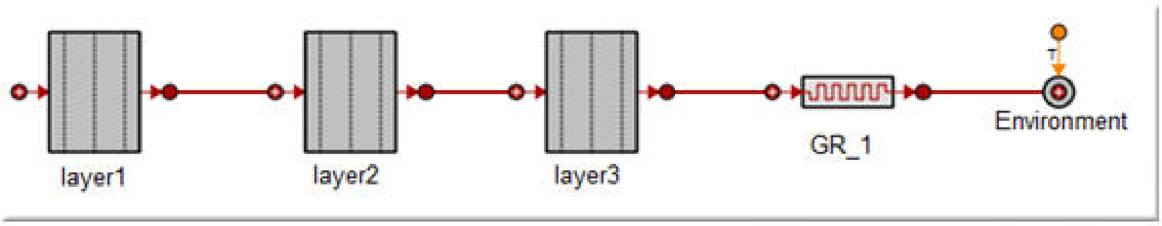
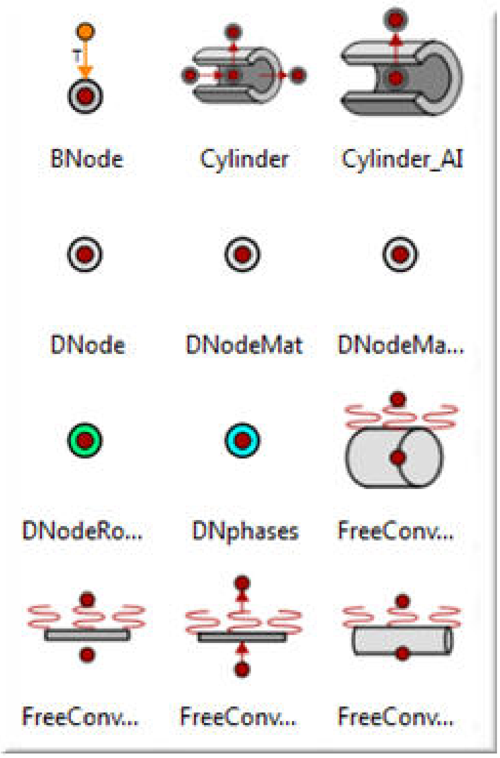
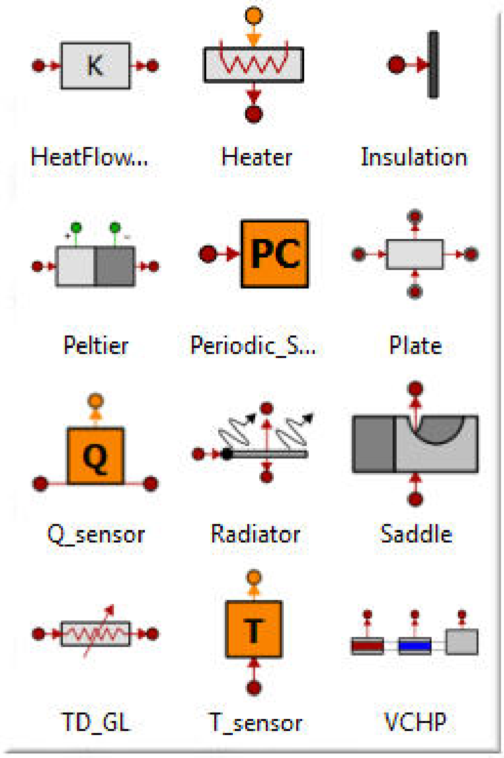
THERMAL Library 3.5.2

Reference Manual







EMPRESARIOS AGRUPADOS

Edition 2016

This page intentionally left blank.

Contents

This page intentionally left blank.

1. Overview of the Library

1.1 Introduction

This document is the reference manual for the THERMAL Library (version 3.5).

This Reference Manual will provide:

- List of the components of the library
- Global description of the units of the library
- Icons associated to every component
- Description of the data and variables associated to every component and function
- Limitations of every component and function
- Description of the physical-mathematical model associated to every component or function

The Thermal Library does not provide all of the components that a user may possibly require, but it presents a good starting point for easily building customised components using the components provided by means of inheritance.

The THERMAL library, supplied with the professional version of this software, contains components to predict temperature distributions and heat flows in systems and devices using the thermal network method.

The thermal network method is also known as the lumped parameter method or the resistance/capacity method. It is essentially a finite difference method and entails modelling a continuous medium as a discrete thermal network of nodes representing the capacitance of the system linked by conductors representing its conductance.

Lumped parameter models are easily built. A thermal system with complex geometries can be replaced by a series of heat capacities and conductors (one-dimensional heat transfer) if a very detailed temperature distribution within the system is not required.

Using drag & drop methodology, the user can quickly create a diagram of the thermal system to be analysed, the representation of which is very similar to the physical system. The capabilities of the library are vast, since customized components can be developed from the available components as a function of the modelling needs.

Thanks to the features of this tool, libraries can be built that are easy to configure and extend, adding any components and characteristics as needed. This can be done graphically through a simple, user friendly interface, or through its own object-orientated language which makes it possible to re-use existing codes.

One of the library's biggest advantages is the possibility of its use in the multidisciplinary facet of this tool. This means that we can jointly study, for example, hydraulic systems or energy generation processes, such as combined cycle plants, and the dynamic behaviour of heat transfer through different geometrics and materials. This way, an overall study of the system is obtained with minimal design time.

Either steady-state or transient analyses can be handled with this library. The heat transfer modes that can be considered in the thermal model are conduction, radiation and natural convection. Material properties such as specific heat or conductivity may also vary with temperature.

1.2 Components in the Thermal Library

1.2.1 List of Components

Abstract Components

Component Type Name	Items Represented
Node	Thermal node
Conductor	Thermal conductor
ADNode	Abstract diffusive thermal node
AGL	Abstract linear thermal conductor
AGR	Abstract radiative conductor

Table 1-1. Abstract Component Types in the Thermal Library

Operative Components

Component Type Name	Items Represented
BNode	Boundary thermal node
Cylinder	Cylinder divided into longitudinal and radial nodes
Cylinder_AI	Axially isolated cylinder divided into longitudinal and radial nodes
DNode	Diffusive thermal node with constant capacity
DNodeMat	Diffusive thermal node having a mass of a given material
DNodeMat_vec	Diffusive vectorized thermal node calculating n temperatures with no heat exchange between nodes
DNodeRoom	Diffusive thermal node that represents a volume filled with a fluid
DNphases	Diffusive thermal node taking into account phase changes
FreeConvec_HorizontalCylinder	Coupling to calculate the natural convection from a surface of a horizontal cylinder to a fluid
FreeConvec_Plate	Coupling to calculate the natural convection from a surface of a plate to a fluid
FreeConvec_UserDefined	Coupling to calculate the natural convection specifying the heat transfer coefficient as a function of the temperature
FreeConvec_Wire	Coupling to calculate the natural convection from a surface of a wire to a fluid
FreeConvec_simple	Coupling to calculate natural convection from a surface to a simplified fluid
GL	Linear conductor with constant conductance
GL_mat	Linear conductor of a given material
GR	Radiative conductor with constant REF (Radiative Exchange Factor)
HeatFlowMultiplier	Device to multiply input heat flow by a specified gain
Heater	Heater
Insulation	Thermal insulator
Peltier	Peltier element
Periodic_Stop_Control	Device to check if the dynamic response of node temperature is cyclic
Plate	Plate divided into a number of longitudinal and transversal nodes
Q_sensor	Heat flow sensor
Radiator	Radiative plate
Saddle	Element to fasten heat pipes and tubes to a surface
TD_GL	Linear conductor with temperature dependant conductance
Th_Demux	Thermal demultiplexer: splitting a thermal port in two thermal ports
Th_Mux	Thermal multiplexer, bringing together two thermal ports into one port
T_sensor	Temperature sensor
Tube	Pipe section required for constructing heat pipe component
VCHP	Variable conductance heat pipe
View_Factors	Radiative exchange between thermal nodes, the radiative exchange is calculated from the view factors.
Wall	Thermal wall

Table 1-2. Component Types in the Thermal Library

1.2.2 Hierarchy of the Components

The following figures show the relationships between inherited components.

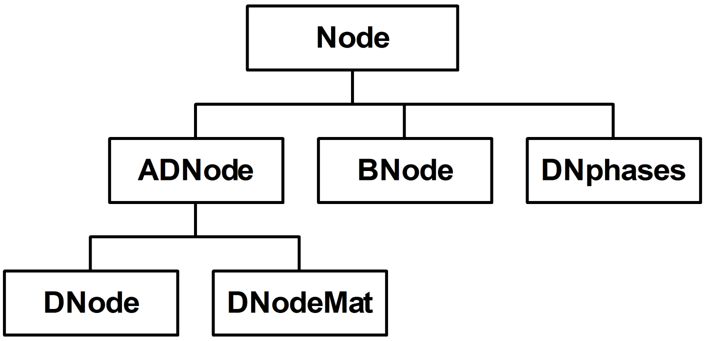


Figure 1-1. Hierarchy of nodes

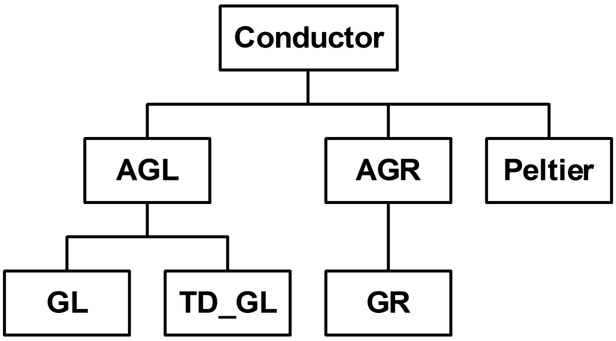


Figure 1-2. Hierarchy of conductors

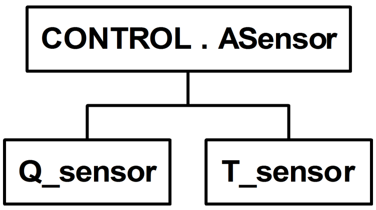


Figure 1-3. Hierarchy of sensors

1.2.3 Ports

This library uses the following types of elementary standard ports that are defined in the library PORTS_LIB:

Port Type	Description	Symbol
thermal	1-dimension array of thermal exchanges between two components. Physically, it corresponds to a line of contact or a surface of contact where the variables only exchange in one direction. The number of points where thermal exchange takes place can be varied through the "n" parameter; thus this type of port is referred to as a vectorised port.	
analog_signal	1-dimension array of analog signal	
elec	electrical pin	

The variables defined in each port are the following:

- **thermal port**

Name	Description	Units
Tc[n]	Array of temperatures in Celsius	ºC
Tk[n]	Array of temperatures in Kelvin	K
q[n]	Array of heat flows	W

- **analog_signal port**

Name	Description	Units
signal[n]	Array of analog signals	-

- **elec port**

Name	Description	Units
i	Electrical Current	A
v	Voltage (electrical potential)	V

2. Global Items

2.1 Global Constants

Name	Type	Value	Description	Units
SOLAR_CONSTANT	CONST REAL	1410	Solar power received at Earth orbit	*W/m2*
STEFAN	CONST REAL	5.67E-8	Stefan Boltzmann Constant	*W/m2 K4*
TMAX	CONST REAL	100000	Maximum temperature to report a warning	*K*
TZERO	CONST REAL	273.15	Zero Temperature Shift	*K*

2.2 Enumeration Data Types

2.2.1 Enumeration Type GasFlProps

Enumerative data type that defines fluid properties in gas state.

```
ENUM GasFlProps = {GAS_cp, GAS_viscosity, GAS_thermal_conductivity}
```

2.2.2 Enumeration Type LiqFlProps

Enumerative data type for fluid properties in liquid state.

```
ENUM LiqFlProps = {LIQ_density, LIQ_cp, LIQ_vsound, LIQ_beta, LIQ_viscosity,
   LIQ_thermal_conductivity, LIQ_surface_tension}
```

2.2.3 Enumeration Type Material

Enumerative data type that defines the list of available materials in the library.

```
ENUM Material = { None, Aluminum,AL2219,AL3003_F,AL5083_O,AL6061_T6,AL7020,
  Carbon, Carbon_Steel, Copper, Cryosof, Dacron_filled_with_helium,
  Dacron_filled_with_nitrogen, Epoxy_fiberglass, H920A, SS_304, SS_304L,
  SS_310, SS_316, SS_321, Titanium, Vacuum_Insulated_Panel, SA_210_A1,
  SA_213_T22, T409, SA_106_GrB, SA_335_P22,Al5083,GCF, Steel321,
VIP, MatUsr1, MatUsr2, MatUsr3}
```

2.2.4 Enumeration Type PropSolid

Enumerative data type that describes properties of material.

```
ENUM PropSolid = {Density, ElasticityModulus, PoissonCoeff, SpecificHeat,
  Conductivity}
```

2.2.5 Enumeration Type ThFluids

Enumerative data type that describes the list of working fluids in the library.

```
ENUM ThFluids = {Air, Nitrogen, Oxygen, Water_ReheatedVap, UsrDef_gas1,
  UsrDef_gas2, UsrDef_gas3, Water_liq, UsrDef_liq1, UsrDef_liq2, UsrDef_liq3}
```

2.2.6 Enumeration Type WALL_INIT_MODE

Enumerative data type that defines the available types of initialization.

```
ENUM WALL_INIT_MODE = {Constant_Temp, Linear_Temp, Table_Temp}
```

option	Description
Constant_Temp	Nodes are initialized at a constant user-defined temperature
Linear_Temp	Nodes on both sides are initialized at the user-defined temperature. Temperatures of intermediate nodes are a linear interpolation between the values of both sides.
Table_Temp	Nodes are initialized by means of interpolation in a user-defined table

2.2.7 Enumeration Type WIRE_POSITION

Enumeration data type that defines the position of the component wire.

```
ENUM WIRE_POSITION = {Horizontal, Vertical}
```

2.2.8 Enumeration Type WallType

Enumerative data type that defines the type of wall.

```
ENUM WallType = {wall, floor, ceil}
```

2.3 SET_OF Data Types

2.3.1 SET_OF Type ConstProp

SET_OF type that defines the group of elements belonging to the enumerative data type PropSolid which are constant.

```
SET_OF (PropSolid) ConstProp ={Density, ElasticityModulus, PoissonCoeff}
```

2.3.2 SET_OF Type PipeMat

SET_OF type that sorts the elements of the enumerative type Material that are available for the component Pipe.

```
SET_OF (Material) PipeMat = { None, Aluminum, AL2219, AL3003_F, AL5083_O,
AL6061_T6, AL7020, Carbon_Steel, Copper, Epoxy_fiberglass, SS_304 , SS_304L,
   10,SS_316,SS_321,Titanium,Al5083,Steel321,MatUsr1,MatUsr2,MatUsr3}
```

2.3.3 SET_OF Type VarProp

SET_OF type that defines the group of elements belonging to the enumerative data type PropSolid which are time-dependent.

```
SET_OF (PropSolid) VarProp = {SpecificHeat, Conductivity}
```

2.4 Current Available Materials

Current available materials are:

Material name	File Name
Aluminium	Aluminium.txt
AL2219	AL2219.txt
Al5083_O or Al5083	Al5083_O.txt
AL3003_F	AL3003_F.txt
AL6061_T6	AL6061_T6.txt
AL7020	AL7020.txt
Carbon	Carbon.txt
Carbon_Steel	Carbon_Steel.txt
Copper	Copper.txt
Cryosof	Cryosof.txt
Dacron_filled_with_helium	Dacron_filled_with_helium.txt
Dacron_filled_with_nitrogen	Dacron_filled_with_nitrogen.txt
GCF or Epoy fiberglass	Epoxy_fiberglass.txt
H920A	H920A.txt
SA_106_GrB	SA_106_GrB.txt
SA_210_A1	SA_210_A1.txt
SA_213_T22	SA_213_T22.txt
SA_335_P22	SA_335_P22.txt
SS_304	SS_304.txt
SS_304L	SS_304L.txt
SS_310	SS_310.txt
SS_316	SS_316.txt
SS_321	SS_321.txt
T409	T409.txt
Titanium	Titanium.txt
MatUsr1.txt tp MatUsr3	MatUsr1.txt tp MatUsr3-txt

2.5 Global Variables

Name	Type	Default	Description	Units
Orbit_Period	BOUND REAL	7200	Orbit Period	*s*
PRINT_TEMP	BOOLEAN	FALSE	Print label to display the name and temperature of the nodes	-
setRaleighWarnings	INTE-GER	1	Test validity ranges in natural convection correlations (0 warnings are not shown ; 1 warnings are shown)	

2.6 Current Available Fluids

The following table depicts the available 1-phase pure fluids in the library for free-convection calculations:

Fluid name	File Name	Description
Air	Air.txt	Air in gas phase
Nitrogen	Nitrogen.txt	Nitrogen in gas phase
Oxygen	Oxygen.txt	Oxygen in gas phase
Water_ReheatedVap	Water_ReheatedVap.txt	Reheated water vapour
UsrDef_gas1	UsrDef_gas1.txt	Gas defined by the user
UsrDef_gas2	UsrDef_gas2.txt	Gas defined by the user
UsrDef_gas3	UsrDef_gas3.txt	Gas defined by the user
Water_liq	Water_liq.txt	Liquid water
UsrDef_liq1	UsrDef_liq1.txt	Liquid defined by the user
UsrDef_liq2	UsrDef_liq2.txt	Liquid defined by the user
UsrDef_liq3	UsrDef_liq3.txt	Liquid defined by the user

The file of every fluid saves the physical-thermodynamic properties of the fluid needed for free-convection calculation like specific heat, conductivity, etc.

The list of available fluid is defined in the global enumerative variable called ThFluids.

3. Abstract Components

3.1 Abstract Component Node

3.1.1 Description

An abstract class for the definition of thermal nodes, with the ability to label nodes and calculate the minimum and the maximum temperatures reached.

3.1.2 Construction Parameters

Name	Type	Default	Description	Units
n	CONST INTEGER	1	Dimension of thermal port	-

3.1.3 Ports

Name	Type	Parameters	Direction	Cardinality	Description
tp_in	thermal	(n = n)	IN	[1, 10000]	Thermal inlet port

3.1.4 Data

Name	Type	Default	Description	Units
Label	STRING	"Node Label"	Node label identifier	-
qi	REAL	0	Impressed heat	*W*

3.1.5 Variables

Name	Type	Initial	Description	Units
q	REAL		Total heat into node	*W*
Tmin	REAL		Minimum temperature reached	*K*
Tmax	REAL		Maximum temperature reached	*K*

3.1.6 Topology

The following statement helps to simplify the connecting statements when defining thermal networks using the EcosimPro Language (EL language).

PATH tp_in TO tp_in

3.1.7 Formulation

Print the Node Label and its Temperature

It is possible to print on the screen the label and the temperature of the nodes at a specific point in time when one switches the global variable PRINT_TEMP to TRUE.

Calculation of the Maximum and the Minimum Temperatures

After each integration interval the maximum and minimum temperatures in the node are calculated by the thermal functions Bmax(xmax, x) and Bmin(xmin, x). In this case:

Bmax(Tmax, T)

Bmin(Tmin, T)

These functions Bmax(xmax, x) and Bmin(xmin, x) calculate the maximum and the minimum values between the two function arguments and are stored in the variables xmax and xmin respectively.

Calculation of the Total Heat into the Node

The total heat into the node is equal to the sum of the impressed heat plus the overall heat flow coming through the inlet thermal port:

$$q_{tot} = q_{impress} + q_{in}$$

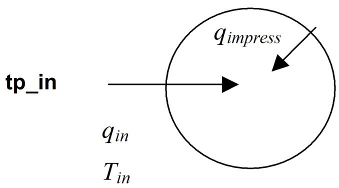
where:

qimpress: Impressed heat

qin: Total heat flow coming into the node through the vectorised thermal inlet port, therefore qin is the sum of all "n" contributions

$$q_{in} = \sum_{i \in 1,n} q_{in,i}$$

Qtot: Total heat flow into the node



3.2 Abstract Component Conductor

3.2.1 Description

An abstract class to derive the different types of thermal conductors by inheritance.

3.2.2 Construction Parameters

Name	Type	Default	Description	Units
n	CONST INTEGER	1	Dimension of thermal port	-

3.2.3 Ports

Name	Type	Parameters	Direction	Cardinality	Description
tp_in	thermal	(n = n)	IN	[0, 1]	Thermal inlet port
tp_out	thermal	(n = n)	OUT	[0, 1]	Thermal outlet port

3.2.4 Variables

Name	Type	Description	Units
q[n]	REAL	Heat flow	W

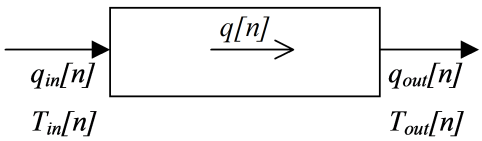
3.2.5 Topology

The following statement helps to simplify the connecting statements when defining thermal networks using the EcosimPro Language.

PATH tp_in TO tp_out

3.2.6 Formulation

Thermal conductors do not have any heat storage capacity by hypothesis, so the inlet heat flow is equal to the outlet heat flow, and it is also equal to the internal heat flow through the conductor:



$$q_{in,i} = q_i \qquad \textit{for all}\ i \in 1, n$$
$$q_{in,i} = q_{out,i} \qquad \textit{for all}\ i \in 1, n$$

3.3 Abstract Component ADnode

Inherited from component Node.

3.3.1 Description

This component type, named ADNode, represents a thermal diffusive node with no equation to calculate the thermal capacity. Its purpose is to enable the user to derive different types of diffusive thermal nodes by inheritance. The thermal capacitance can be provided as a datum or it can be calculated from an equation.

This component has the option of being used as a boundary node that is at constant temperature.

3.3.2 Construction Parameters

Name	Type	Default	Description	Units
n	CONST INTEGER	1	Dimension of thermal port	-

3.3.3 Ports

Name	Type	Parameters	Direction	Cardinality	Description
tp_in	thermal	(n = n)	IN	[1, 10000]	Thermal inlet port

3.3.4 Data

Name	Type	Default	Description	Units
Label	STRING	"Node Label"	Node label identifier	-
qi	REAL	0	Heater power	W
To	REAL	290	Initial temperature	K
Boundary	BOOLEAN	FALSE	Flag used to set boundary mode	-

3.3.5 Variables

Name	Type	Initial	Description	Units
q	REAL		Total Heat into node	W
Tmin	REAL		Minimum temperature reached	K
Tmax	REAL		Maximum temperature reached	K
T	REAL		Temperature	K
VC	REAL		Variable Heat Capacity	J/K

3.3.6 Formulation

The following mathematical expressions apply to this component type:

Initialisation of the Temperature:

$$T = To$$

The rate of change of the node temperature is calculated depending on the value of the Boolean variable named Boundary. If this variable is false then the rate of change of the node temperature is calculated from:

$$\frac{\partial T}{\partial t} = \frac{q}{VC}$$

where:

q: Total heat flow into the node

VC: Thermal capacitance of the node

But if this variable is true then the rate of change of the node temperature is considered equal to zero.

The temperature of the thermal inlet port is equal to the diffusive node temperature:

$$Tin = T$$

Inherited from Abstract Component Node

It is possible to print on the screen the label and the temperature of the nodes at a specific point in time when one switches the global variable PRINT_TEMP to TRUE.

After each integration interval the maximum and minimum temperatures in the node are calculated by the thermal functions Bmax(xmax, x) and Bmin(xmin, x). In this case:

Bmax(Tmax, T)

Bmin(Tmin, T)

These functions Bmax(xmax, x) and Bmin(xmin, x) calculate the maximum and the minimum values between the two function arguments and are stored in the variables xmax and xmin respectively.

The total heat into the node is equal to the sum of the impressed heat plus the overall heat flow coming through the inlet thermal port:

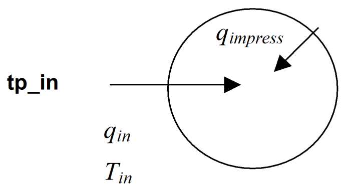
where:

qimpress: Impressed heat

qin: Total heat flow coming into the node through the vectorised thermal inlet port, therefore qin is the sum of all "n" contributions

$$q_{in} = \sum_{i \in 1,n} q_{in,i}$$

qtot: Total heat flow into the node



3.4 Abstract Component AGL

Inherited from component Conductor.

3.4.1 Description

This component type, named AGL, represents a thermal linear conductor where the heat flow is the temperature difference times a thermal conductance. This component does not provide any equation to calculate the thermal conductance, so its purpose is to enable the library developer to derive different types of linear conductors.

3.4.2 Construction Parameters

Name	Type	Default	Description	Units
n	CONST INTEGER	1	Dimension of thermal port	-

3.4.3 Ports

Name	Type	Parameters	Direction	Cardinality	Description
tp_in	thermal	(n = n)	IN	[0, 1]	Thermal inlet port
tp_out	thermal	(n = n)	OUT	[0, 1]	Thermal outlet port

3.4.4 Variables

Name	Type	Description	Units
q[n]	REAL	Heat flow	*W*
Vcond[n]	REAL	Variable thermal conductance per array item	*W/K*

3.4.5 Topology

The following statement helps to simplify the connecting statements when defining thermal networks using the EcosimPro Language.

PATH tp_in TO tp_out

3.4.6 Formulation

The mathematical equations described in this section are employed to represent the behaviour of this component.

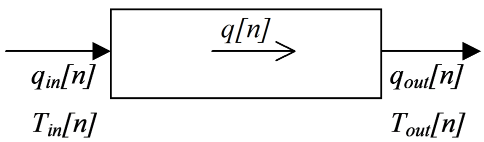
Heat Transport by Conduction

$$q_i = Vcond_i \left(T_{in,i} - T_{out,i} \right)$$

where Vcond is the thermal conductance at each array item.

Inherited from Abstract Component Conductor

Thermal conductors do not have any heat storage capacity by hypothesis, so the inlet heat flow is equal to the outlet heat flow, and it is also equal to the internal heat flow through the conductor:



$$q_{in,i} = q_i \qquad for\,all\, i \in 1, n$$

$$q_{in,i} = q_{out,i} \qquad for\,all\, i \in 1, n$$

3.5 Abstract Component AGR

Inherited from component Conductor.

3.5.1 Description

This component type, named AGR, represents a thermal radiative connection. This component does not provide the value of the Radiative Exchange Factor, and its purpose is to enable the user to derive different types of radiative conductors.

3.5.2 Construction Parameters

Name	Type	Default	Description	Units
n	CONST INTEGER	1	Dimension of thermal port	-

3.5.3 Ports

Name	Type	Parameters	Direction	Cardinality	Description
tp_in	thermal	(n = n)	IN	[0, 1]	Thermal inlet port
tp_out	thermal	(n = n)	OUT	[0, 1]	Thermal outlet port

3.5.4 Variables

Name	Type	Description	Units
Q[n]	REAL	Heat flow	*W*
VREF[n]	REAL	Variable Radiative Exchange Factor per array item	*m2*

3.5.5 Topology

PATH tp_in TO tp_out

3.5.6 Formulation

The mathematical equations described in this section are employed to represent the behaviour of this component.

Heat Conduction by Radiation

$$q_i = \sigma \cdot VREF_i \left(T_{in,i}^{\ 4} - T_{out,i}^{\ 4} \right)$$

where:

σ : Stefan Boltzmann constant, i.e. 5.67.10-8 W/(m2 K4)

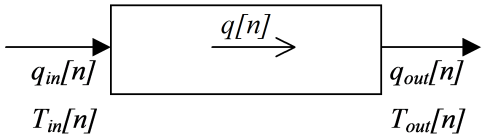
VREFi: Radiation Exchange Factor per array item, in m2

Tin: Inlet absolute temperature in K

Tout: Outlet absolute temperature in K

Inherited from Abstract Component Conductor

Thermal conductors do not have any heat storage capacity by hypothesis, so the inlet heat flow is equal to the outlet heat flow, and it is also equal to the internal heat flow through the conductor:



$$q_{in,i} = q_i \qquad for\, all\, i \in 1, n$$
$$q_{in,i} = q_{out,i} \qquad for\, all\, i \in 1, n$$

3.6 Component Tube

3.6.1 Description

Strictly speaking this component is not an Abstract Component, but it is presented in this section because it is a building block for the definition of the VCHP (variable conductance heat pipe) component. Hence it is not expected that it will be used on its own.

This component represents a section of a VCHP, i.e. a thin-walled tube containing a certain amount of condensable fluid as well as a quantity of non-condensing gas for control purposes.

The component calculates the different heat flows taking place in and out of the tube and through the fluid. The purpose of the control gas is to vary the length of active pipe determining the total inner contact surface of pipe with fluid and heat transfer characteristics of tube.

This component makes use of the Lvap function to calculate the active length of the pipe under given operational conditions and, additionally, the amount of non-condensing gas moles contained in tube.

3.6.2 Symbol

No symbol is needed for this component because it is only the basis of VCHP component.

3.6.3 Construction Parameters

Name	Type	Default	Description
n	CONST INTEGER	5	Number of nodes in tube

3.6.4 Ports

Name	Type	Parameters	Direction	Description
tp_1	thermal	(n = 1)	IN	Tube axial thermal inlet port
tp_N	thermal	(n = 1)	OUT	Tube axial thermal outlet port
tp_vapour	thermal	(n = n)	IN	Thermal inlet port for the radial connection of the inner side of the tube to the vapour
tp_wall	thermal_n	(n = n)	OUT	Thermal outlet port for radial connecting of the outer side of the tube
s_p	analog_signal	-	IN	Input signal port for the total pressure
s_ngas_in	analog_signal	-	IN	Input port for the number of moles of gas
s_ngas_out	analog_signal	-	OUT	Output port for the number of moles of gas
s_lvap	analog_signal	-	OUT	Vapour length measured from start of tube

3.6.5 Data

Name	Type	Default	Description	Units
l	REAL	1	Length of tube	*m*
D_o	REAL	0.02	Outer diameter of tube	*m*
D_i	REAL	0.018	Inner diameter of tube	*m*
A_vap	REAL	0.0001	Fluid cross section area in tube	*m2*
mat	ENUM Material	None	Material used for tube	-
rho	REAL	1000	Wall density if material is None	*Kg/m3*
cp	REAL	500	Wall specific heat if material is None	*J/kg K*
k	REAL	0.1	Wall conductivity if material is None	*W/m K*
h	REAL	1000	Vapour-tube heat transfer coefficient	*W/m2 K*

3.6.6 Variables

Name	Type	Description	Units
dx	REAL	Node length	*m*
P_outer	REAL	Perimeter of outer section	*m*
P_inner	REAL	Perimeter of inner section	*m*
A	REAL	Cross sectional area of tube wall	*m2*
A_i	REAL	Cross sectional inner area of tube	*m2*
rho_var	REAL	Density of wall material	*kg/m3*
cp_var	REAL	Specific heat wall material	*J/kg K*
k	REAL	Thermal conductivity of wall material	*W/m K*
C	REAL	Thermal capacitance of nodes	*J/K*
q[n+1]	REAL	Axial heat flow through nodes	*W*
q_vapour[n]	REAL	Heat flow from vapour to each node	*W*
T[n]	REAL	Temperature of nodes	*K*
chi[n]	REAL	Vapour-gas moles ratio at each section	-
l_vap	REAL	Vapour length measured from start of tube	*m*

3.6.7 Topology

The following statement helps to simplify the connecting statements when defining thermal networks using the EcosimPro Language.

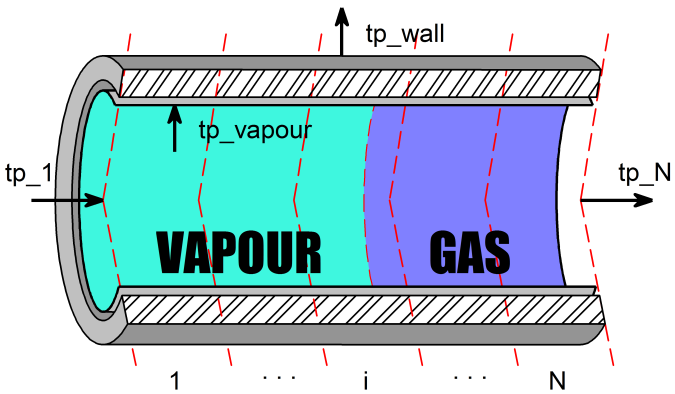
PATH tp_1 TO tp_N

3.6.8 Formulation

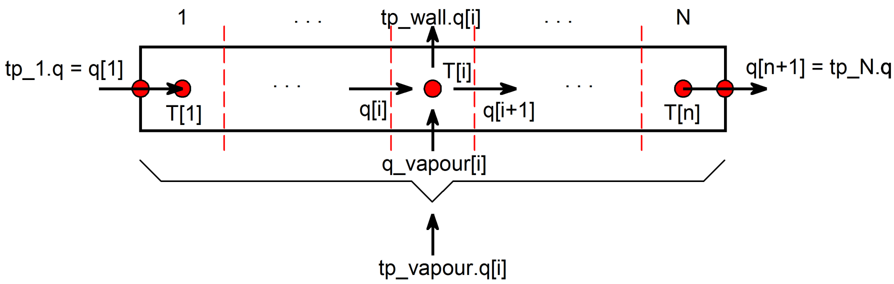
Nodalisation

The tube is divided into a number *n* of sections.

Array of variables *T[n]* contains the temperatures of the wall at each section, array of variables *q[n+1]* contains the heat flows across the wall sections plus the heat flows at the ends of the tube, and array of variables *q_-vapour[n]* contains the heat flows coming from the fluid. On the other hand, the vectorised port *tp_wall* contains the temperatures of the wall (*T[i]*) and outgoing heat flows through the wall at each section. Additionally, the vectorised port *tp_vapour* contains the temperature of the vapour and the total heat flow coming from the fluid.



Each section is modelled as a thermal node as shown below:



It is known that at the ends of the tube:

$$q_1 = tp\_1.q$$

$$q_{n+1} = tp\_N.q$$

We can divide the heat flows into their axial (through nodes) and radial components:

The axial internal heat flows are given by:

$$q_i = K \cdot A \cdot \frac{T_{i-1} - T_i}{dx}$$

and at the ends of the tube:

$$q_1 = K \cdot A \cdot \frac{tp\_1.T - T_1}{dx/2}$$

$$q_{n+1} = K \cdot A \cdot \frac{T_n - tp\_N.T}{dx/2}$$

The radial heat fluxes are given by:

$$q\_vapour_i = h \cdot (\xi_i \cdot P_{inner} \cdot dx) \cdot (tp\_vapour.T - T_i)$$

where ξi depends on the gas front position, it is equal to 1 when the node is filled with vapour, it is equal to zero when the node it is filled with gas, and its value is linearly interpolated between 0 and 1 when the front is within node i.

The corresponding energy equation for any section is:

$$cp_i \cdot \rho_i \cdot A \cdot dx \cdot \frac{dT_i}{dt} = (q_i - q_{i+1}) + (q\_vapour_i - tp\_wall.q_i)$$

Where:

cp: Specific heat of each node in the tube

ρ: Density of each node in the tube

A: Cross sectional are of tube wall

This page intentionally left blank.

4. Components

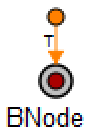
4.1 Component BNode

Inherited from component Node.

4.1.1 Description

This component type, named BNode, represents a thermal boundary node at which the temperature is a time-dependent boundary condition.

4.1.2 Symbol



4.1.3 Construction parameters

Name	Type	Default	Description
n	CONST INTEGER	1	Dimension of thermal port

4.1.4 Ports

Name	Type	Parameters	Direction	Cardinality	Description
s_temperature	analog signal	(n = n)	IN		Input signal port for temperature values
tp_in	thermal	(n = n)	IN	[1, 10000]	Thermal inlet port

4.1.5 Data

Name	Type	Default	Description	Units
Label	STRING	"Node Label"	Node label	-
qi	REAL	0	Impressed heat	W

4.1.6 Variables

Name	Type	Initial	Description	Units
q	REAL		Total heat into node	*W*
Tmin	REAL		Minimum temperature reached	*K*
Tmax	REAL		Maximum temperature reached	*K*

4.1.7 Formulation

The vectorised thermal port temperature is equal to the time-varying temperature signal:

```
tp_in[j] = s_temperature.signal[j]
```

Inherited from Abstract Component Node

It is possible to print on the screen the label and the temperature of the nodes at a specific point in time when one switches the global variable PRINT_TEMP to TRUE.

After each integration interval the maximum and minimum temperatures in the node are calculated by the thermal functions Bmax(xmax, x) and Bmin(xmin, x). In this case:

Bmax(Tmax, T)

Bmin(Tmin, T)

These functions Bmax(xmax, x) and Bmin(xmin, x) calculate the maximum and the minimum values between the two function arguments and are stored in the variables xmax and xmin respectively.

The total heat into the node is equal to the sum of the impressed heat plus the overall heat flow coming through the inlet thermal port:

$$q_{tot} = q_{impress} + q_{in}$$

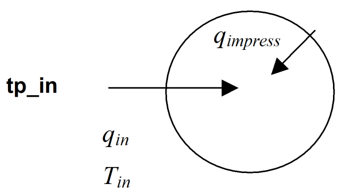
where:

qimpress: Impressed heat

qin: Total heat flow coming into the node through the vectorised thermal inlet port, therefore qin is the sum of all "n" contributions

$$q_{in} = \sum_{i \in 1,n} q_{in,i}$$

qtot: Total heat flow into the node

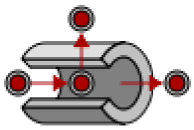


4.2 Component Cylinder

4.2.1 Description

This component type, named Cylinder, represents a pipe that is divided into a number of longitudinal and radial shapes (nodes). Heat flows are considered positive in the direction of increasing node index. For axial nodes this is from left to right, and for radial nodes this is from the centre of the cross-section to the outside.

4.2.2 Symbol



4.2.3 Construction parameters

Name	Type	Default	Description
nr	CONST INTEGER	3	Number of radial nodes
nz	CONST INTEGER	5	Number of axial nodes

4.2.4 Ports

Name	Type	Parameters	Direction	Description
tpr_in	thermal	(n = nz)	IN	Thermal inlet port - radial direction
tpr_out	thermal	(n = nz)	OUT	Thermal outlet port - radial direction
tpz_in	thermal	(n = nr)	IN	Thermal inlet port - axially
tpz_out	thermal	(n=nr)	OUT	Thermal outlet port - axially

4.2.5 Data

Name	Type	Default	Description	Units
Do	REAL	0.12	Outer diameter	*m*
Di	REAL	0.1	Inner diameter	*m*
L	REAL	1	Length	*m*
mat	ENUM Material	None	Material	-
cp	REAL	500	Wall specific heat if material is None	*J/kg K*
k	REAL	0.1	Wall conductivity if material is None	*W/m K*
rho	REAL	1000	Wall density if material is None	*Kg/m3*
TemperatureDependance	BOOLEAN	TRUE	Option to consider or not the temperature dependence on the material properties	
init_mode	ENUM WALL_INIT_MODE	Const_Temp	Option for the initialisation of the nodal temperatures	-
To	REAL	293	Initial temperature	*K*
To_linear	REAL [2]	{290, 290}	Initial temperatures at both sides if init_mode = Linear_Temp	*K*
To_table	TABLE_1D		Table with initial wall temperature versus non-dimensional position if init_mode = Table_Temp	*K*

Input data cp, k and rho will be used only if "mat" = None. If not, the properties will be interpolated using the corresponding data file.

4.2.6 Variables

Name	Type	Initial	Description	Units
F	CONST REAL	0.0001	Thermal capacity ratio assigned to the surface	-
T[nr,nz]	REAL		Nodal temperature	*K*
V[nr]	REAL		Nodal volume	*m3*
cp_var[nr, nz]	REAL		Nodal specific heat of wall material	*J/kg K*
cp_tpr[2, nz]	REAL		Nodal specific heat of outer-inner skin	*J/kg K*
cp_tpz[nr,2]	REAL		Nodal specific heat of first-last section skin	*J/kg K*
dr	REAL		Nodal distance in r	*m*
dz	REAL		Nodal distance in z	*m*
k_var[nr, nz]	REAL		Nodal thermal conductivities	*W/m K*
C_rad[nr+1,nz]	REAL		Radial equivalent conductance between the thermal nodes	*W/K*
C_axi[nr, nz+1]	REAL		Axial equivalent conductance between the thermal nodes	*W/K*
qr[nr +1, nz]	REAL		Heat flows between nodes in radial direction	*W*
qz[nr, nz+1]	REAL		Heat flows between nodes in axial direction	*W*
rho_var	REAL		Density of wall material	*Kg/m3*
ito	INTEGER		Pointer to time interval for To in property tables	-
icpk[nr,nz]	INTEGER		Pointer to last interpolation interval for each node	-
icpk_tpr[2, nz]	INTEGER		Pointer to last interpolation interval for each radial port node	-
icpk_tpz[nr, 2]	INTEGER		Pointer to last interpolation interval for each axial port node	-

4.2.7 Formulation

Nodalisation

The dimensions of the inlet and outlet vectorised thermal ports are defined by the values of *nz* (number of nodes in the axial direction into which the pipe is divided) and nr (number of nodes in the radial direction). The temperature at each node is stored in an array called *T[nr, nz]*.

Array *qr[nr + 1, nz]* contains the heat flows across nodes in the radial direction, whereas array *qz[nr, nz + 1]* contains the heat flows across nodes in the axial direction.

The nodal distance along the axis of the cylinder is then:

$$dz = \frac{L}{nz}$$

And the nodal distance in the radial direction of the cylinder is:

$$dr = \frac{(D_{out} - D_{in})}{2 \cdot nr}$$

where:

Dout: Outer diameter

Din: Inner diameter

The volume of each node is given by:

$$V_{i,j} = \frac{1}{4}\pi \cdot \left\{ (D_{in} + 2 \cdot i \cdot dr)^2 - (D_{in} + 2 \cdot (i-1) \cdot dr)^2 \right\} \cdot dz$$

The volume assigned to the ports is a percentage of the thermal capacity assigned to the closest node specified by the constant F.

Initialisation

The data named init_mode allows the user to specify the initial temperature distribution in the nodes. There are three options::

- Uniform initial temperature for all the nodes (init_mode = Constant_Temp)
- Linear initial temperature distribution in the radial direction and uniform initial temperature in the axial direction (init_mode = Linear_Temp)
- Initial temperature distribution given by a data table of the initial temperature as a function of the non-dimensional position in the radial direction (init_mode = Table_Temp). For the axial direction, the temperature distribution is considered uniform.

Heat balance

The following heat balance equation is applied on each node in the radial and axial directions

$$\rho_{\mathrm{var}} \cdot cp_{\mathrm{var}\,i,j} \cdot V_{i,j} \cdot \frac{\partial T_{i,j}}{\partial t} = qr_{i,j} - qr_{i+1,j} + qz_{i,j} - qz_{i,j+1}$$

Heat flows

The axial heat flows are given by

$$qz_{i,j} = \frac{k_{\mathrm{var}\,i,j-1} \cdot k_{\mathrm{var}\,i,j}}{\frac{k_{\mathrm{var}\,i,j}}{2} \cdot dz + \frac{k_{\mathrm{var}\,i,j-1}}{2} \cdot dz} \left(\frac{V_{i,j}}{dz} \right) \left(\frac{T_{i,j-1} - T_{i,j}}{dz} \right)$$

And the radial heat flows are given by

$$qr_{i,j} = 2 \cdot \pi \cdot dz \cdot \frac{1}{\frac{\log\left(\frac{D_{in} + (2 \cdot i - 2)dr}{D_{in} + (2 \cdot i - 3)dr} \right)}{k_{\mathrm{var}\,i-1,j}} + \frac{\log\left(\frac{D_{in} + (2 \cdot i - 1)dr}{D_{in} + (2 \cdot i - 2)dr} \right)}{k_{\mathrm{var}\,i,j}}} \cdot \left(T_{i-1,j} - T_{i,j} \right)$$

where:

i: Index for the radial discretization

j: Index for the axial discretization

Ti,j: Temperature in the node [i,j]

kvar i,j Conductivity of the node [i,j]

qr i,j: Radial heat flow between the nodes [i-1,j] y [i,j] qz i,j: Axial heat flow between the nodes [i, j-1] y [i,j]

Axial boundary conditions

In the radial direction, a certain thermal inertia of the closest node (given by F) is assigned to the thermal inlet and outlet ports. This is done to prevent the appearance of algebraic loops when calculating the heat flows at the ports.

$$cp\_tpz_{i,1} \cdot F \cdot V_{i,1} \cdot \rho_{\text{var}} \frac{\partial\, tpz\_in.Tk_i}{\partial t} = tpz\_in.q_i - qz_{i,1}$$

$$cp\_tpz_{i,2} \cdot F \cdot V_{i,nz} \cdot \rho_{\text{var}} \frac{\partial\, tpz\_out.Tk_i}{\partial t} = qz_{i,nz+1} - tpz\_out.q_i$$

where:

F: Thermal capacity ratio assigned to the ports

tpz_in.q: Heat flow at the axial inlet thermal port

tpz_out.q: Heat flow at the axial outlet thermal port

tpz_in.Tk: Temperature at the axial inlet thermal port

tpz_out.Tk: Temperature at the axial outlet thermal port

qzi,1: Heat flow between the thermal port and the node [i,1]

qzi,nz+1: Heat flow between the thermal port and the node [i,nz]

These last heat flows are computed as follows:

$$qz_{i,1} = 2 \cdot k_{\text{var}\, i,j} \cdot \frac{V_{i,1}}{dz} \cdot \frac{(tpz\_in.Tk_i - T_{i,1})}{dz}$$

$$qz_{i,nz+1} = 2 \cdot k_{\text{var}\, i,nz} \cdot \frac{V_{i,nz}}{dz} \frac{(T_{i,nz} - tpz\_out.Tk_i)}{dz}$$

Radial boundary conditions

In the radial direction, a certain thermal inertia (given by F) is assigned to the thermal inlet and outlet ports. This is done to prevent the appearance of algebraic loops when calculating the heat flows at the ports.

$$cp\_tpr_{1,j} \cdot F \cdot V_{1,j} \cdot \rho_{\text{var}} \frac{\partial\, tpr\_in.Tk_j}{\partial t} = tpr\_in.q_i - qr_{1,j}$$

$$cp\_tpr_{2,j} \cdot F \cdot V_{nr,j} \cdot \rho_{\text{var}} \frac{\partial\, tpr\_out.Tk_j}{\partial t} = qr_{nr+1,j} - tpr\_out.q_j$$

where:

F: Thermal capacity ratio assigned to the ports

tpr_in.q: Heat flow at the radial inlet thermal port tpr_out.q: Heat flow at the radial outlet thermal port

tpr_in.Tk: Temperature at the radial inlet thermal port

tpr_out.Tk: Temperature at the radial outlet thermal port

qr1,j: Heat flow between the thermal port and the node [1,j]

qrnr+1,j: Heat flow between the thermal port and the node [nr+1,j]

These last heat flows are computed as follows:

$$qr_{1,j} = \frac{2 \cdot \pi \cdot dz \cdot k_{\mathrm{var}\,1,j}}{\log\left(1 + \frac{dr}{D_{in}}\right)} \cdot \left(tpr\_in.Tk_j - T_{1,j}\right)$$

$$qr_{nr+1,j} = \frac{2 \cdot \pi \cdot dz \cdot k_{\mathrm{var}\,nr,j}}{\log\left(\frac{D_{out}}{D_{out} - dr}\right)} \cdot \left(T_{nr,j} - tpr\_out.Tk_j\right)$$

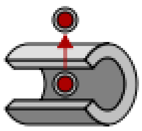
4.3 Component Cylinder_AI

Inherited from component Cylinder.

4.3.1 Description

This component, named Cylinder_AI, is similar to component Cylinder but its axial ends are isolated, which means that the axial ports tpz_in and tpz_out are closed.

4.3.2 Symbol



Cylinder_AI

4.3.3 Construction Parameters

Name	Type	Default	Description
nr	CONST INTEGER	3	Number of radial nodes
nz	CONST INTEGER	5	Number of axial nodes

4.3.4 Ports

Name	Type	Parameters	Direction	Description
tpr_in	thermal	(n = nz)	IN	Thermal inlet port - radial direction
tpr_out	thermal	(n = nz)	OUT	Thermal outlet port - radial direction

4.3.5 Data

Name	Type	Default	Description	Units
Do	REAL	0.12	Outer diameter	*m*
Di	REAL	0.1	Inner diameter	*m*
L	REAL	1	Length	*m*
mat	ENUM Material	None	Material	-
cp	REAL	500	Wall specific heat if material is None	*J/kg K*
k	REAL	0.1	Wall conductivity if material is None	*W/m K*
rho	REAL	1000	Wall density if material is None	*Kg/m3*
TemperatureDependance	BOOLEAN	TRUE	Option to consider or not the temperature dependence on the material properties	
init_mode	ENUM WALL_INIT_MODE	Const_Temp	Option for the initialisation of the nodal temperatures	-
To	REAL	293	Initial temperature	*K*
To_linear	REAL [2]	{290, 290}	Initial temperatures at both sides if init_mode = Linear_Temp	*K*
To_table	TABLE_1D		Table with initial wall temperature versus non-dimensional position if init_mode = Table_Temp	*K*

Input data cp, k and rho will be used only if "mat" = None. If not, the properties will be interpolated using the corresponding data file.

4.3.6 Variables

Name	Type	Initial	Description	Units
F	CONST REAL	0.0001	Thermal capacity ratio assigned to the surface	-
T[nr,nz]	REAL		Nodal temperature	*K*
V[nr]	REAL		Nodal volume	*m3*
cp_var[nr, nz]	REAL		Nodal specific heat of wall material	*J/kg K*
cp_tpr[2, nz]	REAL		Nodal specific heat of outer-inner skin	*J/kg K*
cp_tpz[nr,2]	REAL		Nodal specific heat of first-last section skin	*J/kg K*
dr	REAL		Nodal distance in r	*m*
dz	REAL		Nodal distance in z	*m*
k_var[nr, nz]	REAL		Nodal thermal conductivities	*W/m K*
C_rad[nr+1,nz]	REAL		Radial equivalent conductance between the thermal nodes	*W/K*
C_axi[nr, nz+1]	REAL		Axial equivalent conductance between the thermal nodes	*W/K*
qr[nr +1, nz]	REAL		Heat flows between nodes in radial direction	*W*
qz[nr, nz+1]	REAL		Heat flows between nodes in axial direction	*W*
rho_var	REAL		Density of wall material	*Kg/m3*
ito	INTEGER		Pointer to time interval for To in property tables	
icpk[nr,nz]	INTEGER		Pointer to last interpolation interval for each node	
icpk_tpr[2, nz]	INTEGER		Pointer to last interpolation interval for each radial port node	
icpk_tpz[nr, 2]	INTEGER		Pointer to last interpolation interval for each axial port node	

4.3.7 Formulation

The cylinder is isolated at its ends by removing the axial ports as shown below:

```
DECLS
        CLOSE tpz_in
           CLOSE tpz_out
```

Inherited from component Cylinder

Nodalisation

The dimensions of the inlet and outlet vectorised thermal ports are defined by the values of *nz* (number of nodes in the axial direction into which the pipe is divided) and nr (number of nodes in the radial direction). The temperature at each node is stored in an array called *T[nr, nz]*.

Array *qr[nr + 1, nz]* contains the heat flows across nodes in the radial direction, whereas array *qz[nr, nz + 1]* contains the heat flows across nodes in the axial direction.

The nodal distance along the axis of the cylinder is then:

$$dz = \frac{L}{nz}$$

And the nodal distance in the radial direction of the cylinder is:

$$dr = \frac{(D_{out} - D_{in})}{2 \cdot nr}$$

where:

Dout: Outer diameter

Din: Inner diameter

The volume of each radial node is given by:

$$V_{i,j} = \frac{1}{4}\pi \cdot \left\{(D_{in} + 2 \cdot i \cdot dr)^2 - (D_{in} + 2 \cdot (i-1) \cdot dr)^2\right\} \cdot dz$$

The volume assigned to the ports is a percentage of the thermal capacity assigned to the closest node specified by the constant F.

Initialisation

The data named init_mode allows the user to specify the initial temperature distribution in the nodes. There are three options:

- Uniform initial temperature for all the nodes (init_mode = Constant_Temp)
- Linear initial temperature distribution in the radial direction and uniform initial temperature in the axial direction (init_mode = Linear_Temp)
- Initial temperature distribution given by a data table of the initial temperature as a function of the non-dimensional position in the radial direction (init_mode = Table_Temp). For the axial direction, the temperature distribution is considered uniform.

Heat balance

The following heat balance equation is applied on each node in the radial and axial directions

$$\rho_{\text{var}} \cdot cp_{\text{var}\,i,j} \cdot V_{i,j} \cdot \frac{\partial T_{i,j}}{\partial t} = qr_{i,j} - qr_{i+1,j} + qz_{i,j} - qz_{i,j+1}$$

Heat flows

The axial heat flows are given by

$$qz_{i,j} = \frac{k_{\text{var}\,i,j-1} \cdot k_{\text{var}\,i,j}}{\frac{k_{\text{var}\,i,j}}{2} \cdot dz + \frac{k_{\text{var}\,i,j-1}}{2} \cdot dz} \left(\frac{V_{i,j}}{dz}\right)\left(\frac{T_{i,j-1} - T_{i,j}}{dz}\right)$$

And the radial heat flows are given by

$$qr_{i,j} = 2 \cdot \pi \cdot dz \cdot \frac{1}{\frac{\log\left(\frac{D_{in} + (2 \cdot i - 2)dr}{D_{in} + (2 \cdot i - 3)dr}\right)}{k_{\text{var}\,i-1,j}} + \frac{\log\left(\frac{D_{in} + (2 \cdot i - 1)dr}{D_{in} + (2 \cdot i - 2)dr}\right)}{k_{\text{var}\,i,j}}} \cdot \left(T_{i-1,j} - T_{i,j}\right)$$

where:

i: Index for the radial discretization

j: Index for the axial discretization

Ti,j: Temperature in the node [i,j]

kvar i,j: Conductivity of the node [i,j]

qr i,j: Radial heat flow between the nodes [i-1,j] y [i,j]

qz i,j: Axial heat flow between the nodes [i, j-1] y [i,j]

Axial boundary conditions

In the radial direction, a certain thermal inertia of the closest node (given by F) is assigned to the thermal inlet and outlet ports. This is done to prevent the appearance of algebraic loops when calculating the heat flows at the ports.

$$cp\_tpz_{i,1} \cdot F \cdot V_{i,1} \cdot \rho_{\text{var}} \frac{\partial tpz\_in.Tk_i}{\partial t} = tpz\_in.q_i - qz_{i,1}$$

$$cp\_tpz_{i,2} \cdot F \cdot V_{i,nz} \cdot \rho_{\text{var}} \frac{\partial tpz\_out.Tk_i}{\partial t} = qz_{i,nz+1} - tpz\_out.q_i$$

where:

F: Thermal capacity ratio assigned to the ports

tpz_in.q: Heat flow at the axial inlet thermal port

tpz_out.q: Heat flow at the axial outlet thermal port

tpz_in.Tk: Temperature at the axial inlet thermal port

tpz_out.Tk: Temperature at the axial outlet thermal port

qzi,1: Heat flow between the thermal port and the node [i,1]

qzi,nz+1: Heat flow between the thermal port and the node [i,nz]

These last heat flows are computed as follows:

$$qz_{i,1} = 2 \cdot k_{\text{var}\,i,j} \cdot \frac{V_{i,1}}{dz} \cdot \frac{(tpz\_in.Tk_i - T_{i,1})}{dz}$$

$$qz_{i,nz+1} = 2 \cdot k_{\text{var}\,i,nz} \cdot \frac{V_{i,nz}}{dz} \cdot \frac{(T_{i,nz} - tpz\_out.Tk_i)}{dz}$$

Radial boundary conditions

In the radial direction, a certain thermal inertia (given by F) is assigned to the thermal inlet and outlet ports. This is done to prevent the appearance of algebraic loops when calculating the heat flows at the ports.

$$cp\_tpr_{1,j} \cdot F \cdot V_{1,j} \cdot \rho_{\text{var}} \frac{\partial\, tpr\_in.Tk_j}{\partial t} = tpr\_in.q_i - qr_{1,j}$$

$$cp\_tpr_{2,j} \cdot F \cdot V_{nr,j} \cdot \rho_{\text{var}} \frac{\partial\, tpr\_out.Tk_j}{\partial t} = qr_{nr+1,j} - tpr\_out.q_j$$

where:

F: Thermal capacity ratio assigned to the ports

tpr_in.q: Heat flow at the radial inlet thermal port

tpr_out.q: Heat flow at the radial outlet thermal port

tpr_in.Tk: Temperature at the radial inlet thermal port

tpr_out.Tk: Temperature at the radial outlet thermal port

qr1,j: Heat flow between the thermal port and the node [1,j]

qrnr+1,j: Heat flow between the thermal port and the node [nr+1,j]

These last heat flows are computed as follows:

$$qr_{1,j} = \frac{2 \cdot \pi \cdot dz \cdot k_{\text{var}\,1,j}}{\log\left(1 + \frac{dr}{D_{in}}\right)} \cdot (tpr\_in.Tk_j - T_{1,j})$$

$$qr_{nr+1,j} = \frac{2 \cdot \pi \cdot dz \cdot k_{\text{var}\,nr,j}}{\log\left(\frac{D_{out}}{D_{out} - dr}\right)} \cdot (T_{nr,j} - tpr\_out.Tk_j)$$

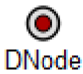
4.4 Component DNode

Inherited from component ADNode.

4.4.1 Description

This component type, named DNode, represents a thermal diffusive node with constant thermal capacity; i.e. the thermal capacity is given as a value.

4.4.2 Symbol



4.4.3 Construction Parameters

Name	Type	Default	Description
n	CONST INTEGER	1	Dimension of thermal port

4.4.4 Ports

Name	Type	Parameters	Direction	Cardinality	Description
tp_in	thermal	(n = n)	IN	[1, 10000]	Thermal inlet port

4.4.5 Data

Name	Type	Default	Description	Units
Label	STRING	"Node Label"	Node label	-
qi	REAL	0	Impressed heat	*W*
C	REAL	1.e-3	Heat capacity	*J/K*
To	REAL	290	Initial temperature	*K*
Boundary	BOOLEAN	FALSE	Flag used to set boundary mode	-

4.4.6 Variables

Name	Type	Description	Units
q	REAL	Total heat into node	*W*
Tmin	REAL	Minimum temperature reached	*K*
Tmax	REAL	Maximum temperature reached	*K*
T	REAL	Temperature	*K*
VC	REAL	Variable heat capacity	*J/K*

4.4.7 Formulation

The following mathematical expressions apply to this component type:

Initialisation of the Temperature:

T = To

If the node is considered as a boundary node with constant temperature (Boundary is TRUE) then the rate of change of the node temperature is zero. If the node is considered as a diffusive node the rate of change of the node temperature is calculated from:

$$\frac{\partial T}{\partial t} = \frac{q}{VC}$$

where:

q: Total heat flow into the node

VC: Thermal capacitance of the node; i.e. it is a datum:

VC= C

The temperature of the thermal inlet port is equal to the diffusive node temperature:

Tin = T

Inherited from Abstract Component Node

It is possible to print on the screen the label and the temperature of the nodes at a specific point in time when one switches the global variable PRINT_TEMP to TRUE.

After each integration interval the maximum and minimum temperatures in the node are calculated by the thermal functions Bmax(xmax, x) and Bmin(xmin, x). In this case:

Bmax(Tmax, T)

Bmin(Tmin, T)

These functions Bmax(xmax, x) and Bmin(xmin, x) calculate the maximum and the minimum values between the two function arguments and are stored in the variables xmax and xmin respectively.

The total heat into the node is equal to the sum of the impressed heat plus the overall heat flow coming through the inlet thermal port:

$$q_{tot} = q_{impress} + q_{in}$$

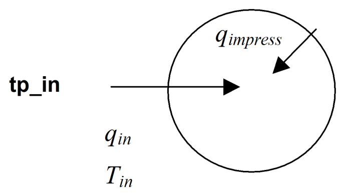
where:

qimpress: Impressed heat

qin: Total heat flow coming into the node through the vectorised thermal inlet port, therefore qin is the sum of all "n" contributions

$$q_{in} = \sum_{i \in 1,n} q_{in,i}$$

qtot: Total heat flow into the node



4.5 Component DNodeMat

Inherited from Component ADNode.

4.5.1 Description

This component type, named DNodeMat, represents a thermal diffusive node containing a mass of a specified material.

4.5.2 Symbol

4.5.3 Construction Parameters

Name	Type	Default	Description
n	CONST INTEGER	1	Dimension of thermal port

4.5.4 Ports

Name	Type	Parameters	Direction	Cardinality	Description
tp_in	thermal	(n = n)	IN	[1, 10000]	Thermal inlet port

4.5.5 Data

Name	Type	Default	Description	Units
Label	STRING	"Node Label"	Node label	-
cp	REAL	500	Specific heat if material is None	*J/kg K*
mass	REAL	1	Mass of the thermal node	*kg*
mat	ENUM Material	None	Material	-
qi	REAL	0	Impressed heat	*W*
To	REAL	290	Initial temperature	*K*
Boundary	BOOLEAN	FALSE	Flat used to set boundary node	-

Input data cp will be used only if "mat" = None. If not, the properties will be interpolated using the corresponding data file.

4.5.6 Variables

Name	Type	Initial	Description	Units
q	REAL		Total heat into node	*W*
Tmin	REAL		Minimum temperature reached	*K*
Tmax	REAL		Maximum temperature reached	*K*
T	REAL		Temperature	*K*
VC	REAL		Variable heat capacity	*J/K*
cp_var	REAL		Specific heat of wall material	*J/kg K*
ipt	REAL		Pointer to last table position	-

4.5.7 Formulation

The following mathematical expressions apply to this component type:

Initialisation of the Temperature:

T = To

If the node is considered as a boundary node with constant temperature (Boundary is TRUE) then the rate of change of the node temperature is zero. If the node is considered as a diffusive node the rate of change of the node temperature is calculated from:

$$\frac{\partial T}{\partial t} = \frac{q}{VC}$$

where:

q: Total heat flow into the node

VC: Thermal capacitance of the node, and its value is given by:

cp_var: Specific heat that is calculated interpolating in a data file if a material is specified in the thermal node (mat $\neq$ None)

The temperature of the thermal inlet port is equal to the diffusive node temperature:

Tin = T

Inherited from Abstract Component Node

It is possible to print on the screen the label and the temperature of the nodes at a specific point in time when one switches the global variable PRINT_TEMP to TRUE.

After each integration interval the maximum and minimum temperatures in the node are calculated by the thermal functions Bmax(xmax, x) and Bmin(xmin, x). In this case:

Bmax(Tmax, T)

Bmin(Tmin, T)

These functions Bmax(xmax, x) and Bmin(xmin, x) calculate the maximum and the minimum values between the two function arguments and are stored in the variables xmax and xmin respectively.

The total heat into the node is equal to the sum of the impressed heat plus the overall heat flow coming through the inlet thermal port:

$$q_{tot} = q_{impress} + q_{in}$$

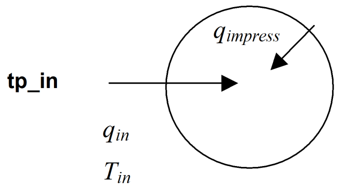
where:

qimpress Impressed heat

qin Total heat flow coming into the node through the vectorised thermal inlet port, therefore qin is the sum of all "n" contributions

$$q_{in} = \sum_{i \in 1,n} q_{in,i}$$

qtot Total heat flow into the node



4.6 Component DNodeMat_vec

4.6.1 Description

This component type, named DNodeMat_vec, represents a vectorized thermal diffusive node calculating n temperatures without considering heat exchange between the nodes.

4.6.2 Symbol

4.6.3 Construction Parameters

Name	Type	Default	Description
n	CONST INTEGER	1	Dimension of thermal port

4.6.4 Ports

Name	Type	Parameters	Direction	Cardinality	Description
tp_in	thermal	(n = n)	IN	[1, 10000]	Thermal inlet port

4.6.5 Data

Name	Type	Default	Description	Units
cp	REAL	500	Specific heat if material is None	*J/kg K*
mass	REAL	1	Mass of the thermal node	*kg*
mat	ENUM Material	None	Material	-
To	REAL	290	Initial temperature	*K*
Boundary	BOOLEAN	FALSE	Flat used to set boundary node	-

Input data cp will be used only if "mat" = None. If not, the properties will be interpolated using the corresponding data file.

4.6.6 Variables

Name	Type	Initial	Description	Units
q[n]	REAL		Heat into node	*W*
T[n]	REAL		Node temperature	*K*
VC[n]	REAL		Variable heat capacity	*J/K*
cp_var[n]	REAL		Specific heat of wall material	*J/kg K*
ipt	REAL		Pointer to last table position	-

4.6.7 Formulation

The following mathematical expressions apply to this component type:

Initialisation of the Temperature:

$$T[i] = To$$

If the nodes are considered as boundary nodes with constant temperature (Boundary is TRUE) then the rate of change of the node temperature is zero. If the nodes are considered as a diffusive node the rate of change of the node temperature is calculated from:

$$\frac{\partial T_i}{\partial t} = \frac{q_i}{VC_i}$$

where:

q_i: Heat flow into the node

VC_i: Thermal capacitance of the node, and its value is given by:

cp_var_i: Specific heat that is calculated interpolating in a data file if a material is specified in the thermal node (mat $\neq$ None)

The temperature of the thermal inlet port is equal to the diffusive node temperature.

The heat into the node is equal to the heat flow coming through the inlet thermal port. The component does not take into account conduction between the nodes.

4.7 Component DNodeRoom

Inherited from component ADNode.

4.7.1 Description

This component type, named DNodeRoom, is a thermal diffusive node which represents a room filled with a fluid.

4.7.2 Symbol

4.7.3 Construction Parameters

Name	Type	Default	Description
n	CONST INTEGER	1	Dimension of thermal port

4.7.4 Ports

Name	Type	Parameters	Direction	Cardinality	Description
tp_in	Thermal	(n = n)	IN	[1, 10000]	Thermal inlet port

4.7.5 Data

Name	Type	Default	Description	Units
Label	STRING	"Node Label"	Node label	-
qi	REAL	0	Impressed heat	*W*
To	REAL	290	Initial temperature	*K*
Boundary	BOOLEAN	FALSE	Flag used to set boundary mode	-
V	REAL	100	Volume of the room node	*m3*
P	REAL	101325	Fluid pressure in the room	*Pa*
thermal_fluid	ENUM ThFluids	Air	Fluid in the room	-

4.7.6 Variables

Name	Type	Description	Units
q	REAL	Total heat into node	*W*
Tmin	REAL	Minimum temperature reached	*K*
Tmax	REAL	Maximum temperature reached	*K*
T	REAL	Temperature	*K*
VC	REAL	Variable heat capacity	*J/K*
cp	REAL	Specific heat of the fluid in the room	*J/kg K*
rho	REAL	Density of the fluid in the room	*kg/m3*
ier	INTEGER	Function error identifier	-

4.7.7 Formulation

The following mathematical expressions apply to this component type:

Initialisation of the Temperature:

T = To

If the node is considered as a boundary node with constant temperature (Boundary is TRUE) then the rate of change of the node temperature is zero. If the node is considered as a diffusive node the rate of change of the node temperature is calculated from:

$$\frac{\partial T}{\partial t} = \frac{q}{VC}$$

where:

q: Total heat flow into the node

VC: Thermal capacitance of the node is calculated from the fluid properties and the volume of the node (V):

$$VC = V \cdot \rho \cdot cp$$

ρ: Density of the fluid calculated from the thermodynamic-physical property functions defined in the THERMAL library

cp: Specific heat of the fluid calculated from the thermodynamic-physical property functions defined in the THERMAL library

The temperature of the thermal inlet port is equal to the diffusive node temperature:

Tin = T

Inherited from Abstract Component Node

It is possible to print on the screen the label and the temperature of the nodes at a specific point in time when one switches the global variable PRINT_TEMP to TRUE.

After each integration interval the maximum and minimum temperatures in the node are calculated by the thermal functions Bmax(xmax, x) and Bmin(xmin, x). In this case:

Bmax(Tmax, T)

Bmin(Tmin, T)

These functions Bmax(xmax, x) and Bmin(xmin, x) calculate the maximum and the minimum values between the two function arguments and are stored in the variables xmax and xmin respectively.

The total heat into the node is equal to the sum of the impressed heat plus the overall heat flow coming through the inlet thermal port:

$$q_{tot} = q_{impress} + q_{in}$$

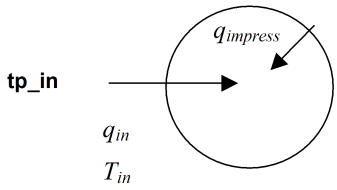
where:

qimpress: Impressed heat

qin: Total heat flow coming into the node through the vectorised thermal inlet port, therefore qin is the sum of all "n" contributions

$$q_{in} = \sum_{i \in 1,n} q_{in,i}$$

qtot: Total heat flow into the node



4.8 DNphases

Inherited from component Node.

4.8.1 Description

This component type, named DNphases, represents a diffusive thermal node that takes into account possible phase changes of the material.

4.8.2 Symbol

4.8.3 Construction Parameters

Name	Type	Default	Description
n	CONST INTEGER	1	Dimension of thermal port

4.8.4 Ports

Name	Type	Parameters	Direction	Cardinality	Description
tp_in	Thermal	(n = n)	IN	[1, 10000]	Thermal inlet port

4.8.5 Data

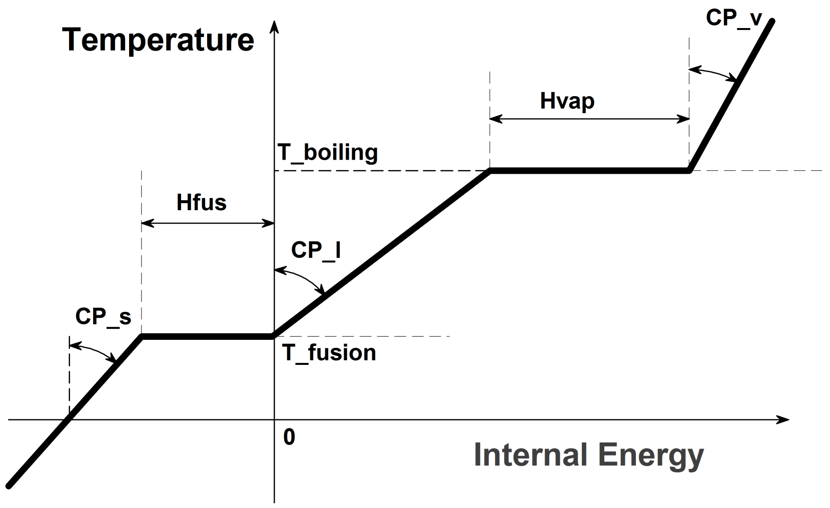
Name	Type	Default	Description	Units
Label	STRING	"Node Label"	Node label	-
qi	REAL	0	Impressed heat	*W*
Mass	REAL	1	Mass of water	*Kg*
To	REAL	290	Initial temperature of water	*K*
CP_l	REAL	4186	Specific heat of liquid	*J/kg K*
CP_s	REAL	2039	Specific heat of solid	*J/kg K*
CP_v	REAL	1805	Specific heat of vapour	*J/kg K*
Hfus	REAL	334000	Enthalpy of fusion	*J/kg*
Hvap	REAL	2501000	Enthalpy of vaporisation	*J/kg*
T_fusion	REAL	273.15	Fusion or melting temperature	*K*
T_boiling	REAL	373.15	Boiling temperature	*K*

4.8.6 Variables

Name	Type	Description	Units
q	REAL	Total heat into node	*W*
Tmin	REAL	Minimum temperature reached	*K*
Tmax	REAL	Maximum temperature reached	*K*
u	REAL	Internal Energy	*J/kg*
Uv	REAL	Internal energy corresponding to saturated vapour	*J/kg*
Ul	REAL	Internal energy corresponding to saturated liquid	*J/kg*
Us	REAL	Internal energy corresponding to solid at fusion temperature	*J/kg*
T	REAL	Temperature	*K*

4.8.7 Formulation

It is assumed that the node material has two phase changes: solid-liquid and liquid-vapour, and that the reference state for zero internal energy is the saturated liquid (see next figure).



The derivative of the specific internal energy (u) is equal to the total heat (q) divided by the mass (M):

$$\frac{\partial u}{\partial t} = \frac{q}{M}$$

From the specific internal energy, the temperature can be calculated using the relationship expressed by the above figure:

T = f(u)

Inherited from Abstract Component Node

It is possible to print on the screen the label and the temperature of the nodes at a specific point in time when one switches the global variable PRINT_TEMP to TRUE.

After each integration interval the maximum and minimum temperatures in the node are calculated by the thermal functions Bmax(xmax, x) and Bmin(xmin, x). In this case:

Bmax(Tmax, T)

Bmin(Tmin, T)

These functions Bmax(xmax, x) and Bmin(xmin, x) calculate the maximum and the minimum values between the two function arguments and are stored in the variables xmax and xmin respectively.

The total heat into the node is equal to the sum of the impressed heat plus the overall heat flow coming through the inlet thermal port:

$$q_{tot} = q_{impress} + q_{in} \quad \frac{\partial u}{\partial t} = \frac{q}{M}$$

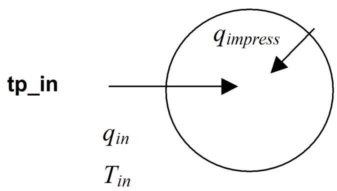
where:

qimpress: Impressed heat

qin: Total heat flow coming into the node through the vectorised thermal inlet port, therefore qin is the sum of all "n" contributions

$$q_{in} = \sum_{i \in 1,n} q_{in,i}$$

qtot: Total heat flow into the node

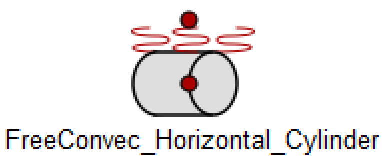


4.9 Component FreeConvec_Horizontal_Cylinder

4.9.1 Description

This component, named FreeConvec_Horizontal_Cylinder, represents a coupling to calculate natural convection from a surface of a horizontal cylinder to an undisturbed fluid.

4.9.2 Symbol



4.9.3 Construction Parameters

Name	Type	Default	Description
n	CONST INTEGER	1	Dimension of thermal port
thermal_fluid	ENUM ThFluids	Air	Interface fluid

4.9.4 Ports

Name	Type	Parameters	Direction	Description
tp_surface	thermal	(n = n)	IN	Surface port
tp_fluid	thermal	(n = n)	OUT	Fluid port

4.9.5 Data

Name	Type	Default	Description	Units
d	REAL	1	Diameter of the cylinder	*m*
L	REAL	0.1	Length of the cylinder	*m*
P	REAL	101325	Ambient pressure	*Pa*
Grav	REAL	9.80665	Gravity	*m/s2*

4.9.6 Variables

Name	Type	Default	Description	Units
q[n]	REAL		Heat flow	*W*
DT[n]	REAL	-	Temperature difference	*K*
Cp[n]	REAL		Specific heat	*J/kg K*
vis[n]	REAL		Viscosity	*Pa s*
rho[n]	REAL		Density	*kg/m3*
Tm[n]	REAL		Average temperature	*K*
B[n]	REAL		Beta factor	*1/K*
Gr[n]	REAL		Grashof number	-
Pr[n]	REAL		Prandtl number	-
Ra[n]	REAL		Rayleigh number	-
K[n]	REAL		Thermal conductivity	*W/m K*
h[n]	REAL		Convective heat transfer coefficient	*W/m2 K*
Nu[n]	REAL		Nusselt number	-
A	REAL		Surface	*m2*
ier	INTEGER	0	Error code from fluid data calculation	-

4.9.7 Formulation

The heat flow exchanged between the cylinder surface and the fluid is calculated as follows:

$$q_i = \left(\frac{A}{n}\right) \cdot h_i \cdot \left(T_{i,surface} - T_{i,fluid}\right)$$

where:

hi: Heat transfer coefficient

Ti, surface: Temperature of the cylinder surface

Ti, fluid: Temperature of the fluid

A: Heat transfer surface: $A = \pi \cdot d \cdot L$

d: Diameter of the cylinder

L: Length of the cylinder

The heat transfer coefficient is computed from the Nusselt number calculated from the THERMAL library function called NusseltCylinder.

```
Nu[i] = NusseltCylinder(Pr[i],Ra[i])
```

where:

Nu: Nusselt number

Pr: Prandtl number

Ra: Rayleigh number

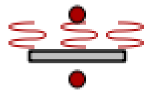
For a description of this function see reference [5.5.

4.10 Component FreeConvec_Plate

4.10.1 Description

This component, named FreeConvec_Plate, represents a coupling to calculate natural convection from a surface of a plate to an undisturbed fluid.

4.10.2 Symbol



4.10.3 Construction Parameters

Name	Type	Default	Description
n	CONST INTEGER	1	Dimension of thermal port
position	ENUM WallType	wall	Position of the plate with regard to the fluid
thermal_fluid	ENUM ThFluids	Air	Interface fluid

4.10.4 Ports

Name	Type	Parameters	Direction	Description
tp_surface	thermal	(n = n)	IN	Surface port
tp_fluid	thermal	(n = n)	OUT	Fluid port

4.10.5 Data

Name	Type	Default	Description	Units
A	REAL	1	Convective heat exchange are	*m2*
Lc	REAL	0.1	Height for vertical plate OR perimeter for floor and ceiling option	*m*
P	REAL	101325	Ambient pressure	*Pa*
Grav	REAL	9.80665	Gravity	*m/s2*

4.10.6 Variables

Name	Type	Default	Description	Units
q[n]	REAL		Heat flow	*W*
DT[n]	REAL	-	Temperature difference	*K*
Cp[n]	REAL		Specific heat	*J/kg K*
vis[n]	REAL		Viscosity	*Pa s*
rho[n]	REAL		Density	*kg/m3*
Tm[n]	REAL		Average temperature	*K*
B[n]	REAL		Beta factor	*1/K*
Gr[n]	REAL		Grashof number	-
Pr[n]	REAL		Prandtl number	-
Ra[n]	REAL		Rayleigh number	-
K[n]	REAL		Thermal conductivity	*W/m K*
h[n]	REAL		Convective heat transfer coefficient	*W/m2 K*
Nu[n]	REAL		Nusselt number	-
L[n]	REAL		Characteristic length	*m*
A	REAL		Surface	*m2*
ier	INTEGER	0	Error code from fluid data calculation	-

4.10.7 Formulation

The heat flow exchanged between the plate surface and the fluid is calculated as follows:

$$q_i = \left(\frac{A}{n}\right) \cdot h_i \cdot \left(T_{i,surface} - T_{i,fluid}\right)$$

where:

hi: Heat transfer coefficient

Ti, surface: Temperature of the cylinder surface

Ti, fluid: Temperature of the fluid

A: Convective heat exchange area

The heat transfer coefficient is computed from the Nusselt number calculated from the THERMAL library functions depending on the position of the plate with regard to the fluid:

```
EXPAND (position == wall) Nu[i] = NusseltWall(Pr[i],Ra[i])
EXPAND (position == floor) Nu[i] = NusseltFloor(Ra[i], tp_surface.Tk[i],
  tp_fluid.Tk[i])
EXPAND (position == ceil) Nu[i] = NusseltCeiling(Ra[i],tp_surface.Tk[i],
  tp_fluid.Tk[i])
```

where:

Nu: Nusselt number

NusseltWall: Function to calculate the natural convection correlation for a vertical plate

NusseltFloor: Function to calculate the natural convection correlation for a upward facing plate

NusseltCeiling: Function to calculate the natural convection correlation for a downward facing plate

Pr: Prandtl number

Ra: Rayleigh number

tp_surface.Tk: Temperature of the plate surface

tp_fluid.Tk: Temperature of the undisturbed fluid

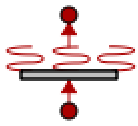
For a description of the aforementioned functions see references 5.4. , 5.6. and 5.7.

4.11 Component FreeConvec_UserDefined

4.11.1 Description

This component, named FreeConvec_UserDefined, represents a coupling to calculate natural convection between two surfaces with a fluid between them. The user specifies the free convection heat transfer coefficient as a function of the temperature difference between the two surfaces.

4.11.2 Symbol



4.11.3 Construction Parameters

Name	Type	Default	Description	Units
n	CONST INTEGER	1	Dimension of thermal port	-

4.11.4 Ports

Name	Type	Parameters	Direction	Cardinality	Description
tp_in	thermal	(n = n)	IN	[0, 1]	Thermal inlet port
tp_out	thermal	(n = n)	OUT	[0, 1]	Thermal outlet port

4.11.5 Data

Name	Type	Default	Description	Units
A	REAL	1	Area to thermal path ratio	*m2*
h_table	TABLE_1D		Heat transfer coefficient vs Temperature difference	*W/m2 K vs. K*

4.11.6 Variables

Name	Type	Description	Units
q[n]	REAL	Heat flow	*W*
Vcond[n]	REAL	Variable thermal conductance per array item	*W/K*

4.11.7 Topology

The following statement helps to simplify the connecting statements when defining thermal networks using the EcosimPro Language.

PATH tp_in TO tp_out

4.11.8 Formulation

The mathematical equations described in this section are employed to represent the behaviour of this component.

The heat flow by free convection is calculated as follows:

$$q_i = Vcond_i \quad \left(T_{in,i} - T_{out,i}\right)$$

where:

Vcondi: Thermal conductance at each array item that is calculated interpolating in the data table given by the user:

Vcond[i] = (A/n) * linearInterp1D(h_table, abs(tp_in.Tk[i] - tp_out.Tk[i]))

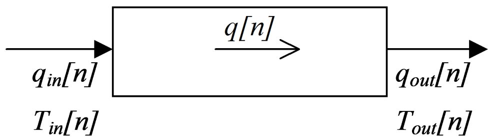
A: Area to thermal path ratio

tp_in.Tk: Temperature of the hot surface

tp_out.Tk: Temperature of the cold surface

Inherited from Abstract Component Conductor

Thermal conductors do not have any heat storage capacity by hypothesis, so the inlet heat flow is equal to the outlet heat flow, and it is also equal to the internal heat flow through the conductor:



$$q_{in,i} = q_i \qquad for\,all\, i \in 1, n$$

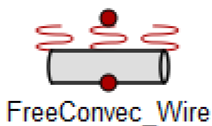
$$q_{in,i} = q_{out,i} \qquad for\,all\, i \in 1, n$$

4.12 Component FreeConvec_Wire

4.12.1 Description

This component, named FreeConvec_Wire, represents a coupling to calculate natural convection from a surface of a wire to an undisturbed fluid.

4.12.2 Symbol



4.12.3 Construction Parameters

Name	Type	Default	Description
n	CONST INTEGER	1	Dimension of thermal port
position	ENUM WIRE_POSITION	Horizontal	Wire Position
thermal_fluid	ENUM ThFluids	Air	Interface fluid

4.12.4 Ports

Name	Type	Parameters	Direction	Description
tp_surface	thermal	(n = n)	IN	Surface port
tp_fluid	thermal	(n = n)	OUT	Fluid port

4.12.5 Data

Name	Type	Default	Description	Units
d	REAL	0.1	Diameter of the wire	*m*
L	REAL	1	Length of the wire	*m*
P	REAL	101325	Ambient pressure	*Pa*
Grav	REAL	9.80665	Gravity	*m/s2*

4.12.6 Variables

Name	Type	Default	Description	Units
q[n]	REAL		Heat flow	*W*
DT[n]	REAL	-	Temperature difference	*K*
Cp[n]	REAL		Specific heat	*J/kg K*
vis[n]	REAL		Viscosity	*Pa s*
rho[n]	REAL		Density	*kg/m3*
Tm[n]	REAL		Average temperature	*K*
B[n]	REAL		Beta factor	*1/K*
Gr[n]	REAL		Grashof number	-
Pr[n]	REAL		Prandtl number	-
Ra[n]	REAL		Rayleigh number	-
K[n]	REAL		Thermal conductivity	*W/m K*
h[n]	REAL		Convective heat transfer coefficient	*W/m2 K*
Nu[n]	REAL		Nusselt number	-
A	REAL		Surface	*m2*
ier	INTEGER	0	Error code from fluid data calculation	-

4.12.7 Formulation

The heat flow exchanged between the plate surface and the fluid is calculated as follows:

$$q_i = \left(\frac{A}{n}\right) \cdot h_i \cdot \left(T_{i,surface} - T_{i,fluid}\right)$$

where:

hi: Heat transfer coefficient

Ti, surface: Temperature of the cylinder surface

Ti, fluid: Temperature of the fluid

A: Heat transfer surface of the wire calculated as follows: $A = \pi \cdot d \cdot L$

d: Wire diameter

L: Wire length

The heat transfer coefficient is computed from the Nusselt number calculated from the THERMAL library functions depending on the position of the wire:

```
EXPAND (position == Horizontal) Nu[i] = NusseltWireHorizontal(Pr[i],Ra[i], d
  )
EXPAND (position == Vertical) Nu[i] = NusseltWireVertical(Pr[i],Ra[i], d, L)
```

where:

Nu: Nusselt number

NusseltWireHorizontal: Function to calculate the natural convection correlation for a vertical wire

NusseltWireVertical: Function to calculate the natural convection correlation for a horizontal wire

Pr: Prandtl number

Ra: Rayleigh number

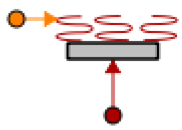
For a description of these functions see reference [5.8.], [5.9.]

4.13 Component FreeConvec_simple

4.13.1 Description

This component, named FreeConvec_simple, represents a coupling to calculate natural convection from a surface to a simplified fluid. The temperature of the fluid is given as an input control signal s_fluid_tem.

4.13.2 Symbol



FreeConvec_simple

4.13.3 Construction Parameters

Name	Type	Default	Description
n	CONST INTEGER	1	Dimension of thermal port

4.13.4 Ports

Name	Type	Parameters	Direction	Description
s_fluid_tem	analog_signal	(n = 1)	IN	Signal port for the fluid temperature
tp_in	thermal	(n = n)	IN	Thermal outlet port

4.13.5 Data

Name	Type	Default	Description	Units
A	REAL	1	Convective heat exchange area	*m2*
Lc	REAL	0.1	Characteristic length	*m*
b_l	REAL	1.32	Coefficient b in hc = b*(DT/Lc)^0.25 for laminar regime	*W/m1.75 K1.25*
b_t	REAL	1.24	Coefficient b in hc = b*DT^0.3333 for turbulent regime	*W/m2 K1.333*

4.13.6 Variables

Name	Type	Default	Description	Units
DT[n]	REAL	-	Temperature difference	*K*
T_fluid	REAL	293.15	Fluid temperature	*K*
hc[n]	REAL	-	Heat transfer coefficient	*W/m2/K*
hc_l[n]	REAL	-	Laminar heat transfer coefficient	*W/m2/K*
hc_t[n]	REAL	-	Turbulent heat transfer coefficient	*W/m2/K*
n_exp	CONST REAL	20	Exponent for laminar turbulent transition	-

4.13.7 Formulation

The temperature difference DT is considered as the absolute value of the difference between the input port temperature and the fluid temperature,

$$DT_i = abs(tp\_in.Tk_i - T_{fluid})$$

Then the heat transfer coefficient for laminar flow is given by:

$$hc\_l_i = b\_l \cdot \left(\frac{DT_i}{L_c}\right)^{\frac{1}{4}}$$

And the heat transfer coefficient for turbulent flow is given by:

$$hc\_t_i = b\_t \cdot (DT_i)^{\frac{1}{3}}$$

So the overall heat transfer coefficient is:

$$hc_i = \left\{hc\_l_i^{n\exp} + hc\_t_i^{n\exp}\right\}^{\frac{1}{n\exp}}$$

The heat flow is then calculated as:

$$tp\_in.q_i = hc_i \cdot \frac{A}{n} \cdot (tp\_in.Tk_i - T_{fluid})$$

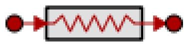
4.14 Component GL

Inherited from component AGL.

4.14.1 Description

This component type, named GL, represents a linear thermal conductor with constant thermal conductance.

4.14.2 Symbol



4.14.3 Construction Parameters

Name	Type	Default	Description
n	CONST INTEGER	1	Dimension of thermal port

4.14.4 Ports

Name	Type	Parameters	Direction	Cardinality	Description
tp_in	thermal	(n = n)	IN	[0, 1]	Thermal inlet port
tp_out	thermal	(n = n)	OUT	[0, 1]	Thermal outlet port

4.14.5 Data

Name	Type	Default	Description	Units
cond	REAL	0	Total thermal conductance	*W/K*

4.14.6 Variables

Name	Type	Description	Units
q[n]	REAL	Total heat	*W*
Vcond[n]	REAL	Variable thermal conductance	*W/K*

4.14.7 Topology

The following statement helps to simplify the connecting statements when defining thermal networks using the EcosimPro Language.

PATH tp_in TO tp_out

4.14.8 Formulation

The mathematical equations described in this section are employed to represent the behaviour of this component.

The total thermal conductance is distributed among the port connections

Since the thermal port is in vectorised form, connecting a GL component (with total thermal conductance cond) is equivalent to connecting n GL components in parallel, each having thermal conductance:

$$Vcond_i = \frac{cond}{n}$$

Heat Transport by Conduction for each node is calculated as follows:

$$q_i = Vcond_i \left(T_{in,i} - T_{out,i} \right)$$

where:

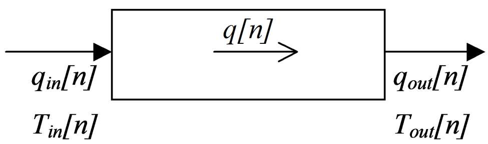
Vcondi: Thermal conductance at each array element.

Tin,i: Temperature of the element i of the inlet thermal port

Tout, i: Temperature of the element i of the outlet thermal port

Inherited from Abstract Component Conductor

Thermal conductors do not have any heat storage capacity by hypothesis, so the inlet heat flow is equal to the outlet heat flow, and it is also equal to the internal heat flow through the conductor:



$$q_{in,i} = q_i \qquad for\ all\ i \in 1, n$$
$$q_{in,i} = q_{out,i} \qquad for\ all\ i \in 1, n$$

4.15 Component GL_mat

Inherited from component AGL.

4.15.1 Description

This component type, named GL_mat, represents a linear thermal conductor of a given material.

4.15.2 Symbol

4.15.3 Construction Parameters

Name	Type	Default	Description
n	CONST INTEGER	1	Dimension of thermal port

4.15.4 Ports

Name	Type	Parameters	Direction	Cardinality	Description
tp_in	thermal	(n = n)	IN	[0, 1]	Thermal inlet port
tp_out	thermal	(n = n)	OUT	[0, 1]	Thermal outlet port

4.15.5 Data

Name	Type	Default	Description	Units
mat	ENUM Material	None	Material	*W/K*
F	REAL	0.1	Area to thermal path ratio	*m*
k	REAL	0.1	Thermal conductivity if material is None	*W/m K*

4.15.6 Variables

Name	Type	Description	Units
q[n]	REAL	Total heat	*W*
Vcond[n]	REAL	Variable thermal conductance	*W/K*
k_var[n]	REAL	Thermal conductivity	*W/m K*
ipt[n]	REAL	Pointer to last table position	-

4.15.7 Topology

The following statement helps to simplify the connecting statements when defining thermal networks using the EcosimPro Language.

PATH tp_in TO tp_out

4.15.8 Formulation

The mathematical equations described in this section are employed to represent the behaviour of this component.

Thermal Conductance is distributed among the port connections

Since the thermal port is in vectorised form, the thermal conductance of each element is computed as follows:

$$Vcond_i = \frac{F}{n} \cdot k\_\mathrm{var}_i$$

where:

F: Area to thermal path ratio

k_var: Variable thermal conductivity at mean conductor temperature calculated interpolating in the data files

Heat Transport by Conduction for each node is calculated as follows:

$$q_i = Vcond_i \left(T_{in,i} - T_{out,i}\right)$$

where:

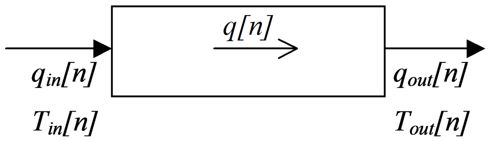
Vcond: Thermal conductance at each array item.

Tin,i: Temperature of the element i of the inlet thermal port

Tout, i: Temperature of the element i of the outlet thermal port

Inherited from Abstract Component Conductor

Thermal conductors do not have any heat storage capacity by hypothesis, so the inlet heat flow is equal to the outlet heat flow, and it is also equal to the internal heat flow through the conductor:



$$q_{in,i} = q_i \qquad for\ all\ i \in 1,n$$
$$q_{in,i} = q_{out,i} \qquad for\ all\ i \in 1,n$$

4.16 Component GR

Inherited from component AGR.

4.16.1 Description

This component type, named GR, represents a thermal radiative connection where REF is a constant value or datum.

4.16.2 Symbol

4.16.3 Construction Parameters

Name	Type	Default	Description
n	CONST INTEGER	1	Dimension of thermal port

4.16.4 Ports

Name	Type	Parameters	Direction	Cardinality	Description
tp_in	thermal	(n = n)	IN	[0, 1]	Thermal inlet port
tp_out	thermal	(n = n)	OUT	[0, 1]	Thermal outlet port

4.16.5 Data

Name	Type	Default	Description	Units
REF	REAL	0	Radiative Exchange Factor	m2

4.16.6 Variables

Name	Type	Description	Units
q[n]	REAL	Heat flow	W
VREF[n]	REAL	Variable radiative exchange factor	m2

4.16.7 Topology

The following statement helps to simplify the connecting statements when defining thermal networks using the EcosimPro Language.

PATH tp_in TO tp_out

4.16.8 Formulation

The mathematical equations described in this section are employed to represent the behaviour of this component.

The Radiative Exchange Factor is distributed along the port connections:

$$VREF_i = \frac{REF}{n}$$

Heat Conduction by Radiation

$$q_i = \sigma \cdot VREF_i \cdot \left(T_{in,i}^4 - T_{out,i}^4\right)$$

where:

σ: Stefan Boltzmann constant = 5.67.10-8 W/m2*K4

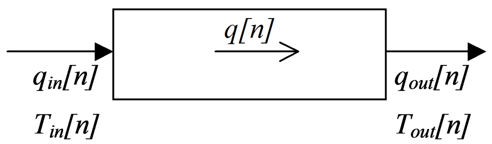
VREFi: Radiative exchange factor that is equal to a datum (VREF = REF)

Tin,i: Temperature of the element i of the inlet thermal port

Tout,i: Temperature of the element i of the outlet thermal port

Conservation of Energy (Inherited from Abstract Component Conductor)

Thermal conductors do not have any heat storage capacity by hypothesis, so the inlet heat flow is equal to the outlet heat flow, and it is also equal to the internal heat flow through the conductor:



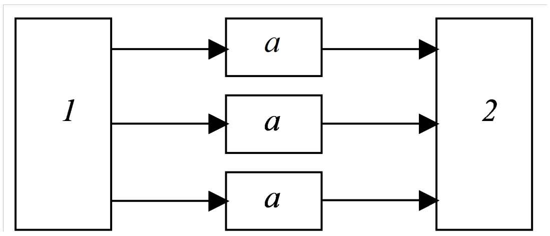
$$q_{in,i} = q_i \qquad for\ all\ i \in 1, n$$
$$q_{in,i} = q_{out,i} \qquad for\ all\ i \in 1, n$$

4.17 Component HeatFlowMultiplier

4.17.1 Description

This component, named HeatFlowMultiplier, produces an outlet heat flow, which is equal to the inlet heat flow times a factor k, provided as a datum.

This component is useful when modelling a system with repeated components, such as that shown below:



Here, a number of identical "a" components are receiving a heat flow from 1 and are ejecting heat towards 2. Rather than including all identical "a" components in the model, it is much simpler to substitute the redundant components by an equivalent arrangement using HeatFlowMultiplier components.

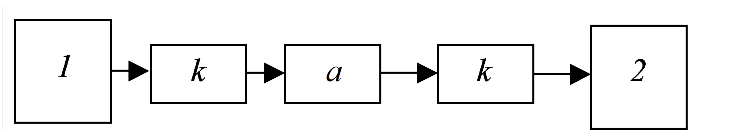
In the equivalent arrangement, only one component of type "a" is required, the rest of the identical components being represented by placing HeatFlowMultiplier components at either side of "a". The presence of multiple "a" components is modelled by adjusting the values of k:

At the inlet of "a",

$$k = \frac{1}{n}$$

At the outlet of "a",

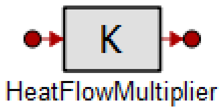
$$k = n$$



Connecting HeatFlowMultiplier components

Special care must be taken when connecting this type of component to others, since its ports are neither capacitive nor resistive. However, in order to comply with the general rule for connecting elements in the thermal library (the arrangement C-R-C-R-etc), this component must be inserted in between a capacitive and a resistive port, in any order.

4.17.2 Symbol



4.17.3 Construction Parameters

Name	Type	Default	Description
n	CONST INTEGER	1	Dimension of thermal port

4.17.4 Ports

Name	Type	Parameters	Direction	Cardinality	Description
tp_in	thermal	(n = n)	IN		Thermal inlet port
tp_out	thermal	(n = n)	OUT		Thermal outlet port

4.17.5 Data

Name	Type	Default	Description	Units
K	REAL	1	Heat flow multiplying factor: outlet heat / inlet heat	-

4.17.6 Topology

The following statement helps to simplify the connecting statements when defining thermal networks using the EcosimPro Language.

PATH tp_in TO tp_out

4.17.7 Formulation

The mathematical equations described in this section are employed to represent the behaviour of this component.

The heat flow at the outlet is equal to the heat flow at the inlet times a factor k:

$$q_{outlet} = k \cdot q_{inlet}$$

The temperatures at outlet and inlet are the same:

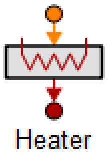
$$T_{outlet} = T_{inlet}$$

4.18 Component Heater

4.18.1 Description

This component represents an electrical heater connected to the control system, and it produces a heat flow equal to the value of the command signal. This component is an actuator, i.e. an interface between the Thermal System and the Control System.

4.18.2 Symbol



4.18.3 Construction Parameters

Name	Type	Default	Description
n	CONST INTEGER	1	Dimension of thermal port

4.18.4 Ports

Name	Type	Parameters	Direction	Cardinality	Description
tp_out	thermal	(n = n)	OUT	1,1	Thermal outlet port
s_power	analog_signal	-	IN	-	Input power signal

4.18.5 Formulation

The heat flow produced by the heater is equal to the value of the input signal:

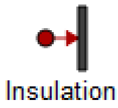
```
tp_out.q = s_power.signal
```

4.19 Component Insulation

4.19.1 Description

This component represents a thermal insulator.

4.19.2 Symbol



4.19.3 Construction Parameters

Name	Type	Default	Description
n	CONST INTEGER	1	Dimension of thermal port

4.19.4 Ports

Name	Type	Parameters	Direction	Cardinality	Description
tp	thermal	(n = n)	IN	1,1	Thermal inlet port

4.19.5 Formulation

The heat flow at the port is equal to zero:

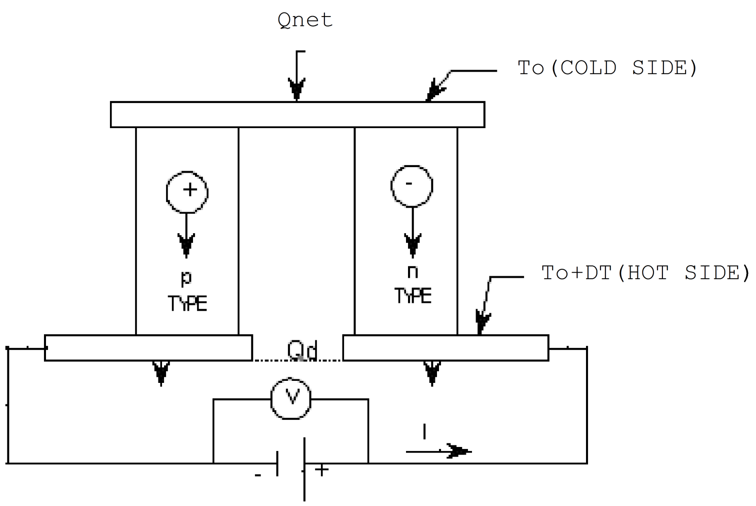
```
tp_in.q[i] = 0
```

4.20 Component Peltier

4.20.1 Description

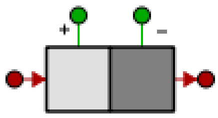
This component represents a thermoelectric element (TE).

This component represents a thermoelectric heat pump consisting of a number (NTE) of n- and p-type semiconductor material. A diagram of a thermoelectric heat pump in cooling mode is shown in the next figure:



There are several operation modes of thermoelectric devices depending on the direction of the current and the sign of the temperature difference between the two sides of the device.

4.20.2 Symbol



4.20.3 Construction Parameters

Name	Type	Default	Description
n	CONST INTEGER	1	Dimension of thermal port

4.20.4 Ports

Name	Type	Parameters	Direction	Cardinality	Description
tp_in	thermal	(n = n)	IN	[0, 1]	Thermal inlet port
tp_out	thermal	(n = n)	OUT	[0, 1]	Thermal outlet port
e_p	elec		IN	-	Positive electrical port connected to the n_type semiconductor
e_n	elec		IN	-	Negative electrical port connected to the p_type semiconductor

4.20.5 Data

Name	Type	Default	Description	Units
NTE	INTEGER	1	Number of TE elements	-
G	REAL	0.04	Area length of the TE elements	m
a_coef[3]	REAL	2.2224E-05 9.306E-07 - 9.905E-10	Temperature coefficients of Seebeck effect	V/ K V / K2 V / K3
r_coef[3]	REAL	5.112E-07 1.634E-08 6.279E-11	Temperature coefficients of resistivity	Ohm m Ohm m / K Ohm m / K2
k_coef[3]	REAL	6.2605 -0.02777 4.131E-005	Temperature coefficients of thermal conductivity	W / m K W / m K2 W / m K3

4.20.6 Variables

Name	Type	Description	Units
qin	REAL	Inlet heat flow	*W*
qout	REAL	Outlet heat flow	*W*
a	REAL	Seebeck coefficient	*V / K*
COP	REAL	Coefficient of performance	-
COPopt	REAL	Optimum coefficient of performance	-
DT	REAL	Temperature jump	*K*
DTmax	REAL	Maximum temperature jump	*K*
Imax	REAL	Maximum current	*A*
I	REAL	Current	*A*
Iopt	REAL	Optimum current	*A*
Pel	REAL	Electrical power	*W*
r	REAL	Resistivity	*Ohm m*
k	REAL	Thermal conductivity	*W / m K*
Tk_ave	REAL	Average temperature	*K*
Tk_out	REAL	Hot temperature	*K*
Tk_in	REAL	Cold temperature	*K*
V	REAL	Voltage	*V*
Z	REAL	Figure of merit	*K-1*

4.20.7 Topology

The following statement helps to simplify the connecting statements when defining thermal networks using the EcosimPro Language.

PATH tp_in TO tp_out

4.20.8 Formulation

The formulation of the Peltier element is based on the information provided in Reference [1].

It is assumed that in cooling mode as refrigerator the cold side is the inlet and the hot side is the outlet. The temperature at the cold side is the mean temperature at the inlet port and the temperature at the hot side is the mean temperature at the outlet port.

The average absolute temperature is calculated from:

$$\mathrm{T_{avg}} = 0.5 \cdot (\mathrm{T_{avg,\,in}} + \mathrm{T_{avg,out}})$$

The Seebeck coefficient, a, the resistivity, r, and the thermal conductivity, k, are polynomial functions of the average temperature:

$$\begin{aligned}
\mathrm{a} &= \mathrm{a_coef[1]} + \mathrm{a_coef[2]} \cdot \mathrm{T_{ave}} + \mathrm{a_coef[3]} \cdot (\mathrm{T_{ave}})^2 \\
\mathrm{r} &= \mathrm{r_coef[1]} + \mathrm{r_coef[2]} \cdot \mathrm{T_{ave}} + \mathrm{r_coef[3]} \cdot (\mathrm{T_{ave}})^2 \\
\mathrm{k} &= \mathrm{k_coef[1]} + \mathrm{k_coef[2]} \cdot \mathrm{T_{ave}} + \mathrm{k_coef[3]} \cdot (\mathrm{T_{ave}})^2
\end{aligned}$$

The figure of merit of the element is calculated from:

$$Z = \frac{a^2}{r \cdot k}$$

The temperature jump is:

$$DT = T_{avg,out} - T_{avg,in}$$

The temperature difference (DT) is positive when the thermoelectric device operates as a cooling device (refrigerating).

The heat pumped is:

$$q_{in} = 2 \quad NTE \left(a \quad I \, T_{avg,in} - \frac{I^2 \quad r}{2 \quad G} - \quad k \quad DT \quad G \right)$$

$$q_{out} = 2 \quad NTE \left(a \quad I \, Tk_{out} - \frac{I^2 \quad r}{2 \quad G} - \quad k \quad DT \quad G \right)$$

The potential difference is the sum of the resistivity and Seebeck effects:

$$V = 2\ N \left(\frac{I\ r}{G} + a\ DT \right)$$

The electrical power is equal to the current times the voltage difference:

$$P_{el} = I \cdot V$$

The maximum temperature DT with q = 0 is:

$$DT_{max} = 0.5\ \ Z\ T^{2}_{avg.in}$$

The maximum current corresponding to DTmax with q = 0 is:

$$I_{max} = \frac{k\ G}{a} \left(\sqrt{1 + 2\ Z\ T_{avg,out}} - 1 \right)$$

The coefficient of performance of the Peltier is defined as:

$$COP = \frac{q}{P_{el}}$$

The optimum coefficient of performance is:

$$COP_{opt} = \frac{Tk_{ave}}{DT} \frac{\sqrt{1 + Z\ Tk_{ave}} - 1}{\sqrt{1 + Z\ Tk_{ave}} + 1} - 0.5$$

The optimum current corresponding to the optimum performance:

$$I_{opt} = k\ \ DT\ G \left(\frac{1 + \sqrt{1 + Z\,Tk_{ave}}}{a\ Tk_{ave}} \right)$$

The equations in the electrical ports are the following:

$$I_p = I_n$$
$$I_p = I$$

It is considered that intensity is positive from the n-type material to the p-type material when the thermoelectric device is in cooling mode as a refrigerator.

And the potential difference between the ports is defined as:

$$V = V_p - V_n$$

4.21 Component Periodic_Stop_Control

4.21.1 Description

This component type, named Periodic_Stop_Control, represents a device that can be attached to a particular node allowing stopping the simulation when the dynamic response of the measured temperature becomes periodic.

4.21.2 Symbol

4.21.3 Construction Parameters

Name	Type	Default	Description
n	CONST INTEGER	1	Dimension of thermal port
node	CONST INTEGER	1	Number of nodes to sample

4.21.4 Ports

Name	Type	Parameters	Direction	Cardinality	Description
tp_in	thermal	(n = n)	IN	1,1	Thermal inlet port

4.21.5 Data

Name	Type	Default	Description	Units
To	REAL	290	Initial temperature	K
T_error	REAL	0.001	Average temperature error to consider cyclic behaviour	K

4.21.6 Variables

Name	Type	Initial	Description	Units
T_After_Period	REAL	To	Temperature delayed by one period	K
ErrorLimit	REAL		Integrated error limit between one period and previous one to consider cyclic behaviour	K s
ErrorPeriod	REAL	1	Integrated absolute error between one period and previous one	K s
Cycled	BOOLEAN	FALSE	Cycling status	-
Check_Error	BOOLEAN	TRUE	Auxiliary variable to check error at the end of each period	-

4.21.7 Topology

The following statement helps to simplify the connecting statements when defining thermal networks using the EcosimPro Language.

PATH tp_in TO tp_in

4.21.8 Formulation

The following mathematical expressions apply to this component type:

Initialisation of the Temperature:

T = To

Determine if the Temperature Dynamics is Cyclic

It is checked after five periods if the temperature dynamics is cyclic at the end of each new period. The condition to know whether the temperature dynamics is cyclic is that the integrated absolute error between one period and the previous one is less than the maximum integrate absolute error allowed to consider cyclic behaviour. When this happens it is fixed the end of the simulation.

```
If (Ep < Elimit)
tstop = TIME + period
Cycled = TRUE
```

where:

Ep: Integrated absolute error between one period and previous one

Elimit: Maximum integrated absolute error allowed

Tstop: Final simulation time

period : Orbit period

The integrated absolute error between one period and the previous one is calculated from:

$$\frac{dE_p}{dt} = \left| T(t) - T(t - period) \right|$$

And the maximum integrated absolute error allowed to consider cyclic behaviour is calculated from:

$$E_{\lim it} = \left| T_{error} \right| \cdot period$$

where:

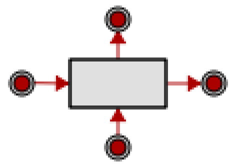
Terror: Average temperature error to consider cycling

4.22 Component Plate

4.22.1 Description

This component, named Plate, represents a panel divided into a number of longitudinal and transversal slabs (nodes). Heat flow is considered at the sides of the plate (positive for increasing node index), and it is assumed that the remaining ends are isolated.

4.22.2 Symbol



4.22.3 Construction Parameters

Name	Type	Default	Description
nx	CONST INTEGER	1	OX-Number of nodes for plate
ny	CONST INTEGER	1	OY-Number of nodes for plate

4.22.4 Ports

Name	Type	Parameters	Direction	Description
tpx_in	thermal	(n = ny)	IN	OX- thermal inlet port
tpx_out	thermal	(n = ny)	OUT	OX- thermal outlet port
tpy_in	thermal	(n = nx)	IN	OY- thermal inlet port
tpy_out	thermal	(n = nx)	OUT	OY- thermal outlet port

4.22.5 Data

Name	Type	Default	Description	Units
L	REAL	1	OX - Plate length	*m*
W	REAL	1	OY - Plate width	*m*
e	REAL	0.003	OZ - Plate thickness	*m*
mat	ENUM Material	None	Material	-
cp	REAL	500	Plate specific heat if material is None	*J/kg K*
k	REAL	0.1	Plate conductivity if material is None	*W/m K*
rho	REAL	1000	Plate density if material is None	*Kg /m3*
TemperatureDependance	BOOLEAN	TRUE	Option to consider or not the temperature dependence on the material properties	
init_mode	ENUM WALL_INIT_MODE	Constant_Temp	Option for the initialisation of the nodal temperatures	
To	REAL	290	Initial temperature of nodes	*K*
To_linear[2]	REAL	{290, 290}	Initial temperatures at both sides if init_mode = Linear_Temp	*K*
To_table	TABLE_1D		Table with the initial temperatures versus non-dimensional position if init_mode = Table_Temp	*K*

Input data cp, k and rho will be used only if "mat" = None. If not, the properties will be interpolated using the corresponding data file.

4.22.6 Variables

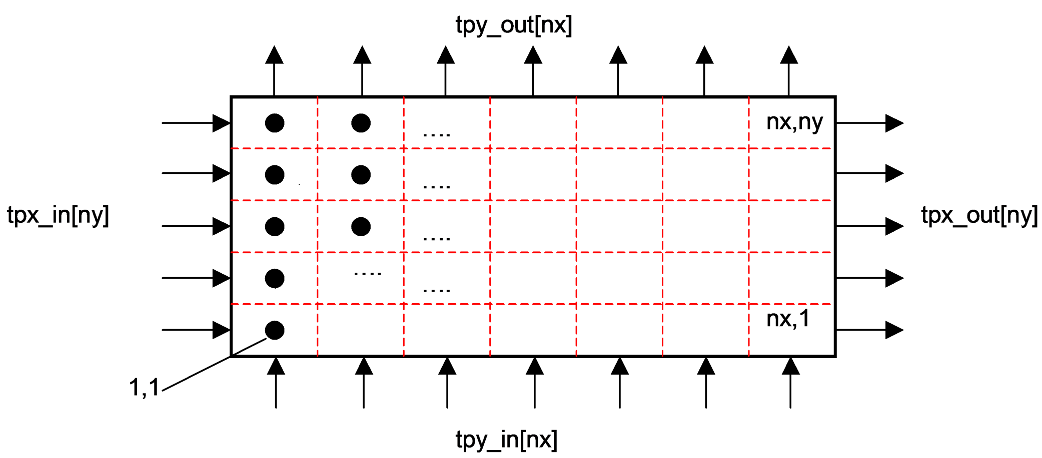
Name	Type	Initial	Description	Units
T[nx,ny]	REAL		Nodal temperatures	*K*
cp_var[nx,ny]	REAL		Specific heat of nodes	*J/kg K*
cp_tpx[2, nx]	REAL		OX-Nodal specific heat of first last section skin	*J/kg K*
cp_tpy[nx,2]	REAL		OY-Nodal specific heat of first last section skin	*J/kg K*
dx	REAL		OX - Node width or nodal distance	*m*
dy	REAL		OY - Node width or nodal distance	*m*
k_var[nx,ny]	REAL		Thermal conductivity of nodes	*W/m K*
C_x[nx+1, ny]	REAL		OX - Equivalent conductance between the thermal nodes	*W/K*
C_y[nx+1, ny]	REAL		OY - Equivalent conductance between the thermal nodes	*W/K*
qx[nx + 1, ny]	REAL		OX - heat flow between nodes	*W*
qy[nx, ny + 1]	REAL		OY - heat flow between nodes	*W*
rho_var	REAL		Density of plate material	*Kg /m3*
V	REAL		Node volume	*m3*
Ax	REAL		OX - Cross Area	*m2*
Ay	REAL		OY - Cross Area	*m2*
icpk[nx, ny]	REAL		Last pointer to speed up the nodal interpolations of material properties	
icpk_tpx[2, ny]	REAL		OX - Pointer to last interpolation interval for each port node	
icpk_tpy[nx, 2]	REAL		OY - Pointer to last interpolation interval for each port node	
it0	REAL		Pointer to To	
F	REAL		Thermal capacity ratio assigned to the surface	-

4.22.7 Formulation

Nodalisation

The dimensions of the inlet and outlet vectorised thermal ports are defined by the values of nx (number of nodes in the x direction into which the plate is divided) and ny (number of nodes in the y direction). The temperature at each node is stored in an array called T[nx,ny].

Array qx[nx + 1, ny] contains the heat flows across nodes in the x direction, and array qy[nx, ny+1] contains the heat flows across nodes in the y direction.



Initialisation

The data named init_mode allows the user to specify the initial temperature distribution in the nodes. There are three options::

- Uniform initial temperature for all the nodes (init_mode = Constant_Temp)
- Linear initial temperature distribution in the OX direction and uniform initial temperature in the OY direction (init_mode = Linear_Temp)

- Initial temperature distribution given by a data table of the initial temperature as a function of the non-dimensional position in the OX direction (init_mode = Table_Temp). For the OY direction, the temperature distribution is considered uniform.

Heat balance

The following heat balance equation is applied on each node in the OX and the OY directions:

$$\rho_{\text{var}} \cdot cp_{\text{var}\,i,j} \cdot V \cdot \left(1 - \frac{F}{nx \cdot ny}\right) \cdot \frac{\partial T_{i,j}}{\partial t} = qx_{i,j} - qx_{i+1,j} + qy_{i,j} - qy_{i,j+1}$$

where

ρ: Density of the plate material

cpvar i,j: Specific heat of the node [i,j]

V: Node volume

F: Thermal capacity ratio assigned to the ports (surface)

dx: Node width or nodal distance in the OX direction:

$$dx = \frac{L}{nx}$$

dy: Node width or nodal distance in the OY direction

$$dy = W / ny$$

L: Plate length (OX dimension)

W: Plate width (OY dimension)

nx: Number of nodes in the direction OX

ny: Number of nodes in the direction OY

V: Node volume

Tij: Temperature of the node [i,j]

qx ij: Heat flow between node [i,j] y [i-1,j] in the direction OX

qy ij: Heat flow between node [i, j] y [i, j-1] in the direction OY

Heat Flows

The heat flows are then considered in the x and y directions:

The internal heat flows in the X direction are given by:

$$qx_{i,j} = \frac{k_{\text{var}\,i-1,j} \cdot k_{\text{var}\,i,j}}{\frac{k_{\text{var}\,i,j} + k_{\text{var}\,i-1,j}}{2}} \cdot (dy \cdot e) \cdot \frac{T_{i-1,j} - T_{i,j}}{dx}$$

and at the ends of the plate:

$$qx_{1,j} = k_{\text{var}\,1,j} \cdot (dy \cdot e) \cdot \frac{tpx\_in.Tk_j - T_{1,j}}{dx/2}$$

$$qx_{n+1,j} = k_{\text{var}\,nx,j} \cdot (dy \cdot e) \cdot \frac{T_{nx,j} - tpx\_out.Tk_j}{dx/2}$$

The internal heat flows in the Y direction are given by:

$$qx_{i,j} = \frac{k_{\text{var}\,i,j-1} \cdot k_{\text{var}\,i,j}}{\frac{k_{\text{var}\,i,j} + k_{\text{var}\,,j-1}}{2}} \cdot (dy \cdot e) \cdot \frac{T_{i,j-1} - T_{i,j}}{dx}$$

and at the ends of the plate:

$$qy_{i,1} = k_{\text{var}\,i,1} \cdot (dx \cdot e) \cdot \frac{tpy\_in.Tk_i - T_{i,1}}{dy/2}$$

$$qy_{i,j+1} = k_{\text{var}\,i,ny} \cdot (dx \cdot e) \cdot \frac{T_{i,ny} - tpy\_out.Tk_i}{dy/2}$$

Equations in the ports

A certain thermal inertia (given by the datum F) is assigned to the thermal inlet and outlet ports. This is done to prevent the appearance of algebraic loops when calculating the heat flows at the ports. The heat balances applied to the ports are the following:

$$\rho_{\text{var}} \cdot cp\_tpx_{1,j} \cdot V \cdot \frac{F}{2 \cdot (nx+ny)} \cdot \frac{\partial\, tpx\_in.Tx_j}{\partial t} = tpx\_in.q_j - qx_{1,j}$$

$$\rho_{\text{var}} \cdot cp\_tpx_{2,j} \cdot V \cdot \frac{F}{2 \cdot (nx+ny)} \cdot \frac{\partial\, tpx\_out.Tx_j}{\partial t} = qx_{nx+1,j} - tpx\_out.q_j$$

$$\rho_{\text{var}} \cdot cp\_tpy_{i,1} \cdot V \cdot \frac{F}{2 \cdot (nx+ny)} \cdot \frac{\partial\, tpy\_in.Tx_j}{\partial t} = tpy\_in.q_i - qy_{i,1}$$

$$\rho_{\text{var}} \cdot cp\_tpx_{i,2} \cdot V \cdot \frac{F}{2 \cdot (nx+ny)} \cdot \frac{\partial\, tpy\_out.Tx_j}{\partial t} = qy_{i,ny+1} - tpy\_out.q_i$$

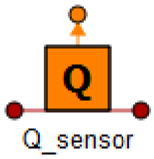
4.23 Component Q_sensor

Inherited from components Conductor and CONTROL.Asensor.

4.23.1 Description

This component represents a heat flow sensor, and it is an interface between the thermal system and the control system.

4.23.2 Symbol



4.23.3 Construction Parameters

Name	Type	Default	Description
in	CONST INTEGER	1	Index of item in the heat flow array to be measured
n	CONST INTEGER	1	Dimension of thermal port
n_out	CONST INTEGER	1	Dimension of outputs

4.23.4 Ports

Name	Type	Parameters	Direction	Cardinality	Description
s_out	analog_signal	(n = n_out)	OUT		Outlet signal
tp_in	thermal	(n = n)	IN	[0, 1]	Thermal inlet port
tp_out	thermal	(n = n)	OUT	[0, 1]	Thermal outlet port

4.23.5 Data

Name	Type	Default	Description	Units
Gain[n_out]	REAL	1	Gain	*W-1*
Bias[n_out]	REAL	0	Bias	-

4.23.6 Variables

Name	Type	Description	Units
q[n]	REAL	Heat flow	*W*
Var[n_out]	REAL	Variable being measured	*W*

4.23.7 Formulation

The value being measured is:

```
s_out.signal = gain · tp_in.q + bias
```

and it is assumed that the heat flow sensor does not produce any temperature difference:

```
tp_in.T = tp_out.T
```

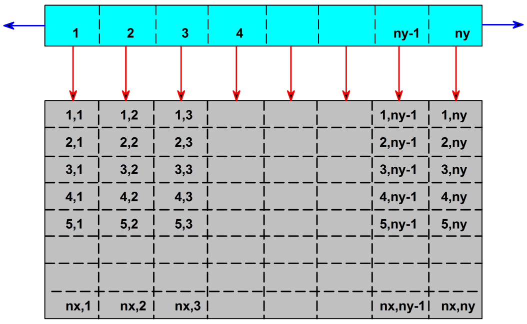
and the conservation of energy is inherited from the conductor component:

```
tp_in.q = tp_out.q = q
```

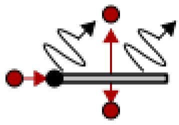
4.24 Component Radiator

4.24.1 Description

This component represents a radiator device for cooling purposes, with the purpose of being connected to a pipe or to a heat pipe.



4.24.2 Symbol



4.24.3 Construction Parameters

Name	Type	Default	Description	Units
nx	CONST INTEGER	5	Number of nodes in cross direction	-
ny	CONST INTEGER	5	Number of nodes in tube direction	-

4.24.4 Ports

Name	Type	Parameters	Direction	Description
tp_in	thermal	(n = ny)	IN	Thermal inlet port for connection to a pipe or to heat pipe
tp_1	thermal	(n = 1)	OUT	Thermal port for radiative sink on side 1
tp_2	thermal	(n = 1)	OUT	Thermal port for radiative sink on side 2

4.24.5 Data

Name	Type	Default	Description	Units
n_fin	INTEGER	1	Number of fins (see figure in Formulation)	-
cp	REAL	500	Radiator specific heat if material is None	*J/kg K*
k	REAL	0.1	Radiator conductivity if material is None	*W/m K*
rho	REAL	1000	Radiator density if material is None	*Kg/m3*
Lx	REAL	0.2	Length of the fin	*m*
Ly	REAL	5	Width of the fin	*m*
A	REAL	25	Area of 1 fin	*m2*
e	REAL	0.012	Fin thickness	*m*
mat	ENUM Material	None	Material used for radiator	-
emiss1	REAL	0.85	Emissivity of side 1 of the fin	-
alpha1	REAL	0.25	Solar absorptance of side 1 of the fin	-
F1	REAL	0.9	Average view factor of sink node 1 from side 1	-
beta1	REAL	0	Incidence angle of the sun in side 1	*o*
emiss2	REAL	0.85	Emissivity of side 2 of the fin	-
alpha2	REAL	0.25	Solar absorptance of side 2 of the fin	-
F2	REAL	0	Average view factor of sink node 2 from side 2	-
beta2	REAL	0	Incidence angle of the sun in side 2	*o*
TemperatureDependance	BOOLEAN	TRUE	Option to consider or not the temperature dependence on the material properties	
To	REAL	290	Initial temperature	*K*

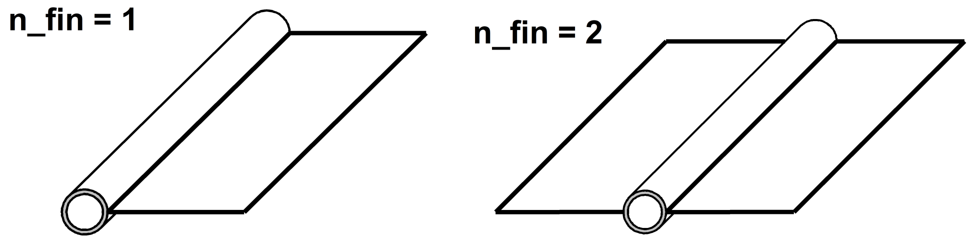
Input data cp, k and rho will be used only if "mat" = None. If not, the properties will be interpolated using the corresponding data file.

4.24.6 Variables

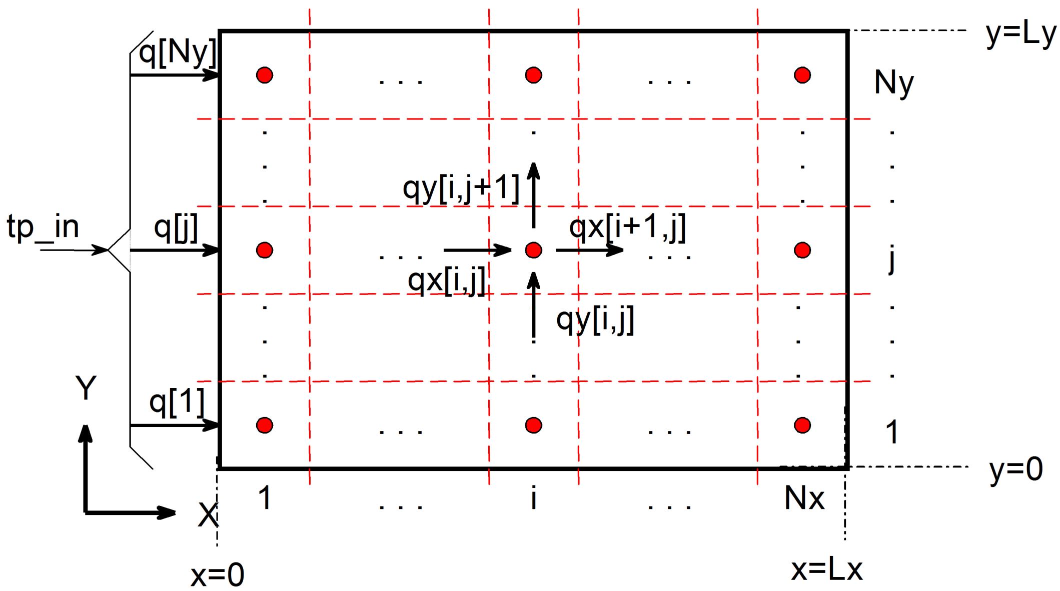
Name	Type	Description	Units
A	REAL	Area of 1 fin	*m2*
dx	REAL	Node length in cross direction	*m*
dy	REAL	Node length in tube direction	*m*
Ax	REAL	OX - Cross Area	*m2*
Ay	REAL	OY - Cross Area	*m2*
rho_var	REAL	Density of radiator material	*kg/m3*
cp_var[nx,ny]	REAL	Specific heat of nodes	*J/kg K*
k_var[nx,ny]	REAL	Thermal conductivity of radiator nodes	*W/m K*
C_x[nx+1, ny]	REAL	Equivalent conductance in the cross direction between the thermal nodes	*W/K*
C_y[nx, ny+1]	REAL	Equivalent conductance in the tube direction between the thermal nodes	*W/K*
Tsink1	REAL	Equivalent sink temperature for side 1	*K*
Tsink2	REAL	Equivalent sink temperature for side 2	*K*
q_tot_sun	REAL	Total solar heat flow received by radiator	*W*
qx[nx+1,ny]	REAL	Cross direction heat flow through nodes	*W*
qy[nx,ny+1]	REAL	Tube direction heat flow through nodes	*W*
q_rad_1[nx,ny]	REAL	Heat flow radiated by each node on side 1	*W*
q_rad_2[nx,ny]	REAL	Heat flow radiated by each node on side 2	*W*
T[nx,ny]	REAL	Temperature of nodes	*K*
q_tot_rad_1	REAL	Total heat flow radiated by side 1	*W*
q_tot_rad_2	REAL	Total heat flow radiated by side 2	*W*
q_tot_fluid	REAL	Total heat flow received from fluid by radiator	*W*
icpk[nx, ny]	INTEGER	Last pointer to speed up nodal interpolations	-
it0	INTEGER	Pointer to To	-

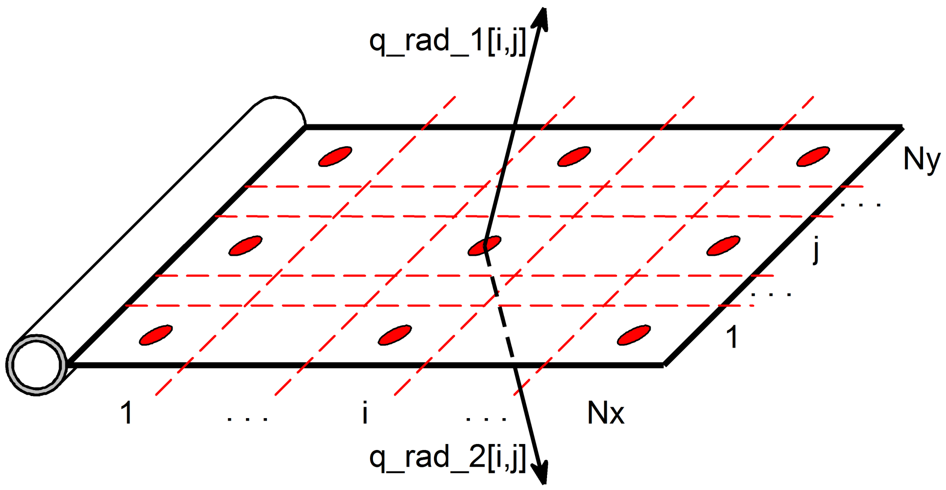
4.24.7 Formulation

Depending on the parameter n_fin (number of fins), the component can represent any of the two configurations shown in the next figure.



Nodalisation





Geometry

Area of one fin:

$$A = Lx \cdot Ly$$

Calculation of node lengths:

$$dx = \frac{Lx}{nx}$$
$$dy = \frac{Ly}{ny}$$

Calculation of the heat transfer area for each direction:

$$Ax = e \cdot dy$$
$$Ay = e \cdot dx$$

where:

Lx: Length of the fin

Ly: Width of the fin

e: Thickness of the fin

nx: Number of nodes in cross direction

ny: Number of nodes in tube direction

Equivalent Sink Temperature for Side 1 (Tsink1):

$$T_{\sin k1} = (tp\_1.Tk_1)\left(1 + \frac{\alpha_1 \; SC \; \sin(\beta_1)}{\varepsilon_1 F_1 \; (tp\_1.Tk_1)^4}\right)^{1/4}$$

where

tp_1.Tk1 : Temperature of the radiative environment for node 1

$\alpha 1$: Solar absorptance of side 1 of the fin

SC : Solar constant

$\beta 1$: Incidence angle of the sun in side 1 of the fin ($0 \leq \beta 1 \leq 90$)

$\varepsilon 1$: Emissivity of side 1 of the fin

F1: Average view factor of the radiation environment from the fin

Equivalent Sink Temperature for Side 2

$$T_{\sin k2} = (tp\_2.T)\left(1 + \frac{\alpha_2 \; SC \; \sin(\beta_2)}{\varepsilon_2 F_2 \; (tp\_2.T)^4}\right)^{1/4}$$

Total Solar Heat Flux Received by one Fin of the Radiator

$$q_{tot,sun} = \alpha_1 SC \; A \sin(\beta_1) + \alpha_2 SC \; A \sin(\beta_2)$$

Heat Flows in X direction

$$qx_{1,j} = k_{\text{var}1,j} \cdot Ax \cdot \frac{(tp\_in.Tk_j - T_{1,j})}{dx/2} \qquad for\ j\ in\ 1, ny$$

$$qx_{i,j} = \frac{k_{\text{var}\,i-1,j} \cdot k_{\text{var}\,i,j}}{k_{\text{var}\,i,j} + k_{\text{var}\,i-1,j}} \cdot Ax \cdot \frac{k \cdot \cdot (T_{i-1,j} - T_{i,j})}{\frac{dx}{2}} \qquad for\ i\ in\ 2, nx\ and\ for\ j\ in\ 1, ny$$

End x=Lx is isolated

$$qx_{nx+1,j} = 0 \qquad for\ j\ in\ 1, ny$$

where:

tp_in.Tkj : Temperature of the node j of the inlet port

qx i,j: Cross direction heat flow between the nodes [i-1, j] and [i, j]

kvar i,j: Thermal conductivity of node [i, j]

Heat flow in y direction

$$qy_{i,j} = \frac{k_{\text{var}\,i,j-1} \cdot k_{\text{var}\,i,j}}{k_{\text{var}\,i,j} + k_{\text{var}\,i,j-1}} \cdot Ay \cdot \frac{(T_{i,j-1} - T_{i,j})}{\frac{dy}{2}} \qquad for\ i\ in\ 1, nx\ and\ for\ j\ in\ 2, ny$$

$$qy_{i,ny+1} = 0 \qquad for\ j\ in\ 1, nx$$

$$qy_{i,1} = 0 \qquad for\ i\ in\ 1, nx$$

Heat Flux radiated by each Node in Side 1

$$q_{rad1,i,j} = \varepsilon_1 \cdot \sigma \cdot F1 \cdot dx \cdot dy \cdot \left(T_{i,j}{}^4 - tp\_1.Tk_1{}^4\right)$$

Heat Flux radiated by each Node in Side 2

$$q_{rad2,i,j} = \varepsilon_2 \cdot \sigma \cdot F2 \cdot dx \cdot dy \cdot \left(T_{i,j}{}^4 - tp\_2.Tk_1{}^4\right)$$

where:

ε1: Emissivity of side 1 of the fin

ε2: Emissivity of side 2 of the fin

F1: Average view factor of sink node 1 from side 1

F2: Average view factor of sink node 2 from side 2

σ: Stefan Boltzman constant

Energy Equation applied to each node:

$$C_{i,j} \cdot \frac{\partial T_{i,j}}{\partial t} = qx_{i,j} - qx_{i+1,j} + qy_{i,j} - qy_{i,j+1} + \frac{q_{tot,sun}}{nx \cdot ny} - q_{rad1,i,j} - q_{rad2,i,j}$$

where

Ci,j: The thermal capacitance of one node

$$C_{i,j} = cp_{\mathrm{var}\,i,j} \cdot \rho \cdot dx \cdot dy \cdot e$$

qtot sun: Total solar heat flow received by radiator

Calculation of Total Heat Flows

Total Heat Flow Radiated by Side 1

$$q_{tot,rad1} = n_{fin} \sum_{i=1}^{nx} \sum_{j=1}^{ny} q_{rad1,i,j}$$

Total Heat Flow Radiated by Side 2

$$q_{tot,rad2} = n_{fin} \sum_{i=1}^{nx} \sum_{j=1}^{ny} q_{rad2,i,j}$$

Total Heat Flow Received from Fluid by Radiator

$$q_{tot,fluid} = n_{fin} \sum_{j=1}^{ny} qx_{1,j}$$

Equations in the Ports

$$tp\_in.q_j = qx_{1,j} \qquad for\ j\ in\ 1, ny$$
$$tp\_1.q = q_{tot\ rad1}$$
$$tp\_2.q = q_{tot\ rad2}$$

where:

tp_in.qj: Heat flow across the element j of the thermal inlet port

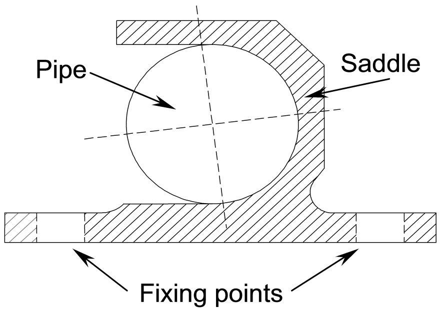
tp_1.q: Heat flow across the thermal port to sink 1

tp_2.q: Heat flow across the thermal port to sink 2

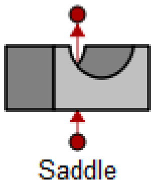
4.25 Component Saddle

4.25.1 Description

This represents a typical saddle device used to fasten heat pipes to a surface, as shown below:



4.25.2 Symbol



4.25.3 Construction Parameters

Name	Type	Default	Description	Units
n	CONST INTEGER	5	Number of nodes	-

4.25.4 Ports

Name	Type	Parameters	Direction	Description
tp_in	thermal_n	(n = n)	IN	Thermal inlet port
tp_out	thermal_n	(n = n)	OUT	Thermal outlet port

4.25.5 Data

Name	Type	Default	Description	Units
l	REAL	1	Length of saddle	m
A	REAL	0.001	Cross sectional area of saddle	m2
mat	ENUM Material	None	Material used for saddle	-
rho	REAL	1000	Saddle density if material is None	Kg/m3
cp	REAL	500	Saddle specific heat if material is None	J/kg K
k	REAL	0.1	Saddle conductivity if material is None	W/m K
G	REAL	50	Conductance of saddle per unit length	W/m K

Input data cp, k and rho will be used only if "mat" = None. If not, the properties will be interpolated using the corresponding data file.

4.25.6 Variables

Name	Type	Description	Units
dx	REAL	Node length	m
rho_var	REAL	Density of saddle material	kg/m3
cp_var	REAL	Specific heat of saddle material	J/kg K
k_var	REAL	Thermal conductivity of saddle material	W/m K
C	REAL	Thermal capacitance of nodes	J/K
q[n+1]	REAL	Axial heat flow through nodes	W
T[n]	REAL	Temperature of nodes	K

4.25.7 Topology

PATH tp_in TO tp_out

4.25.8 Formulation

The formulation used in this component is similar to that in tube with the exception that there is a very low resistance to radial heat flow.

Heat flow through nodes:

$$q_i = \frac{k_{\mathrm{var},i-1} \cdot k_{\mathrm{var}\,i}}{k_{\mathrm{var}\,i} \cdot k_{\mathrm{var}\,i-1}} \cdot A \cdot \frac{T_{i-1} - T_i}{dx} \qquad for\, i\, in\, 2, n$$

$$q_1 = 0$$

$$q_{n+1} = 0$$

Radial heat flow at each node due to saddle-tube contact:

$$tp\_in.q_i = 2 \cdot G \cdot dx \cdot (tp\_in.Tk_i - T_i)$$

$$tp\_out.q_i = 2 \cdot G \cdot dx \cdot (T_i - tp\_out.Tk_i)$$

Energy balance due all heat flows at each node:

$$\rho \cdot cp_i \cdot A \cdot dx \cdot \frac{dT_i}{dt} = (q_i - q_{i+1}) + (tp\_in.q_i - tp\_out.q_i)$$

Where:

tp_in.qi: Heat flow across the element [i] of the thermal inlet port

tp_out.qi: Heat flow across the element [i] of the thermal outlet port

tp_in.Tki: Temperature of the element [i] of the thermal inlet port

tp_out.Tki: Temperature of the element [i] of the thermal outlet port

dx: Node length

G: Conductance of saddle per unit length

Ti: Temperature of node [i]

kvar i: Conductivity of node [i]

qi: Heat flow between node [i-1] and node [i]

cpi: Specific heat of node [i]

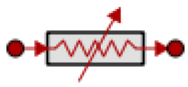
4.26 Component TD_GL

Inherited from component AGL.

4.26.1 Description

This component type, named TD_GL, represents a linear thermal conductor, where conductance is a tabulated function of the average temperature.

4.26.2 Symbol



TD_GL

4.26.3 Construction parameters

Name	Type	Default	Description
n	CONST INTEGER	1	Dimension of thermal port

4.26.4 Ports

Name	Type	Parameters	Direction	Cardinality	Description
tp_in	thermal	(n = n)	IN	[0, 1]	Thermal inlet port
tp_out	thermal	(n = n)	OUT	[0, 1]	Thermal outlet port

4.26.5 Data

Name	Type	Default	Description	Units
F	REAL	0	Area to length ratio	m
k_table	TABLE_1D	{{-273.15, 1000}, {1, 1}}	Thermal conductivity versus average temperature	W/m K vs °C

4.26.6 Variables

Name	Type	Description	Units
q[n]	REAL	Heat flow	W
Vcond[n]	REAL	Variable thermal conductance per node	W/K

4.26.7 Topology

PATH tp_in TO tp_out

4.26.8 Formulation

The mathematical equations described in this section are employed to represent the behaviour of this component.

Thermal conductance

The thermal conductivity is a function of the average temperature, Tave, given by linear interpolation from a table:

$$k = \mathrm{f}(T_{ave})$$

$$T_{ave} = \frac{T_{in} + T_{out}}{2}$$

The thermal conductance is equal to the thermal conductivity times the area to length ratio:

$$Vcond = k\,F$$

Heat Transport by Conduction for each node is calculated as follows:

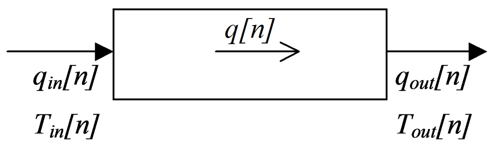
$$q_i = Vcond_i \quad (T_{in,i} - T_{out,i})$$

where:

Vcondi: Thermal conductance at each array item.

Inherited from Abstract Component Conductor

Thermal conductors do not have any heat storage capacity by hypothesis, so the inlet heat flow is equal to the outlet heat flow, and it is also equal to the internal heat flow through the conductor:



$$q_{in,i} = q_i \qquad for\,all\,i \in 1,n$$

$$q_{in,i} = q_{out,i} \qquad for\,all\,i \in 1,n$$

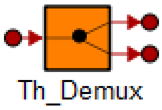
4.27 Component Th_Demux

This abstract component is used for deriving demultiplexer of different number of outputs by inheritance.

4.27.1 Description

This component represents a thermal demultiplexer of different number of outputs. It splits a thermal port of dimension m+n into two ports with dimension m and n.

4.27.2 Symbol



4.27.3 Construction Parameters

Name	Type	Default	Description
n1	CONST INTEGER	1	First outlet thermal port dimension
n2	CONST INTEGER	1	Second outlet thermal port dimension

4.27.4 Ports

Name	Type	Parameters	Direction	Cardinality	Description
tp_in	thermal	(n = n1+n2)	IN	1,1	Thermal inlet port
tp_out1	thermal	(n = n1)	IN	1,1	Thermal outlet port
tp_out2	thermal	(n = n2)	IN	1,1	Thermal outlet port

4.27.5 Formulation

The elements of the input signal vector is divided in two outlet thermal ports. The outlet heat flows and temperatures are calculated as follows:

```
EXPAND(j IN 1, n1) tp_out1.q[j] = tp_in.q[j]
EXPAND(j IN 1, n1) tp_out1.Tk[j] = tp_in.Tk[j]
EXPAND(j IN 1, n2) tp_out2.q[j] = tp_in.q[j+n1]
EXPAND(j IN 1, n2) tp_out2.Tk[j] = tp_in.Tk[j+n1]
```

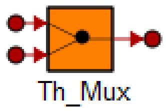
4.28 Component Th_Mux

This component is a thermal multiplexer, bringing together two thermal ports into one port

4.28.1 Description

This component represents a temperature sensor, and it is an interface between the thermal system and the control system.

4.28.2 Symbol



4.28.3 Construction Parameters

Name	Type	Default	Description
n1	CONST INTEGER	1	First inlet thermal port dimension
n2	CONST INTEGER	1	Second inlet thermal port dimension

4.28.4 Ports

Name	Type	Parameters	Direction	Cardinality	Description
tp_in1	thermal	(n = n1)	IN	1,1	Thermal inlet port
tp_in2	thermal	(n = n2)	IN	1,1	Thermal inlet port
tp_out	thermal	(n = n1+n2)	IN	1,1	Thermal outlet port

4.28.5 Formulation

The output signal vector is built concatenating the elements of the input ports. The first dimension (n1) elements of the output signal port will be the set of elements of the first input signal port, the second dimension (n2) elements of the output signal port will be the set of elements of the second input signal port.

```
EXPAND(j IN 1, n1) tp_in1.q[j] = tp_out.q[j]
EXPAND(j IN 1, n1) tp_in1.Tk[j] = tp_out.Tk[j]
EXPAND(j IN 1, n2) tp_in2.q[j] = tp_out.q[j+n1]
EXPAND(j IN 1, n2) tp_in2.Tk[j] = tp_out.Tk[j+n1]
```

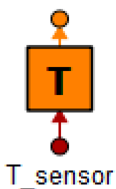
4.29 Component T_sensor

Inherited from component CONTROL.Asensor.

4.29.1 Description

This component represents a temperature sensor, and it is an interface between the thermal system and the control system.

4.29.2 Symbol



4.29.3 Construction Parameters

Name	Type	Default	Description
in	CONST INTEGER	1	Index of the item in temperature array to be measured
n	CONST INTEGER	1	Size of the thermal port array
n_out	CONST INTEGER	1	Dimension of outputs

4.29.4 Ports

Name	Type	Parameters	Direction	Cardinality	Description
s_out	analog_signal	(n = n_out)	OUT		Outlet signal
tp_in	thermal	(n = n)	IN	1,1	Thermal inlet port

4.29.5 Data

Name	Type	Default	Description	Units
Gain[n_out]	REAL	1	Gain	K-1
Bias[n_out]	REAL	0	Bias	-

4.29.6 Variables

Name	Type	Description	Units
Var[n_out]	REAL	Variable being measured	K

4.29.7 Formulation

The value being measured is:

```
s_out.signal = gain · tp_in.T +   bias
```

and it is assumed that the sensor does not produce any heat leak, so:

```
tp_in.q = 0
```

4.30 Component VCHP

4.30.1 Description

This component represents a Variable Conductance Heat pipe (VCHP) which is modelled as an assembly of five Tube components connected in the following order: Evaporator, Adiabatic tube, Condenser, Adiabatic tube and Reservoir.

VCHP behaviour is basically the same as Tube component, particularly heat enters the pipe at the Evaporator section, some of this heat is conducted away by the pipe and some is transmitted to the working fluid (here NH3), which evaporates. The resulting vapour travels along the Adiabatic tube to the cold end of the pipe and condenses in the Condenser section, rejecting heat. When the vapour transforms into liquid again this liquid returns to the evaporator due to capillary forces and the cycle is repeated.

In many cases it is desirable to block or reduce the condenser section, so that the heat transfer characteristics of the pipe can be controlled. This can be done in a number of ways, but this model uses non-condensing gas to control the active length of condenser. The gas is stored in the section called Reservoir, and this section can be heated or cooled in order to vary the partial pressure of the gas (here N2). Because the total pressure inside the pipe is constant, if the gas pressure is increased, the vapour pressure will be reduced, therefore decreasing the active condenser length, and vice versa.

A number of assumptions have been made regarding the VCHP model:

In a heat pipe, the working fluid is contained within the pipe wall and a thin mesh called "wick". The wick has pores to allow the fluid to escape when it evaporates, making its cross sectional area very small so any heat transfer along it can be ignored in our model.

There is no constraint to the fluid flow.

The Reservoir is internally wicked, as any section of a heat pipe, and in equilibrium.

The internal pressure is uniform throughout the pipe. Hence the vapour pressure drop along the pipe is negligible, so we can assume that the vapour temperature remains constant along the pipe.

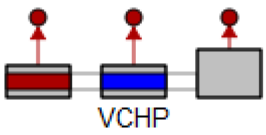
The pipe has a circular cross-section, although other cross-sections can be implemented by modifying the relevant data.

The control gas behaves as an ideal gas.

The vapour behaves as a perfect vapour, obeying in particular the Clapeyron vapour pressure law.

Effects of convection, stratification, diffusion at the vapour-gas front are neglected, i.e. a flat vapour-gas front is assumed. This is called a flat-front model.

4.30.2 Symbol



4.30.3 Parameters

Name	Type	Default	Description	Units
N_evap	CONST INTEGER	1	Number of nodes in evaporator	-
N_ec	CONST INTEGER	1	Number of nodes in evaporator-condenser adiabatic tube	-
N_cond	CONST INTEGER	5	Number of nodes in condenser	-
N_cr	CONST INTEGER	1	Number of nodes in condenser-reservoir adiabatic tube	-
N_rese	CONST INTEGER	1	Number of nodes in reservoir	-

4.30.4 Ports

Name	Type	Parameters	Direction	Description
tp_evap	Thermal	(n = N_evap)	OUT	Evaporator thermal outlet port
tp_cond	Thermal	(n = N_cond)	OUT	Condenser thermal outlet port
tp_rese	thermal	(n = N_rese)	OUT	Reservoir thermal outlet port

4.30.5 Data

Name	Type	Default	Description	Units
l_evap	REAL	1	Length of evaporator	*m*
D_o_evap	REAL	0.02	Outer diameter of evaporator	*m*
D_i_evap	REAL	0.018	Inner diameter of evaporator	*m*
A_vap_evap	REAL	0.0001	Vapour cross section area in evaporator	*m2*
mat_evap	ENUM Material	None	Material used for evaporator	-
h_evap	REAL	1000	Vapour-tube heat transfer coefficient in evaporator	*W/m2 K*
l_ec	REAL	0.5	Length of adiabatic evaporator-condenser tube	*m*
D_o_ec	REAL	0.02	Outer diameter of adiabatic evaporator-condenser tube	*m*
D_i_ec	REAL	0.018	Inner diameter of adiabatic evaporator-condenser tube	*m*
A_vap_ec	REAL	0.0001	Vapour cross section area in adiabatic evaporator-condenser tube	*m2*
mat_ec	ENUM Material	None	Material used for adiabatic evaporator-condenser tube	-
h_ec	REAL	1000	Vapour-tube heat transfer coefficient in evaporator-condenser tube	*W/m2 K*
l_cond	REAL	1	Length of condenser	*m*
D_o_cond	REAL	0.02	Outer diameter of condenser	*m*
D_i_cond	REAL	0.018	Inner diameter of condenser	*m*
A_vap_cond	REAL	0.0001	Vapour cross section area in condenser	*m2*
mat_cond	ENUM Material	None	Material used for condenser	-
h_cond	REAL	1000	Vapour-tube heat transfer coefficient in condenser	*W/m2 K*
l_cr	REAL	0.1	Length of adiabatic condenser-reservoir tube	*m*
D_o_cr	REAL	0.02	Outer diameter of adiabatic condenser-reservoir tube	*m*
D_i_cr	REAL	0.018	Inner diameter of adiabatic condenser-reservoir tube	*m*
A_vap_cr	REAL	0.0001	Vapour cross section area in adiabatic condenser-reservoir tube	*m2*
mat_cr	ENUM Material	None	Material used for adiabatic condenser-reservoir tube	-
h_cr	REAL	1000	Vapour-tube heat transfer coefficient in condenser-reservoir tube	*W/m2 K*
l_rese	REAL	0.1	Length of reservoir	*m*
D_o_rese	REAL	0.05	Outer diameter of reservoir	*m*
D_i_rese	REAL	0.048	Inner diameter of reservoir	*m*
A_vap_rese	REAL	0.0005	Vapour cross section area in reservoir	*m2*
mat_rese	ENUM Material	None	Material used for reservoir	-
h_rese	REAL	1000	Vapour-tube heat transfer coefficient in reservoir	*W/m2 K*
C_vap	REAL	1	Thermal capacitance of vapour	*J/K*
n_gas_o	REAL	0.01	Default number of moles of gas	*mol*
Design	BOOLEAN	FALSE	Flag used to set model on design or transient mode	-
l_vap_o	REAL	2	Default total vapour length for design mode	*m*
T_vap_o	REAL	0	Default vapour temperature for design mode	*K*
T_gas_o	REAL	0	Default gas temperature for design mode	*K*

4.30.6 Variables

Name	Type	Description	Units
RGAS_mol	CONST REAL	Universal gas constant	*J/mol K*
l	REAL	Total heat pipe length	*m*
l_vap	REAL	Total vapour length	*m*
T_vap	REAL	Vapour temperature in pipe	*K*
p	REAL	Total pressure in pipe	*Pa*
n_gas	REAL	Number of moles of gas	*mol*

4.30.7 Formulation

The formulation used here is explained in the tube component, but there are some conditions defining this component:

The axial heat fluxes at the ends of the tube are zero, so

$$evap.tp\_1.q = 0$$

$$rese.tp\_N.q = 0$$

There can be no radial heat flux at the adiabatic sections, hence:

$$adia\_ec.tp\_wall.q_i = 0 \, , \forall i$$

$$adia\_cr.tp\_wall.q_i = 0 \, , \forall i$$

After this the number of ports has been reduced to three thermal type ports. One at the evaporator section, tp_evap, another at the condenser section, tp_cond, and the last one at the reservoir, tp_rese.

The temperature at all tp_fluid must be the same, T_vap, and it is calculated taking into account all heat fluxes from vapour to pipe walls:

$$C_{vap} \cdot \frac{dT_{vap}}{dt} = -(evap.tp\_vapour.q + adia\_ec.tp\_vapour.q +$$
$$+ cond.tp\_vapour.q + adia\_cr.tp\_vapour.q + rese.tp\_vapour.q)$$

The pressure at all s_p.signal must be the same, p, and it is calculated using Psat as a function of vapour temperature, T_vap.

In addition, a certain number of flags have been included to inform about vapour-gas front position, then a warning message is displayed when vapour-gas front goes out of the condenser section indicating in which section it is.

4.31 Component View_Factors

4.31.1 Description

The component type, named View_Factors represents a module that calculates the radiative heat transfer among the nodes by means of the view factors.

4.31.2 Construction Parameters

Name	Type	Default	Description	Units
nports	INTEGER	2	Number of thermal ports connected by the component	-

4.31.3 Ports

Name	Type	Parameters	Direction	Cardinality	Description
tp_in[nports]	thermal	(n = 1)	IN	[0, 1]	Thermal inlet port

4.31.4 Data

Name	Type	Default	Description	Units
VF[nports, nports]	REAL		Array of view factors for each coupling between port arrays	-
e[nports]	REAL		Emissivity	-
A[nports]	REAL		Area	m2

4.31.5 Variables

Name	Type	Description	Units
r[nports]	REAL	Reflectivity of each node	-
B[nports, nports]	REAL	Radiative interchange factor for each GR coupling	-
GR[nports, nports]	REAL	Characteristic value of the radiative conductor for each GR coupling	m2
Q[nports, nports]	REAL	Radiative Heat transfer between two nodes	W
Tk[nports]	REAL	Node absolute temperature	K

4.31.6 Formulation

This component calculated the radiative heat flux between the nodes by means of the data of view factors. The sequence of calculation is the following:

First the radiative interchange factors are calculated:

$$\lfloor B_{i,j} \rfloor = \lfloor r_j F_{i,j} \rfloor \cdot \lfloor B_{i,j} \rfloor + \lfloor e_j F_{i,j} \rfloor$$

where:

Bij: Radiative interchange factor for each node coupling

Fij: View factor for each node coupling

ej: Emissivity of each node

rj: Reflectivity of each node

After that the radiative exchange factors are calculated with the following expression:

$$GR_{i,j} = A_i \cdot B_{i,j} \cdot e_j$$

where:

GRij: Radiative exchange factor for each coupling

Ai: Area of the first node of the coupling

ei: Emissivity of the first node of the coupling

Finally the radiative heat flux between each pair of nodes is calculated in the following way:

$$Q_{i,j} = GR_{i,j} \cdot \sigma \cdot \left(T_i^4 - T_j^4\right)$$

Where Ti and Tj are the absolute temperature of the nodes which forms the coupling and Qij is the radiative heat flux between these nodes.

Ports Equations

The equations in the ports are the following:

$$tp\_in.q_i = \sum_{j=1}^{nports} Q_{ij}$$

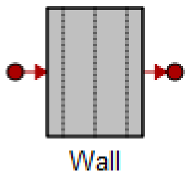
$$T_i = tp\_in.T_i$$

4.32 Component Wall

4.32.1 Description

Defines a wall component divided into a number of nodes.

4.32.2 Symbol



4.32.3 Construction Parameters

Name	Type	Default	Description	Units
n	CONST INTEGER	1	Dimension of thermal port	-
nodes	CONST INTEGER	1	Number of nodes of wall discretisation	-

4.32.4 Ports

Name	Type	Parameters	Direction	Cardinality	Description
tp_in	thermal	(n = n)	IN		Thermal inlet port
tp_out	thermal	(n = n)	OUT		Thermal outlet port

4.32.5 Data

Name	Type	Default	Description	Units
A	REAL	1	Area of the wall	*m2*
e	REAL	0.1	Wall thickness	*m*
mat	ENUM Material	None	Material (see Section)	-
k	REAL	0.1	Thermal conductivity of the wall material (only applies if mat = None)	*W/m K*
rho	REAL	1000	Density of the wall material (only applies if mat = None)	*kg/m3*
cp	REAL	500	Specific heat of the wall material (only applies if mat = None)	*J/kg K*
TemperatureDependance	BOOLEAN	TRUE	Flag to consider temperature dependence of the material properties	
init_mode	ENUM WALL_INIT_MODE	Constant_Temp	Option for the initiliazation of the nodal temperatures	
To	REAL	290	Initial temperature of nodes	*K*
To_linear[2]	REAL	{290,290}	Initial temperatures at both side if init_mode = Linear_Temp	*K*
To_table	TABLE_1D		Table with initial wall temperature versus non-dimensional position if init_mode = Table_Temp	

Input data cp, k and rho will be used only if "mat" = None. If not, the properties will be interpolated using the corresponding data file.

4.32.6 Variables

Name	Type	Description	Units
Asme_DT1	REAL	Linear gradient temperature	*K*
Asme_DT2	REAL	Non linear gradient temperature	*K*
Asme_T_ave	REAL	Average temperature	*K*
F	CONST REAL	Thermal capacity ratio assigned to the surface (ports)	-
k_var [nodes+2]	REAL	Conductivity of wall material	*W/m K*
rho_var	REAL	Density of wall material	*kg/m3*
cp_var[nodes+2]	REAL	Specific heat of wall material	*J/kg*
dx	REAL	Node width or nodal distance	*m*
T[nodes + 2]	REAL	Nodal temperatures	*K*
q[nodes + 1]	REAL	Heat flow through nodes	*W*
icpk[nodes2]	INTEGER	Last table position in cp calculation	
ito	INTEGER	Pointer to time interval for To in Property Tables	

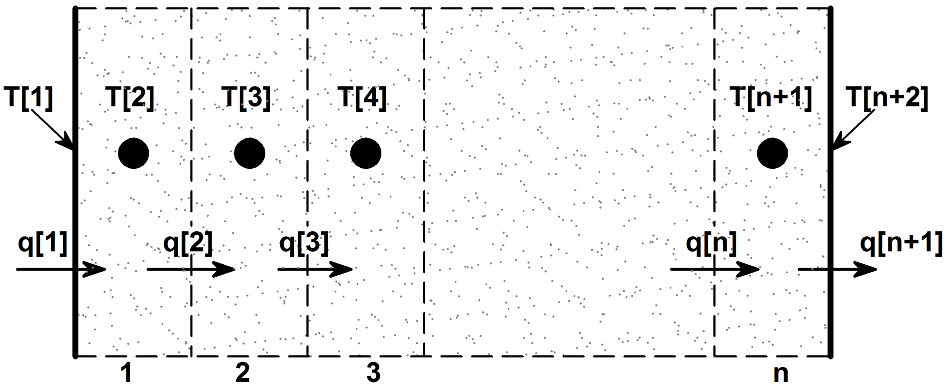
4.32.7 Topology

PATH tp_in TO tp_out

4.32.8 Formulation

Nodalisation

The wall is divided into a number "nodes" of slabs. The array of temperatures contains the temperatures at the mid-points of each slab plus the surface temperature at both sides. The array of heat flows contains the heat flows between slabs plus the heat flows at both sides:



Initialisation

The data named init_mode allows the user to specify the initial temperature distribution in the nodes. There are three options::

- Uniform initial temperature for all the nodes (init_mode = Constant_Temp)
- Linear initial temperature distribution in the wall (init_mode = Linear_Temp)
- Initial temperature distribution given by a data table of the initial temperature as a function of the non-dimensional position in the wall (init_mode = Table_Temp)

Heat flows

The heat flows between the thermal nodes are calculated as follows:

$$q_i = \frac{k_{\mathrm{var}\,i} \cdot k_{\mathrm{var}\,i+1}}{\dfrac{k_{\mathrm{var}\,i} + k_{\mathrm{var}\,i+1}}{2}} \cdot A \cdot \frac{(T_i - T_{i+1})}{(e/nodes)} \qquad for\, i\, in\, 2, nodes$$

$$q_1 = k_{\mathrm{var}\,2} \cdot A \cdot \frac{(T_1 - T_2)}{(e/nodes/2)}$$

$$q_{nodes+1} = k_{\mathrm{var}\,nodes+1} \cdot A \cdot \frac{(T_{nodes+1} - T_{nodes+2})}{(e/nodes/2)}$$

Energy Balance

For all the internal nodes:

$$\rho \cdot cp_i \cdot A \cdot (e/n) \cdot \left(1 - \frac{F}{nodes}\right) \frac{dT_i}{dt} = q_{i-1} - q_i \qquad for\, all\, i\, in\, 2, n+1$$

For the nodes associated to the ports:

$$\rho \cdot cp_1 \cdot A \cdot (e/n) \cdot \left(\frac{F}{2}\right) \cdot \frac{dT_1}{dt} = \sum_{i=1}^{n} tp\_in.q_i - q_1$$

$$\rho \cdot cp_{nodes+2} \cdot A \cdot (e/n) \cdot \left(\frac{F}{2}\right) \cdot \frac{dT_{nodes+2}}{dt} = q_{nodes+1} - \sum_{i=1}^{n} tp\_out.q_i$$

Ports Equations

The temperatures of the thermal ports are:

$$tp\_in.Tk_i = T_1 \qquad for\ i\ in\ 1,n$$
$$tp\_out.Tk_i = T_{n+2} \qquad for\ i\ in\ 1,n$$

5. Functions

These functions provide some complementary functionalities for the calculation of many different values as explained below.

Name	Description
Bmax	It calculates the maximum value between two input values
Bmix	It calculates the minimum value between two input values
Lvap	It calculates the vapour longitude in a tube containing a certain number of moles of non-condensing gas and vapour
NusseltCeiling	It calculates the Nusselt number of a downward facing plate
NusseltCylinder	It computes the natural convection correlation for a horizontal cylinder
NusseltFloor	It calculates the Nusselt number of a upward facing plate
NusseltWall	It calculates the Nusselt number of a vertical plate
NusseltWireHorizontal	It calculates the Nusselt number for a horizontal wire
NusseltWireVertical	It calculates the Nusselt number for a vertical wire
Psat	It calculates NH3 vapour pressure as a function of temperature
FunConstSolidProp	It returns the value of a constant physical property of a material reading it from a text file
FunVarSolidProp	It returns the value of a temperature dependant physical property of a material reading it from a data text file
Ty_integ	It returns calculates the integral of y * T(y) between the limits y1 and y2 assuming T(y) has a linear behaviour between (y1,T1) and (y2, T2).
fun_Wall_Thermal_Gradients	It returns the value of the temperature gradient

5.1 Function Bmax

5.1.1 Description

This function, named Bmax calculates the maximum temperature between two temperatures and stores the value in the variable xmax.

5.1.2 Arguments

Name	Type	Description	Units
xmax	OUT REAL	Variable where the value of the maximum temperature along the simulation is stored	*K*
x	IN REAL	Variable that represents the value of the temperature at each communication interval	*K*

5.1.3 Return Value

This function returns the value of 0. It is a way to give back the control.

5.1.4 Error Messages

None.

5.1.5 Formulation

The function calculates the maximum value between the last maximum value of temperature, xmax, and the present value of temperature, x, in the node. Then the maximum value is stored in xmax.

$$x_{\max} = \max(x_{\max}, x)$$

5.2 Function Bmin

5.2.1 Description

This function, named Bmin, calculates the minimum temperature between two temperatures and stores the value in the variable xmin.

5.2.2 Arguments

Name	Type	Description	Units
xmin	OUT REAL	Variable where the value of the minimum temperature along the simulation is stored	K
x	IN REAL	Variable that represents the value of the temperature at each communication interval	K

5.2.3 Return Value

This function returns 0. It is a way to give back the control.

5.2.4 Error Messages

None.

5.2.5 Formulation

The function calculates the minimum value between the last minimum value of temperature, xmin, and the present value of temperature, x, in the node. Then the minimum value is stored in xmin.

$$x_{\min} = \min(x_{\min}, x)$$

5.3 Function Lvap

5.3.1 Description

This function, named Lvap calculates the vapour longitude in a tube containing a certain number of moles of non-condensing gas and vapour. In addition, Lvap returns as output parameters the vapour-fluid moles ratio at each node used to determine the active length at each tube section for vapour-wall heat exchange evaluation and the remaining number of gas moles if tube is filled.

5.3.2 Arguments

Name	Type	Description	Units
N	IN INTEGER	Number of nodes	-
l	IN REAL	Length of tube	*m*
A	IN REAL	Cross sectional inner area of tube	*m2*
n_gas_in	IN REAL	Number of moles of gas at inlet node	*mol*
p	IN REAL	Total pressure in tube	*Pa*
T[N]	IN REAL	Temperature of gas at each node	*K*
chi[N]	OUT REAL	Vapour-gas moles ratio at each node section	-
n_gas_out	OUT REAL	Number of moles of gas at outlet node	*mol*

5.3.3 Variables

Name	Type	Description	Units
RGAS_mol	CONST REAL	Universal gas constant	*J/mol K*
dx	REAL	Node length	*m*
j	INTEGER	Auxiliary variable used to detect vapour/gas-front node	-
l_vap	REAL	Vapour length measured from the start of the tube	*m*
n_gas	REAL	Total number of moles of gas	*mol*
ngas[N]	REAL	Number of moles of gas in j-node for fully filled node	*mol*

5.3.4 Return Value

This function returns the value of l_vap variable.

5.3.5 Error Messages

None.

5.3.6 Formulation

The formulation in this function is structured in three cases depending on the number of gas moles at the inlet node:

Number of non-condensing gas moles at input node is zero. Then tube is empty and variables are calculated as shown below:

$$\xi_i = 0, \forall i$$
$$n_{out} = 0$$
$$l_{vap} = 1$$

Number of non-condensing gas moles at the input node are sufficient to fill the tube completely. Then tube is fully filled and variables are calculated as shown below, where ntube is the total number of gas moles in tube:

$$\xi_i = 1, \forall i$$
$$n_{out} = n_{in} - n_{tube}$$
$$l_{vap} = 0$$

Number of gas moles at input node only fills tube partially. Then it is necessary to evaluate in which section the vapour-gas front is, supposed j-node, and what the vapour-fluid moles ratio in this section is:

$$\xi_i = 1, \qquad i < j$$

$$\xi_i = \frac{\sum_{i=1}^{j} n_i - n_{in}}{n_j} \qquad i = j$$

$$\xi_i = 0 \qquad i > j$$

$$n_{out} = 0$$

$$l_{vap} = \left((j-1) + \xi_j\right) \cdot dx$$

5.4 Function NusseltCeiling

5.4.1 Description

This function named NusseltCeiling calculates the natural convection correlation for a downward facing plate, i.e. the fluid is placed under the plate.

5.4.2 Arguments

Name	Type	Description	Units
Ra	REAL	Rayleigh Number	-
Ts	REAL	Solid temperature	K
Tf	REAL	Fluid Temperature	K

5.4.3 Variables

Name	Type	Description	Units
Nu	REAL	Nusselt number	-
Nu1	REAL	Nusselt number for first interval	-
Nu2	REAL	Nusselt number for second interval	-
limit1	REAL	Limit for natural convection correlation	
limit2	REAL	Limit for natural convection correlation	
limit3	REAL	Limit for natural convection correlation	
upper_limit_correlation1	REAL	Absolute upper limit for correlation when the temperature of the plate is lower than the temperature of the fluid	
upper_limit_correlation2	REAL	Absolute upper limit for correlation when the temperature of the plate is greater than the temperature of the fluid	

5.4.4 Return Value

This function returns the value of the Nusselt number

5.4.5 Error Messages

```
IF Ts>Tf THEN
ASSERT (Ra <upper_limit_correlation2) WARNING "Raleigh Number in Natural
  Convection Correlation for Ceiling is out of limit"
ELSE
ASSERT (Ra <upper_limit_correlation1) WARNING "Raleigh Number in Natural
  Convection Correlation for Ceiling is out of limit"
```

By default the value of variable setRaleighWarnings is 1 that means the validity range of the function is checked.

5.4.6 Formulation

The function calculates the Nusselt number for a downward facing plate.

If the temperature of the plate is greater than the temperature of the fluid, then the correlation applied is the following:

$$Nu = 0.27 \cdot Ra^{1/4}$$

However, if the temperature of the plate is lower than the temperature of the fluid, then the following correlations are applied depending on the value of the Rayleigh number:

$$\begin{aligned}
&Nu = 0.96 \cdot Ra^{1/6} && 1 < Ra < 200 \\
&Nu = 0.59 \cdot Ra^{1/4} && 200 < Ra < 10^4 \\
&Nu = 0.54 \cdot Ra^{1/4} && 2.2 \times 10^4 < Ra < 8 \times 10^6 \\
&Nu = 0.15 \cdot Ra^{1/3} && 8 \times 10^6 < Ra < 1 \times 10^{11}
\end{aligned}$$

5.5 Function NusseltCylinder

5.5.1 Description

This function named NusseltCylinder computes the natural convection correlation for a horizontal cylinder.

5.5.2 Arguments

Name	Type	Description	Units
Pr	REAL	Prandt Number	-
Ra	REAL	Rayleigh Number	-

5.5.3 Variables

Name	Type	Description	Units
Nu	REAL	Nusselt number	-
lower_limit_correlation	REAL	Lower limit for the correlation validity	-
upper_limit_correlation2	REAL	Upper limit for the correlation validity	-

5.5.4 Return Value

This function returns the value of the Nusselt number for a horizontal cylinder

5.5.5 Error Messages

```
ASSERT (Ra <= upper_limit_correlation) WARNING "Raleigh Number in Natural
  Convection Correlation for Cylinder is out of upper limit"
```

```
ASSERT (Ra >= lower_limit_correlation) WARNING "Raleigh Number in Natural
  Convection Correlation for Cylinder is out of lower limit"
```

5.5.6 Formulation

The function calculates the Nusselt number for a horizontal cylinder using the following correlation.

$$Nu = \left\{ 0.6 + \frac{0.387 \cdot Ra^{1/6}}{\left[1 + (0.559/\mathrm{Pr})^{9/16}\right]^{8/27}} \right\} \qquad 10^9 < Ra < 10^{13}$$

5.6 Function NusseltFloor

5.6.1 Description

This function named NusseltFloor calculates the natural convection correlation for a upward facing plate, i.e. the fluid is placed over the plate.

5.6.2 Arguments

Name	Type	Description	Units
Ra	REAL	Rayleigh Number	-
Ts	REAL	Solid temperature	K
Tf	REAL	Fluid Temperature	K

5.6.3 Variables

Name	Type	Description	Units
Nu	REAL	Nusselt number	-
Nu1	REAL	Nusselt number for first interval	-
Nu2	REAL	Nusselt number for second interval	-
limit1	REAL	Limit for natural convection correlation	
limit2	REAL	Limit for natural convection correlation	
limit3	REAL	Limit for natural convection correlation	
upper_limit_correlation1	REAL	Absolute upper limit for correlation when the temperature of the plate is lower than the temperature of the fluid	
upper_limit_correlation2	REAL	Absolute upper limit for correlation when the temperature of the plate is greater than the temperature of the fluid	

5.6.4 Return Value

This function returns the value of the Nusselt number for a upward facing plate

5.6.5 Error Messages

```
IF Ts<Tf THEN
ASSERT (Ra <upper_limit_correlation2 OR setRaleighWarnings==0) WARNING "
  Raleigh Number in Natural Convection Correlation for Floor is out of limit"
Nu = 0.27*Ra**(1./4.)
ELSE
```

```
ASSERT (Ra <upper_limit_correlation1 OR setRaleighWarnings==0) WARNING "
  Raleigh Number in Natural Convection Correlation for Floor is out of limit"
```

By default the value of variable setRaleighWarnings is 1 that means the validity range of the function is checked.

5.6.6 Formulation

The function calculates the Nusselt number for a upward facing plate.

If the temperature of the plate is lower than the temperature of the fluid, then the correlation applied is the following:

$$Nu = 0.27 \cdot Ra^{1/4}$$

However, if the temperature of the plate is greater than the temperature of the fluid, then the following correlations are applied depending on the value of the Rayleigh number:

$$\begin{aligned}
Nu &= 0.96 \cdot Ra^{1/6} && 1 < Ra < 200 \\
Nu &= 0.59 \cdot Ra^{1/4} && 200 < Ra < 10^4 \\
Nu &= 0.54 \cdot Ra^{1/4} && 2.2 \times 10^4 < Ra < 8 \times 10^6 \\
Nu &= 0.15 \cdot Ra^{1/3} && 8 \times 10^6 < Ra < 1 \times 10^{11}
\end{aligned}$$

5.7 Function NusseltWall

5.7.1 Description

This function named NusseltWall calculates the natural convection correlation for a vertical plate (a wall).

5.7.2 Arguments

Name	Type	Description	Units
Pr	REAL	Prandt Number	-
Ra	REAL	Rayleigh Number	-

5.7.3 Variables

Name	Type	Description	Units
Nu	REAL	Nusselt number	-
Nu1	REAL	Nusselt number for first interval	-
Nu2	REAL	Nusselt number for second interval	-
limit	REAL	Limit for natural convection correlation	
deviation	REAL		
upper_limit_correlation	REAL	Upper limit for correlation validity	

5.7.4 Return Value

This function returns the value of the Nusselt number for a vertical plate.

5.7.5 Error Messages

```
ASSERT (Ra <upper_limit_correlation OR setRaleighWarnings==0) WARNING "
  Raleigh Number in Natural Convection Correlation for Wall is out of limit"
```

By default the value of variable setRaleighWarnings is 1 that means the validity range of the function is checked.

5.7.6 Formulation

The function calculates the Nusselt number for a vertical plate using the following correlations as a function of the Rayleigh number:

$$Nu = \left\{ 0.68 + \frac{0.67 \cdot Ra^{1/4}}{\left[1 + (0.492/\mathrm{Pr})^{9/16}\right]^{4/9}} \right\} \qquad 10^{-1} < Ra < 10^{9}$$

$$Nu = \left\{ 0.825 + \frac{0.387 \cdot Ra^{1/6}}{\left[1 + (0.492/\mathrm{Pr})^{9/16}\right]^{8/27}} \right\} \qquad 10^{9} < Ra < 10^{12}$$

5.8 Function NusseltWireHorizontal

5.8.1 Description

This function named NusseltWireHorizontal computes the natural convection correlation for a horizontal wire.

5.8.2 Arguments

Name	Type	Description	Units
Pr	REAL	Prandt Number	-
Ra	REAL	Rayleigh Number	-

5.8.3 Variables

Name	Type	Description	Units
Nu	REAL	Nusselt number	-
c	REAL	Calculated parameter	-
n	REAL	Calculated exponent	-
lower_limit_correlation	REAL	Lower limit for correlation validity	
upper_limit_correlation	REAL	Upper limit for correlation validity	

5.8.4 Return Value

This function returns the value of the Nusselt number for a horizontal wire.

5.8.5 Error Messages

```
ASSERT (Ra <= upper_limit_correlation) WARNING "Raleigh Number in Natural
  Convection Correlation for Horizontal Wire is out of upper limit"
```

```
ASSERT (Ra >= lower_limit_correlation) WARNING "Raleigh Number in Natural
  Convection Correlation for Horizontal Wire is out of lower limit"
```

5.8.6 Formulation

The function calculates the Nusselt number for a horizontal wire:

$$\frac{2}{Nu} = \ln\left(1 + \frac{3.3}{c \cdot Ra^n}\right) \qquad 10^{-8} < Ra < 10^6$$

where c and n is calculated as follows:

$$c = \frac{0.671}{[1 + (0.492/\mathrm{Pr})^{(9/16)}]^{(4/9)}}$$

$$n = 0.25 + \frac{1}{10 + 5 \cdot (Ra)^{0.175}}$$

5.9 Function NusseltWireVertical

5.9.1 Description

This function named NusseltWireVertical computes the natural convection correlation for a vertical wire.

5.9.2 Arguments

Name	Type	Description	Units
Pr	REAL	Prandt Number	-
Ra	REAL	Rayleigh Number	-
d	REAL	Diameter	m
L	REAL	Length of the wire	m

5.9.3 Variables

Name	Type	Description	Units
Nu	REAL	Nusselt number	-
c	REAL	Calculated parameter	-
n	REAL	Calculated exponent	-
lower_limit_correlation	REAL	Lower limit for correlation validity	

5.9.4 Return Value

This function returns the value of the Nusselt number for a vertical wire.

5.9.5 Error Messages

```
ASSERT (c*(Ra*d/L)**0.25 > lower_limit_correlation) WARNING "Raleigh Number
  in Natural Convection Correlation for Vertical Wire is out of lower limit"
```

5.9.6 Formulation

The function calculates the Nusselt number for a vertical wire:

$$Nu = c \cdot (Ra \cdot D / L)^{0.25} + 0.763 \cdot c^{(1/6)} \cdot (Ra \cdot D / L)^{(1/24)} \qquad c \cdot (Ra \cdot D / L)^{0.25} > 2 \times 10^{-3}$$

where c and n is calculated as follows:

$$c = \frac{0.671}{[1 + (0.492/\Pr)^{(9/16)}]^{(4/9)}}$$

$$n = 0.25 + \frac{1}{10 + 5 \cdot (Ra)^{0.175}}$$

And D and L are the diameter and the length of the wire respectively.

5.10 Function Psat

5.10.1 Description

This function, named Psat calculates the pressure of NH3 vapour as a function of vapour temperature using a polynomial of 7th order.

5.10.2 Arguments

Name	Type	Description	Units
T	IN REAL	Temperature of NH3	K

5.10.3 Return Value

This function returns the value of NH3 vapour pressure in Pa at temperature T.

5.10.4 Error Messages

None.

5.10.5 Formulation

The expression used to evaluate NH3 pressure is:

$$p_{vap} = 4.283253 \cdot 10^{5} + 1.598058 \cdot 10^{4} \cdot T_{vap} + 2.371318 \cdot 10^{2} \cdot T_{vap}{}^{2} + 1.616482 \cdot T_{vap}{}^{3} + \\ + 2.325724 \cdot 10^{-3} \cdot T_{vap}{}^{4} - 1.272831 \cdot 10^{-5} \cdot T_{vap}{}^{5} + 1.313368 \cdot 10^{-7} \cdot T_{vap}{}^{6}$$

where Tvap is expressed in ºC

5.11 Function FunConstSolidProp

5.11.1 Description

This function named FunConstSolidProp returns the value of a constant physical property of a material reading it from a text file.

5.11.2 Arguments

Name	Type	Description	Units
mat	ENUM Material	Material	-
prop	ENUM PropSolid	Property	-
v_None	REAL	Value of the property if Material = None	-

5.11.3 Variables

Name	Type	Description	Units
v	REAL	Output value	-
ier	INTEGER	Error identifier	-

5.11.4 Return Value

This function returns the value of the constant physical property defined by the function parameter "prop" of the material defined by the function parameter "mat".

5.11.5 Error Messages

```
IF (ier != 0) THEN
PRINT("****Error: Problems with Table of Properties for material $mat")
PRINT("           Error code ier = $ier")
END IF
ASSERT(ier == 0) FATAL "Execution aborted: Table for Solid Properties not
  found or wrong"
```

5.11.6 Formulation

If the parameter mat is equal to "None" then the value returned by the function is equal to the parameter v_None.

If the user has specified a material, then the function calls the EL function get_const_prop_from_file in order to read the property defined in the parameter prop from the corresponding text file.

v = get_const_prop_from_file(path_tables, Mat_internal_name[mat], IdProp[prop], IdMat[mat], ier)

5.12 Function FunVarSolidProp

5.12.1 Description

This function named FunVarSolidProp returns the value of a temperature dependant physical property of a material reading it from a data text file.

5.12.2 Arguments

Name	Type	Description	Units
mat	IN ENUM Material	Material	-
prop	IN ENUM PropSolid	Property	-
Tk	IN REAL	Temperature	K
v_None	IN REAL	Value of the property if Material = None	-
itab_pointer	OUT INTEGER	Pointer to last table interval for speed up calculation	-

5.12.3 Variables

Name	Type	Description	Units
tab	TABLE_1D	Table of properties	
Tmin	REAL	Minimum temperature range	K
Tmax	REAL	Maximum temperature range	K
v	REAL	Output value	-
ier	INTEGER	Error identifier	-

5.12.4 Return Value

This function returns the value of the physical property defined by the input parameter "prop" of the material defined by the input parameter "mat" as function of the temperature.

5.12.5 Error Messages

```
IF (ier != 0) THEN
PRINT("****Error: Problems with Table of Properties for material $mat")
END IF
ASSERT(ier == 0) FATAL "Execution aborted: Table for Solid Properties not
  found or wrong"
      IF(Tk < T_min) THEN
         IF(ier_mi[mat] == 0)THEN
            WRITE("\n **** Material: %s. Temperature (%g) below minimum
              range: %g\n\n", Mat_internal_name[mat],Tk,T_min)
         END IF
      END IF
      IF(Tk > T_max) THEN
         IF(ier_ma[mat] == 0)THEN
            WRITE("\n **** Material: %s. Temperature (%g) above maximum
              range: %g\n\n", Mat_internal_name[mat],Tk,T_max)
         END IF
```

5.12.6 Formulation

If the parameter mat is equal to "None" then the value returned by the function is equal to the parameter v_None.

If the user has specified a material, then the function calls the EL function get_table_prop_from_file in order to read the data table of the property as a function of the temperature (TABLE_1D).

get_table_prop_from_file(path_tables, Mat_internal_name[mat], IdProp[prop], tab, T_min,T_max, IdMat[mat], ier)

Then the EL function called "linearInterpHist1D) is called in order to calculate the value of the property at specified temperature interpolating in the data table.

v = linearInterpHist1D(tab, Tk, itab_pointer)

5.13 Function Ty_integ

5.13.1 Description

This function named Ty_integ returns calculates the integral of y * T(y) between the limits y1 and y2 assuming T(y) has a linear behaviour between (y1,T1) and (y2, T2).

5.13.2 Arguments

Name	Type	Description	Units
T1	IN REAL	Temperature at point 1	*K-*
T2	IN REAL	Temperature at point 2	*K*
y1	IN REAL	Position at point 1	-
y2	IN REAL	Position at point 2	-

5.13.3 Variables

Name	Type	Description	Units
alpha	REAL	Slope of the function T(y) in the interval considered	-
integ	REAL	Value of the integral	-

5.13.4 Return Value

This function returns the value of the integral of the following function y * T(y).

5.13.5 Error Messages

None

5.13.6 Formulation

Considering that the function T(y) has a linear behaviour between (y1,T1) and (y2, T2), the integral of the function (y * T(y)) is computed as follows:

$$\int_{y1}^{y2} T(y) \cdot y \cdot dy = \frac{T_1 \cdot \left(y_2^{\ 2} - y_1^{\ 2}\right)}{2} + \alpha \cdot \left(\frac{y_2^{\ 3}}{3} - \frac{y_2^{\ 2} \cdot y_1}{2} + \frac{y_1^{\ 3}}{6} \right)$$

where α is equal to:

$$\alpha = \frac{T_2 - T_1}{y_2 - y_1}$$

And the function T(y) in the interval [y1, y2] is equal to:

$$T(y) = \alpha \cdot (y - y_1) + T_1$$

5.14 Function fun_Wall_Thermal_Gradients

5.14.1 Description

This function named fun_Wall_Thermal_Gradients returns the value of the temperature gradient

5.14.2 Arguments

Name	Type	Description	Units
nodes	IN INTEGER	Number of nodes	-
T[nodes]	IN REAL	Nodal temperatures	*K-*
e	IN REAL	Wall thickness	*m*
T_ave	OUT REAL	Average temperature	*K*
DT1	OUT REAL	Linear thermal gradient	*K*
DT2	OUT REAL	Linear thermal gradient	*K*

5.14.3 Variables

5.14.4 Return Value

None

5.14.5 Error Messages

None

5.14.6 Formulation

This function computes the absolute value of the range for that portion of the nonlinear thermal gradient (ΔT2) through the wall thickness not included in ΔT1 and the absolute value of the range of the temperature difference (ΔT1) between the temperature of the outside surface and the temperature of the inside surface of the piping product assuming moment generating equivalent linear temperature distribution.

$$\Delta T_1 = \left(12/t^2\right)\int_{-t/2}^{t/2} y \cdot T(y) \cdot dy$$

$$\Delta T_2 = \max\left(\left|T_o - T\right| - \tfrac{1}{2} \cdot \left|\Delta T_1\right|, \left|T_i - T\right| - \tfrac{1}{2} \cdot \left|\Delta T_1\right|, 0\right)$$

5.15 Function Fun_prop_vs_pT

5.15.1 Description

This function named Fun_prop_vs_pT computes the value of the specific heat, viscosity, density and thermal conductivity of a 1-phase pure fluid as a function of the temperature and the pressure.

5.15.2 Arguments

Name	Type	Description	Units
chem	IN ENUM ThFluids	Name of the fluid	-
p	IN REAL	Pressure	*Pa*
T	IN REAL	Temperature	*K*
Cp	OUT REAL	Specific heat	*W/kg K*
vis	OUT REAL	Viscosity	*kg/m/s*
rho	OUT REAL	Density	*kg/m3*
K	OUT REAL	Thermal conductivity	*W/m K*
itab	OUT INTEGER	Pointer to last table interval for speed up calculation	-

5.15.3 Variables

Name	Type	Description	Units
MW	REAL	Molecular mass	*kg/kmol*
ier	INTEGER	Error identifier	-

5.15.4 Return Value

None

5.15.5 Error Messages

```
IF (ier != 0) THEN
WRITE(""*****Error: Problems reading fluid properties.)
END IF
ASSERT(ier == 0) FATAL " "Execution aborted: Table for Simplified Fluid
  Properties not found or wrong"
--Test allowed ranges
        IF(Tk < 0.9*T_min) THEN
           IF(ier_fl_mi[liq] == 0)THEN
             WRITE("\n **** fluid: %s. Temperature (%g) below minimum range:
                %g\n\n", FluidFileName[liq],Tk,T_min)
           END IF
           ier_fl_mi[liq] = 1
        END IF
        IF(Tk > 1.1*T_max) THEN
           IF(ier_fl_ma[liq] == 0)THEN
             WRITE("\n **** fluid: %s. Temperature (%g) above maximum range:
                %g\n\n", FluidFileName[liq],Tk,T_max)
           END IF
           ier_fl_ma[liq] = 1
        END IF
```

5.16 Function Fun_beta_vs_T

5.16.1 Description

This function named Fun_beta_vs_T computes the value coefficient of volumetric expansion of a 1-phase pure fluid as a function of the temperature.

5.16.2 Arguments

Name	Type	Description	Units
chem	IN ENUM ThFluids	Name of the fluid	-
T	IN REAL	Temperature	*K*
itab	OUT INTEGER	Pointer to last table interval for speed up calculation	-

5.16.3 Variables

Name	Type	Description	Units
B	REAL	Coefficient of volumetric expansion	*1/K*
ier	INTEGER	Error identifier	-

5.16.4 Return Value

This function returns the value of the coefficient of volumetric expansion of the 1-phase fluid defined by the input parameter "chem" as function of the temperature.

6. References

1. http://www.melcor.com/formula.htm
2. EcosimPro User Manual
4. ESATAN User Manual. UM-ESATAN-004. ESATAN 9.2. October 2004
5. ESATAN Training Manual. TM-ESATAN-008. ESATAN 9.2. October 2004
6. Ingeniería Química. Costa Novella E. Vol 4. Transmisión de calor. Ed. Alhambra. 1986
7. Heat Transfer, A Practical Approach. Yunus A. Çengel. 2nd Edition. McGraw Hill. 2003.
8. 2005 ASHRAE Handbook Fundamentals. SI Edition

This page intentionally left blank.

This page intentionally left blank.

EA Internacional
Magallanes, 3. 28015 Madrid. SPAIN.
E-mail: info@ecosimpro.com
Web: www.ecosimpro.com

EMPRESARIOS AGRUPADOS
Edition 2016