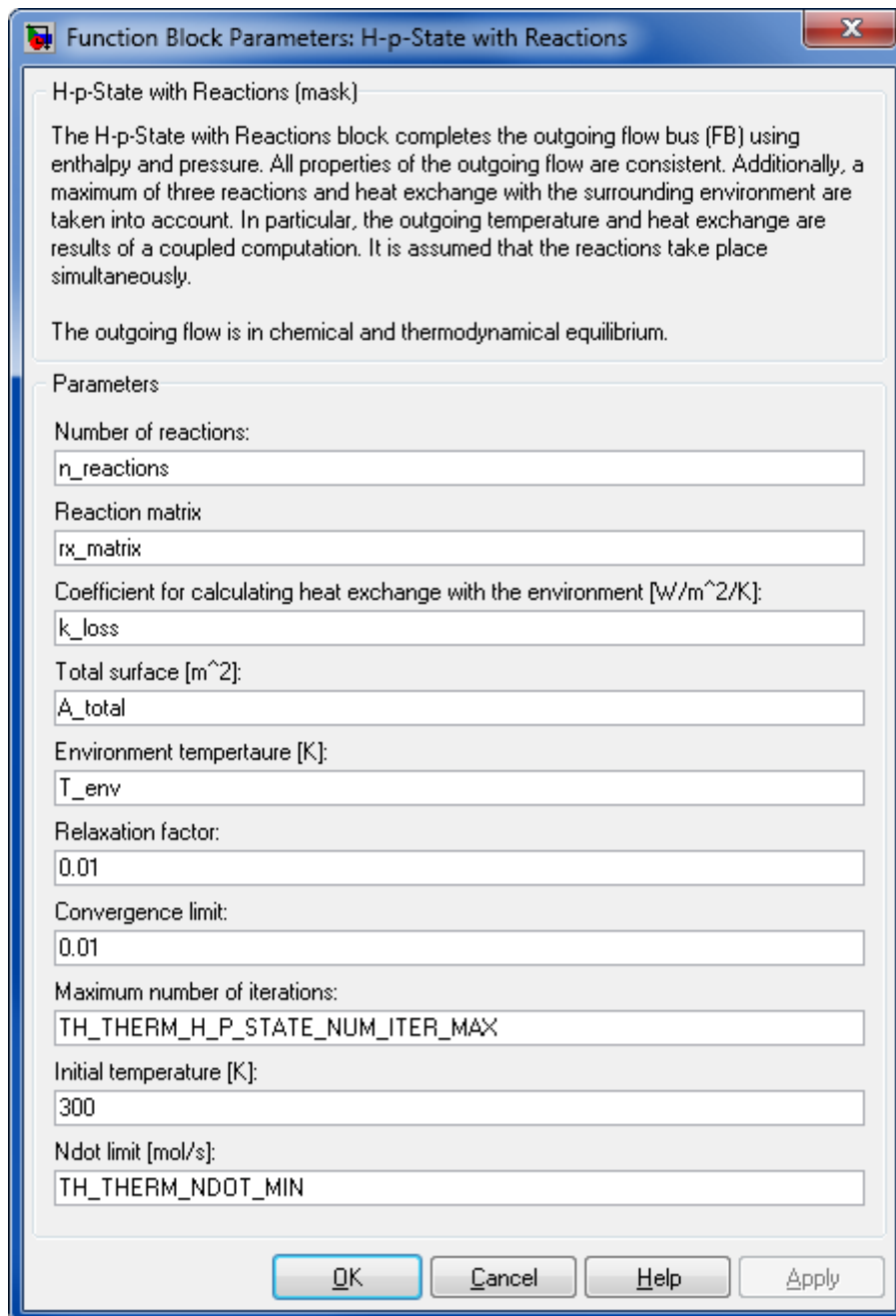
Inputs

IFB	Incoming flow	Incomplete flow bus
-----	---------------	---------------------

Outputs

FB	Outgoing flow	Complete flow bus
Qdot	Heat exchanged with environment	Scalar [W]
info	N.A.	N.A.

Parameters and Dialog Box



Number of reactions

Number of reactions taken into account

Reactionm

A matrix of the reactions simulated in the block. Each row in the reaction matrix represents a separate chemical reaction

Coefficient for calculating heat exchange with the environment [W/(m²K)]

A coefficient for calculating heat exchange with the environment

Total surface [m²]

The external surface at which the heat exchange with the environment takes place. It is assumed that the temperature is constant at the surface.

Environment temperature [K]

Environment temperature

Relaxation factor

A numerical constant used for the computation.

Convergence limit

A numerical constant used for the computation.

Maximal number of iterations

A numerical constant used for the computation.

Initial temperature

The initial value used for the computation.

Ndot limit

The limiting value used for the computation.

Characteristics

Direct Feedthrough	Yes
--------------------	-----

You can look up which property values you will need for using this block in the document [ChemicalMediaData_Dependency_StatesAndProperties.pdf](#)

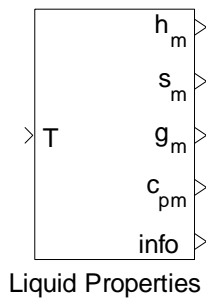
8.8.8 Liquid Properties

Enthalpy, Entropy, free Gibbs Energy and Heat Capacity calculation for liquids.

Library

Thermodynamic States and Properties

Description



The Liquid Properties block determines the

- molar enthalpy h_m ,
- molar entropy s_m ,
- molar free Gibbs energy g_m , and the
- molar heat capacity at constant pressure c_{pm}

at a given temperature.

Species are treated as liquids with temperature-dependent heat capacity.

$$c_{p,f} = A + BT + CT^2 + DT^3$$

The coefficients A , B , C and D could be given in Chemical Media Database. Also limits for the validity of the polynomial can be given there. Outside of the limits heat capacity is assumed to be constant.

To ensure consistency with gas phase calculation, enthalpy of liquid is calculated based to gas enthalpy at a reference temperature (T_{ref}). This means, that enthalpy for fluid is calculated using:

$$h_{m,f} = h_{m,g}(T_{ref}) - h_{fg}(T_{ref}) + \int_{T_{ref}}^T c_{p,m,f} dT$$

Where $h_{fg}(T_{ref})$ is the enthalpy of evaporation at the reference temperature. Same consistency rule applies for calculation of entropy, where:

$$s_{m,f} = s_{m,g}(T_{ref}) - s_{fg}(T_{ref}) + \int_{T_{ref}}^T \frac{c_{p,m,f}}{T} dT$$

is used. The calculation of Gibbs function is based on the entropy and enthalpy as:

$$g_{m,f} = h_{m,f} - s_{m,f}T$$

Note, that there are some blocks, which can only handle liquids (e.g. pump) and some, which can only handle two phase mixtures or gases, but not pure liquids (e. g. Adiabatic Tank). This is stated in the corresponding block documentation.

All coefficients for heat capacity, density, evaporation enthalpy and evaporation entropy as well as the reference temperature (T_{ref}) are given in the Chemical Media Database (see Chapter 4.2)

Data Type Support

Inputs

T	Given temperature	Scalar
---	-------------------	--------

Outputs

h_m	Molar enthalpy	Vector [J/mol]
s_m	Molar entropy	Vector [J/(mol K)]
g_m	Molar free Gibbs energy	Vector [J/mol]
c_{pm}	Molar heat capacity at constant pressure	Vector [J/(mol K)]
info	N.A.	N.A.

Parameters and Dialog Box

Function Block Parameters: Liquid Properties

Liquid Properties (mask)

The Liquid Properties block determines the

- molar enthalpy H_m ,
- molar entropy S_m ,
- molar free Gibbs energy G_m , and the
- molar heat capacity at constant pressure c_{pm}

from a given temperature.

Species are treated as liquids with temperature-independent heat capacity. Enthalpy, Entropy and free Gibbs energy are calculated using reference values at a given reference temperature. To maintain consistency with gas property description the given reference values are adjusted with the property values at the reference temperature.

Parameters

Reference temperatures [K]:

Coefficient of heat capacity in liquid phase :(4xn)

Minimum temperature where Cp-liquid polynomial is valid [K] (1xn)

Maximum temperature where Cp-liquid polynomial is valid [K] (1xn)

Molar enthalpy at reference temperature [J/mol] (1xn)

Molar entropy at reference temperature [J/mol*K] (1xn)

NASA coefficients of gas phase (lower temperature range) (7xn):

NASA coefficients of gas phase (higher temperature range) (7xn)

Reference temperatures [K]

Reference temperature which is used to calculate the enthalpy and entropy of the liquid phase relative to gas phase. This is a vector of the same length as the Chemical Media Data variable Species where each species has its own reference temperature.

Coefficients of heat capacity in liquid phase

Coefficients of heat capacity of each species in the liquid phase

Minimum temperature where c_p -liquid polynomial is valid [K]

Minimum temperature where c_p -liquid polynomial is valid

Maximum temperature where c_p -liquid polynomial is valid [K]

Maximum temperature where c_p -liquid polynomial is valid

Molar enthalpy at reference temperature [J/mol]

Molar enthalpy of each species at the specified reference temperature

Molar entropy at reference temperature [J/(mol K)]

Molar entropy of each species at the specified reference temperature

NASA coefficients of gas phase (lower temperature range)

The coefficients used for temperatures at lower temperature range.

NASA coefficients of gas phase (higher temperature range)

The coefficients used for temperatures at higher temperature range.

Characteristics

Direct Feedthrough	Yes
--------------------	-----

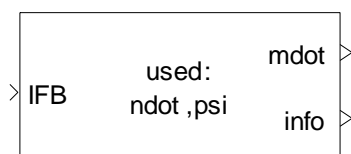
You can look up which property values you will need for using this block in the document [ChemicalMediaData_Dependency_StatesAndProperties.pdf](#)

8.8.9 Mass Flow Rate

Mass flow rate calculation.

Library

Thermodynamic States and Properties

Description

Mass Flow Rate

The Mass Flow Rate block calculates the mass flow rate (mdot) in g/s of a given media flow. The mass flow rate is determined from the total molar flow (ndot) and the molar fractions (psi) provided by the incoming incomplete flow bus (IFB).

Data Type Support

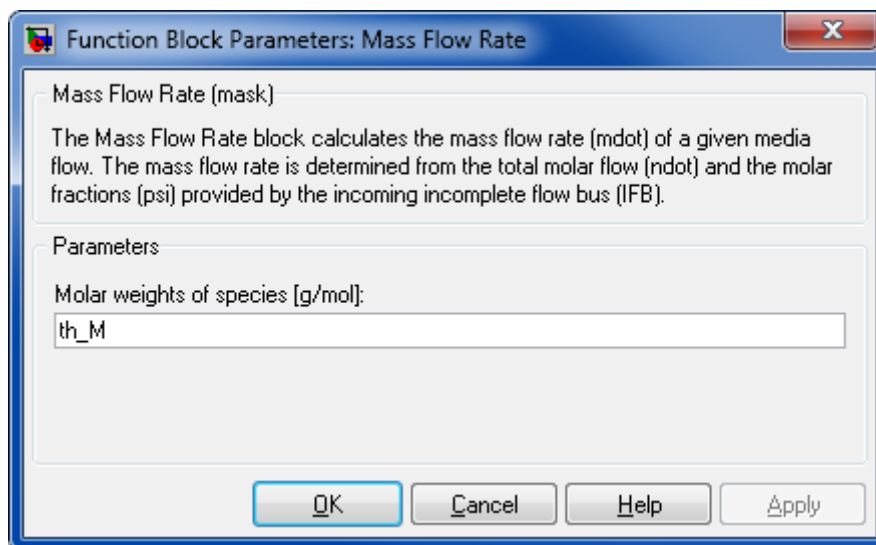
Inputs

IFB	Incoming flow	Incomplete flow bus
-----	---------------	---------------------

Outputs

Mdot	Calculated mass flow rate	Scalar [g/s]
info	N.A.	N.A.

Parameters and Dialog Box



Molar weights of species [g/mol]

A row vector containing the molar weights. This is a vector of the same length as the Species vector.

Characteristics

Direct Feedthrough	Yes
--------------------	-----

You can look up which property values you will need for using this block in the document [ChemicalMediaData_Dependency_StatesAndProperties.pdf](#)

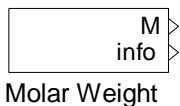
8.8.10 Molar Weight

Molar weight

Library

Thermodynamic States and Properties

Description



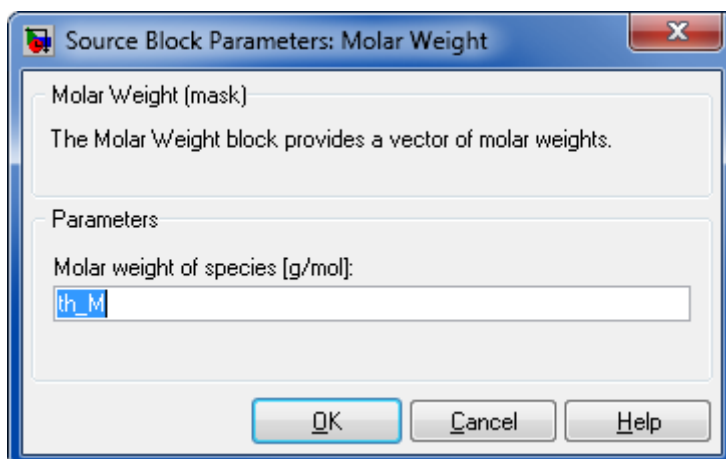
The Molar Weight block provides a vector of molar weights for active species (chosen in Model Setup block).

Data Type Support

Outputs

M	Vector of molar weights	Vector [g/mol]
info	N.A.	N.A.

Parameters and Dialog Box



Molar weight of species [g/mol]

A row vector containing the molar weights. This is a vector of the same length as the Species vector.

Characteristics

Direct Feedthrough	Yes
--------------------	-----

You can look up which property values you will need for using this block in the document [ChemicalMediaData Dependency StatesAndProperties.pdf](#)

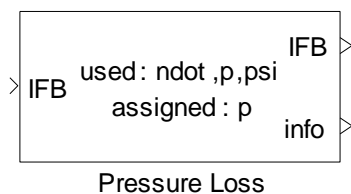
8.8.11 Pressure Loss

Pressure loss calculation

Library

Thermodynamic States and Properties

Description



The Pressure Loss block realizes a decrease in pressure and outputs an incomplete flow bus containing the new pressure.

$$\Delta p = k \dot{m}^2$$

Note: The thermodynamic state of the outgoing flow may be inconsistent.

Data Type Support

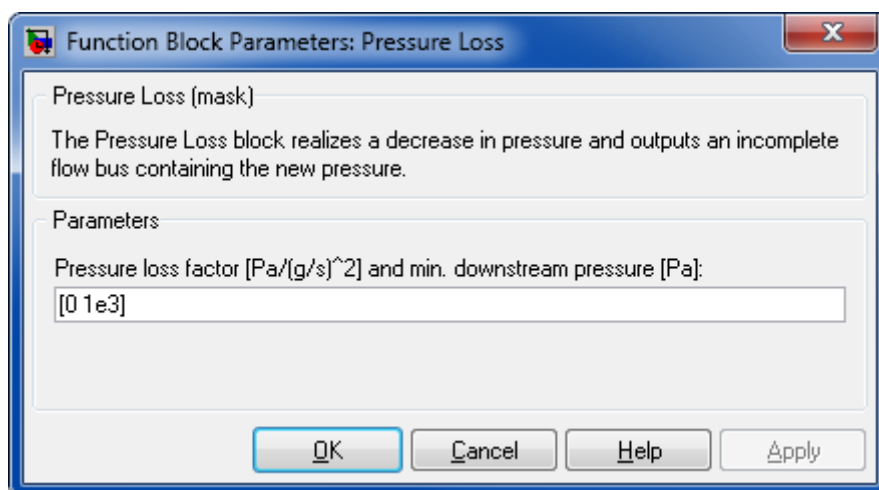
Inputs

IFB	Incoming flow	Incomplete flow bus
-----	---------------	---------------------

Outputs

IFB	Outgoing flow	Incomplete flow bus
info	N.A.	N.A.

Parameters and Dialog Box



Pressure Loss Factor $[\text{Pa}/(\text{g/s})^2]$ and Minimum Downstream Pressure $[\text{Pa}]$

The factor k and the pressure p are given as a vector $[k, p]$. The pressure loss is proportional to the mass flow rate squared ($\Delta p = k \dot{m}^2$).

Characteristics

Direct Feedthrough	Yes
--------------------	-----

You can look up which property values you will need for using this block in the document [ChemicalMediaData_Dependency_StatesAndProperties.pdf](#)

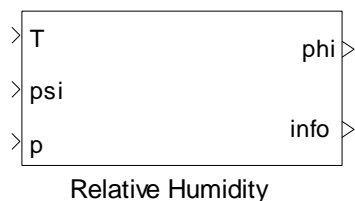
8.8.12 Relative Humidity

Relative humidity calculation

Library

Thermodynamic States and Properties

Description



The Relative Humidity block calculates the relative humidity (ϕ) of a mixture with concentrations (ψ_i) at a given temperature (T) and pressure (p).

Relative Humidity is the ratio of pressure to saturation pressure at a given temperature. The calculation of the saturation pressure requires coefficients for the Antoine equation for all species, in the exact same manner as the Saturation Pressure block.

Note: Usually relative humidity is defined for water only, but this block handles generalized "relative humidity" for all defined species.

Data Type Support

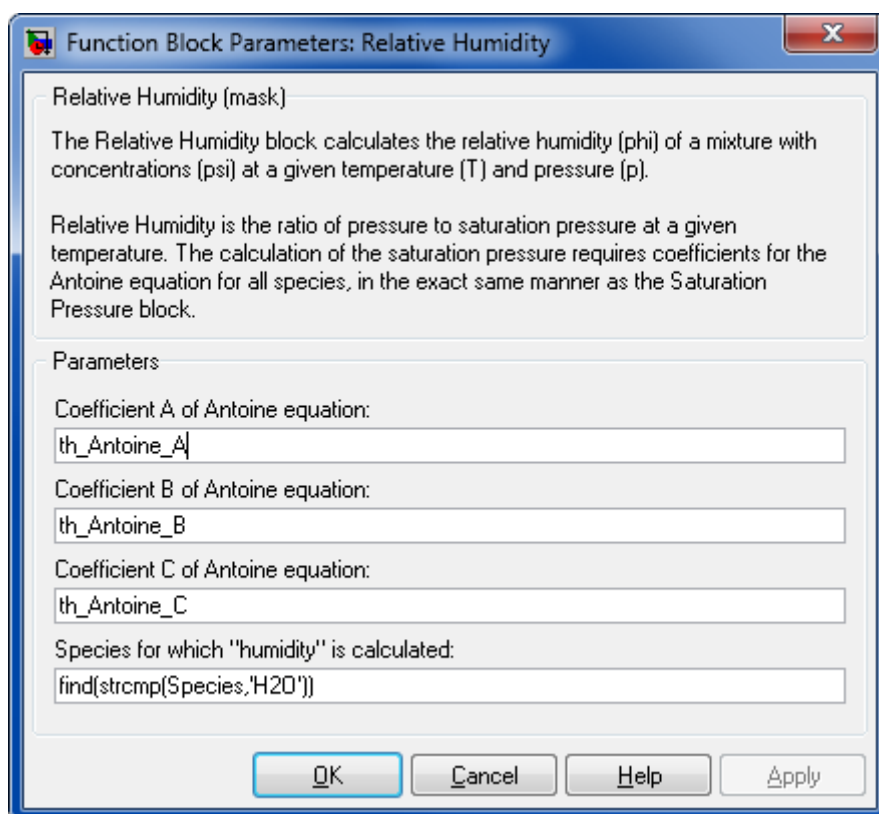
Inputs

T	Given temperature	Scalar [K]
psi	Molar fractions of compounds	Vector [mol/mol]
p	Given pressure	Scalar [Pa]

Outputs

phi	Relative humidity of mixture	Scalar [-]
info	N.A.	N.A.

Parameters and Dialog Box



Coefficient A in Antoine Equation

A vector of "A" coefficients used by the Antoine equation.

Coefficient "B" in Antoine Equation

A vector of B coefficients used by the Antoine equation

Coefficient "C" in Antoine Equation

A vector of C coefficients used by the Antoine equation

Species for which "Humidity" is calculated

The index of the species in the Antoine coefficient row vectors

Characteristics

Direct Feedthrough	Yes
--------------------	-----

You can look up which property values you will need for using this block in the document [ChemicalMediaData_Dependency_StatesAndProperties.pdf](#)

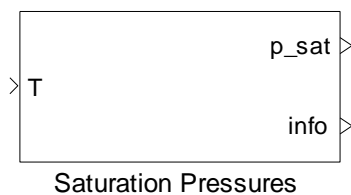
8.8.13 Saturation Pressures

Saturation Pressure Calculation

Library

Thermodynamic States and Properties

Description



The Saturation Pressures block determines the saturation pressures from a given temperature for a set of compounds. This is done with the help of the Antoine equation approximating the saturation vapor pressure curve between the triple point and the critical point for a certain fluid.

The formula reads:

$$p_{sat} = 1e5 \times 10^{A - \frac{B}{C+T}}$$

with fluid-dependent coefficients A, B, and C. T in [K], p_{sat} in [Pa].

The coefficients are assumed to be independent of temperature. It is assumed that the approximation of the saturation pressure based on the chosen constants A, B and C is sufficiently accurate for the intended field of application.

A parallel computation for a set of species will be performed if the Antoine coefficients are given as vectors.

If the default values from Thermolib Chemical Media Database are used, then the block I/O is with T in [K] and p_{sat} is in [Pa].

Data Type Support

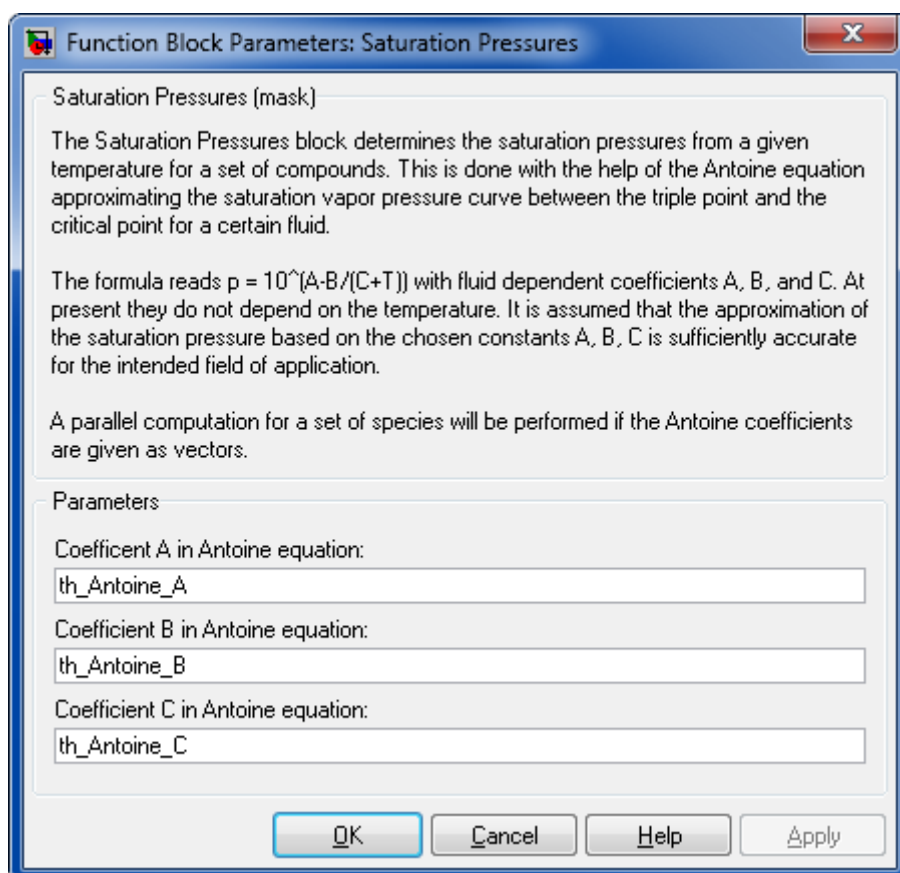
Inputs

T	Given temperature	Scalar [K]
---	-------------------	------------

Outputs

p_sat	Saturation pressures of compounds	Vector [Pa]
info	N.A.	N.A.

Parameters and Dialog Box



Coefficient A in Antoine equation

A vector of "A" coefficients used by the Antoine equation

Coefficient B in Antoine equation

A vector of "B" coefficients used by the Antoine equation

Coefficient C in Antoine equation

A vector of "C" coefficients used by the Antoine equation

Characteristics

Direct Feedthrough	Yes
--------------------	-----

You can look up which property values you will need for using this block in the document [ChemicalMediaData Dependency StatesAndProperties.pdf](#)

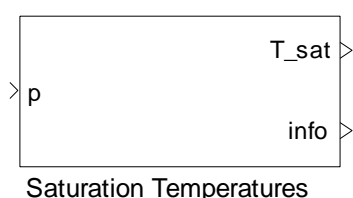
8.8.14 Saturation Temperatures

Saturation Temperatures Calculation

Library

Thermodynamic States and Properties

Description



The Saturation Temperatures block determines the saturation temperatures from a given pressure for a set of compounds. This is done with the help of the Antoine equation approximating the saturation vapor pressure curve between the triple point and the critical point for a certain fluid.

The formula reads

$$T_{sat} = \frac{B}{A - \log_{10}(p \times 1e(-5))} - C$$

with fluid dependent coefficients A, B, and C. T_{sat} in [K] and p in [Pa]. The coefficients A, B, C do not depend on the pressure. It is assumed that the approximation of the saturation temperature based on the chosen constants A, B and C is sufficiently accurate for the intended field of application.

A parallel computation for a set of species will be performed if the Antoine coefficients are given as vectors.

If the default values from Thermolib species database are used, then the block I/O is with T in [K] and p_sat is in [Pa]. Note that the coefficients A, B, C in the database are for T in [K] and p in [pa].

Data Type Support

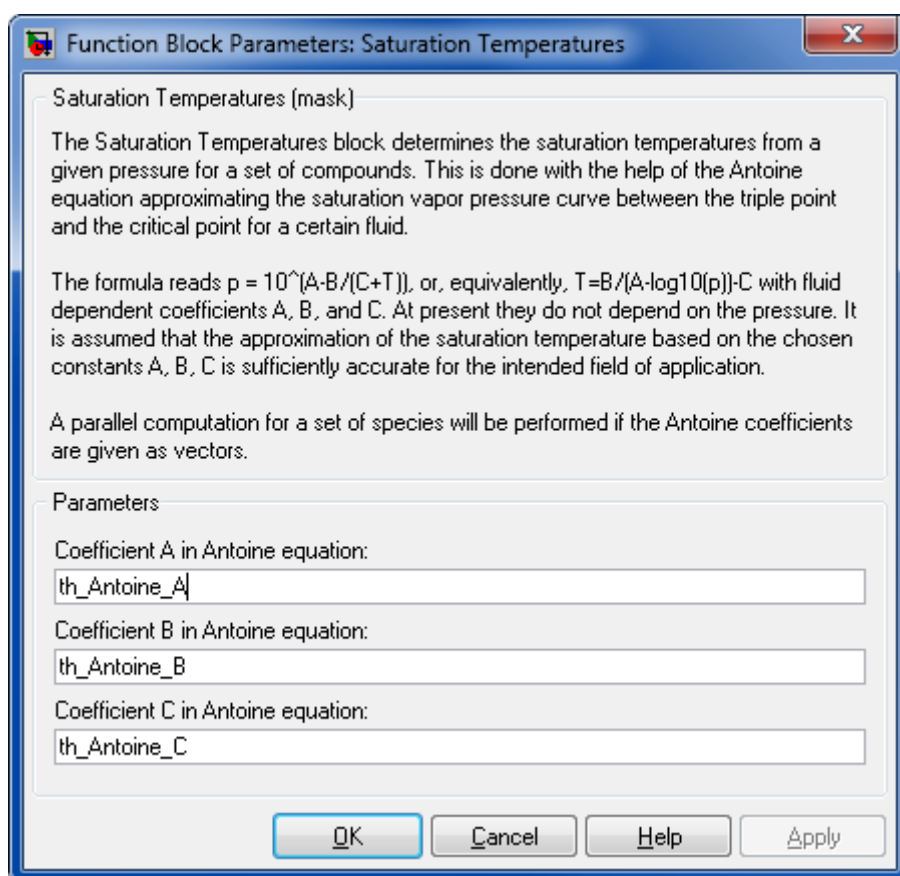
Inputs

p	Given pressure	Scalar [Pa]
---	----------------	-------------

Outputs

T_sat	Saturation temperature of compounds	Vector [K]
info	N.A.	N.A.

Parameters and Dialog Box



Coefficient A in Antoine equation

A vector of "A" coefficients used by the Antoine equation.

Coefficient B in Antoine equation

A vector of "B" coefficients used by the Antoine equation

Coefficient C in Antoine equation

A vector of "C" coefficients used by the Antoine equation

Characteristics

Direct Feedthrough	Yes
--------------------	-----

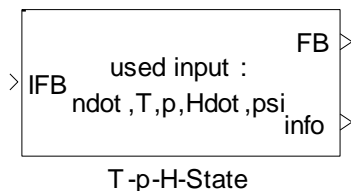
You can look up which property values you will need for using this block in the document [ChemicalMediaData_Dependency_StatesAndProperties.pdf](#)

8.8.15 T-p-H-State

Consistent thermodynamic state calculation from temperature, pressure and enthalpy

Library

Thermodynamic States and Properties

Description

The T-p-H-State block determines a consistent thermodynamic state for either a pure substance or a mixture flow from its temperature, pressure and enthalpy. Temperature and pressure are used to calculate vapor fractions of mixture, but in case of pure substances, temperature and enthalpy are used.

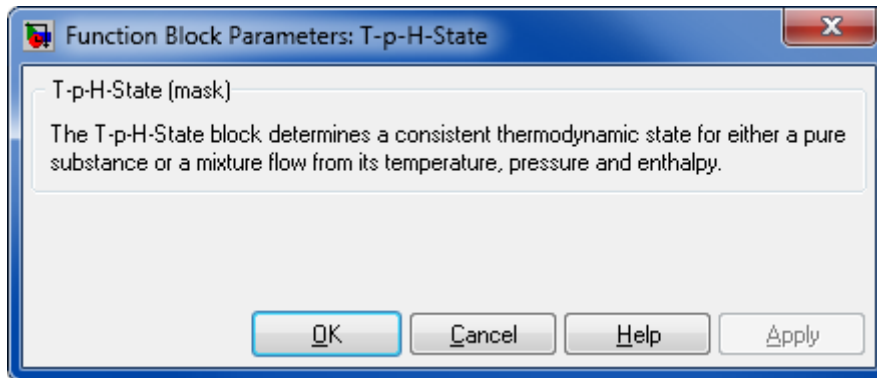
Data Type Support**Inputs**

IFB	Incoming flow	Incomplete flow bus
-----	---------------	---------------------

Outputs

FB	Outgoing flow	Complete flow bus
info	N.A.	N.A.

Parameters and Dialog Box



Characteristics

Direct Feedthrough	Yes
--------------------	-----

You can look up which property values you will need for using this block in the document [ChemicalMediaData_Dependency_StatesAndProperties.pdf](#)

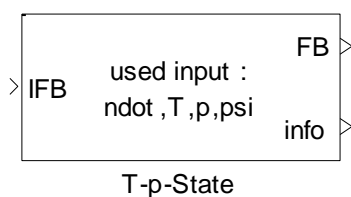
8.8.16 T-p-State

Complete flow bus using temperature and pressure.

Library

Thermodynamic States and Properties

Description



The T-p-State block completes the outgoing flow bus (FB) from temperature and pressures, i.e., all properties of the outgoing flow are consistent.

For pure substances in two phase region cannot be calculated because T and p are not independent to each other, i.e., x is either 0 if the substance is liquid or 1 if it is gaseous.

Data Type Support

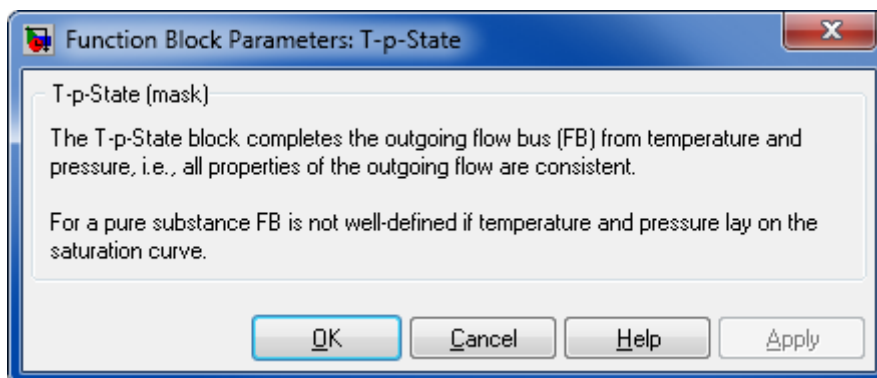
Inputs

IFB	Incoming flow	Incomplete flow bus
-----	---------------	---------------------

Outputs

FB	Outgoing flow	Complete flow bus
info	N.A.	N.A.

Parameters and Dialog Box



Characteristics

Direct Feedthrough	Yes
--------------------	-----

You can look up which property values you will need for using this block in the document [ChemicalMediaData_Dependency_StatesAndProperties.pdf](#)

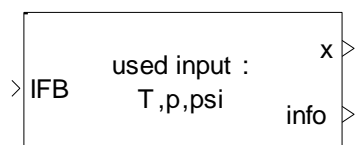
8.8.17 T-p-Vapor Fractions

Calculation of vapor mass fractions

Library

Thermodynamic States and Properties

Description



T-p-Vapor Fractions

The T-p-Vapor Fractions block determines the fluid's vapor fractions (x) from temperature and pressure.

For pure substances in two phase region cannot be calculated because T and p are not independent to each other, i.e., x is either 0 if the substance is liquid or 1 if it is gaseous.

Data Type Support

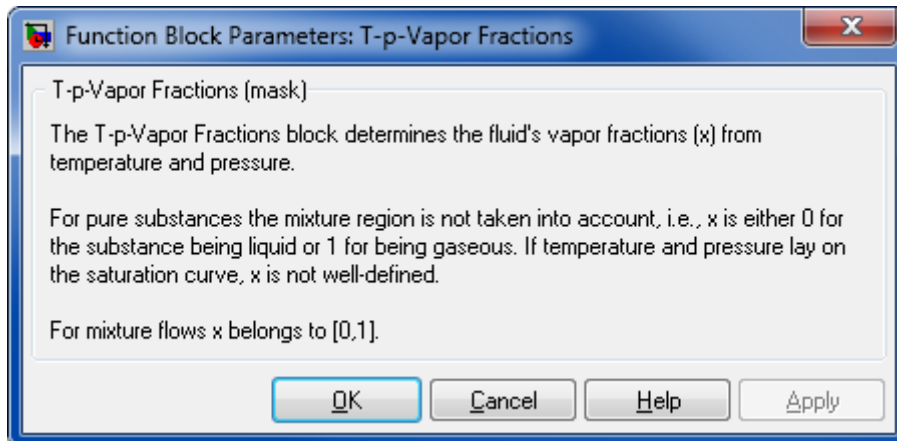
Inputs

IFB	Incoming flow	Incomplete flow bus
-----	---------------	---------------------

Outputs

x	Vapor fractions of compounds	Vector [mol/mol]
info	N.A.	N.A.

Parameters and Dialog Box



Characteristics

Direct Feedthrough	Yes
--------------------	-----

You can look up which property values you will need for using this block in the document [ChemicalMediaData Dependency StatesAndProperties.pdf](#)

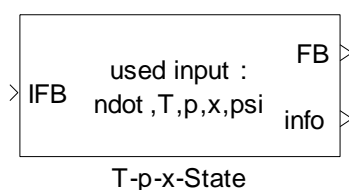
8.8.18 T-p-x-State

Consistent thermodynamic state calculated from temperature, pressure and vapor fractions.

Library

Thermodynamic States and Properties

Description



The T-p-x-State block determines a consistent thermodynamic state for either a pure substance or a mixture flow from its temperature, pressure and vapor fractions.

Data Type Support

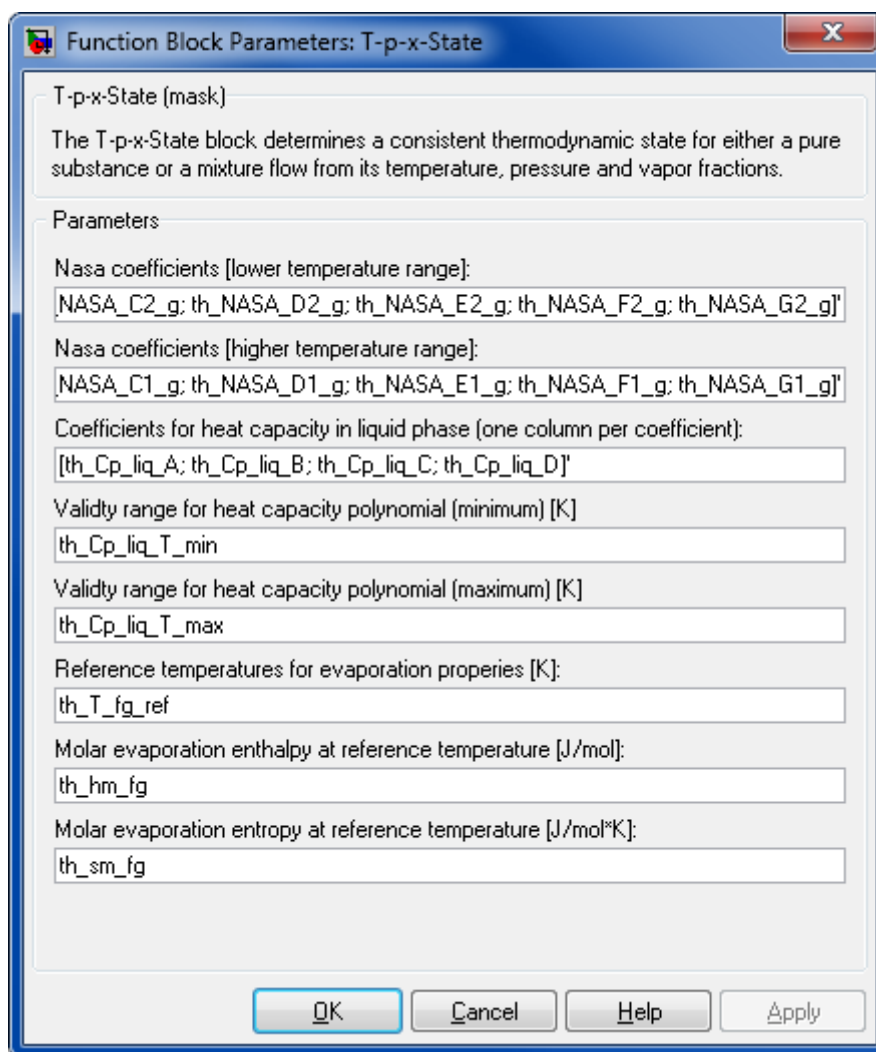
Inputs

IFB	Incoming flow	Incomplete flow bus
-----	---------------	---------------------

Outputs

FB	Outgoing flow	Complete flow bus
info	N.A.	N.A.

Parameters and Dialog Box



NASA coefficients (lower temperature range):

The coefficients used for temperatures at lower temperature range.

NASA coefficients (higher temperature range):

The coefficients used for temperatures at higher temperature range.

Coefficient of heat capacity in liquid phase

Coefficients of heat capacity of each species in the liquid phase

Validity range for heat capacity polynomial (minimum) [K]

Minimum temperature where c_p -liquid polynomial is valid

Validity range for heat capacity polynomial (maximum) [K]

Maximum temperature where c_p -liquid polynomial is valid

Reference temperatures [K]

Reference temperatures to calculate the entropy and entropy of the liquid mixture.
This is a vector of the same length as the Chemical Media Data variable Species
where each species has its own reference temperature.

Molar evaporation enthalpy at reference temperature [J/mol]

Molar evaporation enthalpy of each species at the specified reference temperature

Molar evaporation entropy at reference temperature [J/(mol K)]

Molar evaporation entropy of each species at the specified reference temperature

Characteristics

Direct Feedthrough	Yes
--------------------	-----

You can look up which property values you will need for using this block in the document [ChemicalMediaData_Dependency_StatesAndProperties.pdf](#)

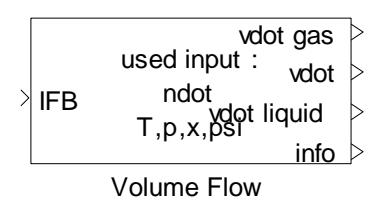
8.8.19 Volume Flow

Volume flow calculation

Library

Thermodynamic States and Properties

Description



The Volume Flow block determines the volume flow [m³/s] of a given media flow in both the gaseous and the liquid phase as well as for overall flow. From these, the overall volume flow is computed depending on the respective vapor and molar fractions of the species.

Data Type Support

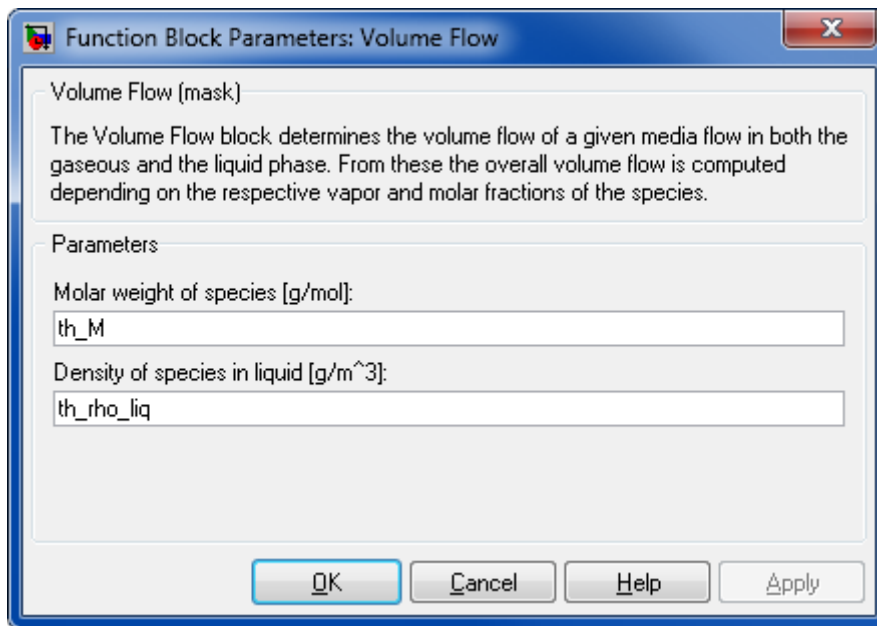
Inputs

IFB	Incoming flow	Incomplete flow bus
-----	---------------	---------------------

Outputs

vdot gas	Volume flow for the gaseous portion	Scalar [m ³ /s]
vdot	Overall volume flow	Scalar [m ³ /s]
vdot liquid	Volume flow for the liquid portion	Scalar [m ³ /s]
info	N.A.	N.A.

Parameters and Dialog Box



Molar Weight of Species [g/mol]

A row vector containing the molar weights. This is a vector of the same length as the Species vector.

Density of Species in Liquid [g/m³]

A vector containing the species' density in the liquid phase

Characteristics

Direct Feedthrough	Yes
--------------------	-----

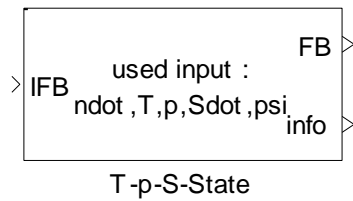
You can look up which property values you will need for using this block in the document [ChemicalMediaData_Dependency_StatesAndProperties.pdf](#)

8.8.20 T-p-S-State

Library

Thermodynamic States and Properties

Description



The T-p-S-State block determines a consistent thermodynamic state for either a pure substance or a mixture flow from its temperature, pressure and entropy.

Temperature and pressure are used to calculate vapor fractions of mixture, but in case of pure substances, temperature and enthalpy are used.

Data Type Support

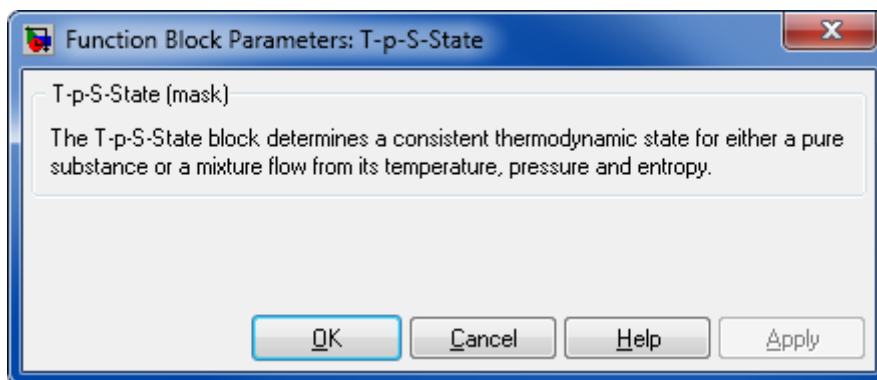
Inputs

IFB	Incoming flow	Incomplete flow bus
-----	---------------	---------------------

Outputs

FB	Outgoing flow	Flow bus
info	N.A.	N.A.

Parameters and Dialog Box



Characteristics

Direct Feedthrough	Yes
--------------------	-----

You can look up which property values you will need for using this block in the document [ChemicalMediaData_Dependency_StatesAndProperties.pdf](#)

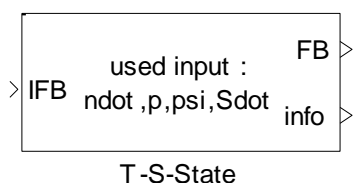
8.8.21 T-S-State

Complete flow bus using temperature and entropy.

Library

Thermodynamic States and Properties

Description



The T-S-State completes the outgoing flow bus (FB) using temperature and entropy. All properties of the outgoing flow are consistent.

Data Type Support

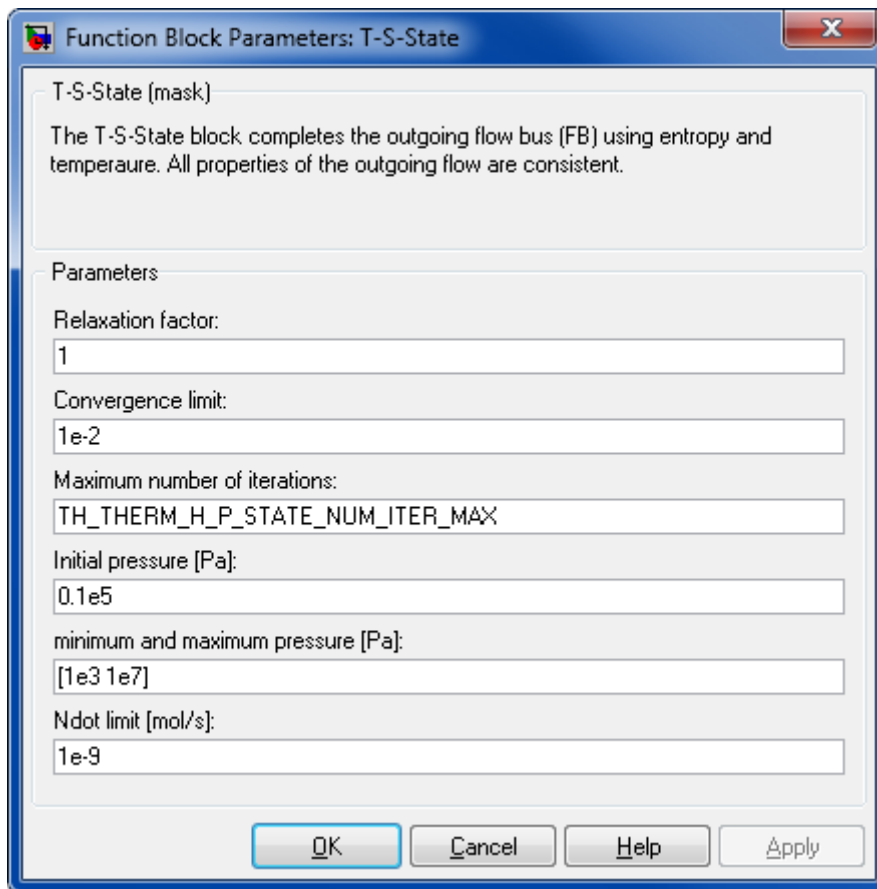
Inputs

IFB	Incoming flow	Incomplete flow bus
-----	---------------	---------------------

Outputs

FB	Outgoing flow	Complete flow bus
info	N.A.	N.A.

Parameters and Dialog Box

**Relaxation factor**

A factor that is used to iterate pressure. Default value is 1, the higher this value, the more aggressive the iteration behaves. This may lead to instable iterations. The lower the value the less aggressive the iteration behaves. This results in slower convergence.

Convergence limit

A numerical constant used for the computation.

Maximal number of iterations

A numerical constant used for the computation.

Initial pressure [Pa]

The initial value used for the computation.

Minimum and maximum pressure [Pa]

The valid range of pressure.

Ndot limit [mol/s]

A limiting value used for the computation. If ndot is less than “ndot limit”, the block does not iterate to calculate a valid pressure. The output flow bus would be same as input flow bus.

Characteristics

Direct Feedthrough	Yes
--------------------	-----

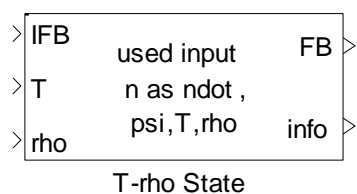
You can look up which property values you will need for using this block in the document [ChemicalMediaData_Dependency_StatesAndProperties.pdf](#)

8.8.22 T-rho State

Complete flow bus using temperature and density.

Library

Thermodynamic States and Properties

Description

The T-rho-State completes the outgoing flow bus (FB) using temperature and density. All properties of the outgoing flow are consistent.

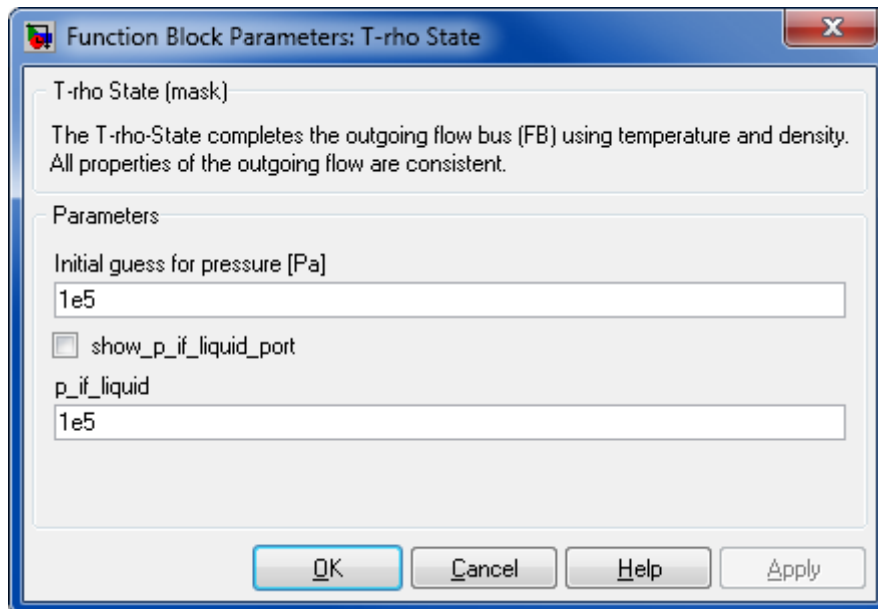
Data Type Support**Inputs**

IFB	Incoming flow	Incomplete flow bus
rho	Density	[g/m ³]

Outputs

FB	Outgoing flow	Complete flow bus
info	N.A.	N.A.

Parameters and Dialog Box



Initial pressure [Pa]

The initial value used for the computation

Show p_if_liquid_port

Show or hide p_if_liquid_input port

p_if_liquid [Pa]

Value of pressure if state is liquid

Characteristics

Direct Feedthrough	Yes
--------------------	-----

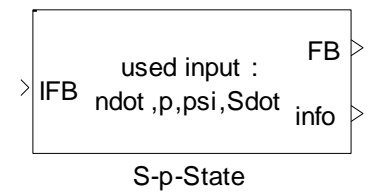
You can look up which property values you will need for using this block in the document [ChemicalMediaData_Dependency_StatesAndProperties.pdf](#)

8.8.23 S-p-State

Library

Thermodynamic States and Properties

Description



The S-p-State block completes the outgoing flow bus (FB) using entropy and pressure. All properties of the outgoing flow are consistent.

Data Type Support

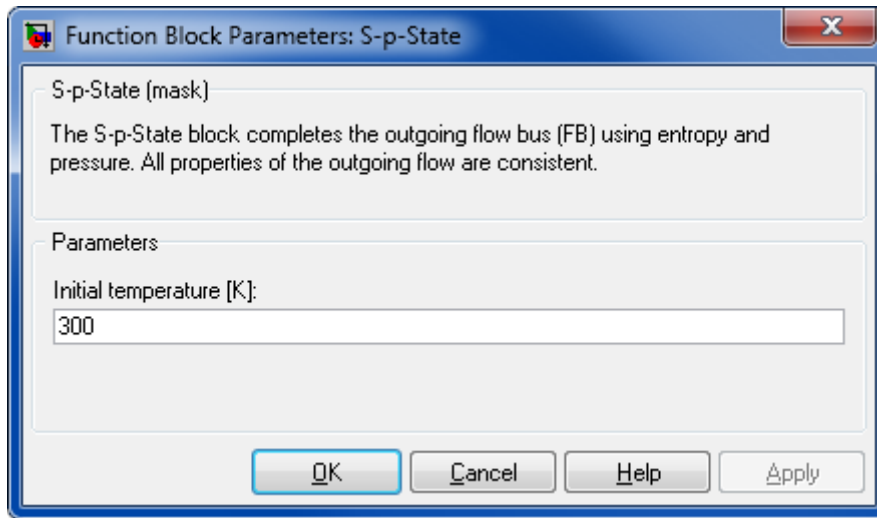
Inputs

IFB	Incoming flow	Incomplete flow bus
-----	---------------	---------------------

Outputs

FB	Outgoing flow	Flow bus
info	N.A.	N.A.

Parameters and Dialog Box



Initial temperature [K]

The initial value used for the computation.

Characteristics

Direct Feedthrough	Yes
--------------------	-----

You can look up which property values you will need for using this block in the document [ChemicalMediaData_Dependency_StatesAndProperties.pdf](#)

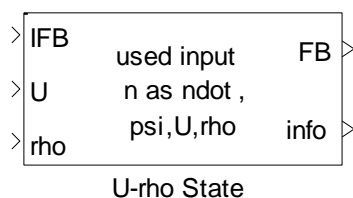
8.8.24 U-rho State

Compute flow bus using internal energy and density.

Library

Thermodynamic States and Properties

Description



The U-rho-State calculates the outgoing flow bus (FB) using internal energy and density given at input ports. All properties of the outgoing flow are consistent. Note that, in this case, \dot{n} in flow bus is used as n .

Data Type Support

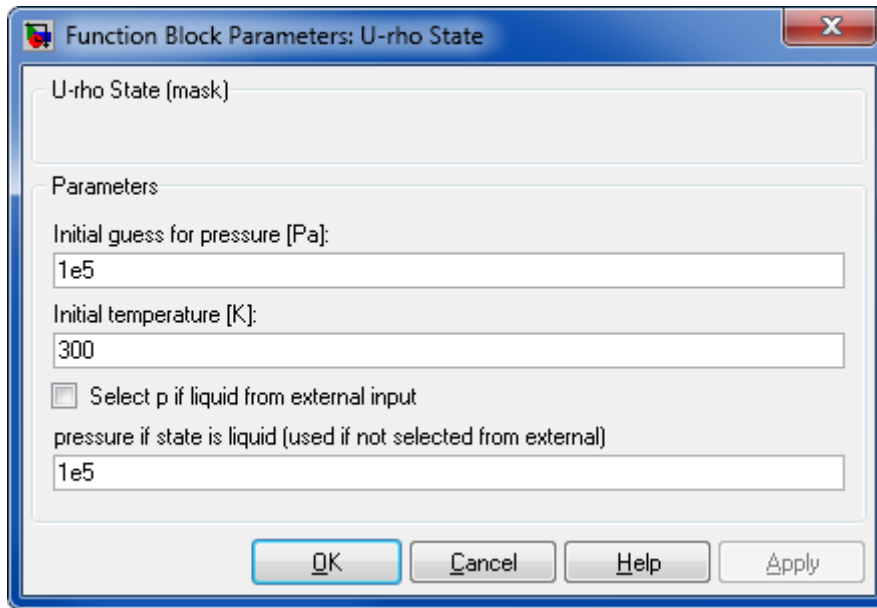
Inputs

IFB	Incoming flow	flow bus
U	Internal energy	[Joule]
rho	Density	[g/m ³]

Outputs

FB	Outgoing flow	Complete flow bus
	Bus with internal information e. g.:	
	<i>error_code</i> ... vector with error codes	
info	<i>p_iter</i> ... number of pressure iterations	Simulink bus
	<i>T_iter</i> ... number of temperature iterations	

Parameters and Dialog Box

**Initial guess for pressure [Pa]**

The initial value used for the computation.

Initial guess for temperature [T]

The initial value used for the computation.

Show p_if_liquid_port

Show or hide p_if_liquid_input port

p_if_liquid [Pa]

Value of pressure if state is liquid

Characteristics

Direct Feedthrough	Yes
--------------------	-----

You can look up which property values you will need for using this block in the document [ChemicalMediaData_Dependency_StatesAndProperties.pdf](#)

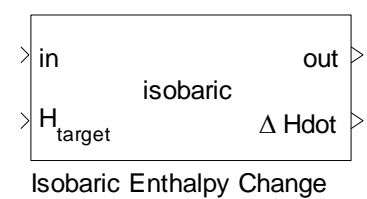
8.9 THERMODYNAMIC STATE CHANGES

8.9.1 Isobaric Enthalpy Change

Library

Thermodynamic State Changes

Description



This block calculates the new state in an isobaric process. The enthalpy of the target state is given at input " H_{target} ". The difference in enthalpy between input and output is calculated at output " ΔH_{dot} ".

Data Type Support

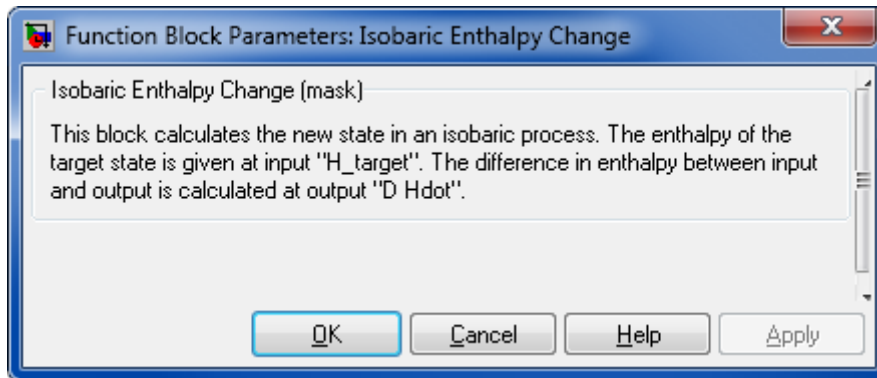
Inputs

in	Incoming flow	flow bus
H_{target}	Target enthalpy	[W]

Outputs

out	Outgoing flow	flow bus
ΔH_{dot}	Enthalpy difference	[W]

Parameters and Dialog Box

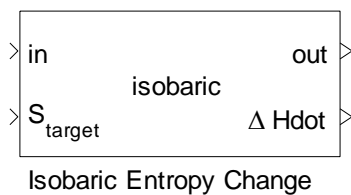


8.9.2 Isobaric Entropy Change

Library

Thermodynamic State Changes

Description



This block calculates the new state in an isobaric process. The entropy of the target state is given at input "S_target". The difference in enthalpy between input and output is calculated at output "Δ Hdot ".

Data Type Support

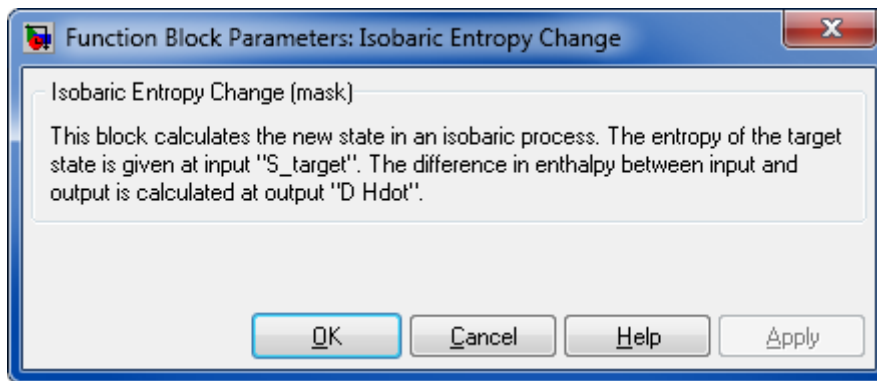
Inputs

in	Incoming flow	flow bus
S-target	Target entropy	[W/K]

Outputs

out	Outgoing flow	flow bus
Δ Hdot	Enthalpy defference	[W]

Parameters and Dialog Box

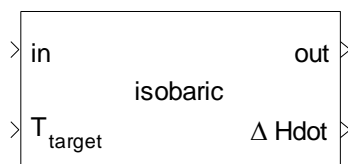


8.9.3 Isobaric Temperature Change

Library

Thermodynamic State Changes

Description



Isobaric Temperature Change

This block calculates the new state in an isobaric process. The temperature of the target state is given at input "T_target". The difference in enthalpy between input and output is calculated at output "Δ Hdot".

Data Type Support

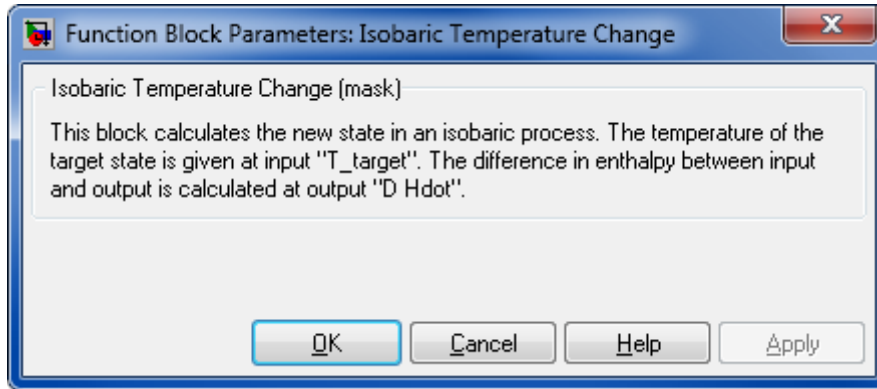
Inputs

in	Incoming flow	flow bus
T-target	Target temperature	[K]

Outputs

out	Outgoing flow	flow bus
$\Delta \dot{H}$	Enthalpy difference	[W]

Parameters and Dialog Box

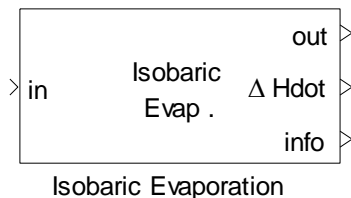


8.9.4 Isobaric Evaporation

Library

Thermodynamic State Changes

Description



This block simulates a complete isobaric evaporation/condensation process.

If you choose "evaporation" the output state will be a saturated vapor. But if the input flow is already gas, it remains unchanged. If you choose "condensation" the output state will be saturated liquid. But if the input is already liquid, it remains unchanged. The enthalpy difference, which is necessary to perform this evaporation/condensation will be given at output " $\Delta \dot{H}$ ".

Note: The block works only for pure substances!

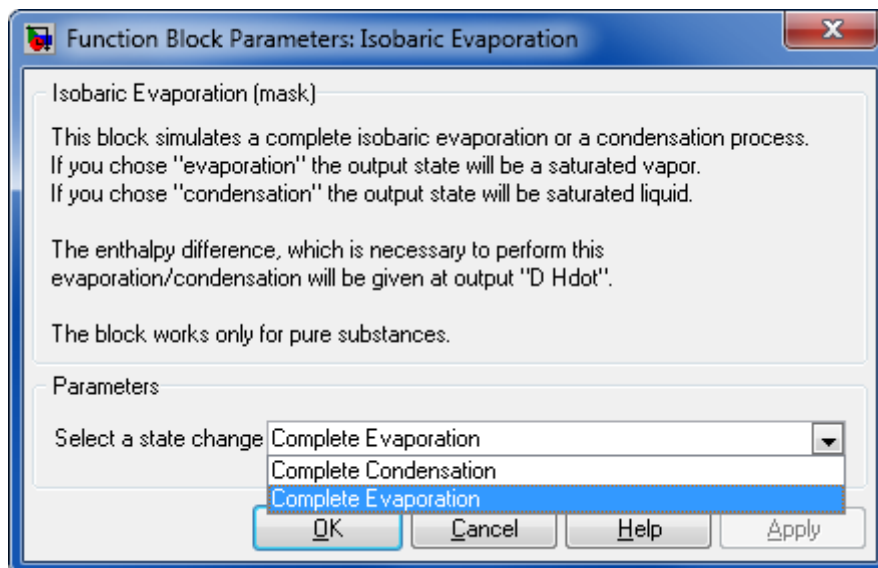
Data Type Support

Inputs

in	Incoming flow	flow bus
----	---------------	----------

Outputs

out	Outgoing flow	flow bus
ΔH_{dot}	Enthalpy difference	[W]
info	-	[-]

Parameters and Dialog Box**State change**

Select "Complete Evaporation" then the block functions as a evaporator. If you select "Complete Condensation" the block functions as condenser.

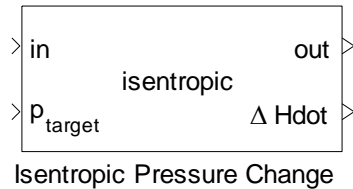
8.9.5 Isobaric Condensation

The Isobaric Condensation block is a link to the Isobaric Evaporation block.

8.9.6 Isentropic Pressure Change**Library**

Thermodynamic State Changes

Description



This block calculates the new state in an isentropic process. The pressure of the target state is given at input "p_target". The difference in enthalpy between input and output is calculated at output "Δ Hdot".

Data Type Support

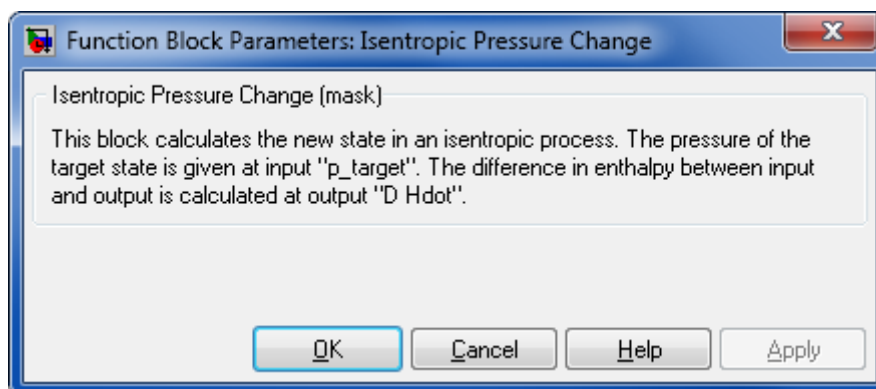
Inputs

in	Incoming flow	flow bus
p-target	Target pressure	[pa]

Outputs

out	Outgoing flow	flow bus
Δ Hdot	Enthalpy difference	[W]

Parameters and Dialog Box

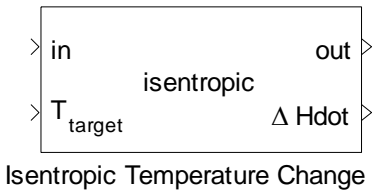


8.9.7 Isentropic Temperature Change

Library

Thermodynamic State Changes

Description



This block calculates the new state in an isentropic process. The temperature of the target state is given at input "T_target". The difference in enthalpy between input and output is calculated at output "Δ Hdot".

Data Type Support

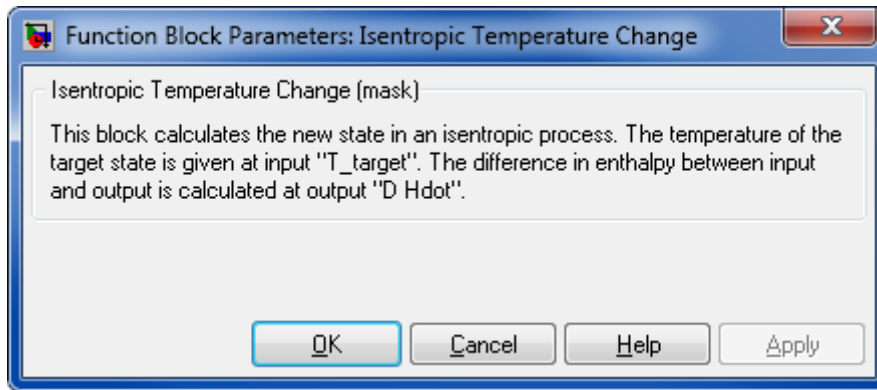
Inputs

in	Incoming flow	flow bus
T-target	Target temperature	[K]

Outputs

out	Outgoing flow	flow bus
Δ Hdot	Enthalpy difference	[W]

Parameters and Dialog Box

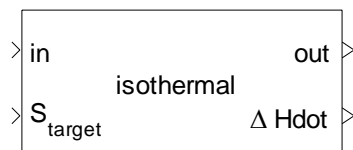


8.9.8 Isothermal Entropy Change

Library

Thermodynamic State Changes

Description



Isothermal Entropy Change

This block calculates the new state in an isothermal process. The entropy of the target state is given at input "S_target". The difference in enthalpy between input and output is calculated at output "Δ Hdot".

Data Type Support

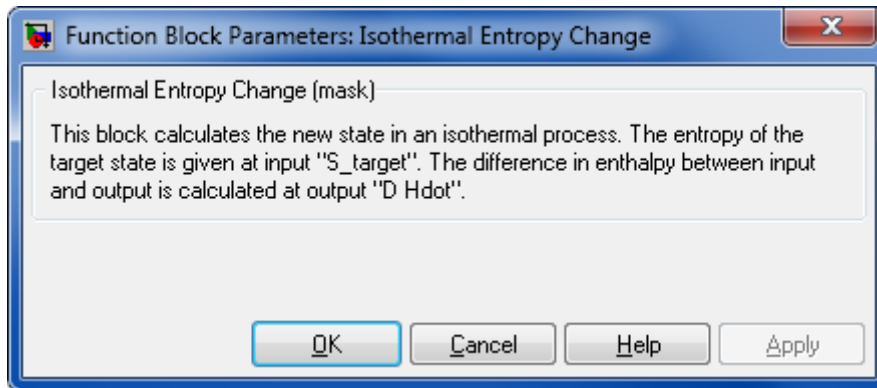
Inputs

in	Incoming flow	flow bus
S-target	Target entropy	[W/K]

Outputs

out	Outgoing flow	flow bus
Δ Hdot	Enthalpy difference	[W]

Parameters and Dialog Box

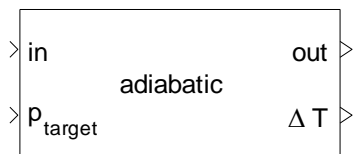


8.9.9 Adiabatic Pressure Change

Library

Thermodynamic State Changes

Description



Adiabatic Pressure Change

This block calculates the new state in an adiabatic process. The pressure of the target state is given at input "p_target". The temperature difference between input and output is calculated at output "D T".

Data Type Support

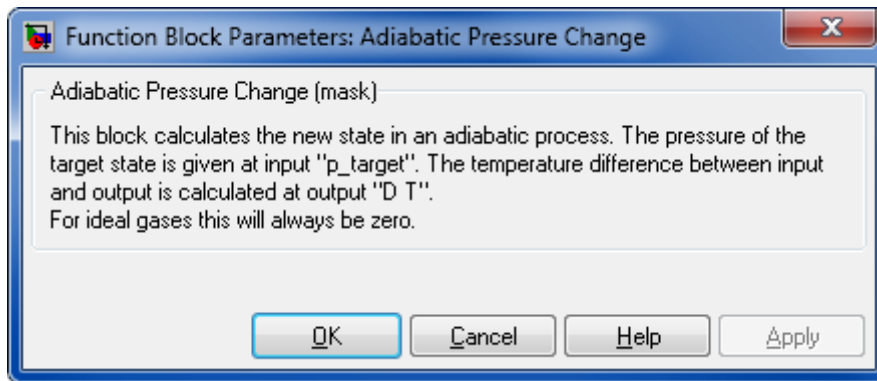
Inputs

in	Incoming flow	flow bus
p-target	Target pressure	[pa]

Outputs

out	Outgoing flow	flow bus
ΔT	Temperature difference	[K]

Parameters and Dialog Box



8.10 COMPATIBILITY BLOCKS

This section contains documentation for blocks of former Thermolib/FClib versions, which are still available at Thermolib\Compatibility subfolders of Thermolib for compatibility purpose.

8.10.1 FClib - 4-1-8-4

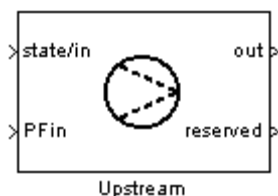
8.10.1.1 Upstream

Upstream pressure control.

Library

Components / Hydraulic with Pressure Feedback

Description



The Downstream and Upstream blocks should be used together in a hydraulic system with pressure feedback. In combination they act like a pressure controller.

Pressure-feedback information from the Downstream block is used by the Upstream block, which adjusts the amount of mass flow passing through it so that the difference between the reference pressure (P_{ref}) and the current pressure (P_{is}) approaches zero. In other words, the Upstream block corrects the overall mass stream in such a way that the pressure reference set in the Downstream block is reached.

The maximum molar flow that the Upstream block can output, is defined by the user in the mask.

Data Type Support

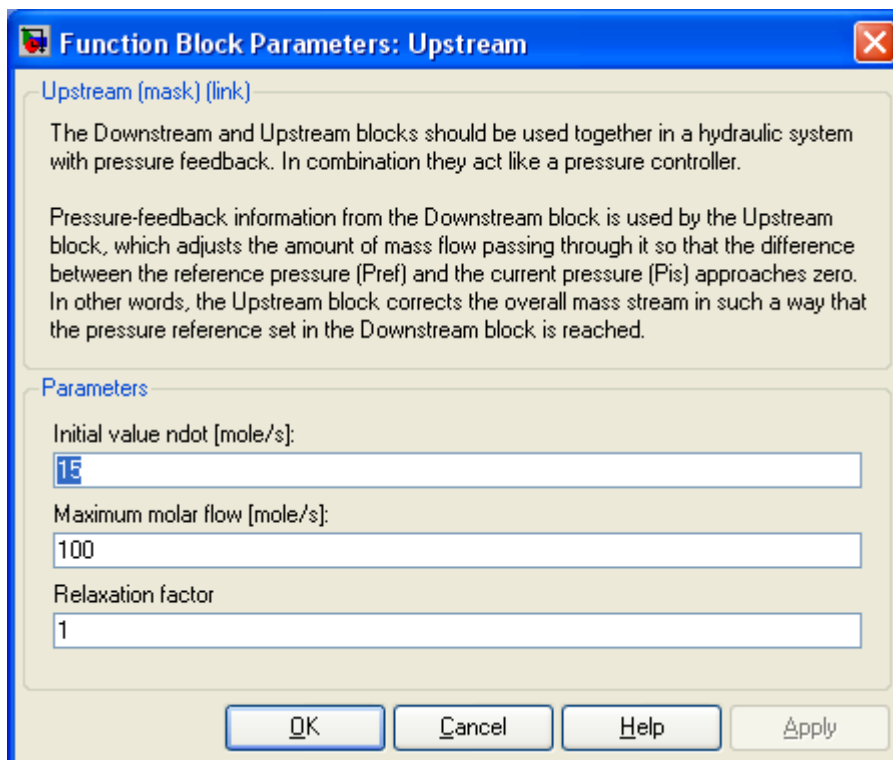
Inputs

state/in	Incoming flow	Flow bus
PFin	Pressure feedback for output	4-doubles vector

Outputs

out	Outgoing flow	Flow bus
reserved	N.A.	N.A.

Parameters and Dialog Box



Initial value Ndot

Initial value ndot.

Maximum molar flow [mol/s]

Maximum molar flow.

Relaxation factor

A factor, that is used to iterate pressure. Default value is 1, the higher this value, the more aggressive the iteration behaves. This may lead to instable iterations. The lower the value the less aggressive the iteration behaves. This results in slower convergence.

Characteristics

Direct Feedthrough	Yes
--------------------	-----

9 COMMAND REFERENCE

9.1 THERMOLIB_SETUP

Add Thermolib directories to the MATLAB path

Syntax

```
Thermolib_Setup
```

```
Thermolib_Setup('SavePaths','off','ShowDialog','off');
```

Description

`Thermolib_Setup` adds all necessary directories of the current Thermolib installation to the MATLAB path. In a dialog displayed one can chose to save the paths permanently to the MATLAB path.

To avoid showing the dialog you can use the option 'ShowDialog' with setting 'off'. Use 'SavePaths' to decide then if the paths should be saved or not.

9.2 THERMOLIB_VERSION

Syntax

```
Thermolib_Version
```

```
v = Thermolib_Version
```

```
[v,r] = Thermolib_Version
```

```
v = Thermolib_Version('string')
```

```
[v,r] = Thermolib_Version('string')
```

Description

```
Thermolib_Version;
```

display version information

```
v = Thermolib_Version;
```

return version number v as vector [H,U,B,I]

```
[v,r] = Thermolib_Version;
```



```
    return version number v as vector [H,U,B,I] and release string r  
v = Thermolib_Version('string');  
    return version number v as string 'H.U.B.I'  
[v,r] = Thermolib_Version('string');  
    return version number v as string 'H.U.B.I' and release string r
```

9.3 THERMOLIB_OPEN

Syntax

```
Thermolib_Open
```

Description

Thermolib_Open opens the Thermolib.mdl Library file and checks if all paths are correctly set.

9.4 TH_CREATEFEEDBACK

Open the Dialog for automatic creation of pressure feedback

Syntax

```
th_CreateFeedback
```

Description

This command opens the dialog for automatically connect pressure feedback ports with GOTO and FROM blocks. If you created a Thermolib model and you connected the components with forward flow busses in direction of the flow you can use this dialog to add connections between the "PFin" and "PFout" ports of the blocks in backward direction.

There are an automatic and a manual mode for creating the connections.

Executing the command from the MATLAB command window opens the Feedback Creator GUI, from which operations on the model are controlled and initiated.

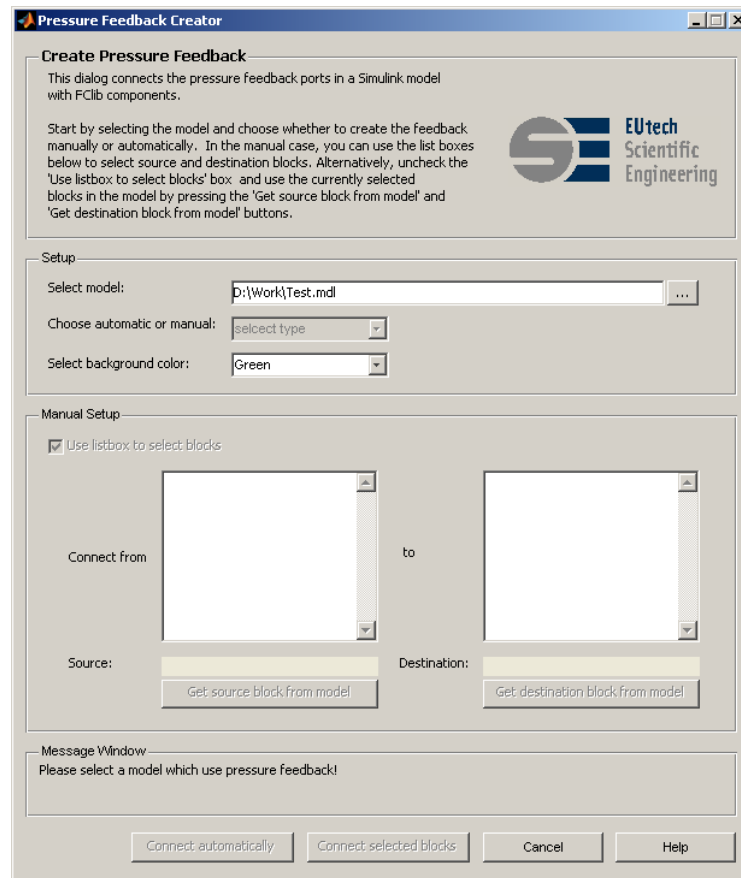


Figure 15: GUI for automatic creation of pressure feedback connection

For the Thermolib, pressure feedback means to use pressure information from a block downstream as input for the direct predecessor block upstream. This can best be modeled with Goto and From blocks to avoid too many signal lines in the model.

The Feedback Creator GUI supports two modes:

- Manual mode
- Automatic mode

In manual mode, a Goto-From connection are created after selecting an appropriate input and output block. All Thermolib blocks from the model that support pressure feedback can be chosen – either from list boxes containing these blocks or from the Simulink model window.

In automatic mode, the Goto-From connections are created automatically – where possible.

Parameters

Creation Mode

The modes available for selection are:

- Manual mode
- Automatic mode

The Manual Setup region is only accessible if mode is set to Manual.

Add Grounds and Terminators when ready

If selected, Ground and Terminator blocks are added to free inputs and outputs after creating pressure feedback.

Background Color

The color used for newly created Goto and From blocks.

Use Listbox to select Blocks

In manual mode you can select this option to chose the blocks to be connected from the listboxes below. A new connection creates a Goto block at the output block that is the "Connected from" selection. The From block is generated at the input block, selected as "to".

Connect automatically

This button is enabled only in Automatic mode. It starts the automatic creation of pressure feedback connections.

Connect selected Blocks

This button is enabled only in the manual mode. It starts the creation of a pressure backfeed connection based on the current selection seen in Source and Destination, respectively.

9.5 TH_EXPORTBALANCE

Export balance information after a simulation into Excel, ASCII file and into a Workspace variable.

Syntax

```
S = th_ExportBalance(SModelBalance, [], [])  
S = th_ExportBalance(SModelBalance, 'mybalance.xls', 'mybalance.txt')  
th_ExportBalance(SModelBalance, 'mybalance.xls', 'mybalance.txt')
```

Description

Use this command to export balancing information collected during a simulation run of a model containing Thermolib components. Information is collected automatically in the `SModelBalance` workspace variable. To make this information a complete balancing use the

```
S = th_ExportBalance(SModelBalance, [], []).
```

The command

```
output = th_ExportBalance (SModelBalance, xls_file, ascii_file)
```

saves the results of balancing in "output" and exports those in to Excel file "xls_file" and to ASCII file "ascii_file". The structure of "output" is explained in detail in chapter 6 "Thermodynamic Balancing".

If you leave `xls_file` or `ascii_file` empty, then the corresponding file will not be created.

Example

The following code commands open a demo model, run the simulation and export balance to Excel file "C:\temp\mybalance.xls" and to ASCII file "c:\temp\mybalance.txt".

Open a model with Thermolib components

```
DemoCombinedCyclePowerPlant
sim(gcs)
S = th_ExportBalance(SModelBalance, ...
'c:\temp\mybalance.xls', 'c:\temp\mybalance.txt')
```

9.6 TH_MODIFYCONSTANTS

Opens the dialog for modifying global, numerical constants

Syntax

```
th_ModifyConstants
```

default Constants.mat in Data directory is used

```
th_ModifyConstants('filename.mat')
```

custom mat file that contains constants is used

Description

`th_ModifyConstants` opens a dialog for modifying several global constants in Thermolib. These constants are numerical settings and limits for thermodynamical values as well as

settings for numerical accuracy. The mat file containing these constants is saved by default in your Thermolib installation directory under Data\Constants.mat. If there is no such file or you want to create a custom constants file please execute

```
th_CreateConstants('filename.mat').
```

List of constants

Constant	Description
TH_THERM_NDOT_EPS	1-by-n vector of molar flows [mole/s]
TH_THERM_NDOT_MIN	no calculation of states if mole flow below this, pumps/compressors will output at least this or 0
TH_THERM_NDOT_MAX	pumps/compressors/sources will not output more than this.
TH_THERM_PSI_EPS	used to check if pure substance or mixture $\psi < (1-\text{eps}) \Rightarrow$ mixture
TH_THERM_H_P_STATE_NUM_ITER_MAX	default number of iterations for H-P-State blocks
TH_THERM_T_S_STATE_NUM_ITER_MAX	default number of iterations for T-S-State blocks
TH_THERM_T_RHO_STATE_NUM_ITER_MAX	default number of iterations for T-Rho-State blocks
TH_THERM_U_RHO_STATE_NUM_ITER_MAX	default number of iterations for U-Rho-State blocks
TH_THERM_S_P_STATE_NUM_ITER_MAX	default number of iterations for S-P-State blocks
TH_THERM_H_P_STATE_T_CONV_LIMIT	[K] ... Accuracy for temperature calculation in H-P-State blocks
TH_THERM_H_P_STATE_HM_CONV_LIMIT	[J/mol] ... Accuracy for molar enthalpy in H-P-State blocks
TH_THERM_H_P_STATE_T_CONV_LIMIT	[K] ... Accuracy for temperature calculation in H-P-State blocks
TH_THERM_H_P_STATE_HM_CONV_LIMIT	[J/mol] ... Accuracy for molar enthalpy in H-P-State blocks

TH_THERM_S_P_STATE_T_CONV_LIMIT	[K] ... Accuracy for temperature calculation in S-P-State blocks
TH_THERM_S_P_STATE_SM_CONV_LIMIT	[J/mol] ... Accuracy for molar entropy in S-P-State blocks
TH_THERM_T_RHO_STATE_P_CONV_LIMIT	[Pa] ... Accuracy for pressure calculation in T-Rho-State blocks
TH_THERM_T_RHO_STATE_RHO_CONV_LIMIT	[g/m ³] ... Accuracy for density calculation in T-Rho-State blocks
TH_THERM_U_RHO_STATE_T_CONV_LIMIT	[K] ... Accuracy for temperature calculation in U-Rho-State blocks
TH_THERM_U_RHO_STATE_UM_CONV_LIMIT	[J/mol] ... Accuracy for molar enthalpy in U-Rho-State blocks

10 COMMAND LINE FUNCTIONS

10.1 TH_DENSITY

th_Density calculates the density of n flows at given temperature, pressure and vapor fractions.

Syntax:

```
[S,err] = th_Density('field1',VALUES1,'field2',VALUES2,...)
```

Fields:

The values given must describe N complete flows, by giving the following fields:

Field	Description
T	1-by-n matrix of temperatures [K]
p	1-by-n matrix of pressures [Pa]
species	cell array of length s with species names
psi	s-by-n matrix of molar fractions
gas_eos_method	1-by-1 matrix that defines the eos method
MediaData	Structure with the media data to use (see ChemicalMediaData.mat)
x	(optional) s-by-n matrix of vapor fractions (default is to calculate them using TpVaporFractions)
density_mode	(optional) 1-by-1 matrix that defines the way of calculating density; 0 = 5.1.0, 1 = <= 5.0

s ... number of species

n ... number of flows

Return:

The return value "S" is a struct with the following members:

Field	Description
S.rho	1-by-n matrix of densities of the overall flow [g/m ³]
S.rho_gas	1-by-n matrix of densities of gas phase only [g/m ³]
S.rho_liquid	1-by-n matrix of densities of liquid phase only [g/m ³]

If gas or liquid phase are not present, then the corresponding rho-value will be Inf or Nan.

The return value "err" is an error code, which indicates errors.
It is 0 if no error occurred.

Example:

The following example returns the rho-struct for two flows.

The first one with T1=80°C, of Water=0.5 and Methanol=0.5 and N2=0

the second flow with 25°C, containing Water=0.95 and Methanol=0 and N2=0.05 ,both in p=1 bar.

```
load ChemicalMediaData;
[S,err] = th_Density('T',[273.15+80,273.15+25],...
'p',[1e5,1e5],...
'psi',[0.5,0.95;0.5,0;0,0.05],...
'species',{'H2O','Methanol','N2' },...
```



```
'MediaData',SMediaData,...
'gas_eos_method',0);
```

See also ChemicalMediaData

This function is a Thermolib function.

Copyright 2003-2011 EUTech Scientific Engineering GmbH

10.2 TH_HPSTATE

th_HpState calculates complete state at given pressure and enthalpy.

Syntax:

```
[State_str,err] = th_HpState('field1',VALUES1,'field2',VALUES2,...)
```

Fields:

The values given must describe N complete flows, by giving the following fields:

Field	Description
ndot	1-by-n matrix of molar flow [mole/s]
Hdot	1-by-n matrix of enthalpy flow [W]
pressure	1-by-n matrix of pressures [Pa]
psi	s-by-n matrix of molar fraction
gas_eos_method	1-by-1 matrix that defines the eos method; 0 = ideal gas (default), 1 = Peng Robinson, 2 = RK
species	cell array of length s with species names
MediaData	Structure with the media data to use (see ChemicalMediaData.mat)
T_initial	1-by-n matrix of initial temperatures [K]

s ... number of species

n ... number of flows

Return:

The return value "State_str" is a struct with the following members:

Field	Description
State_str.ndot	1-by-n matrix of the molar flow [mole/s]
State_str.T	1-by-n matrix of the temperature [K]
State_str.p	1-by-n matrix of the pressure [Pa]
State_str.Hdot	1-by-n matrix of the Hdot [W]
State_str.Sdot	1-by-n matrix of the Sdot [W/K]
State_str.Gdot	1-by-n matrix of the Gdot [W]
State_str.Cpdot	1-by-n matrix of the Cpdot [W/K]
State_str.x	s-by-n matrix of the vapor fractions
State_str.psi	s-by-n matrix of the molar fractions

The return value "err" is an error code, which indicates errors.

It is 0 if no error occurred.

Example:

The following example returns the State_str for two flows.

The first one with H1=-1e6, of Water=0.5 and O2=0.25 and N2=0.25

the second flow with H2=-1e5, containing Water and O2 and N2 ,both in

p=1 bar and ndot=1.5 mole/s.

```
load ChemicalMediaData
```

```
[State_str,err] = th_HpState('Hdot',[-1e6,-1e5],...
```

```
'species',{'H2O','O2','N2' },...
```

```
'MediaData',SMediaData,...
'psi',[0.5,0.5;0.25,0.25;0.25,0.25],...
'pressure',[1e5,1e5],...
'ndot',[1.5,1.5],...
'gas_eos_method',0,...
'T_initial',[300,400]);
```

See also ChemicalMediaData

This function is a Thermolib function.

Copyright 2003-2011 EUTech Scientific Engineering GmbH

10.3TH_PENGROBINSON

th_PengRobinson calculates the Peng Robinson Z values for given temperature and pressure.

Syntax:

```
[Result,err] = th_PengRobinson('field1',VALUES1,'field2',VALUES2,...)
```

Fields:

The values given must describe N complete flows, by giving the following fields:

Field	Description
T	1-by-n matrix of temperatures [K]
p	1-by-n matrix of pressure [Pa]
species	cell array of length s with species names
MediaData	Structure with the media data to use (see ChemicalMediaData.mat)

s ... number of species

n ... number of flows

Return:

The return value "Result" is a struct with 3 complex points containing real part and imag part.

The return value "err" is an error code, which indicates errors.

It is 0 if no error occurred.

Example:

The following example returns the Z values for H2O with 80 degrees and 1e6 Pa.

```
load ChemicalMediaData
[Result,err] = th_PengRobinson('T',[273.15+80], ...
'p',[1e6], ...
'species',{'H2O'}, ...
'MediaData',SMediaData);
```

See also ChemicalMediaData

This function is a Thermolib function.

Copyright 2003-2011 EUTech Scientific Engineering GmbH

10.4 TH_SATURATIONPRESSURE

th_SaturationPressure calculates saturation pressure at given temperature.

Syntax:

```
[Psat,err] = th_SaturationPressure('field1',VALUES1,'field2',VALUES2,...)
```

Fields:

The values given must describe N complete flows, by giving the following fields:

Field	Description
T	1-by-n matrix of temperatures [K]
species	cell array of length s with species names
MediaData	Structure with the media data to use (see ChemicalMediaData.mat)

s ... number of species

n ... number of flows

Return:

The return value "Psat" is a s-by-n matrix of the Saturation Pressure.

The return value "err" is an error code, which indicates errors.

It is 0 if no error occurred.

Example:

The following example returns the Saturation Pressure matrix for two flows.

The first one with T1=80°C, of Water and Methanol and N2

the second flow with 25°C, containing Water and Methanol and N2

```
load ChemicalMediaData
```

```
[Psat,err] = th_SaturationPressure('T',[273.15+80,273.15+25],...
```

```
'species',{'H2O','Methanol','N2' },...
```

```
'MediaData',SMediaData);
```

See also ChemicalMediaData

This function is a Thermolib function.

Copyright 2003-2011 EUTech Scientific Engineering GmbH

10.5 TH_SATURATIONTEMPERATURE

th_SaturationTemperature calculates saturation temperature at given pressure

Syntax:

```
[Tsat,err] =  
th_SaturationTemperature('field1',VALUES1,'field2',VALUES2,...)
```

Fields:

The values given must describe N complete flows, by giving the following fields:

Field	Description
p	1-by-n matrix of pressures [Pa]
species	cell array of length s with species names
MediaData	Structure with the media data to use (see ChemicalMediaData.mat)

s ... number of species

n ... number of flows

Return:

The return value "Tsat" is a s-by-n matrix of the Saturation Temperature.

The return value "err" is an error code, which indicates errors.

It is 0 if no error occurred.

Example:

The following example returns the Saturation Temperature matrix for two flows.

The first one with $p_1=1e5$, of Water and Methanol and N2

the second flow with $2e4$, containing Water and Methanol and N2

```
load ChemicalMediaData
```

```
[Tsat,err] = th_SaturationTemperature('p',[1e5,2e4],...
'species',{'H2O','Methanol','N2' },...
'MediaData',SMediaData);
```

See also ChemicalMediaData

This function is a Thermolib function.

Copyright 2003-2011 EUTech Scientific Engineering GmbH

10.6 TH_SPSTATE

th_SpState calculates complete state at given pressure and Sdot.

Syntax:

```
[State_str,err] = th_SpState('field1',VALUES1,'field2',VALUES2,...)
```

Fields:

The values given must describe N complete flows, by giving the following fields:

Field	Description
ndot	1-by-n matrix of molar flow [mole/s]
p	1-by-n matrix of pressures [Pa]
psi	s-by-n matrix of molar fraction
Sdot	1-by-n matrix of entropy flow [W/K]

gas_eos_method	1-by-1 matrix that defines the eos method; 0 = ideal gas (default), 1 = Peng Robinson, 2 = RK
species	cell array of length s with species names
MediaData	Structure with the media data to use (see ChemicalMediaData.mat)

s ... number of species

n ... number of flows

Return:

The return value "State_str" is a struct with the following members:

Field	Description
State_str.ndot	1-by-n matrix of the molar flow [mole/s]
State_str.T	1-by-n matrix of the temperature [K]
State_str.p	1-by-n matrix of the pressure [Pa]
State_str.Hdot	1-by-n matrix of the Hdot [W]
State_str.Sdot	1-by-n matrix of the Sdot [W/K]
State_str.Gdot	1-by-n matrix of the Gdot [W]
State_str.Cpdot	1-by-n matrix of the Cpdot [W/K]
State_str.x	s-by-n matrix of the vapor fractions
State_str.psi	s-by-n matrix of the molar fractions

The return value "err" is an error code, which indicates errors.

It is 0 if no error occurred.

Example:

The following example returns the State_str for two flows.

The first one with Sdot=285.6 W/K, of Water=0.1 and N2=0.74 and O2=0.16

the second flow with $\dot{S}=322.7$ W/K, with the same mixture. Both flows at $p=1$ bar and $\dot{n}=1.5$ mole/s.

```
load ChemicalMediaData
[State_str,err] = th_SpState('Sdot',[285.6 322.7],...
'p',[1e5 1e5],...
'psi',[0.1, 0.1; 0.74, 0.74; 0.16, 0.16;],...
'ndot',[1.5 1.5], ...
'species',{'H2O','N2','O2'},...
'gas_eos_method',1, ...
'MediaData',SMediaData);
```

See also ChemicalMediaData

This function is a Thermolib function.

Copyright 2003-2011 EUTech Scientific Engineering GmbH

10.7 TH_TPHSTATE

th_TpHState calculates complete state at given temperature, pressure and enthalpy.

Syntax:

```
[State_str,err] = th_TpHState('field1',VALUES1,'field2',VALUES2,...)
```

Fields:

The values given must describe N complete flows, by giving the following fields:

Field	Description
ndot	1-by-n matrix of molar flow [mole/s]
T	1-by-n matrix of temperatures [K]

p	1-by-n matrix of pressures [Pa]
Hdot	1-by-n matrix of enthalpy flow [watt]
psi	s-by-n matrix of molar fraction
gas_eos_method	1-by-1 matrix that defines the eos method; 0 = ideal gas, 1 = Peng Robinson, 2 = RK
species	cell array of length s with species names
MediaData	Structure with the media data to use (see ChemicalMediaData.mat)

s ... number of species

n ... number of flows

Return:

The return value "State_str" is a struct with the following members:

Field	Description
State_str.ndot	1-by-n matrix of the molar flow [mole/s]
State_str.T	1-by-n matrix of the temperature [K]
State_str.p	1-by-n matrix of the pressure [Pa]
State_str.Hdot	1-by-n matrix of the Hdot [W]
State_str.Sdot	1-by-n matrix of the Sdot [W/K]
State_str.Gdot	1-by-n matrix of the Gdot [W]
State_str.Cpdot	1-by-n matrix of the Cpdot [W/K]
State_str.X	s-by-n matrix of the vapor fractions
State_str.psi	s-by-n matrix of the molar fractions

The return value "err" is an error code, which indicates errors.

It is 0 if no error occurred.

Example:

The following example returns the State_str for two flows.

The first one with T1=80°C, of Water=0.5 and O2=0.25 and N2=0.25
the second flow with 25°C, containing Water and O2 and N2 ,both in
p=1 bar and ndot=1.5 mole/s.

```
load ChemicalMediaData
[State_str,err] = th_TpHState('T',[273.15+80,273.15+25],...
'species',{'H2O','O2','N2' },...
'MediaData',SMediaData,...
'psi',[0.5,0.5;0.25,0.25;0.25,0.25],...
'pressure',[1e5,1e5],...
'ndot',[1.5,1.5],...
'Hdot',[-1e5,-1e5],...
'gas_eos_method',0);
```

See also ChemicalMediaData

This function is a Thermolib function.

Copyright 2003-2011 EUTech Scientific Engineering GmbH

10.8 TH_TPSTATE

th_TpState calculates complete state at given temperature and pressure.

Syntax:

```
[State_str,err] = th_TpState('field1',VALUES1,'field2',VALUES2,...)
```

Fields:

The values given must describe N complete flows, by giving the following fields:

Field	Description
ndot	1-by-n matrix of molar flow [mole/s]
T	1-by-n matrix of temperatures [K]
p	1-by-n matrix of pressures [Pa]
psi	s-by-n matrix of molar fraction
gas_eos_method	1-by-1 matrix that defines the eos method;
species	cell array of length s with species names
MediaData	Structure with the media data to use (see ChemicalMediaData.mat)
liquid_eos_method	1-by-1 matrix that defines the liquid eos method; 0 = using T_ref, 1 = using hfg

s ... number of species

n ... number of flows

Return:

The return value "State_str" is a struct with the following members:

Field	Description
State_str.ndot	1-by-n matrix of the molar flow [mole/s]
State_str.T	1-by-n matrix of the temperature [K]
State_str.p	1-by-n matrix of the pressure [Pa]
State_str.Hdot	1-by-n matrix of the Hdot [W]
State_str.Sdot	1-by-n matrix of the Sdot [W/K]
State_str.Gdot	1-by-n matrix of the Gdot [W]
State_str.Cpdot	1-by-n matrix of the Cpdot [W/K]
State_str.x	s-by-n matrix of the vapor fractions
State_str.psi	s-by-n matrix of the molar fractions

The return value "err" is an error code, which indicates errors.

It is 0 if no error occurred.

Example:

The following example returns the State_str for two flows.

The first one with T1=80°C, of Water=0.5 and O2=0.25 and N2=0.25

the second flow with 25°C, containing Water and O2 and N2, both in

p=1 bar and ndot=1.5 mole/s.

```
load ChemicalMediaData
[State_str,err] = th_TpState('T',[273.15+80,273.15+25],...
'species',{'H2O','O2','N2' },...
'MediaData',SMediaData,...
'psi',[0.5,0.5;0.25,0.25;0.25,0.25],...
'pressure',[1e5,1e5],...
'ndot',[1.5,1.5],...
'gas_eos_method',0);
```

See also ChemicalMediaData

This function is a Thermolib function.

Copyright 2003-2011 EUTech Scientific Engineering GmbH

10.9 TH_TP VAPOR FRACTIONS

th_TpVaporFractions calculates molar fractions at given temperature and pressure.

Syntax:

```
[S,err] = th_TpVaporFractions('field1',VALUES1,'field2',VALUES2,...)
```

Fields:

The values given must describe N complete flows, by giving the following fields:

Field	Description
T	1-by-n matrix of temperatures [K]
p	1-by-n matrix of pressures [Pa]
psi	s-by-n matrix of molar fraction
gas_eos_method	1-by-1 matrix that defines the eos method; 0 = ideal gas, 1 = Peng Robinson, 2 = RK
species	cell array of length s with species names
MediaData	Structure with the media data to use (see ChemicalMediaData.mat)

s ... number of species

n ... number of flows

Return:

The return value "State_str" is a struct with the following members:

Field	Description
State_str.x	s-by-n matrix of the vapor fractions

The return value "err" is an error code, which indicates errors.

It is 0 if no error occurred. If an error occurred, use function

S = th_GetErrorInfo(err) to get detailed error informations.

Example:

The following example returns the State_str for two flows.

The first one with T1=80°C, of Water=0.5 and Methanol=0.5 and N2=0

the second flow with 25°C, containing Water=0.95 and Methanol=0 and N2=0.05 ,both in p=1 bar.

```
load ChemicalMediaData
```

```
[S,err] = th_TpVaporFractions('T',[273.15+80,273.15+25],...
'species',{'H2O','Methanol','N2' },...
'MediaData',SMediaData,...
'psi',[0.5,0.95;0.5,0;0,0.05],...
'pressure',[1e5,1e5],...
'gas_eos_method',0);
```

See also th_GetErrorInfo, ChemicalMediaData

This function is a Thermolib function.

Copyright 2003-2011 EUTech Scientific Engineering GmbH

10.10 TH_TPXSTATE

th_TpxState calculates complete state at given temperature and pressure and vapor fractions.

Syntax:

```
[State_str,err] = th_TpxState('field1',VALUES1,'field2',VALUES2,...)
```

Fields:

The values given must describe N complete flows, by giving the following fields:

Field	Description
ndot	1-by-n matrix of molar flow [mole/s]
T	1-by-n matrix of temperatures [K]

p	1-by-n matrix of pressures [Pa]
x	s_by_n matrix of vapor fractions
psi	s-by-n matrix of molar fraction
gas_eos_method	1-by-1 matrix that defines the eos method; 0 = ideal gas, 1 = Peng Robinson, 2 = RK
species	cell array of length s with species names
MediaData	Structure with the media data to use (see ChemicalMediaData.mat)

s ... number of species

n ... number of flows

Return:

The return value "State_str" is a struct with the following members:

Field	Description
State_str.ndot	1-by-n matrix of the molar flow [mole/s]
State_str.T	1-by-n matrix of the temperature [K]
State_str.p	1-by-n matrix of the pressure [Pa]
State_str.Hdot	1-by-n matrix of the Hdot [W]
State_str.Sdot	1-by-n matrix of the Sdot [W/K]
State_str.Gdot	1-by-n matrix of the Gdot [W]
State_str.Cpdot	1-by-n matrix of the Cpdot [W/K]
State_str.x	s-by-n matrix of the vapor fractions
State_str.psi	s-by-n matrix of the molar fractions

The return value "err" is an error code, which indicates errors.

It is 0 if no error occurred.

Example:

The following example returns the State_str for two flows.

The first one with T1=80°C, of Water=0.5 and Methanol=0.5 and N2=0

the second flow with 25°C, containing Water=0.95 and Methanol=0 and N2=0.05 ,both in
p=1 bar and ndot=1.5 mole/s.

```
load ChemicalMediaData
```

```
[State_str,err] = th_TpxState('T',[273.15+80,273.15+25],...
```

```
'species',{ 'H2O','Methanol','N2' },...
```

```
'MediaData',SMediaData,...
```

```
'psi',[0.5,0.95;0.5,0;0,0.05],...
```

```
'pressure',[1e5,1e5],...
```

```
'ndot',[1.5,1.5],...
```

```
'x',[0.1289,0.0017;0.3421,1;1,1],...
```

```
'gas_eos_method',0);
```

See also ChemicalMediaData

This function is a Thermolib function.

Copyright 2003-2011 EUTech Scientific Engineering GmbH

10.11 TH_TRHOSTATE

th_TrhoState calculates complete state at given densities and temperature.

Syntax:

```
[State_str,err] = th_TrhoState('field1',VALUES1,'field2',VALUES2,...)
```

Fields:

The values given must describe N complete flows, by giving the following fields:

Field	Description
ndot	1-by-n matrix of molar flow [mole/s]
T	1-by-n matrix of temperatures [K]
psi	s-by-n matrix of molar fraction
rho	1-by-n matrix of densities [g/m ³]
p_if_liquid	1-by-n matrix of pressures [Pa]
gas_eos_method	1-by-1 matrix that defines the eos method; 0 = ideal gas (default), 1 = Peng Robinson, 2 = RK
species	cell array of length s with species names
MediaData	Structure with the media data to use (see ChemicalMediaData.mat)
density_mode	(optional) 1-by-1 matrix that defines the way of calculating density; 0 = 5.1.0, 1 = <= 5.0

s ... number of species

n ... number of flows

Return:

The return value "State_str" is a struct with the following members:

Field	Description
State_str.ndot	1-by-n matrix of the molar flow [mole/s]
State_str.T	1-by-n matrix of the temperature [K]
State_str.p	1-by-n matrix of the pressure [Pa]
State_str.Hdot	1-by-n matrix of the Hdot [W]
State_str.Sdot	1-by-n matrix of the Sdot [W/K]
State_str.Gdot	1-by-n matrix of the Gdot [W]
State_str.Cpdot	1-by-n matrix of the Cpdot [W/K]
State_str.x	s-by-n matrix of the vapor fractions

State_str.psi	s-by-n matrix of the molar fractions
---------------	--------------------------------------

The return value "err" is an error code, which indicates errors.

It is 0 if no error occurred.

Example:

The following example returns the State_str for two flows.

The first one with $T_1=300\text{K}$ and $\rho = 1198\text{g/m}^3$, of Water=0.1 and $\text{O}_2=0.16$ and $\text{N}_2=0.74$ the second flow with 500K and 665.1 g/m^3 containing Water and O_2 and N_2 , both with $\dot{n}=1.5\text{mole/s}$.

```
load ChemicalMediaData
[STrho,err] = th_TrhoState('T',[300 500],...
'psi',[0.1, 0.1; 0.74, 0.74; 0.16, 0.16;],...
'rho',[1189, 665.1], ...
'ndot',[1.5 1.5], ...
'species',{'H2O','N2','O2'},...
'gas_eos_method',1, ...
'p_if_liquid',[1e6,1e6],...
'MediaData',SMediaData);
```

See also ChemicalMediaData

This function is a Thermolib function.

Copyright 2003-2011 EUTech Scientific Engineering GmbH

10.12 TH_URHOSTATE

th_UrhoState calculates complete state at given density and the internal energy.

Syntax:

```
[State_str,err] = th_UrhoState('field1',VALUES1,'field2',VALUES2,...)
```

Fields:

The values given must describe N complete flows, by giving the following fields:

Field	Description
ndot	1-by-n matrix of molar flow [mole/s]
U	1-by-n matrix of internal energy [W]
psi	s-by-n matrix of molar fraction
rho	1-by-n matrix of densities [g/m ³]
p_if_liquid	1-by-n matrix of pressures [Pa]
gas_eos_method	1-by-1 matrix that defines the eos method; 0 = ideal gas (default), 1 = Peng Robinson, 2 = RK
species	cell array of length s with species names
MediaData	Structure with the media data to use (see ChemicalMediaData.mat)
density_mode	(optional) 1-by-1 matrix that defines the way of calculating density; 0 = 5.1.0, 1 = <= 5.0

s ... number of species

n ... number of flows

Return:

The return value "State_str" is a struct with the following members:

Field	Description
State_str.ndot	1-by-n matrix of the molar flow [mole/s]
State_str.T	1-by-n matrix of the temperature [K]

State_str.p	1-by-n matrix of the pressure [Pa]
State_str.Hdot	1-by-n matrix of the Hdot [W]
State_str.Sdot	1-by-n matrix of the Sdot [W/K]
State_str.Gdot	1-by-n matrix of the Gdot [W]
State_str.Cpdot	1-by-n matrix of the Cpdot [W/K]
State_str.x	s-by-n matrix of the vapor fractions
State_str.psi	s-by-n matrix of the molar fractions
State_str.u_rho_iterations	1-by-1 matrix of the number of iterations in UrhoState
State_str.t_rho_iterations	1-by-1 matrix of the number of all iterations in TrhoState

The return value "err" is an error code, which indicates errors.

It is 0 if no error occurred.

Example:

The following example returns the State_str for two flows.

The first one with $U1 = -4e4W$ and $\rho_{o1}=1189 \text{ g/m}^3$ of Water=0.1 and $O2=0.16$ and $N2=0.74$. The second flow with $U2 = -2.7e4W$ and $\rho_{o2} = 665.1 \text{ g/m}^3$, containing Water and $O2$ and $N2$, both with $\dot{n}=1.5 \text{ mole/s}$.

```
load ChemicalMediaData
[Surho,err] = th_UrhoState('U',[-4e4 -2.7e4],...
'psi',[0.1, 0.1; 0.74, 0.74; 0.16, 0.16;],...
'rho',[1189, 665.1], ...
'ndot',[1.5 1.5], ...
'species',{'H2O','N2','O2'},...
'gas_eos_method', 1, ...
```

```
'p_if_liquid',[1e6,1e6],...  
'MediaData',SMediaData);
```

See also ChemicalMediaData

This function is a Thermolib function.

Copyright 2003-2011 EUTech Scientific Engineering GmbH

11 REFERENCES

- [1] <http://garfield.chem.elte.hu/Burcat/burcat.html>
- [2] NIST-Webbook, <http://webbook.nist.gov/chemistry>
- [3] Fundamentals of Thermodynamics, Richard E. Sonntag, Claus Borgnakke, Gordon J. Van Wylen, 6.Aufl., 2003, John Wiley & Sons.
- [4] Pukrushpan, Jay T. Control of Fuel Cell Power Systems: principles, modeling, analysis and feedback design. (Advances in industrial control). Springer, 2005.
- [5] Larminie, James. Dicks, Andrew. Fuel Cell Systems Explained, 2nd Edition, Wiley 2003.
- [6] Holman, J. P. Heat Transfer, 9th edition, McGraw-Hill, 2002.
- [7] Cengel, Y.A., Thermodynamic: An Engineering approach, 5th ed, McGraw-Hill, 2006.