THERMICA V4.9

User Guide

September 2024

Ref: UM.000040345.AIRB
Issue: 8

Use of the software and of the present software manual is submitted to a license agreement to be accepted before the software installation on a computer.

All suggestion or error concerning the software or this software manual can be sent to:

AIRBUS DEFENCE AND SPACE

To the attention of systema.business@airbus.com

Z.I. du Palays

31 rue des Cosmonautes

31402 TOULOUSE CEDEX 4

FRANCE

Table of Contents

Generalities

THERMICA User Manual

This Manual describes the THERMICA application and its integration into the SYSTEMA v4 framework.
For the general usage of SYSTEMA,
please refer to the SYSTEMA User Manual

Introduction

The **THERMICA** applications globally transform the geometric model into a mathematical model and perform a complete thermal analysis of a space system orbiting around a planet or along an interplanetary trajectory. On this thermal model, **THERMICA** computes all the elements needed to simulate the temperature. The temperature computation is then performed by a thermal solver which may be **THERMISOL**, the solver of **THERMICA**, or MSC SINDA. The description of the simulation transmitted to the temperature solver is based on a nodal description. Here, the thermal analysis no longer requires any geometry. The main functions of **THERMICA** are as follows:

- **Geometry modeling and physical properties & meshing**
- **Mission modeling: orbit & pointing**
- **Physical simulation: radiative & conductive couplings, solar & planet fluxes**
- **Translation of the geometrical problem to a nodal network problem.**

Within the *SYSTEMA processing* tab, the **THERMICA** application is organized into modules, which are based on 6 sub-applications and used to manage the THERMICA entities:

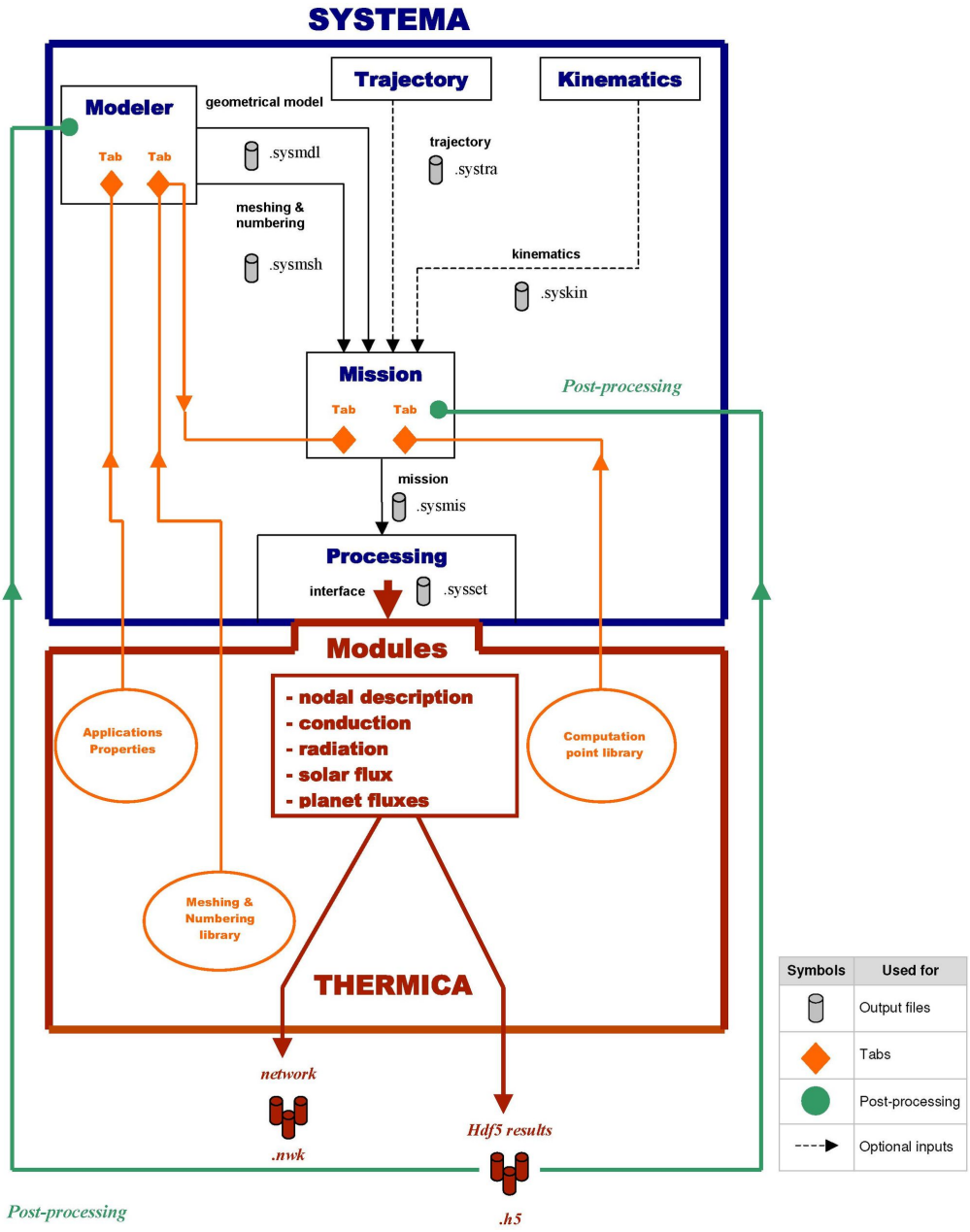
Nodal description:	**this module is used to translate the nodal breakdown including the geometrical mesh and specific elements (except contact resistance which are handled by the conduction module)**
Conduction:	**this module is used to create conductive links between meshes. These links are created through edge nodes that are also created by this module.**
Radiation:	**this module is used to create radiative links.**
Solar flux:	**this module is used to give the quantity of solar flux collected by each mesh.**
Planet fluxes:	**this module is used to give the quantity of planet Albedo and IR fluxes collected by each mesh.**
Aero flux:	**this module is used to give the quantity of planet Albedo and IR fluxes collected by each mesh.**
Emission source:	**this module is used to simulate the thermal flux exchanges between UV/IR lamps and the geometrical nodes of the model.**
Convection:	**this module is used to compute surface contacts between thermal nodes and air nodes, then to estimate the conducto-convective couplings according to standard or Mac Adams formulaes**

Note : Some beta modules such as IR Camera are no longer available in THERMICA.

We finally obtain a complete description of the thermal model which is not expressed in terms of geometry but by means of a mathematical model composed of a certain number of interconnected variables forming a network.

About THERMICA

THERMICA is a plug-in application of the **SYSTEMA** environment. This manual makes therefore reference to the modeler, mission and processing modules of SYSTEMA. This manual contains task-oriented instructions which show you how to use the **THERMICA** program. The different computation techniques supported by **THERMICA** are also presented. The application may be illustrated as follows:



THERMICA Modules

THERMICA Modules

The different modules and corresponding files

In this User Manual, the "**module**" or "**THERMICA module**" terms will be used to refer to the sub-processes managed by the **THERMICA** application, namely ***nodal description***, ***conduction module***, ***radiation module***, ***solar flux, planet fluxes, aero flux, emission source and convection.***
Each module defines its input files, parameters and output files.
The parameters of each module will be detailed in the chapters dealing with the different modules.
The following tables present the different modules and corresponding files:

INPUT File extension

Module	File extension
Nodal description	**".sysset"**
Conduction	**".sysset"**
Radiation	**".sysset"**
Solar flux	**".sysset"** **".box.h5"** (resulting from radiative calculation)
Planet fluxes	**".sysset"** **".box.h5"** (resulting from radiative calculation) **".map"**
Aero flux	**".sysset"** **".box.h5"** (resulting from radiative calculation)
Emission source	**".sysset"**
Convection	**".sysset"**

INPUT file extension
*The **".sysset"** file refers to a mission defined in the SYSTEMA V4 platform.*

OUTPUT File extension

Module	File extension	

Nodal description	**".nod.log"** **".nod.nwk"** (nodal description)	
Conduction	**".cond.log"** **".gl.nwk"** (conductive couplings) **".edg.h5"** (for edge display)	
Radiation	**".rad.h5"** **".rad.log"** **".gr.nwk"** (radiative couplings)	
	Optional:	**".gb.txt"** (view, IR and UV gebhart factors) **".gv.nwk"** (view couplings) **".rrt.h5"** (radiative ray-tracing results)
Solar flux	**".sf.h5"** **".sf.log"** **".sf.nwk"** (absorbed solar flux)	
	Optional:	**".srt.h5"** (radiative ray-tracing results)
Planet fluxes	**".pf.h5"** **".pf.log"** **".pf.nwk"** (absorbed planet fluxes)	
Aero flux	**".af.h5"** **".af.log"** **".af.nwk"** (absorbed aero-thermal flux)	
Emission source	**".emi.h5"** **".emi.log"** **".emi.nwk"** (absorbed flux coming from point source)	
	Optional:	**".uvert.h5"** (UV ray-tracing results) **".irert.h5"** (IR ray-tracing results)
Convection	**".conv.h5"** **".conv.log"** **".gf.nwk"** (convective couplings)	

OUTPUT file extension
*The **".h5"** outputs are HDF5 files used for result interpretation.They can also be loaded into an HDF explorer.*
*The **".log"** file provides general information.*
*The **".nwk"** ascii files are parts of the mathematical model.They can be written in the Thermisol or Sinda format depending on the thermal solver used at a later stage.*
*The **".txt"** files output by the radiation module are extra data written with the format*

	*of the **".nwk"** files but are not taken to complete the mathematical model for the thermal solver.*

THERMICA Model Properties

THERMICA Model Properties

- The Modeler tab is used to work on models and meshing. According to the *SYSTEMA* configuration, some applications can be interfaced with *SYSTEMA*. In this case, the elements of the geometrical model have application-dependent properties.

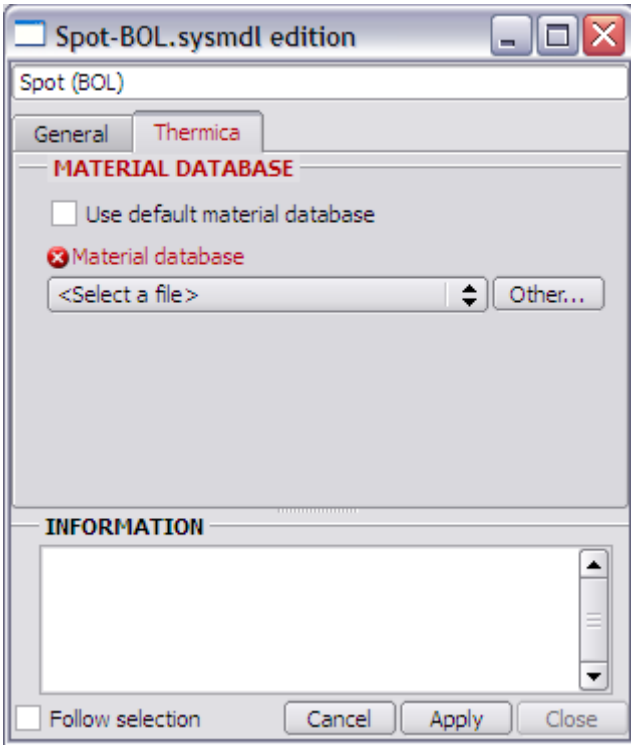
These properties can be consulted or modified using the edition window, via a specific tab (called "**THERMICA** tab" in this document). Each application has its own tab.

	Refer to SYSTEMA User Guide (Chapter 4: "Geometrical Model Management")

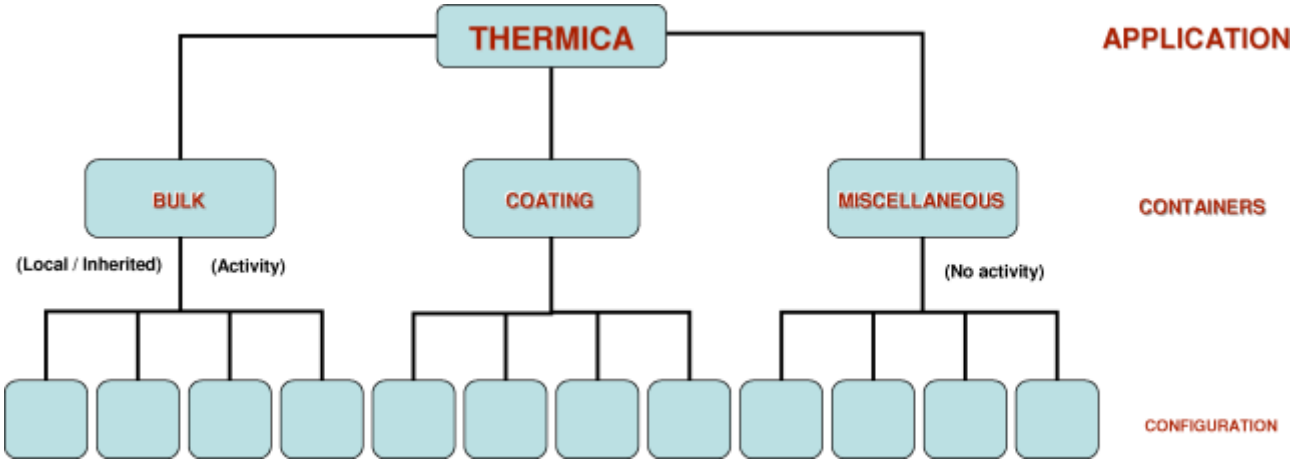
THERMICA Material tab

The user defines material associations via the specific tab of the edition window. In the following figure, the user can associate a material file with the application-dependent properties.

Example
The Material tab (Figure 2) lets the user enter a material filename for the physical properties and the thermo-optical IR & UV.
To get access to this window, open the edit window at the host level.



Applicative Properties



The THERMICA applicative properties are sorted into 5 containers for the shells and 3 for the volumes.

The shell containers are: Bulk, Transverse, Coating, Miscellaneous, Enclosure

The volume containers are: Bulk, Coating, Enclosure

Each container can be in "Inherited configuration" or in "Local configuration". The local configuration allows to modify the "Side management" of the container which may propose to set alike properties for both sides or distinct properties for the positive and negative sides.

Bulk

This container refers to the internal material settings. Shells and volumes can be assigned with bulk properties.

In the case of shells, it is necessary to also indicate a geometrical thickness. In addition, it is possible to switch to a "Distinct sides" management in order to specify 2 layers of bulk properties.

In any case, the physical properties (conductivity, specific heat, density) can be set by a material reference.

The conductivity may be defined as a temperature dependent parameter. In order to ease the export of such a dependent property into the thermal solver format, it is required that this property, if temperature dependent, is defined through a material. This will allows exporting one array by material into the thermal solver output (nwk file) rather than one array by thermal node. Also, in the case of a condensed node defined with atemperature dependent conductivty, all geometrical shapes condensed shall be set with the same bulk material.

This container will affect the following:

- **Nodal description**
- **Conduction module**

The Bulk activity will affect the conductive couplings computation from the conductive module.

Transverse

This container may be used to define transverse couplings between positive and negative side nodes, in case the nodal breakdown of the model (i.e. the meshing) is made of two distinct side nodes.

Both a conductive and radiative contribution may be given so to ease, for example, the MLI efficiency modelling.

Material database can include transverse properties

This container will affect the following:

- **Nodal description**

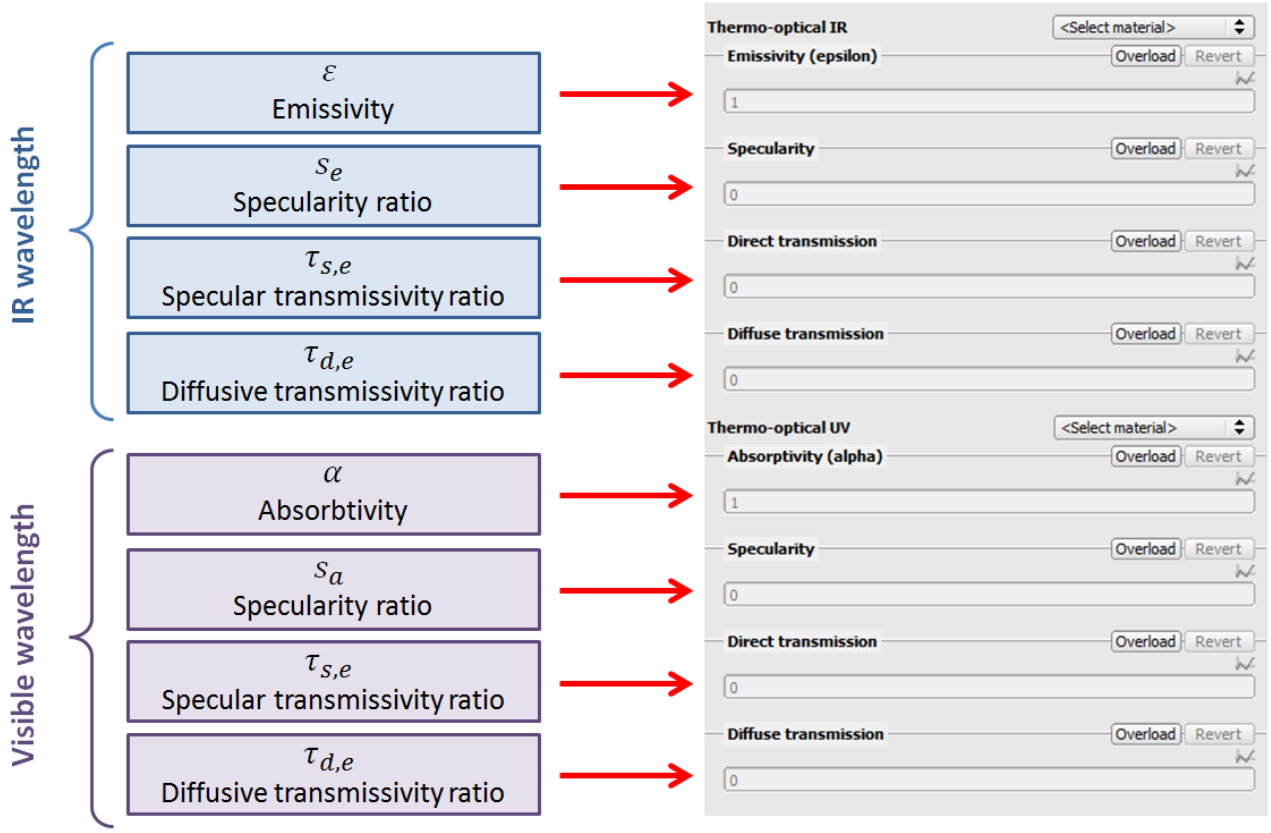
Coating

This container refers to the coating material settings and especially thermo-optical properties. It is also possible to choose the "Ray emission" level (none, low, normal, critical) so to use more or less ray from this geometrical element.

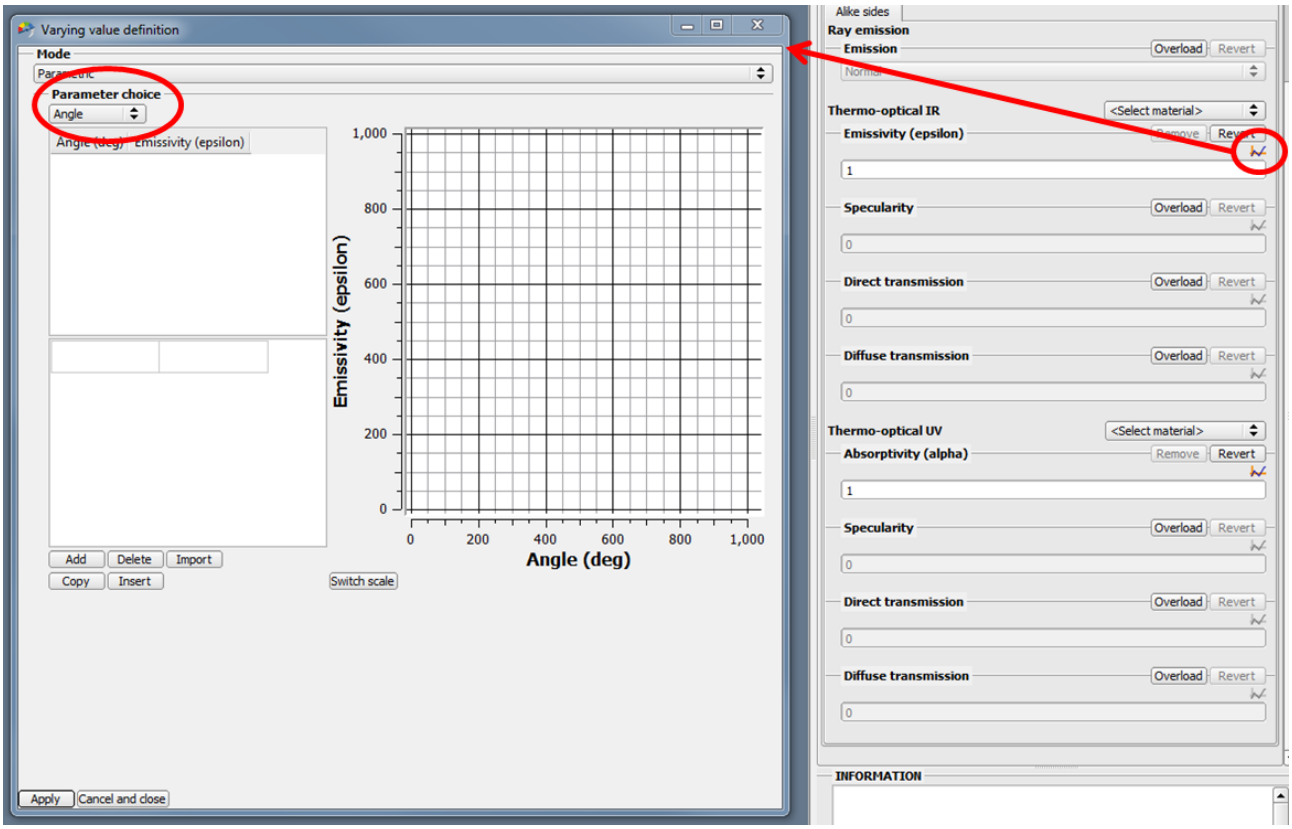
For volumes, there is no notion of side management. The settings are set for the externe part of the volume only.

The thermo-optical properties are, for the IR and Visible wavelength:

- ε/α : emissivity / absorbtivity - correspond to the amount of energy absorbed in the IR and Visible range. In the IR range, the epsilon coefficient also drives the IR radiative emission.
- d_e/d_a : diffusive ratios in the IR and Visible wavelength (this property is in fact deduced from the other).
- s_e/s_a : specularity ratios in the IR and Visible wavelength.
- $\tau_{s,e}/\tau_{s,a}$: specular transmissivity ratios in the IR and Visible wavelength.
- $\tau_{d,e}/\tau_{d,a}$: diffusive transmissivity ratios in the IR and Visible wavelength.



All these coefficients may be defined as incident angle dependent parameters (see figure below). Note that the angle is defined in relation to the tangent to the shape, so that an angle of 0° represents a ray tangent to the surface, and an angle of 90° represents a ray normal to the surface.



All IR coefficient may otherwise be defined with a wavelength dependency (by selecting "wavelength" for the parameter choice in the figure above). In that case, the wavelength dependent profile may be accurately defined, the wavelength band splitting of the computation being done at processing level.

The reflections or transmissions coefficients are specified independently of the absorption coefficients, meaning that the energy budget on the reflected/transmitted is splitted into the 4 modes:

IR wavelength	Visible wavelength
$d_e + s_e + \tau_{s,e} + \tau_{d,e} = 1$	$d_a + s_a + \tau_{s,a} + \tau_{d,a} = 1$

So that the diffusive coefficients are in fact deduced from the other reflection/transmission modes:

IR wavelength	Visible wavelength
$d_e = 1 - s_e - \tau_{s,e} - \tau_{d,e}$	$d_a = 1 - s_a - \tau_{s,a} - \tau_{d,a}$

The complete energy budget at a ray impact is then (for the IR wavelength for example):

$$\varepsilon + (1 - \varepsilon)(d_e + s_e + \tau_{s,e} + \tau_{d,e}) = 1$$

WARNING:

- **THERMICA v3 parameters had a different interpretation** in the cases of non-zero transmitive coefficients (refer to the v3 manual). The import/export with v3 models howevers convert the coefficients if need be.
- **Step-Tas parameters are also different** (based on the total energy budget): $\varepsilon + d_e + s_e + \tau_{s,e} + \tau_{d,e} = 1$. Import/export of step-tas models also convert properly these coefficients according to their respective definitions.

In the case of volumes, transmittive coefficients are automatically set to zero and cannot be modified, i.e. only reflective modes are possible.

This container will affect the following:

- **Nodal description**
- **Radiation**
- **Solar flux**
- **Planet fluxes**

The Coating activity will affect the radiative couplings and external fluxes computation.

Miscellaneous

This container impacts the ray-tracing modules radiation and solar flux. It is meaningful only if the direct transmission ratio is a positive value.
However this container is handled independently because it does not concern the coating itself but defines optical indexes of the positive and negative side environments.

It is only used by shell elements since transmittive coefficients cannot be defined for volumes.

In case of a direct transmission, the ray direction will be modified according to the Snell-Descartes law.

This container will affect the following:

- **Radiation**
- **Solar flux**
- **Planet fluxes**

Enclosure

The enclosure container may be used to defined different enclosures onto a model. The default enclosure, 0, shall always correspond to the external or total part of the spacecraft.

Enclosures 1 to 99 may be used to define internal cavities.

> From Systema 4.9.1, it is possible to define a discrete list of enclosures for a shape. Example of a list of enclosures (for a shape belonging to 3 different ones for example): "0 96 97".
> NB: When importing model from previous version the enclosures are loaded as integer, it is not an issue unless you want to set a list. If so you would need to convert the "int" enclosures in the sysmdl, to do so a python script convertEnclosureInSysmdlFile.py in "public scripts" is available. Run it in a console like this: convertEnclosureInSysmdlFile.py -i <inputFile> -o <outputFile>

At processing level, it will then be possible to filter the radiation computation for a specific enclosure. External fluxes modules will only work on the external enclosure 0 since internal enclosures are not supposed to see space.

Defining enclosures will be useful in the case of a full model composed of internal and external parts where moving bodies are defined in the external part (typically moving solar arrays). Then, using different radiative couplings computations for each module will optimize the computation time since only the external enclosure would lead to dynamic computation (i.e. for each orbital position) while the internal enclosures would be considered as static (i.e. without time dependencies).

THERMICA Meshing

THERMICA Meshing

The meshing structure is manly composed of the geometric model breakdown which is the transformation of the geometry into calculation variables.

Meshing Parameters

Meshing parameters may be used to split a geometrical shape into several mesh. This geometrical sub-meshing is made through the parameters a, b, c to cut geometry on the principal directions.

An "Enable ratios" option allows to customize the cutting lines into each direction. For example, a mesh defined with a paramter a=4 defines default cutting lines at a=0.25;0.5;0.75 - cutting ratios may be then overloaded with customized values.

The "do not mesh" option will remove the geometrical element at computation level.

Condensation

This parameter allows the current element (and sub element of the meshing tree) to be condensed. A condensation error will always be raised if all condensed elements are not flaged with the "Authorize condensation".

Submodel Name

All nodes creating with the submodel name option filled in will be created into a submodel rather than in the main model.

Numbering Parameters

Refer to the SYSTEMA User Guide for a complete description of these settings.

THERMICA Specific items

THERMICA Specific items

A list of non-geometrical elements is available from the meshing structure. It allows defining elements that are usually added into the THERMISOL input files. These elements, called specific items, can be defined in the meshing tab of the graphical user interface. They are available by clicking on the icon  , by right clicking on the meshing tree or in the Modeler menu by selecting "create specific meshing item".
The specific items available in Thermica are described in this section.

WARNING: Do not use submodels when specific items are applied on nodes (instead of shapes).

Non-Geometrical Node

The user can create new nodal entities that will be added to the nodal description. Those nodes are initially not linked with the geometrical nodes and couplings must be also added to complete this network.

A default "Space Node" is always created into the non-geometrical nodes and may be also customized.

Non-Geometrical Node parameters - General tab

- **Submodel name**
- **Node number**

Non-Geometrical Node parameters - Thermica tab

- **Status:** Diffusive (D) or Boundary (B)
- **Capacitance:** Capacitance of the virtual node in W.s/K
- **Temperature:** Temperature of the virtual node in °C

- **Internal dissipation:** Internal dissipation of the virtual node in W

Output file

The non-geometrical node specific item is generated by the "Nodal Description" module. The output file is the nod.nwk file.

Boundary condition

The boundary condition specific item can be used to set the node's status to boundary and fix their temperature or give them an initial temperature.

Boundary condition parameters

- **Type:** Fixed temperature or Initial condition
- **Temperature:** Temperature of the node in °C

Referenced objects

- **Applied on:** geometrical nodes, shapes, volumes can be selected.

The boundary condition specific item cannot be applied on a non-geometrical node or Space node.

Output file

The boundary condition specific item is generated by the "Nodal Description" module. The output file is the nod.nwk file.

Capacitance overload

The capacitance overload specific item can be used to overload the value of a node capacitance normally computed with the shape's Bulk properties.

Capacitance overload parameters

- **Type:** mCp [W.s/K] or mCp/S [W.s/K/m2]
- **Node distribution mode [only for mCp type]:** duplicate or distribute
- **Value:** value of mCp or mCp/S

Note:

- the output in the nod.nwk file is always a capacitance in W.s/K.
- If the mCp/S [W.s/K/m2] type is selected, the mCp/S value specified by the user will be automatically multiplied by the geometrical area of the node.
- If the mCp [W.s/K] type is selected, the value specified by the user can either be duplicated or distributed among the selected nodes in proportion to their geometric area.

Referenced objects

- **Applied on:** geometrical nodes, shapes, volumes (for mCp [W.s/K] only) can be selected.

The capacitance overload specific item cannot be applied on a non-geometrical node or Space node. It cannot be applied on a volume either, if the selected type is mCp/S [W.s/K/m2].

Output file

The capacitance overload specific item is generated by the "Nodal Description" module. The output file is the nod.nwk file.

Internal dissipation

The internal dissipation specific item can be used to set a power or a flux on a node or on a group of nodes using the Thermisol QI nodal entity.

Internal dissipation parameters

- **Type:** Power [W] or Flux [W/m2]
- **Node distribution mode [only for Power type]:** duplicate or distribute
- **Node distribution parameter [only for distribute mode]:** geometric area or capacitance
- **Value:** value of the dissipation power or flux

Note:

- the output in the nod.nwk file is always a power.
- If the flux [W/m2] type is selected, the value of the flux specified by the user will be automatically multiplied by the geometrical area of the node.
- If the Power [W] type is selected, the power value specified by the user can either be duplicated or distributed among the selected nodes in proportion to their geometric area or capacitance.

Referenced objects

- **Applied on:** geometrical nodes, shapes, volumes (for Power [W] only) can be selected.

The internal dissipation specific item cannot be applied on a non-geometrical node or Space node. It cannot be applied on a volume either, if the selected type is Flux [W/m2].

Output file

The internal dissipation specific item is generated by the "Nodal Description" module. The output file is the nod.nwk file.

Thermostat

The thermostat specific item can be used to set a thermostat using the Thermisol THRMST or THSTAT routine.

Thermostat parameters

- **Routine:** THSTAT or THRMST
- **Low Temperature:** low temperature threshold
- **High Temperature:** high temperature threshold
- **Type:** Power [W] or Flux [W/m2]
- **Heater Dissipation:** Dissipation of the thermostat in W or W/m2 depending on the selected type

Note: if the flux [W/m2] type is selected, the power dissipated by the thermostat is equal to the flux multiplied by the geometric area of the selected dissipation node. For a non-geometrical node, the area is set to 1 m^2.

Referenced objects

- **Temperature Control:** geometrical nodes, shapes, non-geometrical nodes can be selected.
- **Applied on:** geometrical nodes, shapes, non-geometrical nodes can be selected.

The thermostat specific item cannot be applied on the space node. The volumes are not supported.

Note:

- The Sinda output format does not support the multi selection in referenced objects and application nodes.
- If the controlled and dissipation nodes are different (i.e if the selections in "Temperature Control" and "Applied on" sections are different), the Thermisol routine used is automatically THRMST.
- If multiple entities are selected in "Temperature Control" section, the Thermisol routine used is automatically THRMST and the controlled temperature considered for the thermostat is the following:

$$T_{CTR} = \frac{\sum_i T_i S_i}{\sum_i S_i}$$

where S is the geometrical area and i is a node belonging to the "Temperature Control" selection. The area of a non-geometrical node is set to 1m^2. This feature is not adapted for volumes.

- If multiple entities are selected in "Applied On" section, the Thermisol routine used is automatically THRMST and the power dissipated by the Thermostat is distributed on the nodes in proportion of their area:

$$QR_i = \frac{A_i}{A_{tot}} * Pow$$

Output file

The thermostat specific item is generated by the "Nodal Description" module. The output file is the nod.nwk file.

Limitations

- The thermostat specific item is not compatible with the use of the $VARIABLES1/$VARIABLES2 mode.
- The THSTAT routine is not compatible with Esatan. If the Esatan output format is selected, the routine is automatically THRMST.

Heater

The heater specific item can be used to set a general heater by a thermostat with the possibility to define a specific law (mean temperature or majority vote of controlled temperatures) or by a PID element.

Heater parameters

The parameters of the heater (or heating line) vary depending on the chosen control mode.

Heater type: Thermostat (or ON/OFF) with THRMST routine

The Thermisol routine used by this heating line control mode is THRMST. The parameters specified by the user are:

- **Heater law:** this corresponds to the activation law that can be "Majority vote" or "Mean temperature" (see description below)
- **Low temperature:** low temperature threshold [°C]
- **High temperature:** high temperature threshold [°C]
- **Heater dissipation:** dissipated power [W]

Mean temperature activation law: The controlled temperature considered for the heating line is the mean temperature of the selected nodes:

$$T_{CTR} = \frac{\sum_i T_i}{N_{nodes}}$$

- The status is set to 1 if $T_{CTR} \leq T_{Low}$
- The status is set to 0 if $T_{CTR} \geq T_{High}$
- The status is unchanged if $T_{Low} \leq T_{CTR} \leq T_{High}$

Finally, the power dissipated by each node selected in the "Applied on" section is equal to:

$$QR_i = Status * Pow * \frac{A_i}{A_{tot}}$$

where Pow is the dissipated power specified by the user.

Majority vote activation law:

- The status is set to 1 if the majority of nodes selected in the "Temperature Control" section have a temperature value lower than T_{Low}
- The status is set to 0 if the majority of nodes selected in the "Temperature Control" section have a temperature value greater than T_{High}
- Otherwise the status is unchanged

Finally, the power dissipated by each node selected in the "Applied on" section is equal to:

$$QR_i = Status * Pow * \frac{A_i}{A_{tot}}$$

where Pow is the dissipated power specified by the user.

Heater type: Thermostat (or ON/OFF) with THSTAT routines

The Thermisol routine used by this heating line control mode is THSTAT2 if there is only one Temperature Control node and THSTAT3 if there is 3 Temperature Control nodes. The Thermostat (THSTAT) type cannot be used with a number of Temperature Control nodes different from 1 or 3. The parameters specified by the user are:

- **Low temperature:** low temperature threshold [°C]
- **High temperature:** high temperature threshold [°C]
- **Heater dissipation:** dissipated power [W]

The user does not have the choice of the activation law. If 3 nodes are selected in the "Temperature Control" section, the THSTAT3 routine will be called instead of THSTAT2: this routine allows to find the node corresponding to the median temperature value and call the THSTAT2 routine with the median node as input (see the Thermisol User Manual for more details about the THSTAT3 routine).

If several application nodes are selected by the user the heater dissipation is distributed on the nodes in proportion of their area. To do this, the total power HTPOW dissipated by these nodes is calculated by the THSTAT2/THSTAT3 routine. Then the power dissipated by each node is calculated as follows:

$$QR_i = \frac{A_i}{A_{tot}} * HTPOW$$

Note: the area of a non-geometrical node is set to $1m^2$

Heater type: PID (only for transient computation)

A PID controller continuously calculates an error value as the difference between the target temperature and the controlled temperature and applies a correction based on proportional, integral and derivative terms (P, I and D respectively). The parameters specified by the user are:

- **Kp:** coefficient for the proportional terms [W/K]
- **Ki:** coefficient for the integral terms [W/K/s]
- **Kd:** coefficient for the derivative terms [W.s/K]
- **Target temperature:** target value [°C]
- **Heater dissipation:** dissipated power threshold [W]

The power to provide is calculated thanks to the sum of these 3 terms:

- The proportional term which is proportional to the current value of the error. $P = K_P\, e(t)$
- The integral term which accounts for past values of the error and integrates them over time. $I = K_I \int_0^t e(t')dt'$
- The derivative term which estimates the future trend of the error and is based on its current rate of change. $D = K_D \frac{de(t)}{dt}$

where $e(t) = T_{target} - T_{CTR}$

For the PID mode, the activation law used is the mean temperature. The controlled temperature considered for the heating line is the mean temperature of the selected nodes:

$$T_{CTR} = \frac{\sum_i T_i}{N_{nodes}}$$

The obtained power is then bounded between 0 and the dissipated power (Pow) specified by the user:

$$QR_{tot} = Min(Max(0, P + I + D), Pow)$$

Finally, the power dissipated by each node selected in the "Applied on" section is equal to:

$$QR_i = QR_{tot} * \frac{A_i}{A_{tot}}$$

Referenced objects

- **Temperature Control:** geometrical nodes, shapes, non-geometrical nodes can be selected.
- **Applied on:** geometrical nodes, shapes, non-geometrical nodes can be selected.

The heater specific item cannot be applied on the space node and volumes.

Output file

The heater specific item is generated by the "Nodal Description" module. The output file is the nod.nwk file.

Limitations

- The heater specific item is not supported by the Sinda and Esatan output format (nod.nwk file).
- The heater specific item is not compatible with the use of the $VARIABLES1/$VARIABLES2 mode.

Linear coupling

This specific item can be used to create linear couplings between nodes or groups of nodes.

Linear coupling parameters

- **GL:** total GL value [W/K]

Note: the total GL value specified by the user is distributed among the nodes selected in the "Pin 1" or "Pin 2" section according to their geometrical area:

$$GL(i_{pin1}, j_{pin2}) = \frac{A_{i_{pin1}}}{A_{tot_{pin1}}} \times \frac{A_{j_{pin2}}}{A_{tot_{pin2}}} \times GL$$

where $A_{tot_{pin1}}$ and $A_{tot_{pin2}}$ are the total areas of nodes selected in the "Pin 1" and "Pin 2" sections respectively. The area of a non-geometrical node or Space node is considered equal to 1m^2.

Referenced objects

- **Pin 1:** geometrical nodes, shapes, non-geometrical nodes, Space node can be selected.
- **Pin 2:** geometrical nodes, shapes, non-geometrical nodes, Space node can be selected.

The linear coupling specific item cannot be applied on a volume.

Output file

The linear coupling specific item is generated by the "Nodal Description" module. The output file is the nod.nwk file.

Limitations

- The temperature dependent values of GL (parametric mode) are not supported by the Sinda output format (nod.nwk file).

Radiative coupling

This specific item can be used to create radiative couplings between nodes or groups of nodes.

Radiative coupling parameters

- **GR:** total GR value [m2]

Note: the total GR value specified by the user is distributed among the nodes selected in the "Pin 1" or "Pin 2" section according to their geometrical area:

$$GR(i_{pin1}, j_{pin2}) = \frac{A_{i_{pin1}}}{A_{tot_{pin1}}} \times \frac{A_{j_{pin2}}}{A_{tot_{pin2}}} \times GR$$

where $A_{tot_{pin1}}$ and $A_{tot_{pin2}}$ are the total areas of nodes selected in the "Pin 1" and "Pin 2" sections respectively. The area of a non-geometrical node or Space node is considered equal to 1m^2.

Referenced objects

- **Pin 1:** geometrical nodes, shapes, non-geometrical nodes, Space node can be selected.
- **Pin 2:** geometrical nodes, shapes, non-geometrical nodes, Space node can be selected.

The radiative coupling specific item cannot be applied on a volume.

Output file

The radiative coupling specific item is generated by the "Nodal Description" module. The output file is the nod.nwk file.

Limitations

- The temperature dependent values of GR (parametric mode) are not supported by the Sinda output format (nod.nwk file).

Surface contact

The surface contact specific item can be used to define a linear coupling and/or a radiative coupling between two surfaces in contact. This specific item should be preferred to the linear coupling or radiative coupling specific items when the meshes of the two surfaces are not coincident.

Surface contact parameters

- **Type:** Contact Quality ([W/K/m2] or [-]) or Contact Value ([W/K/] or [m2]) (see description below)
- **Conductive contact:** GL value in [W/K/m2] or [W/K] depending on the type
- **Additional radiosity:** GR value in [-] or [m2] depending on the type
- **Surface contact threshold:** Surface Contact Threshold value in [m]
- **Ray density:** density of rays fired from each mesh by ray-tracing to compute the contact area [ray/m2]
- **Minimum number of rays per mesh:** Minimum number of rays fired, whatever the chosen density [-]
- **Maximum number of rays per mesh:** Maximum number of rays fired, whatever the chosen density [-]

Note: the surface contact threshold considered in the algorithm is equal to 1.01 times the surface contact threshold specified by the user.

Surface contact type: Contact Quality [W/K/m2] or [-]:

Let's consider a contact between to surfaces 1 and 2. The user can apply a Contact Quality for conductive contact (CQL) and a Contact Quality for additional radiosity (CQR) to this contact. The module "Nodal Description" will compute the linear and radiative couplings between a node i, belonging to surface 1, and a node j, belonging to surface 2, as follows:

$$\mathrm{GL(i,j)} = \mathrm{contact\_area(i,j)} \times CQL$$

$$\mathrm{GR(i,j)} = \mathrm{contact\_area(i,j)} \times CQR$$

where contact_area(i,j) is the area of the contact between nodes i and j.

Note that:

$$\sum_{i \in 1} \sum_{j \in 2} GL(i,j) = \mathrm{contact\_area(1,2)} \times CQL$$

$$\sum_{i \in 1} \sum_{j \in 2} GR(i,j) = \mathrm{contact\_area(1,2)} \times CQR$$

where contact_area(1,2) is the area of the contact between surfaces 1 and 2.

Surface contact type: Contact Value [W/K] or [m2]:

Let's consider a contact between to surfaces 1 and 2. The user can apply a Contact Value for conductive contact (CVL) and a Contact Value for additional radiosity (CVR) to this contact. The module "Nodal Description" will compute the linear and radiative couplings between a node i, belonging to surface 1, and a node j, belonging to surface 2, as follows:

$$\mathrm{GL(i,j)} = \frac{\mathrm{contact\_area(i,j)}}{\mathrm{contact\_area(1,2)}} \times CQL$$

$$\mathrm{GR(i,j)} = \frac{\mathrm{contact\_area(i,j)}}{\mathrm{contact\_area(1,2)}} \times CQR$$

where contact_area(i,j) is the area of the contact between nodes i and j and contact_area(1,2) is the area of the contact between surfaces 1 and 2.

Note that:

$$\sum_{i \in 1} \sum_{j \in 2} GL(i,j) = CVL$$

$$\sum_{i \in 1} \sum_{j \in 2} GR(i,j) = CVR$$

Regardless of the type of surface contact used (contact quality or contact value), the contact area between an element i and an element j is calculated thanks to a ray-tracing method, using an iso distribution law. The number of rays fired from each mesh is computed from the density and the mesh area, but is nevertheless limited between a minimum and a maximum value. This ensures that the calculation is accurate and does not take too long. Each ray is fired from the element i in the direction perpendicular to the surface. Each ray represents a small proportion of the mesh area. If the ray impacts the element j, with a distance traveled smaller than the surface contact threshold, then the little area it represents belongs to the contact.

Referenced objects

- Surface 1: shapes, volumes can be selected.
- Surface 2: shapes, volumes can be selected.

The surface contact specific item cannot be applied on geometrical nodes, non-geometrical nodes and space node.

Output file

The surface contact specific item is generated by the "Nodal Description" module. The output file is the nod.nwk file.

Limitations

- It is not recommended to have a contact area very small compared to the mesh area (less than 5%). This can lead to errors in the results (see the warning above).
- The temperature dependent values of conductive contact or additional radiosity (parametric mode) are not supported by the Sinda output format (nod.nwk file)
- Surface contact computation between two moving bodies is not supported because the coupling values are constant. In this case, the couplings due to surface contact specific items are calculated in the configuration of the first computation point.

Edge contact resistance

The edge contact resistance specific item can be used to modify the linear coupling existing between two group of shapes with edge contacts. This will add a linear coupling GL_{add} in series with the already existing linear coupling GL_{nat} (calculated by the conduction module). The resulting or equivalent linear coupling is equal to:

$$GL_{eq}(i,j) = \frac{1}{\frac{1}{GL_{nat}(i,j)} + \frac{1}{GL_{add}(i,j)}}$$

Adding an edge contact resistance will decrease the overall value of the conductive coupling between the two nodes. The impact is inversely proportional to the value of the edge contact resistance: a very large value will decrease the contact quality very little, while a zero value will result in a zero equivalent conductive coupling value.

Note: using the simplified RCN method for the conduction computation, the GL written in the gl.nwk file correspond directly to the equivalent coupling GL_{eq}. Using the RCN method, the two linear couplings GL_{nat} and GL_{add} are written separately in the gl.nwk file.

Edge contact resistance parameters

- **Unit:** [W/K/m2] (x length x minThickness) or [W/K/m] (x length) or [W/K] (x length ratio) (see description below)
- **Value:** Value of the additional linear coupling in [W/K/m2], [W/K/m] or [W/K] depending on the unit

Edge contact resistance unit: [W/K/m2] (x length x minThickness):

If the user chooses to specify a value in W/K/m2, this value will be multiplied by the length of contact and the thickness of contact between nodes i and j to obtain the additional linear coupling:

$$GL_{add}(i,j) = UserValue \times L_{contact\ i,j} \times th_{contact\ i,j}$$

Edge contact resistance: [W/K/m] (x length):

If the user chooses to specify a value in W/K/m, this value will be multiplied by the length of contact between nodes i and j to obtain the additional linear coupling:

$$GL_{add}(i,j) = UserValue \times L_{contact\ i,j}$$

Edge contact resistance: [W/K] (x length ratio):

If the user chooses to specify a value in W/K, this value will be multiplied by a length ratio (equal to the contact length between node i and j over the contact length between surface 1 and surface 2) to obtain the additional linear coupling:

$$GL_{add}(i,j) = UserValue \times \frac{L_{contact\ i,j}}{\sum_{i\in 1}\sum_{j\in 2} L_{contact\ i,j}}$$

Note: this formula may not be exact if the contact thickness between the two groups of shapes "surface 1" and "surface 2" is not uniform. In this case, the ratio should be calculated with the contact areas and not only with the contact lengths.

Referenced objects

- **Surface 1:** shapes can be selected
- **Surface 2:** shapes can be selected

The edge contact resistance specific item cannot be applied on geometrical nodes, non-geometrical nodes, volumes and space node.

Output file

The edge contact resistance specific item is generated by the "Conduction" module. The output file is the gl.nwk file.

Limitations

- Do not use a variable to define the thickness of a shape when using the edge contact resistance specific item.
- The temperature dependent values of the edge contact resistance (parametric mode) are not supported by the Sinda output format (gl.nwk file).

Cavity by sphere

The cavity by sphere specific item can be used to define a cavity using a spherical geometry (which shall be small and included into the cavity to define). It allows to parametrize the convective method to apply on the cavity and to set fluid characteristics into the cavity.

Cavity by rectangle

This specific item is the same as the "cavity by Sphere" specific item except that the geometry is a rectangle, allowing to define a thin zone as cavity.

Cavity closure

The cavity closure specific item can be used to fill-in holes in the geometry so cavities may be define correctly.

Camera

The camera specific item can be used to set the localisation of an IR camera in the model. But as IR Camera module is no longer available in THERMICA, it is only avalailable for backward compatibility.

Sensor

The sensor specific item can be used to set the localisation of a sensor in the model, its numbering and a few properties like the sensor computation method (average or backward RCN), the surface contact threshold and a contact value.

Point Source

The Point Source specific item is used to model UV/IR lamp.

The shape of the lamp will be determined by the Systema shape to which it is applied. Fluxes are always emitted from the **positive side** of the lamp (the lamp is active on the positive side only regardless of its thermo-optical properties).

Point Source parameters

- **Energy source type:** UV, IR, both.
- **UV Flux:** Value of the flux emitted by the lamp in the UV spectrum [W/m2].
- **IR Flux:** Value of the flux emitted by the lamp in the IR spectrum [W/m2].
- **Ray emission direction:** Lambert's law or normal, meaning that the flux can be emitted either in all directions (following a Lambert's law) or in the direction normal to the surface of the lamp.

Referenced objects

- **Applied on:** geometrical nodes, shapes, volumes can be selected.

Output file

The Point Source specific item is controlled by the Emission Source module. The fluxes emitted by the lamps are available in the emi.nwk and emi.h5 files. For further details, please refer to the "Emission Source module" chapter.

Solar Lamp

Important: *To take full advantage of advanced features and improved performance, you should switch to the Point Source specific item with the Emission Source module. The Point Source item provides the same functionalities as the current item, but with additional features and fewer limitations. The Solar Lamp items remains available only for compatibility purposes.*

The solar lamp specific item can be used to model a surface source emission UV or IR.

The Radiative Exchange Factors are computed following the same logic as in the radiation module.

	*For a definition of the REF calculation, refer to Chapter "Radiation Module".*

Rays are emitted from the **positive side** of the solar lamp and following the normal of the mesh.

Solar lamp parameters

- **Energy source type:** UV or IR.
- **Flux:** Value of the solar lamp flux in [W/m2].
- **Ray number:** Number of rays used by the solar lamp. For computation performances, the total number of rays that shall be used for a node is re-distributed on its mesh so to keep a uniform density of rays.

- **Advanced settings**
    - **Random seed:** Is used to modify the generation of pseudo random numbers used by the ray-tracing. This option may be modified to check the results sensitivity to the random process.

- **Ray threshold:** This option is used to filter the ray bounces according to the remaining energy of a ray. By default, this filtering is done whenever a ray has lost at least 99% of its energy (1% of remaining energy). In such cases, the remaining energy is considered as absorbed by the last impacted node.
- **Maximum ray bounces:** This option is used to stop a ray propagation in the case of too many bounces. It allows in fact to detect a model failure, such as an active closed cavity with an epsilon or alpha value of 0 (theoritically leading to infinite ray bounces). Whenever rays are being filtered due to maximum bounces reached, the total filtered energy from a node is given in the log file for both IR and Visible contributions so to check if energy levels are significant or not.

Note:

- The output in the nod.nwk file is a QS followed by the node numbers.
- The solar lamp is active on the positive side only regardless of its thermo-optical properties.

Referenced objects

- **Applied on:** geometrical nodes, shapes, volumes can be selected.

Output file

The solar lamp specific item is generated by the "Nodal Description" module. The output file is the nod.nwk file.

Radiation Module

The radiation module computes the thermal radiative couplings between the geometrical nodes of a model.

It also exports the radiative exchange factors to a discretize surrounding box representing space directions. This information will then be required for the Planet fluxes module or may be used to speed-up the computation of Solar flux or Aero flux.

The information stored in this *box* output concerned direct view factors to space but also IR and Visible exchange factors. Indeed, when activated, the radiation module will not only work into the IR spectrum but also into the Visible range so to be able to export the corresponding data into the *box* file.

This documentation explains the theoritical background of the radiative couplings computation and the principles of the monte-carlo ray-tracing technic used.

Software parameters and inputs / outputs are also explained in a second part.

Radiation Overview

Radiation Overview

Introduction

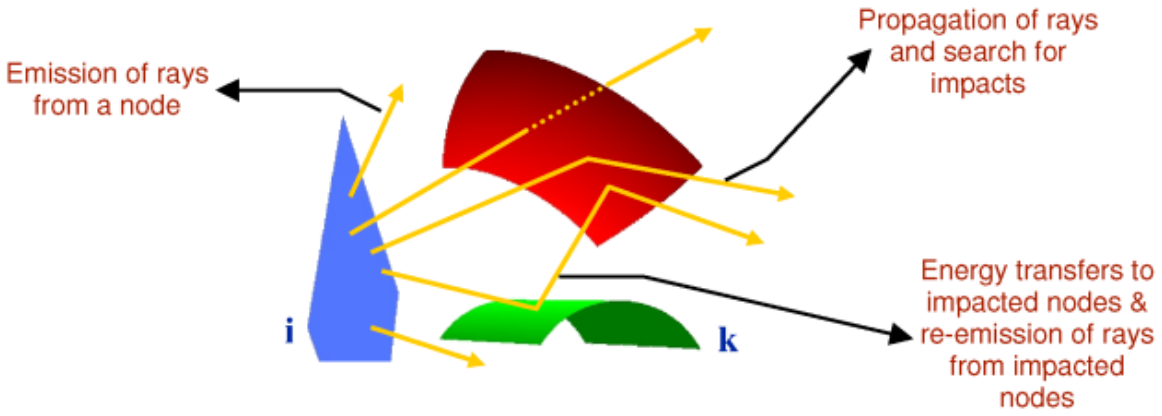
The radiation module describes the radiative exchanges between nodes and is impacted by the Monte-Carlo Ray Tracing method.

Monte-Carlo Ray tracing method

This method contains the following modeling tasks:

- ☐ **Emission of rays from a node**
  - Emission points are sampled randomly, according to an iso-distribution law, with an analytic definition of shapes.
  - Emission directions are sampled randomly, according to the Lambert's law.
- ☐ **Propagation of rays and search for impacts**
  - The main problem is to determine which surface is stuck by the ray.
  - An optimized search is needed as it is impossible to test all the surfaces.
  - The Ray Tracing considers the analytic definition of shapes.
- ☐ **Energy transfers to impacted nodes**
  - The ray has started from the initial emitting node with an energy E_0.
  - During its travel, the ray transfers energy to each impacted node.
  - The ray is re-emitted from the impacted surface if the energy is > extinction threshold.
- ☐ **Re-emission of rays from impacted nodes**
  - Re-emission direction determined by the Monte-Carlo method:
    - ***Random choice between:***
      - Diffuse reflection
      - Specular reflection
      - Diffuse transmission
      - Direct transmission
    - ***This choice is consistent with the thermo-optical properties of the surface and with the analytic definition of shapes surfaces have different optical properties:***

- In IR and UV
- On each side



With the Monte Carlo Ray Tracing method, all the REFs are obtained

- **REF_{ij} has been evaluated by firing rays from node i**
- **REF_{ji} has been evaluated by firing rays from node j**
- **With an infinite number of rays, the reciprocity law is respected**
- **Statistical convergence**
- **In a real computation, this is not the case:**
- $REF_{ji}\ \varepsilon_j\ S_j \neq REF_{ij}\ \varepsilon_i\ S_i$

Therefore, the software forces the reciprocity law to be verified.

- **REF_{ij} and REF_{ji} are modified**
- **The software takes into account the number of rays fired from (i) and from (j), and their respective surfaces**
- **Then, all the GRs are stored and transmitted to the temperature solver.**

Background of REF calculations

Definition of the REF Calculation

Geometric view-factors

Definition
The view factor $F_{i,j}$ between nodes i and j is defined as the fraction of the radiant energy emitted by i, which is directly intercepted by j.
Usually, it is assumed that the emission process follows the cosine distribution law; the analytical expression for the view-factor is (see Figure 6-1):

$$F_{i,j} = \frac{1}{A_i} \int_{A_i} \int_{A_j} \frac{\cos(\theta_i)\cos(\theta_j)}{\pi r_{i,j}^2} X_{i,j} dA_i\, dA_j$$

where

dA_i, dA_j	:	infinitesimal areas of nodes i and j

θ_i, θ_j	:	line-of-sight inclination angles
$r_{i,j}$	:	distance between dA_i and dA_j
$X_{i,j}$	:	line-of-sight occultation indicator (0 or 1)

The following properties of the view-factors are to be introduced:

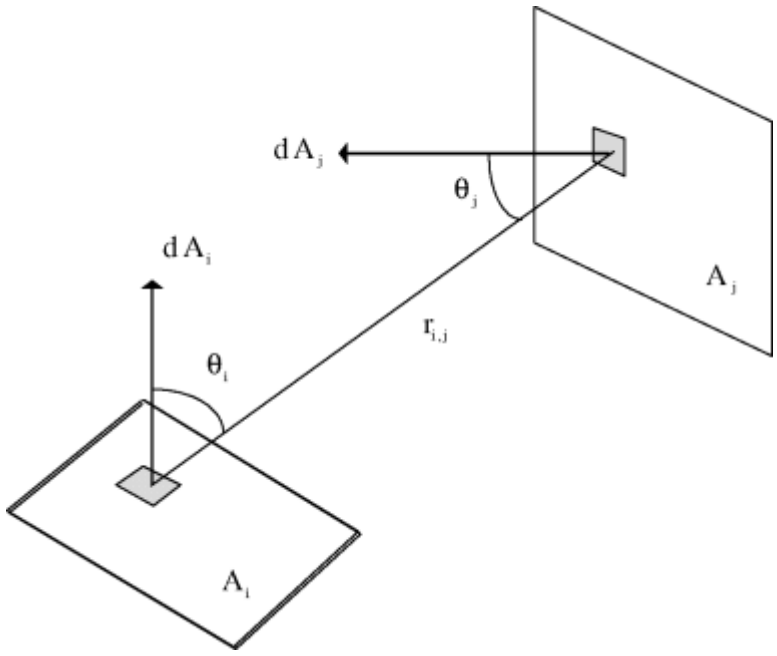
- **The view-factor expression involves only the geometrical definition of the model**
- **The expression shows an i-j symmetry, resulting in the well known reciprocity law:**

$$A_i F_{i,j} = A_j F_{j,i}$$

- **The energy conservation principle implies that the view-factor line sum is strictly equal to 1:**

$$\sum_j F_{i,j} = 1$$

The method used for the geometric view-factors computation is the Monte-Carlo Ray-Tracing technique.



Radiative Exchange Factors (REF)

Definition
Except in the case where all the surfaces are considered as perfect black bodies, the view-factor matrix does not represent the radiative coupling network required for the energy balance. For this purpose, the so-called radiative exchange factors (also called Gebhart factors) are required.
The radiative exchange factor $B_{i,j}$ between nodes i and j is defined as the fraction of the energy emitted by i, which is finally absorbed by j.
The energy absorbed by the node j comes either directly or via reflections from other surfaces of the model.
The net thermal radiation balance between nodes i and j is:

$$Q_{i,j} = A_i B_{i,j} \varepsilon_i \sigma \left(T_i^4 - T_j^4 \right)$$

The radiative exchange factor matrix has the following properties:

- **The values of the radiative exchange factors depend on both the geometrical definition of the model and the thermo-optical properties of the surfaces,**
- **The reciprocity law is written as follows:**

$$Q_{i,j} = -Q_{j,i}$$

$$A_i B_{i,j} \varepsilon_i = A_j B_{j,i} \varepsilon_j$$

The product $A_i B_{i,j} \varepsilon_i$ represents the radiative coupling between i and j.

- **The energy conservation is written as follows:**

$$\sum_j B_{i,j} = 1$$

As for the geometric view factor computation, the computation of the REF is based on the Monte-Carlo Ray-Tracing technique.

Condensation onto the thermal node network

View-factors
Let I and J be two thermal nodes built up with one or several radiative nodes:

$$I = \bigcup_{i \in I} \{i\} and J = \bigcup_{j \in J} \{j\}$$

The view factors between thermal nodes I and J are:

$$F_{i,j} = \frac{1}{A_i} \int_{A_i} \int_{A_j} \frac{\cos(\theta_i)\cos(\theta_j)}{\pi r_{i,j}} X_{i,j} dA_i dA_j$$

where:

dA_I, dA_J	:	infinitesimal areas of nodes I and J
θ_I, θ_J	:	line-of-sight inclination angles
$r_{I,J}$	:	distance between dA_i and dA_j
$X_{I,J}$	:	line-of-sight occultation indicator (is 0 or 1)
A_I, A_J	:	area of node I: $\sum_{i \in I} A_i$, area of node J: $\sum_{j \in J} A_j$

This can be written as follows:

$$A_I F_{I,J} = \int_{\bigcup_{i\in I} A_i} \int_{\bigcup_{j\in J} A_j} \frac{\cos(\theta_i)\cos(\theta_j)}{\pi r_{i,j}} X_{i,j}\, dA_i\, dA_j$$

$$= \sum_{i\in I}\sum_{j\in J} \int_{A_I} \int_{A_J} \frac{\cos(\theta_i)\cos(\theta_j)}{\pi r_{i,j}} X_{i,j}\, dA_i\, dA_j$$

$$= \sum_{i\in I}\sum_{j\in J} A_i F_{i,j}$$

hence:

$$F_{I,J} = \frac{\sum_{i\in I} A_i \sum_{j\in J} F_{i,j}}{\sum_{i\in I} A_i}$$

The view factors between thermal nodes have the following properties:

- **They involve only the geometrical model**
- **The reciprocity law is written as follows:**

$$A_I F_{I,J} = A_J F_{J,I}$$

- **The energy conservation is written as follows:**

$$\sum_J F_{IJ} = 1$$

Radiative Exchange Factors
The net thermal radiation balance between radiative nodes i and j is as follows:

$$Q_{i,j} = A_i B_{i,j} \varepsilon_i \sigma\left(T_i^4 - T_j^4\right)$$

The product $A_i B_{i,j} \varepsilon_i$ represents the radiative coupling from i and j, $R_{i,j}$.
The net interchange of thermal radiation between thermal nodes I and J is:

$$Q_{IJ} = A_I B_{IJ} \varepsilon_I \sigma\left(T_I^4 - T_J^4\right) = \sum_{i\in I}\sum_{j\in J} Q_{i,j}$$

where:

$B_{I,J}$	:	radiative exchange factor between thermal nodes I and J
ε_I	:	mean infrared emissivity on thermal node I: $\varepsilon_I = \frac{\sum_{i\in I} A_i \varepsilon_i}{\sum_{i\in I} A_i}$ (remember that a thermal node is isothermal, that is $\forall i \in I, \quad T_i = T_I$)

The equation can be written as follows:

$$A_I B_{I,J} \varepsilon_I = \sum_{i \in I} \sum_{j \in J} A_i B_{i,j} \varepsilon_i$$

hence

$$B_{I,J} = \frac{\sum_{i \in I} A_i \varepsilon_i \sum_{j \in J} B_{i,j}}{\sum_{i \in I} A_i \varepsilon_i}$$

For the UV wavelength, the radiative exchange factors are computed by analogy with the previous formula (note that they are not used):

$$B_{I,J}^{(uv)} = \frac{\sum_{i \in I} A_i \alpha_i \sum_{j \in J} B_{i,j}^{(uv)}}{\sum_{i \in I} A_i \alpha_i}$$

The radiative exchange factors have the following properties:

- **Their values depend on both the geometrical definition of the model and the thermo-optical properties of the surface.**
- **The reciprocity law is written as follows:**

$$A_I B_{I,J}^{(ir)} \varepsilon_I = A_J B_{J,I}^{(ir)} \varepsilon_J$$
$$A_I B_{I,J}^{(uv)} \alpha_I = A_J B_{J,I}^{(uv)} \alpha_J$$

- **The energy conservation is written as follows:**

$$\sum_J B_{I,J} = 1$$

Radiative exchanges between nodes

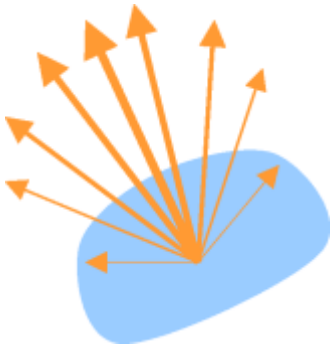
Emission of radiative energy

The total radiative energy emitted by a node is:

$$E = \varepsilon S \sigma T^4$$

where

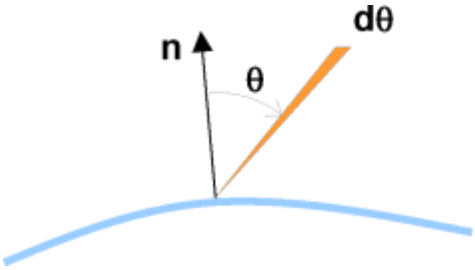
ε	=	node emissivity
S	=	node surface
T	=	node temperature
σ	=	Stefan's constant

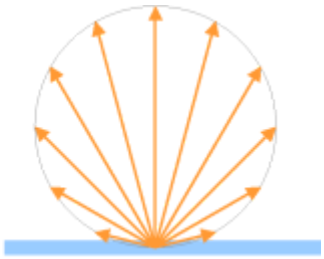


Radiative energy emission

The energy is propagated through electromagnetic waves considering a diffuse emission (Lambert's law):

$$dE(\theta) = \cos(\theta)d\theta$$





Lambert's law

Distribution of the energy emitted by a node

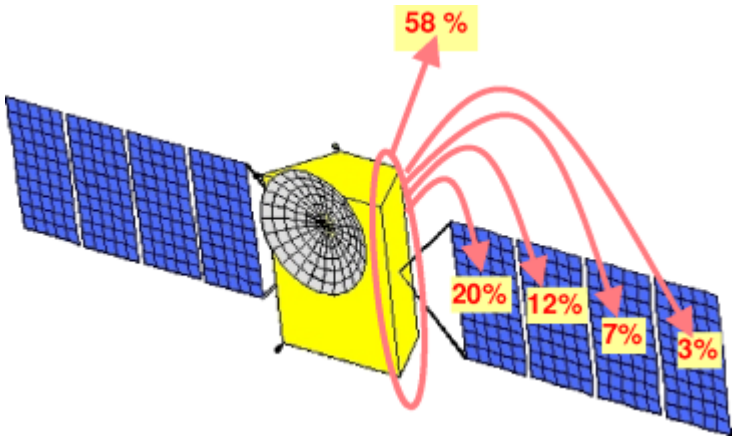
The energy emitted by a node is transferred to the other nodes in the geometrical model, and also to the space:

$$E_i = \varepsilon_i S_i \sigma T_i^4 = (\Sigma_j REF_{ij})\ \varepsilon_i S_i \sigma T_i^4$$

HOW the energy is transferred

WHICH energy is transferred

	REF = Radiative Exchange Factor



Proportion of energy transferred to j ($\Sigma_j REF_{ij} = 1$)

The main task of **THERMICA** in the REF calculation is to study, for each node, this emitted energy distribution.

Reciprocity law: radiative coupling symmetry (GR)

The radiation balance for node i is:

$$\underbrace{\Sigma_j REF_{ji}\varepsilon_j S_j\sigma T_j^4}_{\text{Energy collected by (i) coming from the other nodes}} - \underbrace{\Sigma_j REF_{ij}\,\varepsilon_i S_i\sigma T_i^4}_{\text{Energy emitted by (i) to the other nodes}}$$

$$= \sigma\Sigma_j GR_{ij}(T_j^4 - T_i^4)$$

$$\text{where: } \underbrace{GR_{ij} = REF_{ji}\varepsilon_j S_i = REF_{ij}\,\varepsilon_i S_i}_{\text{Reciprocity law (a theoretically proven physical property)}}$$

After the evaluation and symmetrization of REFs are completed, all the GRs are transmitted to the temperature solver:

$$\sum_j GL_{i,j}(T_j - T_i) + \sigma\sum_j GR_{i,j}\left(T_j^4 - T_i^4\right) + P_{SUN_i} + P_{ALB_i} + P_{PLA_i} + P_{INT_i} = MCp_i\frac{dT_i}{dt}$$

Modeling by Monte-Carlo Ray Tracing

Here, the problem consists in evaluating the Radiative Exchange Factors (REF):

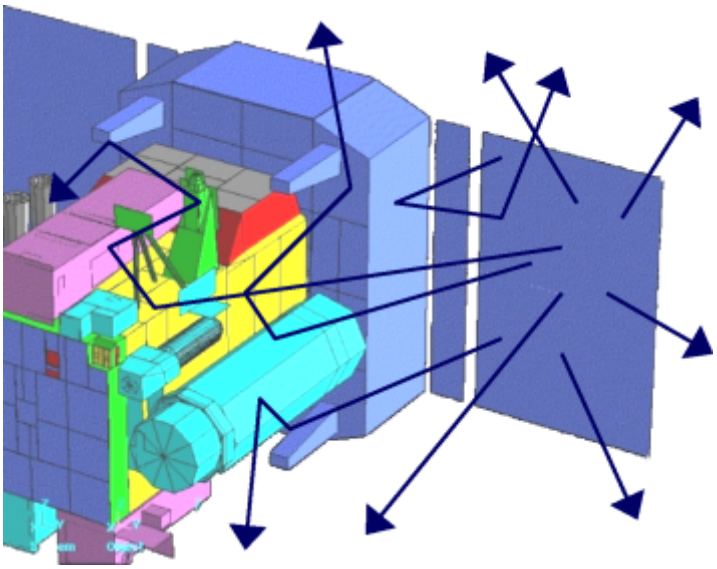
REF_{ij} = energy ratio transferred from node i to node j considering a diffuse emission from node I
all the GR_{ij} will be output to the temperature solver.

It is not so easy to do compute because:

- **The geometry is always a complex data**
- **There are shading effects**
- **There are specularity reflections or transitions.**

Solutions retained

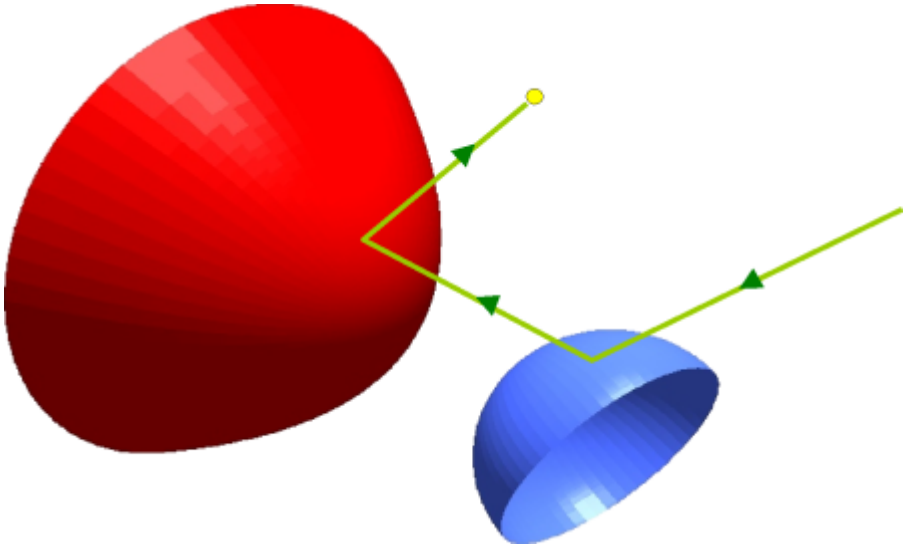
The best method for an accurate modeling is the Monte Carlo Ray-Tracing method. It can indeed manage all the above-mentioned behaviors.



Monte-Carlo Ray Tracing

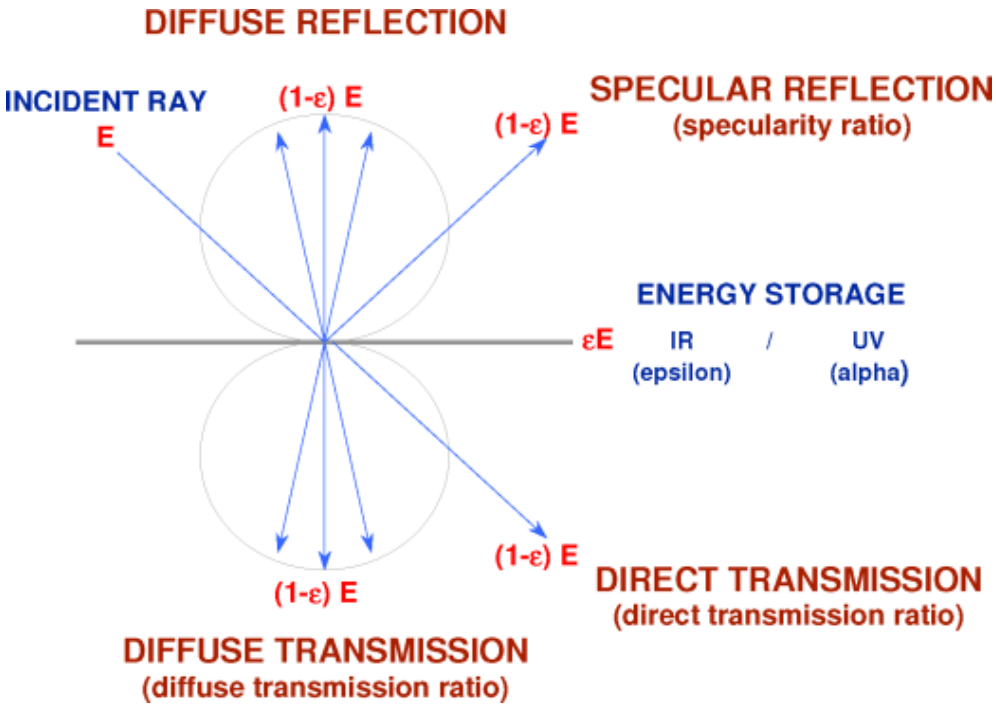
General principle of the Monte-Carlo Ray Tracing method

- Simulation of the light behavior by considering the individual history of a small number of photons
- The photons and various phenomena affecting their history are sampled randomly from the appropriate distribution functions:
- Emission
- Interaction with surfaces.
- The macroscopic behavior is reached statistically after firing a sufficient number of photons



General principle of the Monte-Carlo Ray Tracing method

Energy transfers to impacted nodes



Re-emission of rays from impacted nodes

Radiation Parameters

Radiation Parameters

Input

- **.SYSSET file:** Mission synthesis for the THERMICA application.

Outputs

The following two files are used to process the results in the Modeler or Mission tabs of SYSTEMA:

- **.RAD.LOG file:** This file contains information about the process and has useful summary tables of radiative exchanged factors unbalances and inactive impingements.
- **.RAD.H5 file:** Contains radiative results in HDF5 format with GR symmetric radiative couplings.
- **.GR.NWK:** Radiative couplings expressed in the selected network format.
- **.BOX.H5:** (optionnal) Contains exchange factors with the surrounding box for further planetary fluxes computation or for solar fluxes optimization. If activated, the radiadition module will automatically propagates radiative fluxes into the Visible spectrum in addition to the IR one.
- **.RRT.H5** (optionnal) Contains all computed rays for display in SYSTEMA.
- **.GV.NWK** (optionnal) View couplings expressed in the selected network format.
- **.GB.TXT** (optionnal) Contains all view factors and the GEBHART factors.

Parameters

The parameters are splitted into 2 general categories: Radiation properties (local to the current module) and Common properties (global to the complete diagram).

Common properties

From the **Common** properties, the Radiation module will be influenced by the wavelength dependency options. Depending on the number of spectral bands specified, the computation will be made with a multi-band mode which allows propagating radiative fluxes simulataneously into all wavelength band.

The results will be exported as wavelength dependent or classical radiative couplings depending on the wavelength sensibilities of each coupling.

Note that this option is only compatible with the THERMISOL output format: **other thermal solver formats are not supported for wavelength dependent export.**

Radiation properties

Ray tracing

- **Random seed:** Is used to modify the generation of pseudo random numbers used by the ray-tracing. This option may be modified to check the results sensitivity to the random process.
- **Re-init random seed:** Re-intialize for each node the random number series. This option may be used in rare cases where the computation have to be not sensitive of the nodal ordering.
- **Number of space division:** Corresponds to the discretization of the space environment into the box output. By default the value is 20 (meaning 20x20x6=2400 space elements). It is usually not required to modify this value but it can be increased up to 100 to improve the accuracy of the external fluxes calculation if needed.
- **Ray threshold:** This option is used to filter the ray bounces according to the remaining energy of a ray. By default, this filtering is done whenever a ray has lost at least 99% of its energy (1% of remaining energy). In such cases, the remaining energy is considered as absorbed by the last impacted node.
- **Maximum ray bounces:** This option is used to stop a ray propagation in the case of too many bounces. It allows in fact to detect a model failure, such as an active closed cavity with an epsilon or alpha value of 0 (theoritically leading to infinite ray bounces). Whenever rays are being filtered due to maximum bounces reached, the total filtered energy from a node is given in the log file for both IR and Visible contributions so to check if energy levels are significant or not.

Critical / Normal / Low

- **Ray building method:** Select "Per node" / "Per mesh" / "Density". The default "Per node" value indicates that each node will use the given number of rays. In the case of condensed nodes (and/or with both active sides), this number of rays is distributed to the node's mesh according to their respective area (i.e. providing a uniform density of rays into the node). The "Per mesh" value may be used to specify the number of rays per radiative mesh (a both coating active mesh has in fact two radiative mesh: one per side). A condensed node will then use as many more rays as there are condensed radiative mesh into the node. For computation performances, the total number of rays that shall be used for a node is re-distributed on the mesh so to keep a uniform density of rays. This option shall be avoided when using CAD geometries because of those are build from a geometrical tesselation (and so with lots of mesh).

REF Computation

- **Enclosure:** Set the slected enclosure for the computation (-1 for all, 0 for enclosure 0, ...).
- **Space couplings:** Filter space couplings in the nwk output.
- **REF reciprocal filter:** Filter any GR between two nodes *i* and *j* in the nwk output if both radiative exchange factors from *i* to *j* and from *j* to *i* and less than this value (0.01% by default).
- **REF maximum filter:** This option prevents from letting previous filtering leading to a total of filter couplings for a given node exceeding this value.
- **Time variation filter:** Will output a time dependent coupling if it's variation through time is above this limit (by default 5%).

- **Wavelength variation filter:** Will output a wavelength dependent coupling if it's variation on wavelength bands is above this limit (by default 5%).

Outputs

- **Network format:** Set the format of the nwk output for THERMISOL, ESATAN or MSC/SINDA.
- **Ray display length:** Set the length of ray display to space. By default 2x the size of the model.

Listing

- **Gebhart Sum levels**
- **Impingement summary:** To output an impingement summary in the log file.
- **Low space coupling summary:** To output a list of nodes having small non-zero couplings with space.

Solar Flux Module

The solar flux module computes the absorbed solar flux onto the geometrical nodes of a model.

This documentation explains the theoritical background of the computation and the principles of the monte-carlo ray-tracing technic used.

Software parameters and inputs / outputs are also explained in a second part.

Solar Flux Overview

Solar Flux Overview

Introduction

Flux computation technique

- For flux computation, the definition of an orbit and pointing is required:
  - **The computation is performed at each orbital position.**
  - **At each position, the position and orientation of the spacecraft is automatically computed.**
  - **The planet and sun positions are automatically computed.**

The ray-tracing method of the REF calculation is used for the solar flux computation as for the internodal factors, except for the emission technique.
The ray-tracing optimization technique is the same as that of the REF.

	*For a definition of the REF calculation, refer to Chapter "Radiation Module".*

Solar flux computation

Principle

The direct solar flux received by the node i, Φ_i, is:

$$\Phi_i = \int_{Ai} Csun \chi S dA_i$$

where:

S	=	sun-direction vector
χ	=	local sun visibility indicator
	=	1 if dA_i is directly lit by the sun
	=	0 otherwise

Φ_i can be written as follows:

$$\Phi_i = \mathrm{Csun}\,\mathrm{A}_i \int_{A_i} \chi \mathrm{S}.\frac{dA_i}{A_i} = \mathrm{Csun}\,\mathrm{A}_i \int_{A_i} \chi \cos\theta \mathrm{s}.\frac{dA_i}{A_i}$$

θs being the angle between the local normal and the sun direction.

An estimate of Φ_i can be obtained using the Monte-Carlo method. A number N of sampled points in A_i is picked from the probability density function dA_i/A_i as for the Ref. Φ_i is computed by the arithmetical average:

$$\Phi_i = Csun A_i \sum_{k=1}^{N} \chi_k \cos\theta s_k = \sum_{k=1}^{N} \chi_k \Delta\Phi_{i_k}$$

where:

$\Delta\Phi_{i_k} = \frac{\mathrm{Csun}\ \mathrm{A}_i}{\mathrm{N}} \cos\theta s_k$ is the energy of the k-th sun-ray striking dA_i.

To compute the extended and absorbed fluxes, that is to compute the reflection process, the rays are re-emitted on dA_i from the points (with $\chi=1$), with the energy:

$(1-\alpha_i)\Delta\Phi_{i_k}$

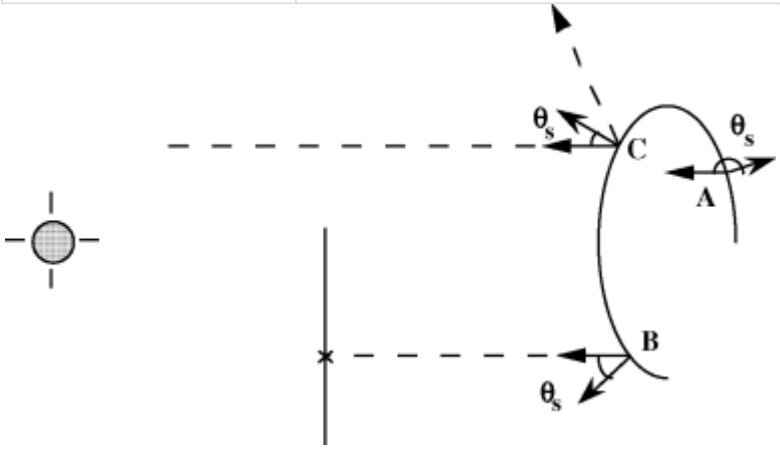
with αi being the solar absorptance of node 'i'.

Computation technique

The computation technique for the solar flux is illustrated on Figure 1.
The main computation steps are as follows:

- **Several points are sampled randomly on each node, as for the REF.**
- **Each point is tested to know whether it is oriented towards the sun or not (i.e. $\cos(\theta_s) > 0$ or not).**
- **Then, ray-tracing is used to test whether the point is occulted or not (i.e. in the shade of a surface or not).**
- **If not, the point is directly lit by the sun and the ray energy is transferred to the node as in the REF computation. Then, the ray is re-emitted from the point to compute extended and absorbed fluxes.**
- **The flux at a radiative node is the sum of all the ray energy transferred to the node.**

	cos θs is the same for every point of a planar node. To avoid useless computations, each planar node is tested before sampling the points.



Solar flux computation technique

Point	Orientation of points to the Sun

A	Not oriented to the Sun
B	Oriented to the Sun but in the shade of a surface
C	Directly lit by the Sun. The Sun ray is then re-emitted

External solar flux

Solar flux

The objective is to estimate on each node the absorbed solar flux:

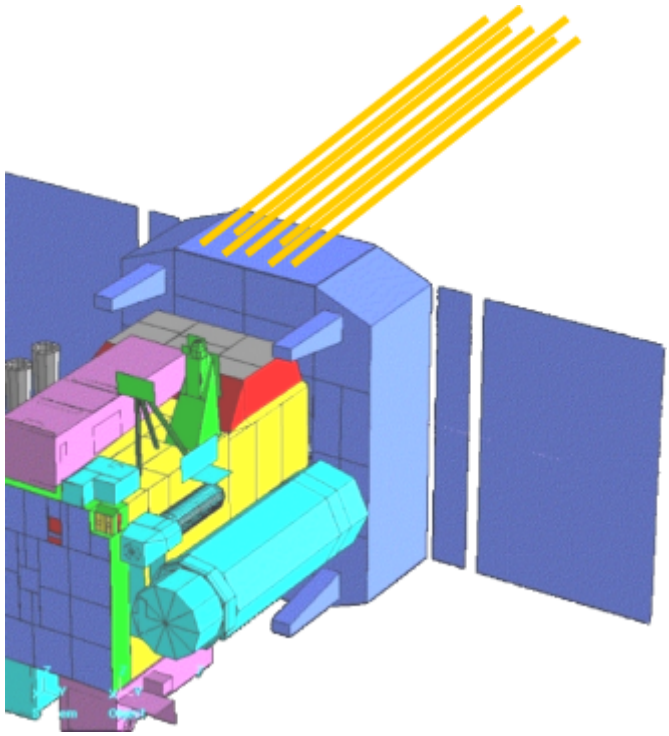
- Solution: use of the Monte Carlo Ray Tracing method.
- Rays are sampled randomly on each node.
- All the rays come from the same direction.

The numerical simulation consists in the following:

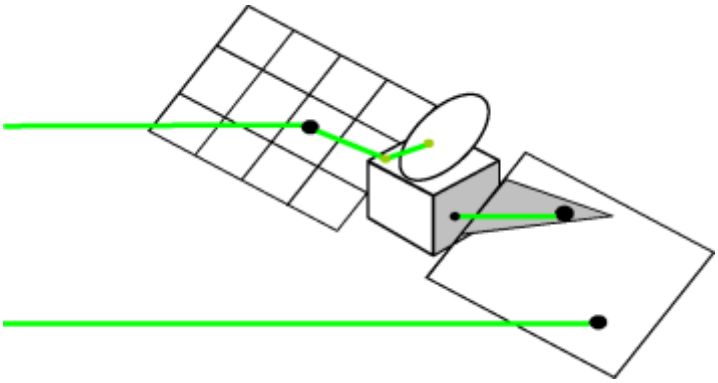
- Propagating the rays.
- Modeling the energy exchanges.
- Using the same approach as in the REF computation.

Differences from the REF computation:

- Thermo-optical properties in UV wave length
- Rays coming from a single direction or using a real sun at a finite distance.



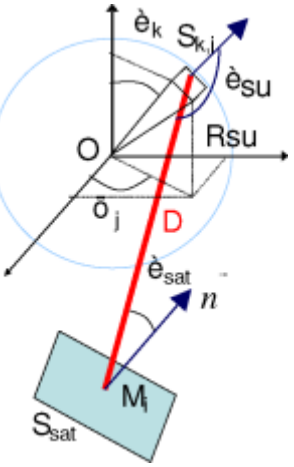
Absorbed solar flux through the Monte Carlo Ray tracing method



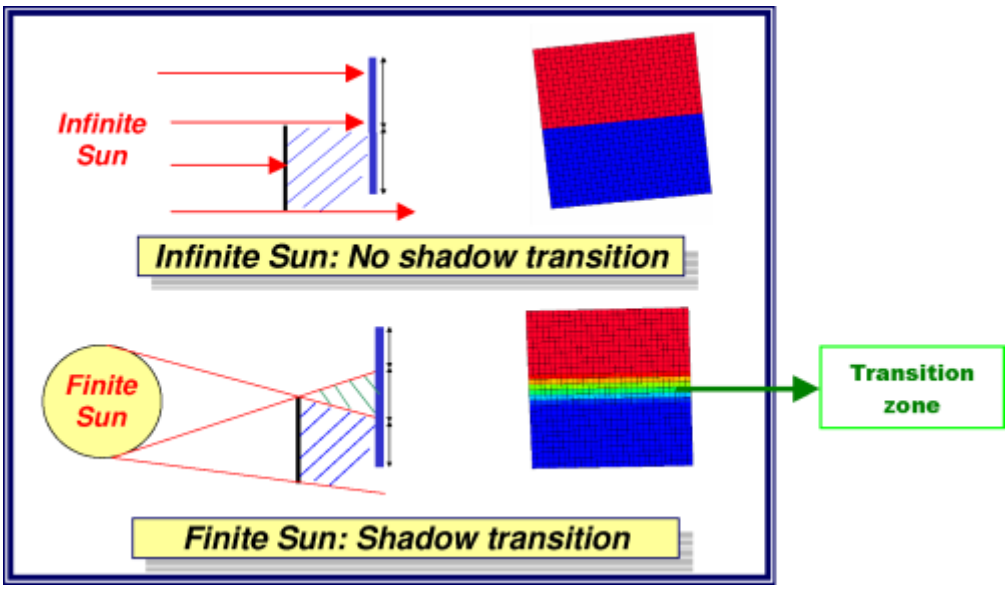
Ray propagation

The sun can now be a Sphere at a finite distance

- A rising need coming from recent & future scientific missions
- Venus Express
- Bepi Colombo
- Solar Orbiter.
- Ray tracing + numerical integration to integrate all geometrical and physical data



Numerical integration of finite Sun at a finite distance



Shadow transition

Solar Flux Parameters

Solar Flux Parameters

Input

- **.SYSSET file:** Mission synthesis for the THERMICA application.
- **.BOX.H5:** (optionnal) Output from the Radiation Module used to speed-up the computation by deactivating the identification of highlightedpoints for nodes without a direct view of space around the Sun's direction.

Outputs

- **.SF.LOG file:** Log file.
- **.SF.H5 file:** Contains fluxes results in HDF5 format to be displayed or post-processed into SYSTEMA.
- **.SF.NWK:** Solar flux expressed in the selected network format.
- **.SRT.H5:** (optionnal) Contains all computed rays for display in SYSTEMA.

Parameters

The parameters are splitted into 2 general categories: Solar Flux properties (local to the current module) and Common properties (global to the complete diagram).

Common properties

From the **Common** properties, the Solar Flux module will be influenced by the "Sun Constant" category. It is possible to configure an automatic or manual Sun Constant.

Solar Flux properties

Ray tracing

- **Random seed:** Is used to modify the generation of pseudo random numbers used by the ray-tracing. This option may be modified to check the results sensitivity to the random process.
- **Ray threshold:** This option is used to filter the ray bounces according to the remaining energy of a ray. By default, this filtering is done whenever a ray has lost at least 99% of its energy (1% of remaining energy). In such cases, the remaining energy is considered as absorbed by the last impacted node.

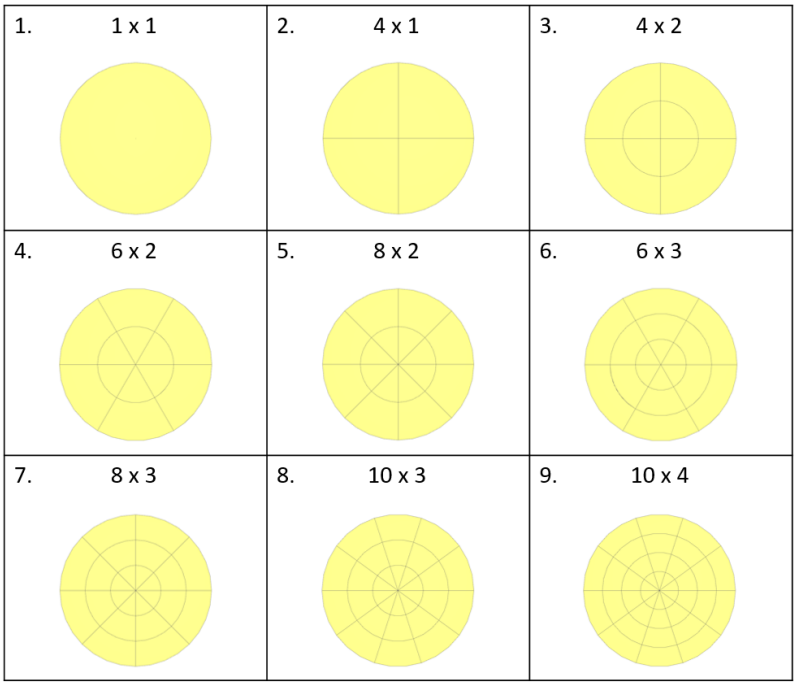
Critical / Normal / Low

- **Ray building method:** Select "Per node" / "Per mesh" / "Density".
  The default "Per node" value indicates that each node will use the given number of rays. In the case of condensed nodes (and/or with both active sides), this number of rays is distributed to the node's mesh according to their respective area (i.e. providing a uniform density of rays into the node).
  The "Per mesh" value may be used to specify the number of rays per radiative mesh (a both coating active mesh has in fact two radiative mesh: one per side). A condensed node will then use as many more rays as there are condensed radiative mesh into the node.

For computation performances, the total number of rays that shall be used for a node is re-distributed on the mesh so to keep a uniform density of rays.
This option shall be avoided when using CAD geometries because of those are build from a geometrical tesselation (and so with lots of mesh).

Flux Computation

- **Direct flux only:** If set to yes, it deactivates the multi-reflection and computes only the direct incident flux.
- **Direct/indirect absorbed fluxes:** If set to yes, it computes the direct and indirect absorbed fluxes and export the obtained values to the h5 file. This option is available only if the option "direct flux only" is set to no.
- **Flux filter:** Filter any QS corresponding to a flux lower than this value (in W/m^2).
- **Sun modelling:** Choose between a finite or infinite Sun modelling. For a finite Sun, an additionnal Sun detail level is required corresponding to a refinment of the Sun meshing.



Outputs

- **Network format:** Set the format of the nwk output for THERMISOL, ESATAN or MSC/SINDA.
- **Export Sun direction vector:** If set to yes , it exports the sun direction vector in three time dependent arrays SUNDIRX, SUNDIRY and SUNDIRZ in the .nwk file.
- **Export Sun constant:** If set to yes, it exports the Sun constant in a time dependant array (for automatic constant) or in a variable (for manual constant) in the .nwk file.
- **Export Sun angle:** If set to yes, It exports the sun angle (angle between the normal of node and the sun direction vector) in degrees for each node and at each computation time. It is stored in the .nwk file in arrays named SANGXXX where XXX is the node number. In addition, a new entity "SA" is available to access the sun angle for each node at each time step of a transient computation performed by the solver Thermisol. Note that the sun angle is not calculated in eclipse and equals to -360.0°. The sun angle export is not supported for fast spin kinematic bodies (in that case, the angle equals to -360.0°).
- **Ray display length:** Set the length of ray display to space. By default 2x (for reflection) and 2.5x (from Sun) the size of the model.

Planet Fluxes Module

The planet fluxes module computes the absorbed IR and Albedo fluxes onto the geometrical nodes of a model.

This documentation explains the theoritical background of the computation and the principles of the monte-carlo ray-tracing technic used.

Software parameters and inputs / outputs are also explained in a second part.

Planet Fluxes Overview

Planet Fluxes Overview

Introduction

External fluxes

The external sources of fluxes are the Sun, the Planet IR (infrared radiation) and the Planet Albedo. In this chapter only the planet fluxes are considered.

- **Planet IR emission**
- **Black body emission, uniform or non uniform.**
- **Planet Albedo**
- **Albedo coefficient, uniform or non uniform.**

The Albedo coefficients can either be constant on the planets or longitude, latitude and/or time dependent (using input maps)
The incoming IR flux from the planets can be specified as a "night and day temperatures" or into a map of longitude, latitude and/or time dependent temperature or flux.

Flux computation

For the flux computation, the definition of an orbit and pointing is required:

- **The computation is performed at each orbital position.**
- **At each position, the position and orientation of the spacecraft is automatically computed.**
- **The planet and sun positions are automatically computed.**

Scope of validity

In Systema, planets are considered as fluxes and not as thermal nodes. There is therefore no defined radiative coupling between a planet and the spacecraft. This is only valid if the temperature of the sky is small compared to the temperature of the planet, so that :

$T_{sky}^4 << T_{planet}^4$

If it is not the case (for very low-altitude missions, startospheric balloons for example), a correction should be applied to take into account the temperature of the sky.

External planet fluxes

Variations of planetary fluxes

- **Planet IR and Albedo can be latitude / longitude / time-dependent:**
- **useful for scientific missions**
- **and also for the Earth if high accuracy is needed.**
- **Extension of the projection algorithm:**
- **For each orbital position, the planet has a non-uniform IR level and Albedo factor.**
- **The global flux level is integrated from all solid angles around the spacecraft.**

Planet flux computation

- **Preliminary calculations of Planet IR & Albedo fluxes:**
- **During the REF computation, rays escaping to space have been collected.**
- **A virtual sphere located at infinity has been meshed.**
- **View factors have been evaluated between each surface and each sphere element.**
- **Projection on the planet:**
- **Each sphere mesh is radially projected on the planet.**
- **The on-ground light ratio is computed for projected meshes.**
- **Night and day temperatures interpolation:**
- **If 2 different temperatures have been specified, the temperature of planet element is linearly interpolated over the Sun oriented axis. This correspond to a cosine interpolation on the planet surface.**

The algorithm for the computation of the planet Albedo and infrared radiations introduces an additional processing in the computation of internal couplings by ray-tracing.
This processing is entirely dedicated to the computation of the planet Albedo and infrared radiations.
It can be summarized as follows:

- **Consider a unit box centered on the spacecraft and located at infinity so that the spacecraft can be considered as a point (see Figure 1).**
- **The unit box is regularly meshed (see Figure 2).**
- **In addition to the computation of couplings between the spacecraft nodes, couplings between the spacecraft nodes and each element of the unit box are computed.**

Main steps of this computation:

- **When a ray does not strike any spacecraft node (i.e. the ray strikes the space node):**
- **the unit box element struck by the ray is determined**
- **the ray energy is transferred to the struck box element**
- **at the end of Ray-tracing, in addition to internal couplings between the spacecraft nodes, the following data have been computed:**
- $F_{s,b}$: (extended) view factor from spacecraft node 's' to box element 'b'
- $B_{s,b}$: radiative exchange factor from spacecraft node 's' to unit box element 'b'.
- **Starting from the previous results, the Planet albedo $\Phi a(s)$ and infrared radiations $\Phi p(s)$ on node 's' of the spacecraft are computed by the following formula:**

$$\Phi_p(s) = A_s \, \sigma \sum_b F_{s,b} \, X_b \, \Phi_b$$

$$\Phi_a(s) = A_s \, Csun \sum_b F_{s,b} \, X'_b \, \rho_b$$

where:

Csun	:	solar flux density
σ	:	Stefan-Boltzmann constant
Φ_b	:	Mean IR flux emission of the planet at the projection of the box element. For a constant emission, $\Phi_b = \sigma T^4$ (where T is the absolute temperature of the planet). For a variable emission, Φ_b is determined by interpolation in the user table
A_s	:	area of node 's'
ρ_b	:	Mean diffuse reflectivity of planet at the projection of the box element. For a constant albedo level, ρ_b has always the same value. For a variable albedo level, ρ_b is determined by interpolation in the user table

X_b and X'_b are defined as follows (see Figure 3 & Figure 4):

$$X_b = \frac{\Omega_{bp}}{\Omega_b}$$

$$X'_b = \frac{\int_{\Omega_{bp}} f(\theta_s) d\Omega}{\Omega_b}$$

Ω_b	:	solid angle of the element 'b' seen from the unit box center S
Ω_{bp}	:	solid angle defined by the intersection of Wb with the planet
$f(\theta_s)$	=	$\max(\cos(\theta_s),0)$

The expression of Ω_b is:

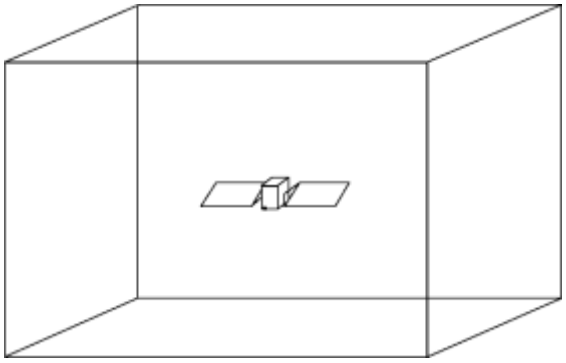
$$\Omega_b = \int_b \frac{c}{l^3} da$$

approached by:

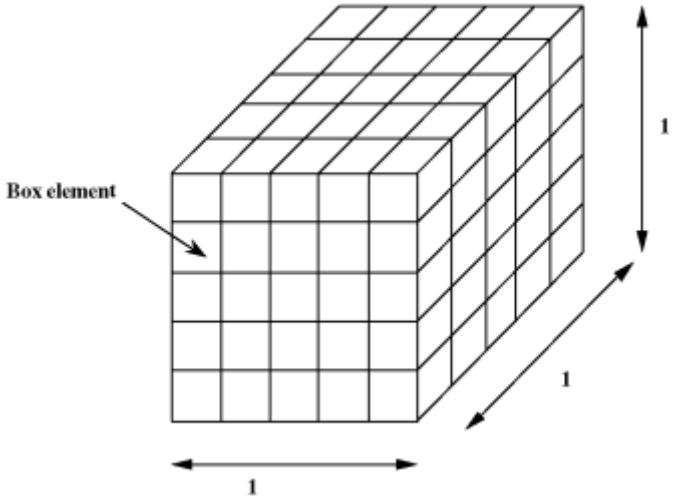
$$\Omega_b = \frac{c A_b}{l^3}$$

where:

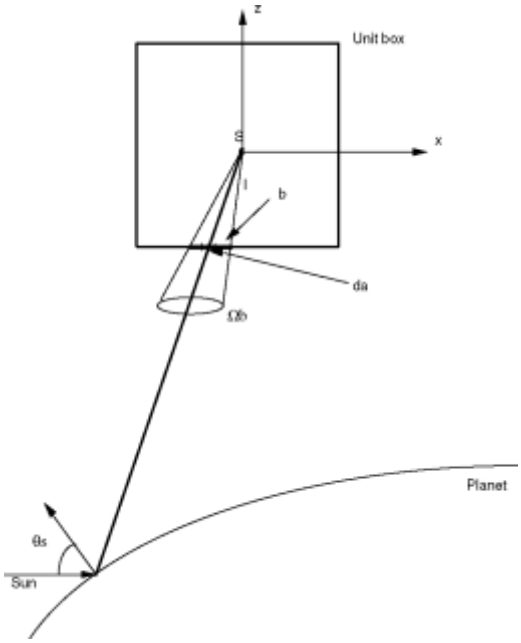
A_b	:	area of the unit box element 'b'
χ	:	unit box half size (c=0.5)
L	=	distance between S, the unit box center and the center of the unit box element



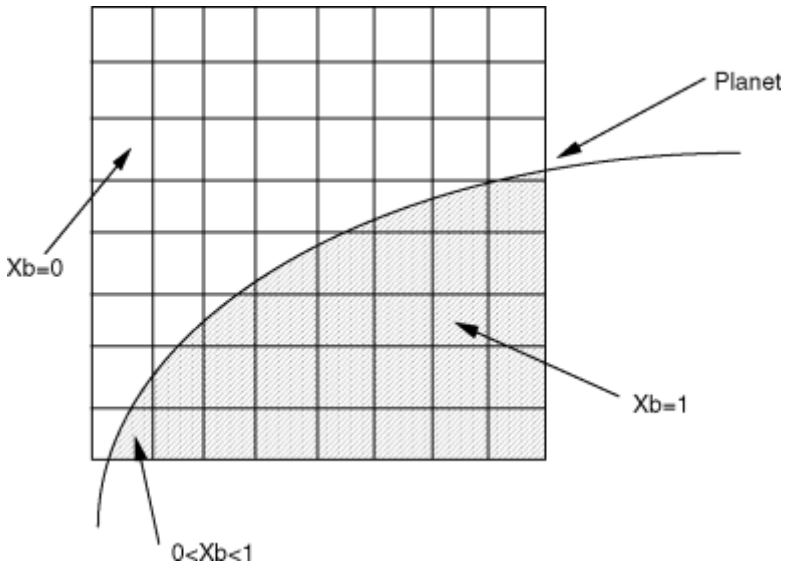
Unit box located at infinity and centered on the spacecraft



Unit box meshing



Box element projection on Planet



Projection of Planet onto a face of the unit box

Planet Fluxes Parameters

Planet Fluxes Parameters

Inputs

- **.SYSSET file:** Mission synthesis for the THERMICA application.
- **.BOX.H5 file:** Results issued from the radiative computation.
- **.MAPS file:** (optionnal) Maps of Albedo and/or IR properties for planets.

Using maps of planet properties

If specified, the properties described into the maps file overload the values given into the parameter menus. The maps are described into a specific format which respects the following rules:

- **'#' starts a comment**
- **'$*planet property*' starts a new map**
  - **Where '*planet*' can take one of the values:**
    - **MERCURY**
    - **VENUS**
    - **EARTH**
    - **MOON**
    - **MARS**
    - **JUPITER**
    - **SATURN**
    - **URANUS**
    - **NEPTUNE**
  - **Where '*property*' can take one of the values**
    - **ALBEDO**
    - **TEMPERATURE**
    - **IR**

For each planet, only one of temperature or IR properties can be defined. By default, the temperatures are considered as degrees Celsius.

It is possible to specifiy the temperature unit by adding the following line in the temperature definition block :

$EARTH TEMPERATURE
 # Temperature unit - Kelvin and Celsius degrees are supported
UNIT = Kelvin

- **'FRAME = *frame*' sets the frame definition of the current map**
  - **Where '*frame*' can take one of the values**
    - **ROTATIONAL (or GREENWHICH for the Earth)**
    - **SUN_ORIENTED**
- **'LONGITUDES = [-180, *[x,]* 180]' sets the longitudes intervals**
- **'LATITUDES = [-90, *[x,]* 90]' sets the latitudes intervals**

The longitudes and latitudes can only be expressed between [-180, 180] and [-90, 90] respectively.
In the rotational frame:

In the Greenwich oriented frame:

- *(0,0) corresponds to the intersection between the equator and the rotational reference (the Greenwich meridian for the Earth)*
- *The longitudes give the West to East direction*
- *The latitudes give the South to North direction*

In the Sun oriented frame:

- *(0,0) correspond to the sub ecliptic north pole*
- *The latitudes give the Night to Day direction (the point of 90° latitude is the sub-solar point)*

	• *The longitude axe is in the ecliptic plane (180° is the planet velocity direction)*

- **'TIME = *time*' sets the time definition of the current map**
  - **Where '*time*' can take one of the values**
    - **EARTHDAY (only for Earth)**
    - **JULIAN**
    - **NO**
- **'TIMES = [x *[,y]*]' sets the times of the properties maps**

	*The values given by the maps are interpolated over the time*

- **'@ *comment*' starts a new set of properties for a given time, followed by an optional comment**

	*Values are then written into a table format, where each line contains the longitude interval values (from -180° to +180°) for a latitude interval (written from -90° to 90°)*

A file with the Albedo and Earth IR is given as an example (earth.maps). Those tables are Earth IR/albedo mapping file extracted from R. Peyrou-Lauga data (Romain Peyrou-Lauga, Using real Earth Albedo and Earth IR Flux for Spacecraft Thermal Analysis, ESA, ESTEC, Noordwijk, The Netherlands, July 2017).

Outputs

The following two files are used to process the results in the Modeler or Mission tabs of SYSTEMA:

- **.PF.LOG file:** Log file.
- **.PF.H5 file:** Contains fluxes results in HDF5 format to be displayed or post-processed into SYSTEMA.
- **.PF.NWK file:** Planet fluxes expressed in the selected network format.
- **.MAPS file:** (optionnal) Maps of Albedo and IR properties for one planet mission, with albedo and IR values computed from each latitude and longitude given in the input file, for every computational time. The .maps output file needs to have a .maps input file as a reference for computation. The .maps output file is limited to 76 computation points.

Parameters

The parameters are splitted into 2 general categories: Planet Fluxes properties (local to the current module) and Common properties (global to the complete diagram).

Common properties

From the **Common** properties, the Planet Fluxes module will be influenced by the "Sun Constant" category for the Albedo reflection. It is possible to configure an automatic or manual Sun Constant.

Planet Fluxes properties

Fluxes Computation

- **Albedo computation:** If set to no, it deactivates the Albedo flux.
- **IR computation:** If set to no, it deactivates the IR flux.

- **Direct/indirect absorbed fluxes:** If set to yes, it computes the direct and indirect absorbed albedo and/or IR fluxes. The obtained values are exported to the h5 file. Note that the direct/indirect albedo and infra-red fluxes are calculated only if the albedo and IR computation are enabled respectively.
- **Flux filter [W/m^2]:** Filter any QS corresponding to a flux lower than this value.
- **Exhaustive contribution from the Box:** Forces to take into account every direction facing the planet to compute the flux contribution. This option is used to improve flux computing for faraway planets (For instance Moon fluxes from Earth orbit).

Planet properties (for each planet)

- **Day temperature:** Temperature of sub-solar point on the planet.
- **Night temperature:** Temperature of eclipsed part of the planet.

	*If Day and Night temperatures are equal, then the temperature profile is constant onto the planet.* *Otherwise, the temperature profile is interpolated between the eclipse part and the sub-solar point so that the emitted flux has a linear onto the planet's ground.*

- **Albedo:** Albedo coefficient.

Outputs

- **Network format:** Set the format of the nwk output for THERMISOL, ESATAN or MSC/SINDA.
- **Export Planet direction vector:** If set to yes, it exports the planet direction vector (for each active planet) in three time dependent arrays.
- **Export nadir solar zenith angle:** If set to yes, it exports the Solar Zenith Angle value at each computation time in the .pf.nwk file. The SZA is defined as the angle between the Planet to Spacecraft vector and the Spacecraft to Sun vector. This option is available only for Thermisol output format.

Aero Flux Module

The Aero Flux module computes the absorbed Free Molecular Heating (FMH) flux onto the geometrical nodes of a model.

This documentation explains the theoretical background of the FMH computation. Software parameters and inputs/outputs are also explained in a second part.

Aero Flux Overview

Introduction

The present section deals with the modelling of the Aero Flux which is a thermal effect, related to the Free Molecular Heating (FMH) and stemming from the interactions between the atmosphere gas molecules and space vehicule surfaces.

FMH is caused by friction in the rarefied upper atmosphere that may be experienced by space vehicules and may lead to important constraints for space vehicule thermal analysis.

Indeed, FMH, when not taken into account into heat flux models, can result in higher-than-predicted heat loads on space vehicule surfaces. There are situations where FMH is a significant environmental factor and must be incorporated into thermal analysis along with solar, Earth infrared and albedo sources.

Free Molecular Heating

Free Molecular Flow Regime

Whereas in the lower atmosphere the kinetic theory of gasses provides the most appropriate description, in the upper atmosphere, one should turn to the Free Molecule Regime.

Indeed, due to the fact that the density of the gas molecules is so low in the upper atmosphere, the intermolecular collisions are considered negligible. As a consequence, the heating process involved in the upper atmosphere is modeled as collisions between a space vehicule with individual molecules rather than as a gas-flow heating problem.

In order to ensure the validity of the free molecular flow regime, one has to take into account the mean-free path, λ, of the molecules which is the average distance traveled by a moving molecule between two successive collisions and depends on both altitude and density. The free molecule flow regime approximation stands if λ is much larger than the characteristic length of the space vehicule, d.

In practice, free molecule flow is assumed for

$$\frac{\lambda}{d} \geq 10$$

Free Molecular Heating Theory

When free molecules collide onto a surface, their kinetic energy is transferred to thermal energy on the surface.

The FMH global heating rate, $\varphi_{FMH,\,abs}$ (in W/m^2), generated by FMH and absorbed by a surface normal to the incident molecular flow is given by

$$\varphi_{FMH,\,abs} = \alpha \frac{1}{2} \rho V^3$$

where ρ is the atmosphere density, V is the vehicule velocity relative to the atmosphere velocity and α is the thermal accommodation/adaptation coefficient.

FMH is an explicit function of both space vehicule velocity and atmosphere density but the impact of the space vehicule velocity variations has proved to be insignificant compared to the effects of the atmosphere density variations.

The vehicule velocity is already provided by the orbital mechanics part of the model while the atmosphere density is computed by the user-selected atmosphere model.

The upper atmosphere molecules density depends primarily on the altitude and on various environmental factors, such as solar activity, geomagnetic activity, location and time.

One of the main assumptions of the free molecular flow regime is that the flow of atmospheric gas molecules incident on a spacecraft surface is unperturbed by the presence of the space vehicule. Moreover, the free molecular regime is characterized by single collisions between the gas molecules and the spacecraft vehicule.

FMH becomes significant in low-pressure altitudes where the atmosphere molecules no longer act like a fluid but the density is still high enough to cause significant interactions with the spacecraft surfaces.

FMH is not restricted to launch but may as well be experienced by spacecrafts in operational orbits with perigee altitudes below 180km, for instance.

FMH Reflections

The accomodation/adaptation coefficient, α, quantifies to what extent the kinetic energy of the free gas molecules is transferred to thermal energy on the surface. α also indicates the diffusive and/or specular nature of the FMH reflections.

When purely diffuse reflections are involved, α=1, the free molecules are momentarily absorbed by the space vehicule surface and then emitted with a velocity which depends on the surface temperature, according to the Maxwell distribution.

In specular reflections, α=0, the nominal velocity of the free molecules is reversed while the tangential component is unchanged.

In situations involving space vehicules, diffuse reflections tend to be predominant and accomodation coefficients of approximately 0.6 to 0.8 are considered appropriate, but α=1.0 is recommended for conservatism considerations.

Nevertheless, in practice, the two extreme situations, α=0 and α=1, can also be used to perform maximum and minimum heating analysis, respectively.

FMH Computation in Systema

In practice, when random molecular trajectories are involved, it is possible to calculate the Aero Flux heat load, Φ_{AF} (in W), on a given spacecraft surface by multiplying the FMH global heating rate by the cross-sectional area of the surface in question, S_c, and the cosine of the angle, θ, between the surface normal and the velocity vector of the incident molecular flux.

$$\Phi_{AF} = S_c \cos\theta \left(\alpha \frac{1}{2} \rho V^3 \right)$$

In Systema, the computation of the Aero Flux is performed under the following assumptions:

- no multiple collisions between a given atmosphere molecule and the space vehicule

- only the aerodynamic velocity vector of the molecules is taken into account (no thermal motion vector)
- no wind effects

Ambient Atmosphere

In order to compute the above-mentioned Aero Flux, it is necessary to have a good knowledge of the atmosphere density experienced by the spacecraft during its mission. Atmospheric densities
are highly difficult quantities to predict because of the large number of parameters they depend on. In practice, the scientific community rather relies on sophisticated atmospheric models with a dozen or more
input parameters to predict realistic and accurate descriptions of the atmospheric environment.

Two different atmosphere models are available in the Aero Flux module: the NRLMSISE-00 model and the standard atmosphere model US76.

The NRLMSISE-00 Atmosphere Model

The NRL Mass Spectrometer, Incoherent Scatter Radar Extended Model (NRLMSISE-00) model describes the neutral temperature and species densities in Earth's atmosphere. It is based on a very large underlying set of supporting data from satellites, rockets and radars, with extensive temporal and spatial distribution. It has been extensively tested by the global scientific community against experimental data. The model has a flexible mathematical formulation. It is valid for use from ground level to the exosphere. The Mass Spectrometer and Incoherent Scatter (MSIS) series of models developed between 1977 and 1990 are used extensively by the scientific community for their superior description of neutral composition. The models utilized atmospheric composition data from instrumented satellites and temperatures from ground-based radars. The initial MSIS 1977 model was based on the Jacchia temperature profile framework, but the density at 120 km varied with local time and other geophysical parameters to fit the measurements. Exospheric temperature and density variations were represented by spherical harmonics resulting in requiring fewer parameters for a given level of accuracy. Subsequent versions of the model include the longitude variations, a refined geomagnetic storm effect, improved high latitude, high solar flux data and a boundary lowered to sea level. The NRLMSISE-00 model of atmospheric composition, temperature, and total mass density from ground to exobase includes the following:

- Drag data based on orbit determination
- More recent accelerometer data sets
- New temperature data derived from Millstone Hill and Arecibo incoherent scatter radar observations
- Observations of O2 by the Solar Maximum Mission (SMM), based on solar ultraviolet occultation.

For this model, the following parameters are used.

Input Parameters

- Year and day in the year (YYDDD)
- Seconds in the day (UT seconds)
- Altitude (kilometres)
- Latitude (degrees)
- Longitude (degrees)
- Local solar time (decimal hours)
- Daily F10.7 solar flux (10^{-22} W/m2/Hz)
- 3 month average F10.7 solar flux (10^{-22} W/m2/Hz)
- Magnetic index (Ap)
- Mass number of the selected species

Output Parameters

- Helium number density (/cm3)

- Atomic oxygen number density (/cm3)
- Molecular nitrogen number density (/cm3)
- Molecular oxygen number density (/cm3)
- Argon number density (/cm3)
- Atomic hydrogen number density (/cm3)
- Atomic nitrogen number density (/cm3)
- Total density (g/cm3)
- Exospheric temperature (K)
- Temperature at altitude (K)

The US76 Standard Atmosphere Model

The US76 standard atmosphere model is a hypothetical, vertical distribution of atmospheric temperature, pressure and density. In this static model, the gasses of the atmosphere are assumed to obey the perfect gas law and hydrostatic equation, which, when taken together, relate temperature, pressure and density with altitude.

Computation Scheme

Monte-Carlo Ray-Tracing

The Aero Flux module is based on the idea to benefit from the similarity of the physics related to free molecules striking a surface and light striking a surface. Consequently, the process used in the Aero Flux module to model FMH phenomena is very similar to the process used in the Solar Flux module, and relies on the Monte-Carlo ray-tracing method where a large number of solar rays are generated randomly. In the Aero Flux module, the Monte Carlo ray tracing method is used to compute random trajectories of large numbers of incident molecules that may collide with the space vehicule surfaces. In particular, the modelling of the FMH phenomenon takes into account shadowing effects, that is when surface elements are being obstructed by others, but does not account for multiple reflections, as opposed to the Solar flux computation. Indeed, free molecules striking a given surface of a space vehicule are assumed to have negligible interactions with other surfaces after the initial collision.

Aero Flux Parameters

Input

- **.SYSSET file**: Mission synthesis for the THERMICA application.
- **.BOX.H5 file**: Results issued from the radiative computation.

Output

- **.AF.LOG file**: Log file.
- **.AF.H5 file**: Contains fluxes results in HDF5 format to be displayed or post-processed into SYSTEMA.
- **.AF.NWK file**: Contains computed Aero fluxes expressed in the selected network format.

Parameters

The parameters are splitted into two general categories: Aero Flux properties (local to the current module) and Common properties (global to the complete diagram).

Common Properties

The Common properties here should not impact the Aero Flux module.

Aero Flux Properties

Ray tracing

- **Random seed**: Is used to modify the generation of pseudo random numbers used by the ray-tracing. This option may be modified to check the results sensitivity to the random process.
- **Ray number per mesh**: The "Per mesh" value indicates that each mesh will use the given number of rays.

  It specifies the number of rays per radiative mesh (a both coating active mesh has in fact two radiative meshes: one per side).

Outputs

- **Network format**: Set the format of the nwk output for THERMISOL, ESATAN or MSC/SINDA.
- **Adaptation coefficient**: Set the adaptation factor (or accommodation factor) used to compute the effective aero flux.
- **Aero flux constant limit**: Set the upper limit of the aero flux.
- **Aero flux table source**: Specify the atmosphere model used for the computation of the aero flux. Choose between the Earth atmosphere models NRLMSISE-00 and US76 (only for Earth orbit). The user-defined atmosphere model option is not available for the Aero Flux module.
- **Ap:** Earth magnetic index, which is an input for the NRLMSISE-00 atmosphere model.
- **F107A:** 81-day average F10.7 solar flux index (10^{-22} W/m2/Hz), which is an input for the NRLMSISE-00 atmosphere model.
- **F107:** Daily F10.7 solar flux index (10^{-22} W/m2/Hz), which is an input for the NRLMSISE-00 atmosphere model.
- **Aero Flux Table Name:** Specify the name of the Aero Flux table that is available in the .AF.NWK, .AF.LOG ("AeroFlux Cst") and .AF.H5 files. This Aero Flux table contains the values of the "bare" incident FMH heat rate (without the adaptation coefficient, cross-sectional section and cosine multiplicative factors), not the effective heat rate absorbed by the space vehicule surfaces.

Conduction Module

Introduction

The conductive energy exchange in a solid is governed by the Fourier's law. The application of the Fourier's law in the nodal network leads to the conductive power exchange between nodes.

Program history

- In the first versions, a geometrical approach could be used for any simulation at system level.
- Advantage: no specific requirements on the model.
- Disadvantage: bad accuracy when using shapes with high distortion.
- In the previous version, a finite element (FE) approach was used for simulation at instrument level.
- Advantage: the method is based on the Fourier's law integration and gives exact results for linear temperature solutions.
- Disadvantage: for non-linear temperature solution, the method lies on a C0 temperature field (i.e. with no continuity of the temperature across edges of the shapes) and, as a consequence, has a strong dependency on the mesh.
- In the actual version, a new volume element approach has been implemented. It uses a "sub mesh – reduction" process. This new method has been named *Reduced Conductive Network* (RCN).
- Advantage: the method lies on a C1 temperature profile, it integrates the fluxes so the node temperatures correspond to the mean temperatures of the associate geometrical elements.

RCN approach requirements and quick computation principles

Conductive network definition

- Introduction of the edge nodes (free borders and contacts)
- Consideration of the surface node

Computation principle

- Decomposition of a node into a detailed sub-mesh
- Computation of elementary couplings
- Assembly of a detailed conductive network
- Reduction of the detailed network

Fourier's law

The conduction deals with heat transport inside the structure of a material. This phenomenon is physically modeled with the Fourier's law. Eq.1.1 :

$$\varphi = -\lambda grad T$$

where

- λ is a material characteristic, called thermal conductivity (W/m/K). It usually depends on the temperature and location:

$$\lambda = \lambda(x, y, z, T)$$

- φ is the density flux vector (W/m2) linked to the flux by the equation: Eq.1.2 :

$$d\phi = \varphi n dS$$

  with *n* being the normal surface
- *T* is the temperature (K)

In the general case of an anisotropic material, Eq.1.1 is written as follows, Eq.1.3 :

$$\varphi = -\overline{\overline{\lambda}} grad T$$

where $\overline{\overline{\lambda}}$ is a tensor

Conduction Overview

Conduction Overview

Overview of the RCN method

Goal of the RCN

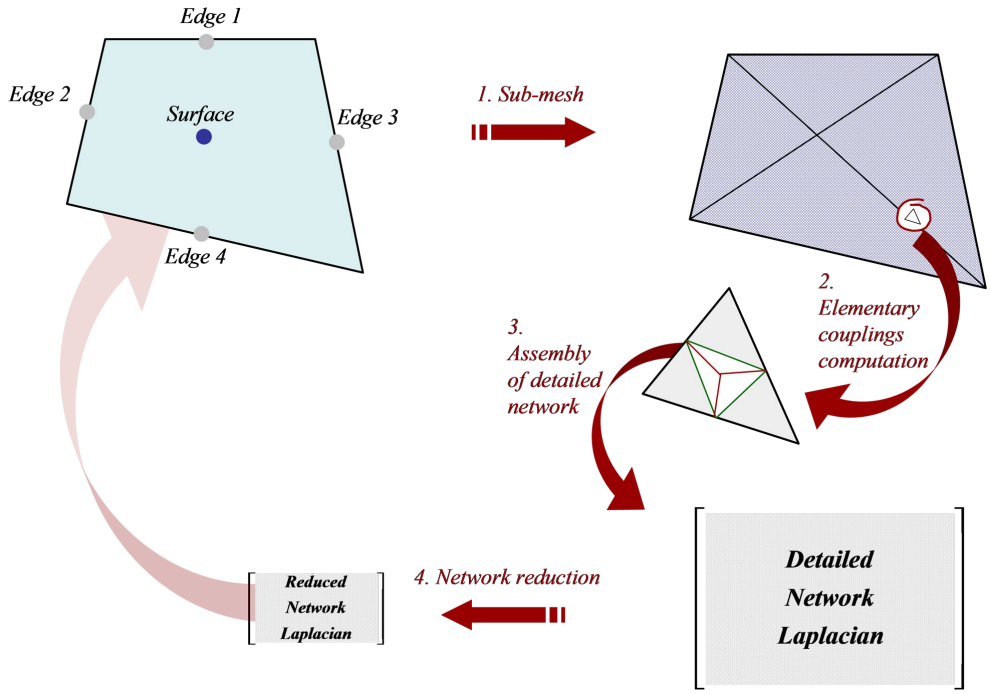
The study of the conduction heat transfer in Thermica should not lead to the calculation of the temperatures but to the conductive exchange factors between the nodes composed of geometrical elements. Those couplings are then integrated into the full mathematical model made by the nodal description, the radiation couplings, the external fluxes as well as other elements added by the user.
In the context of the radiation couplings or external fluxes computations, the nodes representing surfaces are supposed to be isothermal. On the other hand, for the conduction heat transfer, we need to have a gradient of temperature which means that we have to introduce such a gradient into the model based on the radiation computation.
In order to retrieve a temperature gradient into a node, we add interfaces nodes, called edge nodes. Those nodes will represent the mean temperatures of the free borders and contacts. Also the surface node has to represent the mean temperature of the surface (and not the temperature at the center of the surface).

Steps of the RCN

The RCN method is applied sequentially on each node and executes the following main tasks:

1. The surface is sub-meshed
2. The conductive couplings are computed into each sub-triangle
3. The sub-meshed surface leads to a complex conductive network
4. The detailed conductive network is reduced back to original configuration

	This process can handle condensed nodes (i.e. a node composed of many geometrical elements) as well as non-uniform nodal breakdown. *It has also been design to support boolean shapes but their integration will be implemented in a further version.*

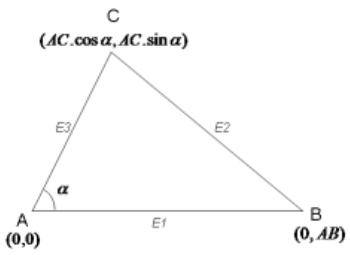


Surface conductive process

Elementary couplings computation

Linear Case Study

The principle of this approach is to compute the linear coupling between the nodes of a mesh element. To do so, let's consider a triangle within which the temperature is linear.



Expression of the temperature:

$$T = T_A + \frac{T_B - T_A}{AB}x + \left(\frac{T_C - T_A}{AC.\sin\alpha} - \frac{T_B - T_A}{AB.\sin\alpha}.\cos\alpha\right)y$$

Expression for the gradient of temperature:

$$\overrightarrow{\nabla T} = \begin{pmatrix} \partial T/\partial x \\ \partial T/\partial y \end{pmatrix} = \begin{pmatrix} \frac{T_B - T_A}{AB} \\ \frac{T_C - T_A}{AC.\sin\alpha} - \frac{T_B - T_A}{AB.\sin\alpha}.\cos\alpha \end{pmatrix}$$

The expression of the external normals is:

$$l_1\overrightarrow{N_1} = (0, -AB)$$

$$l_2\overrightarrow{N_2} = (AC\sin\alpha, AB - AC\cos\alpha)$$

$$l_3\overrightarrow{N_3} = (-AC\sin\alpha, AC\cos\alpha)$$

So the flux crossing the borders are:

$$q_k = -\lambda.t.l_k.\vec{\nabla T}.\vec{N_k}$$

$$q_1 = \lambda.t.\left[\frac{AB}{AC\sin\alpha}(T_C - T_A) - \frac{\cos\alpha}{\sin\alpha}(T_B - T_A)\right]$$

$$q_2 = \lambda.t.\left[\left(\frac{AC - AB\cos\alpha}{AB\sin\alpha}\right)(T_B - T_A) + \left(\frac{AB - AC\cos\alpha}{AC\sin\alpha}\right)(T_C - T_A)\right]$$

$$q_3 = \lambda.t.\left[\frac{AC\sin\alpha + AC\cos^2\alpha}{AB}(T_B - T_A) - (T_A - T_C)\right]$$

Introducing the average temperatures of the edges E1, E2, E3, we can re-write this system:

$$q_1 = \lambda t\ \frac{AB^2 + BC^2 - AC^2}{2.Area}(T_2 - T_1) + \lambda t\ \frac{AB^2 + AC^2 - BC^2}{2.Area}(T_3 - T_1)$$

$$q_2 = \lambda t\ \frac{AB^2 + BC^2 - AC^2}{2.Area}(T_1 - T_2) + \lambda t\ \frac{AC^2 + BC^2 - BC^2}{2.Area}(T_3 - T_2)$$

$$q_3 = \lambda t\ \frac{AB^2 + AC^2 - BC^2}{2.Area}(T_1 - T_3) + \lambda t\ \frac{AC^2 + BC^2 - AB^2}{2.Area}(T_2 - T_3)$$

Then we have expressed the symmetric coefficients:

$$g_{12} = \lambda t\ \frac{AB^2 + BC^2 - AC^2}{2.Area}$$

$$g_{13} = \lambda t\ \frac{AB^2 + AC^2 - BC^2}{2.Area}$$

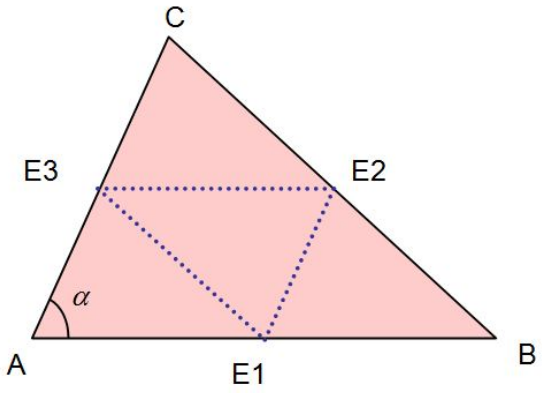
$$g_{23} = \lambda t\ \frac{AC^2 + BC^2 - AB^2}{2.Area}$$

The Conductive Power on the triangle is:

$$Q_S = Q_{E1} + Q_{E2} + Q_{E3}$$

$$= \left[g_{12}(T_2 - T_1) + g_{13}(T_3 - T_1)\right] + \left[g_{12}(T_1 - T_2) + g_{23}(T_3 - T_2)\right] + \left[g_{13}(T_1 - T_3) + g_{23}(T_2 - T_3)\right]$$

$$= 0 \quad \forall T_1, T_2, T_3$$



Remarks on those results:

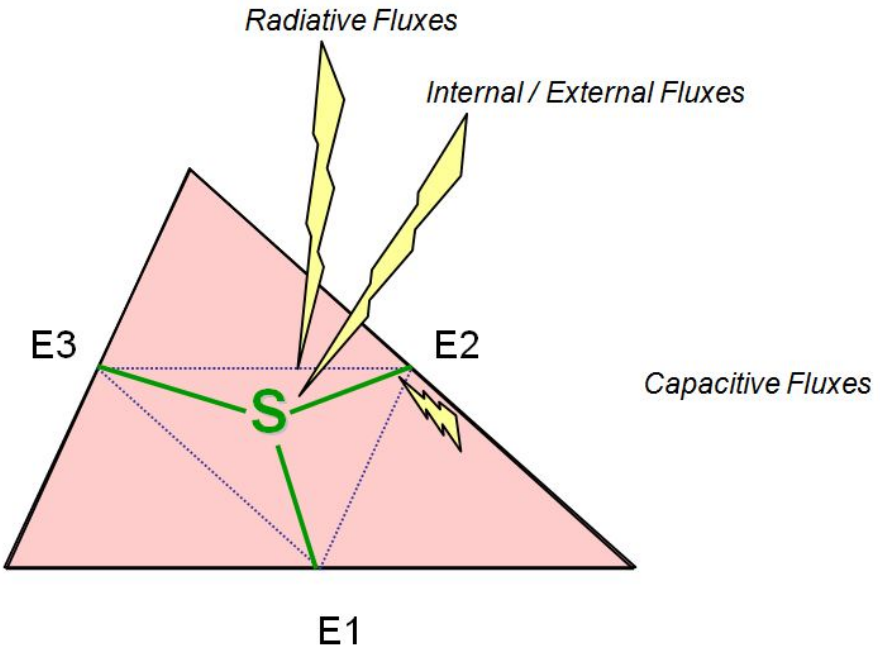
- Considering a linear temperature profile imply that the conductive flux crossing a surface is always null

  Note that the it implies:
  If the conductive flux crossing a surface is not null (i.e. there are other fluxes) then the temperature profile within the shape is not linear

- This problem is a pure conductive problem for which the network is only composed of "edge" elements. Fluxes on the "surface" elements are not yet taken into account.

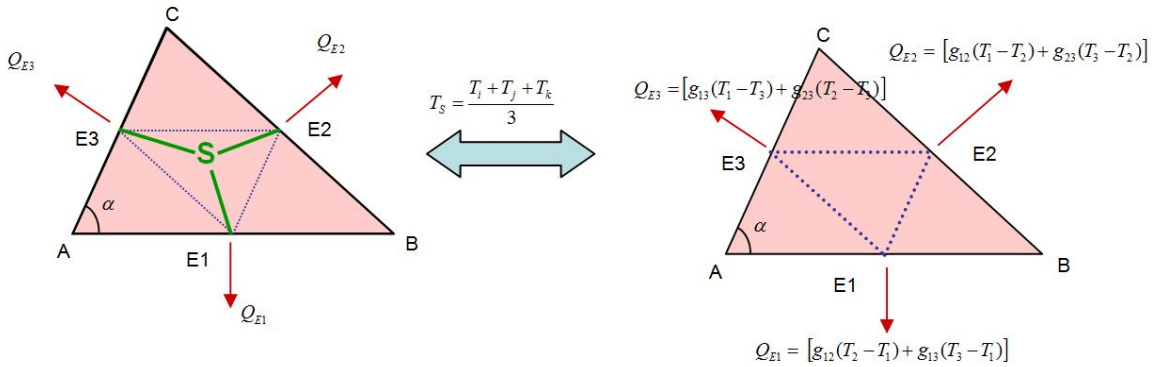
Introduction of the Average Surface Temperature

- In general, other fluxes than purely conductive flux have to be considered
- Those other fluxes lay on the shape's surface itself
- So we need to introduce the surface node in the previous conductive network in order to take into account all the surface fluxes



- Any solution of the general problem (i.e. involving the surface node) shall be equivalent to the purely conductive problem for linear solution

Note : there is no hypothesis of linearity imposed here but a constraint on the general solution to be at least true in linear cases (without surface fluxes)



Edge Powers in the linear Case:

- The preliminary study gave us the flux crossing an edge *i* on a linear problem

$$Q_i = g_{ij}(T_j - T_i) + g_{ik}(T_k - T_i)$$

- Let's consider the formulations

$$g_{ij} = \frac{a}{3} + b_{ij} \qquad g_{ik} = \frac{a}{3} + b_{ik}$$

- Then, introducing

$$T_S = \frac{T_i + T_j + T_k}{3}$$

$$\begin{aligned} Q_i &= \frac{a}{3}(T_j + T_k - 2.T_i) + b_{ij}(T_j - T_i) + b_{ik}(T_k - T_i) \\ &= a(T_S - T_i) + b_{ij}(T_j - T_i) + b_{ik}(T_k - T_i) \end{aligned}$$

Surface Power in the linear Case:

- The conductive power on the surface is equal to the sum of the edge power:

$$\begin{aligned} Q_S &= Q_i + Q_j + Q_k \\ &= a.(3T_S - T_i - T_j - T_k) \\ &= 0 \quad \forall a \end{aligned}$$

Which is of course always equal to zero (because we are in a linear case)

Power System Equation in a Linear Case:
The previous power equations can be summarized by the following matrix equation which is true in linear cases for any coefficient a:

$$\underbrace{\begin{pmatrix} -3a & a & a & a \\ a & -\frac{a}{3}-g_{ij}-g_{ik} & g_{ij}-\frac{a}{3} & g_{ik}-\frac{a}{3} \\ a & g_{ij}-\frac{a}{3} & -\frac{a}{3}-g_{ij}-g_{jk} & g_{jk}-\frac{a}{3} \\ a & g_{ik}-\frac{a}{3} & g_{jk}-\frac{a}{3} & -\frac{a}{3}-g_{ik}-g_{jk} \end{pmatrix}}_{\text{Matrix E}} \cdot \begin{pmatrix} T_S \\ T_i \\ T_j \\ T_k \end{pmatrix} = \begin{pmatrix} Q_S = 0 \\ Q_i \\ Q_j \\ Q_k \end{pmatrix}$$

$$Q_i + Q_j + Q_k = 0$$

Matrix E
"Elementary Matrix"

Quasi-Linear Case Study

- We have introduced the shape average node in the linear case.
- For the further demonstration of the RCN which solves the general problem (which is not linear), we will need to consider quasi-linear cases.
- So now, let's consider that power on the surface is equal to an epsilon:

$$\tilde{Q}_S = \varepsilon_S$$

The new average temperature of the shape is by consequence not equal to the average temperature of the edges but is increased due to the additional surface flux:

$$\tilde{T}_S = T_S + \delta T_S \quad with \quad T_S = \frac{T_i + T_j + T_k}{3} \quad and \quad \delta T_S = f(\varepsilon_S)$$

- Since $\varepsilon_S << 1$ we can approximate the new shape's temperature by

$$\exists k \quad / \quad \tilde{T}_S = T_S + k.\varepsilon_S + o(\varepsilon_S^2) \approx T_S + k.\varepsilon_S$$

Which can also be written

$$\exists a \quad / \quad \tilde{T}_S = T_S + \frac{1}{3a}\varepsilon_S + o(\varepsilon_S^2) \approx T_S + \frac{1}{3a}\varepsilon_S$$

Leading following Power Balance on the surface:

$$\tilde{Q}_S = a\left(3\left(T_S + \frac{1}{3a}\varepsilon_S\right) - T_i - T_j - T_k\right) = \varepsilon_S$$

- Considering that the surface power is uniform we have:

$$\widetilde{Q}_S = \int_S q_S dS = \int_S \text{div}\ \lambda \overrightarrow{\text{grad}} T dS = \lambda \int_C \vec{\nabla} T . \vec{n} dC = \widetilde{Q}_i + \widetilde{Q}_j + \widetilde{Q}_k$$

Where $\widetilde{Q}_i,\ \widetilde{Q}_j,\ \widetilde{Q}_k$ are the additional powers on the edges due to the surface power and which are solution of the following system:

$$\begin{cases} \widetilde{Q}_i = \int_{Ei} \lambda \vec{\nabla} T . \vec{n} dE_i = \alpha_i \widetilde{Q}_S \\ \widetilde{Q}_j = \int_{Ej} \lambda \vec{\nabla} T . \vec{n} dE_j = \alpha_j \widetilde{Q}_S \\ \widetilde{Q}_k = \int_{Ek} \lambda \vec{\nabla} T . \vec{n} dE_k = \alpha_k \widetilde{Q}_S \end{cases}$$

With , $\alpha_i + \alpha_j + \alpha_k = 1$

Wich can approximate on small regular triangle by: $\alpha_i = \alpha_j = \alpha_k = \frac{1}{3}$

- In the linear case we got that

$$\forall a \quad \begin{cases} Q_i = a(T_S - T_i) + b_{ij}(T_j - T_i) + b_{ik}(T_k - T_i) \\ Q_S = a\left(3T_S - T_i - T_j - T_k\right) \end{cases}$$

Now, we have demonstrated that in the quasi-linear case we have:

$$\exists a \quad / \quad \begin{cases} Q_i = a\left(\left(T_S + \frac{1}{3a}\varepsilon_S\right) - T_i\right) + b_{ij}(T_j - T_i) + b_{ik}(T_k - T_i) \\ Q_S = a\left(3.\left(T_S + \frac{1}{3a}\varepsilon_S\right) - T_i - T_j - T_k\right) \end{cases}$$

Power System Equation in a Quasi-Linear Case:
There exist a positive coefficient *a* so:

$$\underbrace{\begin{pmatrix} -3a & a & a & a \\ a & -\frac{a}{3} - g_{ij} - g_{ik} & g_{ij} - \frac{a}{3} & g_{ik} - \frac{a}{3} \\ a & g_{ij} - \frac{a}{3} & -\frac{a}{3} - g_{ij} - g_{jk} & g_{jk} - \frac{a}{3} \\ a & g_{ik} - \frac{a}{3} & g_{jk} - \frac{a}{3} & -\frac{a}{3} - g_{ik} - g_{jk} \end{pmatrix}}_{\text{Matrix E}} \cdot \begin{pmatrix} \widetilde{T}_S \\ T_i \\ T_j \\ T_k \end{pmatrix} = \begin{pmatrix} \widetilde{Q}_S = \varepsilon_S \\ \widetilde{Q}_i = Q_i + \frac{\varepsilon_S}{3} \\ \widetilde{Q}_j = Q_j + \frac{\varepsilon_S}{3} \\ \widetilde{Q}_k = Q_k + \frac{\varepsilon_S}{3} \end{pmatrix}$$

$$Q_i + Q_j + Q_k = 0$$

Matrix E
"Elementary Matrix"

General Problem Definition

- The general problem consist on integrating all powers on a surface to get the average temperature of the surface
- The total power equation is then:

$$Q_{total} = Q_{conductive\ from\ edges} + Q_{radiative} + Q_{surfacic\ conduction} + Q_{int/ext} + Q_{capacitive}$$

$$Q_{total} = \sum_{Ei} Q_{Ei} + Q_S = 0$$

Solving the general problem is then performed by submeshing the original geometrical element on which we create a detail conductive network.

Detailed Conductive Network

The conduction problem on the submesh can be written:

$$\begin{pmatrix} C_{tt} & C_{te} & C_{ti} \\ C_{te}^T & C_{ee} & C_{ei} \\ C_{ti}^T & C_{ei}^T & C_{ii} \end{pmatrix} \cdot \begin{pmatrix} T_t \\ T_e \\ T_i \end{pmatrix} = \begin{pmatrix} Q_t = \kappa_t Q_S \\ Q_e = \kappa_e Q_E \\ Q_i = 0 \end{pmatrix}$$

Matrix C: Detailed Conductive Network

With *{ t }* being the elementary triangle surfaces
{ e } the external edges of the elementary triangles
{ i } the internal edges

Reduced Conuctive Network

Let's integrate the DCN to get a set of couplings on the following nodes:

- Edge Average Nodes

$$E_i \qquad T_i = \frac{1}{l_i} \int_{l_i} T(\rho) d\rho$$

- Surface Average Node

$$S \qquad T_s = \frac{1}{A_i} \int_{A_i} T(\alpha) d\alpha$$

This integration can be approach by discretization:

$$\begin{cases} T_S = \sum_t \alpha_t T_t \\ T_{E1} = \sum_{e1} \beta_{e1} T_{e1} \\ T_{E2} = \sum_{e2} \gamma_{e2} T_{e2} \\ T_{E3} = \sum_{e3} \delta_{e3} T_{e3} \end{cases}$$

With

$$\alpha_t = \frac{a_t}{\sum a_t} \qquad \beta_e = \frac{l_{e1}}{\sum l_{e1}} \qquad \gamma_e = \frac{l_{e2}}{\sum l_{e2}} \qquad \delta_e = \frac{l_{e3}}{\sum l_{e3}}$$

So the average temperature can be wriiten

$$\underbrace{\begin{pmatrix} \alpha_1 & \cdots & \alpha_{nt} & 0 & & 0 & & 0 & 0 \\ 0 & & \beta_1 & \cdots & \beta_{n\beta} & 0 & & 0 & 0 \\ 0 & & 0 & & \gamma_1 & \cdots & \gamma_{n\gamma} & 0 & 0 \\ 0 & & 0 & & 0 & & \delta_1 \cdots & \delta_{n\delta} & 0 \end{pmatrix}}_{\text{Matrix A: Detailed to Reduced Definition}} \cdot \begin{pmatrix} T_t \\ T_e \\ T_i \end{pmatrix} = \begin{pmatrix} T_S \\ T_{E1} \\ T_{E2} \\ T_{E3} \end{pmatrix}$$

- The Detailed Conductive Network + the Detailed to Reduced Network can be expressed into a single matricial equation.

To simplify the notation, let's call *D as* the elements of the Detailed Network and *R* of Reduced one.

$$\begin{pmatrix} C & A^T \\ A & 0 \end{pmatrix} \cdot \begin{pmatrix} T_D \\ 0 \end{pmatrix} = \begin{pmatrix} Q_D \\ T_R \end{pmatrix}$$

- By inverting the equation we obtain:

$$\begin{pmatrix} T_D \\ 0 \end{pmatrix} = \begin{pmatrix} X & Y^T \\ Y & -Z \end{pmatrix} \cdot \begin{pmatrix} Q_D \\ T_R \end{pmatrix}$$

And particularly, we will focus on the following equation:

$$Y.Q_D - Z.T_R = 0$$

- Introducing the hypothesis that the surface flux is uniform, we can demonstrate that

$$Y.Q_D = Q_R$$

Leading to the Reduced Conductive Network:

$$Z.T_R = Q_R$$

Resolution of the RCN

The previous equation system leads to the RCN solution but it is in fact not determined. There is still a coefficient *a* necessary for writing correctly the matrix to invert that we don't know.
The error made on elementary average temperature due to a wrong coefficient is:

$$T_t^{error} \le k.\varepsilon_S = k.a_t.Q_S$$

Then the error made on the reduced average temperature is:

$$T_S^{error} = \sum_t \alpha_t T_t^{error} \le \sum_t \frac{k.a_t^2}{A_S}.Q_S$$

Since

$$\sum_t \frac{a_t}{A_S} = 1$$

We know that

$$\sum_t \frac{a_t^2}{A_S} \xrightarrow[a_t \to 0]{card(t) \to \infty} 0$$

So we have convergence of the RCN matrix whatever the initial coefficient *a* is when the submesh tends to infinitely small surfaces.

Simplified RCN method

The simplified RCN method is derived from the RCN to get shape-to-shape couplings. This topology of couplings gives an approximation of the conductive flux assuming piecewise linear temperature profile normal to each edge.

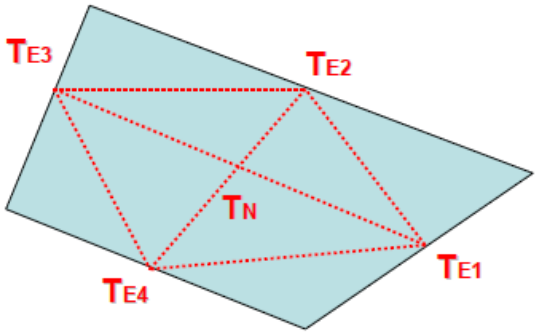
If the conductive fluxes are roughly approximated using a shape-to-shape coupling topology, it is often required in simple cases to use this simplified topology allowing a better understanding network.

The simplified RCN method leads to equivalent results obtained with the classic geometrical approach implemented in THERMICA v3. If this new method inherits of the same limitations than other shape-to-shape expression of the couplings (linearization of the temperature profile and directivity of the flux normal to the edges), it is able to handle non-conformances and condensation properly. Besides, the expressions of the couplings are no longer computed from empirical formulae but by simplifying the RCN results which are physical ones.

To get the simplify RCN couplings, THERMICA computes the real RCN solution for each node, then simplify its expression by introducing the hypothesis of a linear flux normal to one edge:

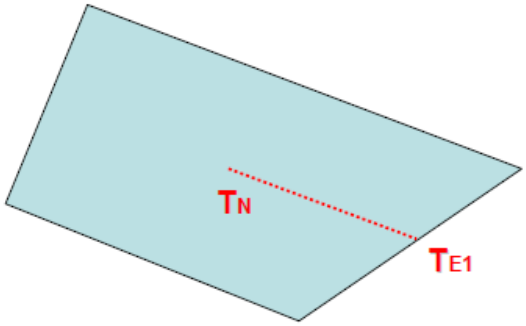
Step 1: Computation of RCN couplings and Expression of conductive flux to each edge:

$$\varphi_{Ei} = GL_{N,Ei}(T_N - T_{Ei}) + \sum_{Ej} GL_{Ei,Ej}(T_{Ej} - T_{Ei})$$



Step 2: Simplification of temperature gradients assuming a normal temperature profile:

$$T_{Ej} - T_{Ei} = \frac{\overrightarrow{E_i N}.\overrightarrow{E_i E_j}}{\left\|E_i N\right\|^2}(T_N - T_{Ei})$$

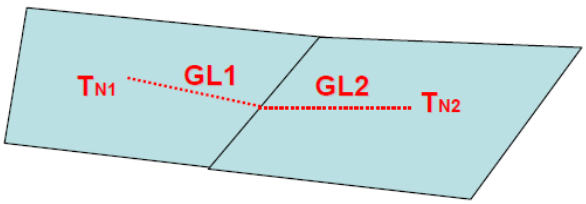


New expression of conductive flux to each edge:

$$\varphi_{Ei} = \left(GL_{N,Ei} + \sum_{Ej} \frac{\overrightarrow{E_i N}.\overrightarrow{E_i E_j}}{\left\|E_i N\right\|^2} GL_{Ei,Ej} \right)(T_N - T_{Ei})$$
$$= GL^{*}_{N,Ei}(T_N - T_{Ei})$$

Step3: Association of half couplings in series:

$$GL_{N1,N2} = \frac{1}{\frac{1}{GL1} + \frac{1}{GL2}}$$

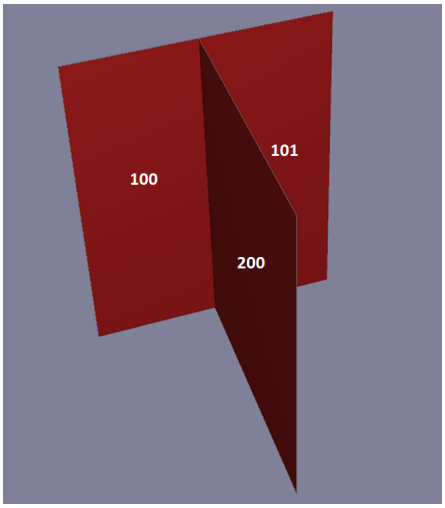
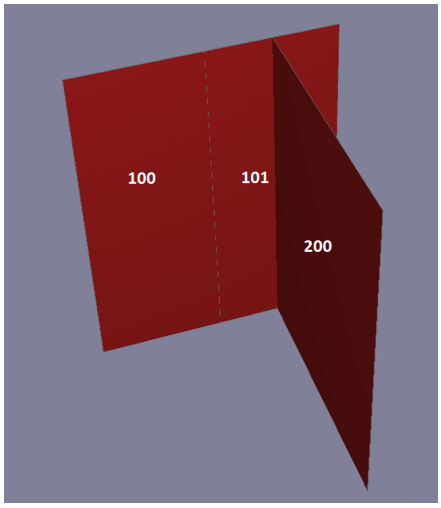


How to use the conduction module

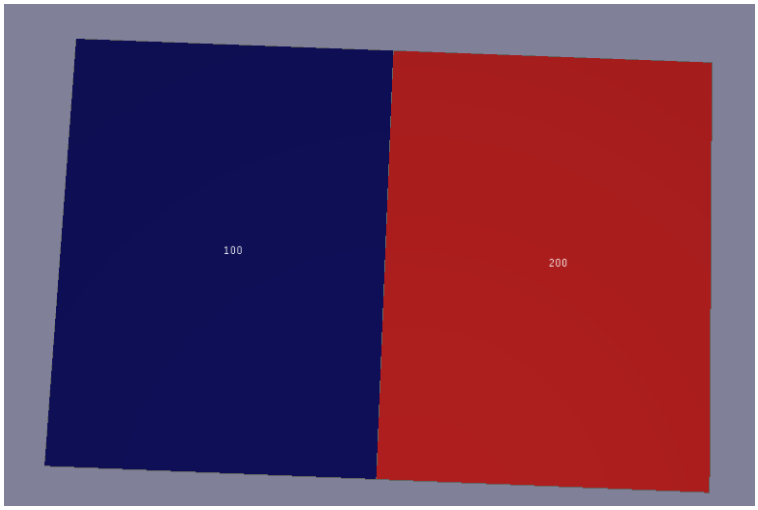
Scope of validity and limitations

- Conduction module algorithm doesn't take into account the boolean transformations (ie. cutters) for the computation of conductive couplings.
- Conduction module does not allow cutters. If a shape is modified by a cutter, the conduction contribution will not be computed throughout this shape.
- The Volumic revolved Quadrangle does not work well with the conduction module. It is therefore advised to not use this particular volumic shape while leading a conductive analysis.
- Do not use the conduction module on polygons.
- Conduction module does not allow to compute conductive couplings between two disjoined kinematic bodies.
- Conduction module does not allow to compute conductive couplings for a non-aligned mesh (with edge-surface contact).

Here is an example of a T-configuration:

Aligned meshes	**Non-aligned meshes**
	
Conductive couplings between nodes : ▪ 100 and 101 ▪ 100 and 200 ▪ 101 and 200	Conductive coupling between nodes : ▪ 100 and 101 No conductive coupling between 101 and 200

- Using distinct side nodes can lead to different results depending on the chosen options. Consider for example 2 shapes in contact with different material for each side of the shapes.



The following table summarizes the different possibilities:

Select distinct side in the model tab	**Select "Enable distinct side numbering" in the meshing tab**	**result in .nod.nwk file**	**result in .gl.nwk file (Simplified RCN method)**
yes	no	• 2 nodes : 100 and 200 • capacitance computation : $C_i = (V\rho c_p)_{i,side\ 1} + (V\rho c_p)_{i,side\ 2}$ *Note : a warning is written to the .nod.log file "Node i has more than 1 bulk material property (2)"*	• 1 conductive coupling : GL(100,200) • GL computation : $GL(i,j) = \left(\frac{GL_{iTOT}GL_{jTOT}}{GL_{iTOT} + GL_{jTOT}}\right)$ where $GL_{iTOT} = GL_{i,side\ 1} + GL_{i,side\ 2}$ and $GL_{i,side\ x} = \lambda_{i,side\ x}\frac{S_{i,side\ x}}{L_i}$
yes	yes	• 4 nodes : 100, 101, 200 and 201 • capacitance computation : $C_{i,side\ x} = (V\rho c_p)_{i,side\ x}$	• 2 conductive couplings : GL(100,200), GL(101,201) • GL computation : $GL(i,j)_{side\ x} = \left(\frac{GL_iGL_j}{GL_i + GL_j}\right)_{side\ x}$ where $GL_{i,side\ x} = \lambda_{i,side\ x}\frac{S_{i,side\ x}}{L_i}$

If in doubt, do not use distinct side nodes for a conduction computation.

How to choose the method (RCN or Simplified RCN)

- The use of Simplified RCN method is recommended for most cases. It allows to minimize the number of GL and nodes.
- RCN method has to be used on complex structures (complex heat flux path, non-orthogonal to the edges) or when high temperature gradient is expected. *Note: RCN method involves EDGES network format that can cause compatibility issues with Esatan and TMRT.*

WARNING:	
	Since Systema 4.9.2 the previous limitations on condensed nodes with Simplified RCN method have been solved. However when using the Simplified RCN method and when high temperature gradient is expected on non orthogonal shapes (triangles, quadrangles), it is recommended to finely mesh those shapes. Do not use the Simplified RCN method on Spheres. Moreover the use of the Simplified RCN method on Antennas, Extrusions, Revolutions can lead to inaccuracies or even errors in the calculation of couplings.

As a general rule, the conduction module is better suited to simple shapes such as rectangles. As far as possible, it is advisable to use simple rather than complex shapes for conduction calculations.

Conduction Parameters

Conduction Parameters

Input

- **.SYSSET file:** Mission synthesis for the THERMICA application.

Outputs

- **.COND.LOG file:** This file contains information about the process.
- **.EDG.H5 file:** Contains conductive edge topology in HDF5 format to be displayed in SYSTEMA.
- **.GL.NWK:** Conductive couplings and eventually edges expressed in the selected network format.
- **.BKW.H5 file:** Contains the backward matrix so to be able to retreive a detailed temperature map. This file may be used as an input of the mapping module or of the nodal description so to resolve sensor's links based on a backward RCn method.

Parameters

Conduction properties

Contact management

- **Edge Contact Threshold [m]:** Threshold for the contact detection. *Note: User specified Edge contact threshold can be locally modified if the size of the cell is too small compared to the contact threshold value. The min/max corrected contact threshold values are written to the ".gl.nwk" file.*

Outputs

- **Network format:** Set the format of the nwk output for THERMISOL, ESATAN or MSC/SINDA. If edges are created (RCN), they will be stored as EDGES in THERMISOL and as classical nodes on other formats.
- **Method:** Choose between the RCN and Simplified RCN methods.
- **Parametric couplings:** (RCN and THERMISOL only) If set to "Yes", couplings may be exported using conductivity variable names instead of values (variables are generated by the Nodal Description module).

Nodal Description Module

The nodal description module exports all geometrical nodes definitions as well as variables, material references and all specific items (excluding the edge contact resistance).

This documentation explains the theoritical background of the nodal method used.

Software parameters and inputs / outputs are also explained in a second part.

Nodal Description Overview

Nodal Description Overview

Introduction

The standard process used in space thermal analysis is the lumped parameter method also named nodal method, which is based on the knowledge of:

- **All the nodes**
- **All the interactions between nodes (energy transfers).**

A thermal problem is divided into several nodes, each of them representing an entity with a uniform temperature, which is computed by writing a heat balance for each node. This leads to a system of differential equations.
The approach is similar to electrical networks.

Nodal method

In a nodal scheme, the nodes are the unknown variables on which we want to compute the solution. A node is therefore an entity with a set of properties connected to its nodal environment by "couplings" and on which external conditions can also be applied.
A node can represent a shape, a group of shapes or a subpart (mesh) of a shape on which the temperature is assumed to be isothermal (only one temperature will be computed for the node, i.e. the mean temperature of the node's geometrical representation).
For a conductive computation, as a temperature gradient is necessary to compute heat fluxes, nodes representing the edges can be added.

	*See Chapter 5 "Conduction Module" for detailed information.*

Computation algorithm

The nodal description is a translation of the geometry into a nodal network. It uses the thermal numbering applied to the geometry and computes, for each node, a set of properties:

- **The Area (geometric or radiative)**
- **The Capacitance**
- **the formula for the capacitance is:**

C = Specific Heat . Density . Thickness . Area

- **The Alpha and Epsilon thermo-optical properties:**
- **If a node is a shape active on only one side, the alpha and epsilon are the values of the coating.**

- **If a node is a shape active on both sides, the alpha and epsilon are the mean of the values of the front and back coating.**
- **If a node is a combination of shapes, the returned values are the mean weighted by the area of all the values computed for each shape (according to points a and b).**

Note: a node set to inactive in the bulk and coating tabs will be commented in the nod.nwk file. The label of non-geomtrical nodes is limited to 24 characters.

Nodal Description Parameters

Nodal Description Parameters

Inputs

- **.SYSSET file**
  Mission synthesis for the THERMICA application

Outputs

The following two files are used to process the results in the Modeler or Mission tabs of SYSTEMA:

- **.NOD.LOG file**
  Log file
- **.NOD.NWK file**
  Nodal description expressed in the selected network format

Parameters

The parameters are splitted into 2 general categories: Nodal Description properties (local to the current module) and Common properties (global to the complete diagram).

Common properties

Mesh filter

- **Filter small area cut meshes:** This option is used to enable or disable the filtering of meshes partially cut by a cutter and whose resulting area is very small. By default, the filtering is activated to be compliant with the behavior of the previous versions (before 4.9.2). When the filtering is activated, a partially cut mesh is filtered when its resulting area is lower than 0.1% of the initial mesh area. In that case, the corresponding thermal node is not defined by Thermica. This option is also available from all the other Thermica modules in Common properties.

Nodal Description properties

Outputs:

- **Network format:**
  Set the format of the nwk output for THERMISOL, ESATAN or MSC/SINDA

- **Export geometrical position:**
  Creates new Entities FX, FY and FZ so to store the geometrical position of the node centers
- **Export geometric areas:**
  Creates a new Entity S so to store the geometrical areas (in adittion to the radiative areas A)
- **Export geometric volumes:**
  Creates a new Entity V so to store the geometrical volumes (for volume nodes)

Convection Module

The convection module computes the thermal fluid coupling between the geometrical nodes of a model and a fluid node, for natural convection, using McAdams correlations for flows on walls.

The generated couplings are exported in the runcase.gf.nwk files.

This documentation explains the theoretical background of the convective couplings computation.

Software parameters and inputs / outputs are also explained in a second part.

Convection Overview

The Convection module describes the convective exchanges between the geometrical nodes of the model, and fluid nodes. The module uses the parametric or Mac Adams correlations to compute conducto-convective couplings to model flows along walls.

Introduction

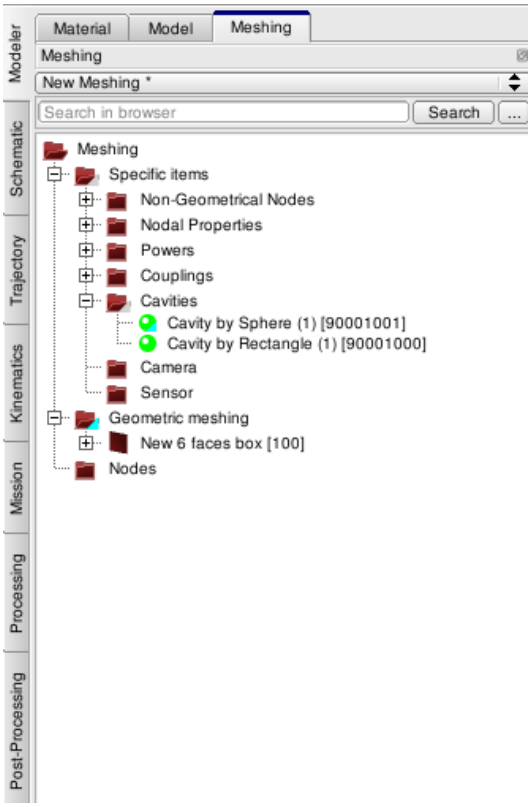
This module shall be used when a fluid has to be accounted for. Indeed, the motion of the fluid can bring new matter to the wall, extending the thermal exchange with the wall not only to conduction but also to convection (the thermal exchange between the fluid and a given wall leads to a natural motion of the fluid).

The heat transfer by convection depends of the thermal properties of the fluid (Fluid dynamic viscosity, Specific heat capacity, Fluid density, Thermal expansion, Temperature, and Volume) and the Wet Surface of the wall.

How to use the Convection module

Cavity by Sphere & Rectangles

In order to use the convection module, the user needs to define fluid nodes. This is done in the Cavities section, in the Specific items in Meshing tab of the Modeler, by creating a specific item called Cavity by sphere or Cavity by rectangle. In order to be fully functional, the cavity has to be contained in a given shape. If several fluid nodes are defined in a cavity and in contact with each other, only the last node defined in the shape will be taken into account, taking over the other nodes. A way around this limitation is to use cavity closure. Note that even though the algorithm detects all cavities available in the computational domain, only the cavities defined by the user will be acknowledged for the computation of the conducto-convective couplings.



In the Thermica tab of a given cavity, the following fluid parameters can be defined:

- Fluid dynamic viscosity [kg/m/s]
- Specific heat capacity [J/Kg/K]
- Fluid thermal conductivity [W/m/K]
- Fluid density [Kg/m3]
- Thermal expansion (beta) [K-1]

The following coupling parameters can be defined:

- Method (parametric or McAdams)
- K coefficient (used for parametric method only)
- Alpha coefficient (used for parametric method only)

The following Air node parameters can be defined:

- Status (Boundary or Diffusive)
- Temperature [K]
- Volume [m3]

Cavity closure

Cavity closure is a specific item allowing to close an open cavity/shape. It will act as a wall, but only for convection purpose. This feature can be used in case stratification is needed, to define several fluid nodes in one cavity. With cavity closure, the fluid nodes will not be in direct contact with each other, and therefore will be acknowledged as different nodes by the algorithm.

Computation of Wet Surfaces

The Wet Surface is the total area of contact between the fluid and the geometrical node.

In order to compute the Wet Surfaces, the algorithm will first divide the volume in 3D cells, called voxels. If a voxel is in contact with a surface, it will be considered as a full voxel, and as an empty voxel otherwise.

Then the algorithm will agglomerate all empty voxels, and look for cavities defined by user to attach the voxels to itself. Once the voxels of the cavities are identified, the Ray tracing method is used to compute the percentage of the cell that sees directly the cavity : If an empty voxel is reached directly by the ray emitted, this means that the surface is in contact with this fluid node.

The Wet Surface corresponds to the area of the shape multiplied by the view factor resulting from the Ray tracing for this shape.

Two parameters can be used to modify this behavior : the Hole Threshold and the Contact Threshold.

Parametric Computation

The parametric correlation will compute the Conducto-Convective couplings between each geometrical nodes and each fluid nodes as follow :

GL (Cavity_Node , Geometrical_Node) = Cavity_Node_K * Wet_Surface_Cavity_Node _ Geometrical_Node*(abs(Cavity_Node_T - Geometrical_Node_T))Cavity_Node_Alpha**

Where Cavity_Node_K, Cavity_Node_Alpha are the K coefficient and Alpha coefficient defined in the Thermica Tab, Cavity_Node_T, Geometrical_Node_T are the cavity and geometrical node temperatures, and Wet_Surface_Cavity_Node _ Geometrical_Node is the total area of thermal exchange between the fluid and the geometrical nodes.

Mac Adams Correlations

The conducto-convective coupling using the Mac Adams correlations are defined as follow:

GL (Cavity_Node , Geometrical_Node) = Wet_Surface_Cavity_Node _ Geometrical_Node * MCADAMS()

The Mac Adams function is computed as:

MCADAMS = Cb * (GPNb) * Lm / Lc**

Where:

- Cb and Nb are constants depending on the orientation of the wall and if the fluid flow is laminar or turbulent:
  - For a Horizontal wall (lower face) : Cb = 0.14 , Nb = 0.25
  - For a Horizontal wall (upper face) - Laminar mode : Cb = 0.54 , Nb = 0.25
  - For a Horizontal wall (upper face) - Turbulent mode : Cb = 0.14 , Nb = 0.33
  - For a Vertical wall - Laminar mode : Cb = 0.555 , Nb = 0.25
  - For a Vertical wall - Turbulent mode : Cb = 0.129, Nb = 0.33
- GP is the Rayleigh number, the product of the Grashof (Ghf) and Prandtl (Pr) numbers
  - $$\mathrm{Pr} = \frac{\mu C_p}{\lambda}$$
  - $$Gr = \frac{\beta g (T_p - T_f) \rho^2 L^3}{\mu^2}$$
- Lm is the Fluid thermal conductivity
- Lc is the characteristic length (computed as Area**0.5) - *Please note that this is an interpretation, and it can be changed by the user in the .gf.nwk file.*

Convection Parameters

Input

- **Mission set - .SYSSET file:** Mission synthesis for the THERMICA application

Outputs

- **LOG - .COND.LOG file:** This file contains information about the process
- **Conducto-Convective couplings - .H5 file:** Contains conducto-Convective couplings in HDF5 format to be displayed in SYSTEMA
- **Conducto-Convective couplings - .GF.NWK:** Conducto-Convective couplings and Mac Adams correlation function expressed in the selected network format (ESATAN or Thermisol).

Parameters

The parameters are split into 2 general categories: Convection properties (local to the current module) and Common properties (global to the complete diagram).

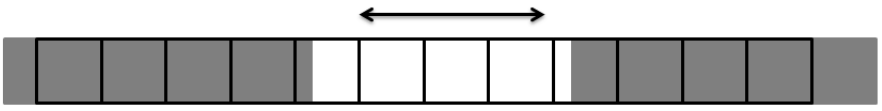
Common properties

The Common properties here should not impact the convection module.

Convection properties

Convection parameters

- **Voxel sizes [m]:** The number of voxels used for the computation of the Wet Surfaces is computed via the Voxel sizes and the size of the computational domain. *Note: the limit minimal size of voxel fixed by the software is 0.01 for user input. On top of this restriction, the software will reduce the number of voxels if the resulting one is too big. Check the .conv.log file for case by case details.*
- **Hole Threshold [m]:** Gives the minimal size for Holes to be taken into account. If the Hole Threshold is less than the actual hole, the voxels corresponding to the hole will not be considered as empty, and will form a natural cavity closure.

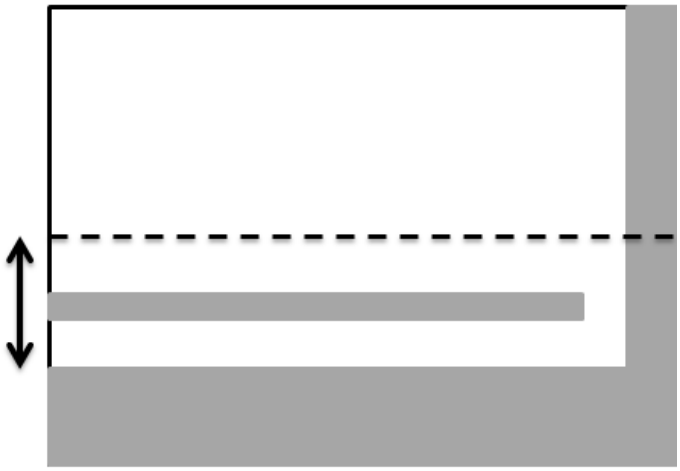


Case 1

Case 2

Exemple of Hole Threshold : Case 1 is where the hole threshold corresponds to 3 voxels, and the cavities can be merged, and Case 2 where the threshold is bigger than the actual hole, and the voxels are not considered empty, thus creating closure.

- **Contact Threshold [m]:** Threshold to avoid shadowing between two close surfaces- allows the wetting of these two close surfaces, where the distance between the two surfaces shall be less than the Contact Threshold.



Exemple of a contact threshold : Even if the two surfaces are close, the lower one will see the cavity, as the rays emitted from this surface will take into account all that is past the Contact Threshold.

- **Gravity:** Gravity vector for each directions in the main frame.

Ray tracing

- **Random seed:** Is used to modify the generation of pseudo random numbers used by the ray-tracing. This option may be modified to check the results sensitivity to the random process.
- **Ray number per node:** Is used to modify the number of rays emitted per node. This option may be modified to check the results sensitivity to the random process for the computation of the Wet Surfaces.

Outputs

- **Network format:** Set the format of the nwk output for THERMISOL or ESATAN.

Emission Source Module

The Emission Source module enhances the simulation of satellite testing by accurately modeling the UV and IR fluxes from UV/IR lamps and their absorption at various geometrical nodes of the model. This capability facilitates the preparation and optimization of physical tests, leading to more precise and effective evaluations.

This section details the theoretical background of the computations performed by the module and the scope of its validity. Additionally, software parameters, inputs, and outputs are explained in a subsequent part.

Emission Source Overview

Introduction

The Emission Source module simulates the thermal flux exchanges between UV/IR lamps and the geometrical nodes of the model.

This module is designed to fully replace the "Solar Lamp" specific item, whose outputs are generated by the Nodal Description module. Although the "Solar Lamp" specific item has been retained for compatibility reasons, it is highly recommended to use the Emission Source module now, as it offers at least the same capabilities and even more due to additional functionalities and fewer limitations. Additionally, this is a standalone module, which allows the generation of various dedicated output files such as .nwk, .h5 and files for ray visualization. The Emission Source module operates with a new specific item named "Point Source", which enables the placement of UV/IR lamps on the surfaces of the model. The user needs to create at least one Point Source specific item to be able to run the Emission Source module.

Below is a table summarizing the differences between the capabilities of the "Solar Lamp" and the new "Emission Source" module:

Feature	Solar Lamp specific item	Emission source module
Multiple lamp simulation	Not supported	Supported
Dual emission capability (UV & IR)	Either UV or IR	UV, IR or both UV and IR simultaneously
Parametric flux definition	Not available	Available in .nwk file
Versatile ray tracing	Ray emission in normal direction only	Ray emission in all directions (Lambert's law) or normal direction
3D ray visualization	Not available	Supported
Data Storage	Fluxes in nwk file	Fluxes in nwk file Fluxes and couplings in h5 file
Time-Dependent emitted flux (user input)	Not available	Supported (only in the .nwk output, not in the emi.h5 file)

By leveraging these enhanced features, the Emission Source module provides a more comprehensive and flexible tool for modeling and predicting satellite test outcomes.

Flux computation

The calculation utilizes a Quasi Monte Carlo ray tracing method. While similar to the Radiation module in principle and methodology, this module exclusively calculates radiative couplings between the lamp nodes and the model's nodes. Additionally, the rays are fired unidirectionally, from the lamp to the model's nodes, resulting in non-symmetrical couplings (meaning that in this module, the software does not force the reciprocity law to be verified). Therefore, a greater number or rays need to be fired to ensure converged calculations.

Following the coupling calculations, the Emission Source module determines the exchanged UV and IR fluxes between the lamp and the model's nodes based on the UV and IR fluxes emittted by the lamp.

The main computation steps are as follows:

- Rays are emitted from each node of each lamp defined by a Source Point specific item:
    - Emission points are sampled randomly
    - Emission direction depends on the option chosen when defining the specific item Point Source. It either follows Lambert's law or is normal to the emission surface.
- The ray tracing method of REF calculation is used to compute the radiative couplings between the lamp's nodes and the model's nodes. Please refer to the "Radiation module" chapter for more details.
- The radiative couplings are multiplied by the emitted UV and/or IR flux (defined by the user in the Point Source specific item) to determine the UV and/or IR fluxes absorbed by each node of the model.
- The UV and IR fluxes and couplings are stored in the .emi.h5 file. The fluxes are also output in the .emi.nwk file:
    - The UV fluxes are stored in the nodal entity QS (generally used to store solar flux)
    - The IR fluxes are stored in the nodal entity QE (generally used to store IR planet flux)

Please note that using the Emission Source module for UV/IR lamp modeling is incompatible with the Solar Flux and Planet Fluxes modules, as they both rely on the same nodal entities for result storage (QS and QE), which could result in overwriting if used together. These modules are not intended to be used simultaneously, so this should not present any problems. The Emission Source module is designed for satellite testing scenarios, whereas the Planet Fluxes and Solar Fluxes modules are meant for simulating orbital conditions.

For this module, the definition of an orbit and pointing is not required (but is possible). In any case, the computation is performed only at the first orbital position. This means that the moving bodies are not supported by this module. The emission source module can be executed in the presence of moving bodies, but the fluxes calculated and provided in the output files will correspond only to those of the first orbital position.

Module's current constraints

While the Emission Source module provides a robust set of features for various simulations, there are certain limitations that users should be aware of:

1. Wavelength dependency not supported

The module does not support wavelength dependency, meaning that calculations cannot account for thermo-optical properties defined with a wavelength dependency.

2. Moving bodies not accounted for in calculations

The presence of moving bodies is not considered in the calculations. The fluxes calculated and provided in the output files correspond only to those of the initial moment.

3. Static results in H5 file

Results in the H5 file are static and represent only the initial moment, regardless of moving bodies or time-varying emitted fluxes.

4. Incompatibility with Solar FLux and Planet Fluxes modules

This module cannot be used simultaneously with the Solar Flux and Planet Fluxes modules because they both store their results in the same nodal entities (QS and QE). For further details, please refer to the "Flux computation" section.

5. Reciprocity law not enforced

The module does not enforce the reciprocity law, as it only traces rays from the lamps to other nodes and does not perform reciprocal tracing. Consequently, to achieve accurate and convergent results, it is necessary to trace a larger number of rays.

Emission Source Parameters

Input

- **.SYSSET file:** Mission synthesis for the THERMICA application.

Outputs

- **.EMI.LOG file:** This file contains information about the process.
- **.EMI.H5 file:** Contains UV and/or IR fluxes and couplings results in HDF5 format.
- **.EMI.NWK file:** UV and/or IR fluxes absorbed by each nodes and expressed in Thermisol network format.
- **.UVERT.H5 file:** (optionnal) Contains all computed UV rays for display in SYSTEMA.
- **.IRERT.H5 file:** (optionnal) Contains all computed IR rays for display in SYSTEMA.

Parameters

The parameters are splitted into 2 general categories: Emission Source properties (local to the current module) and Common properties (global to the complete diagram).

Common properties

From the **Common** properties, the Emission Source module will only be influenced by the mesh filter option:

- **Filter small area cut meshes:** This option is used to enable or disable the filtering of meshes partially cut by a cutter and whose resulting area is very small. By default, the filtering is activated to be compliant with the behavior of the previous versions (before 4.9.2). When the filtering is activated, a partially cut mesh is filtered when its resulting area is lower than 0.1% of the initial mesh area. In that case, the corresponding thermal node is not defined by Thermica. This option is also available from all the other Thermica modules in Common properties.

Emission Source properties

Ray tracing

- **Random seed:** Is used to modify the generation of pseudo random numbers used by the ray-tracing. This option may be modified to check the results sensitivity to the random process.
- **Ray threshold:** This option is used to filter the ray bounces according to the remaining energy of a ray. By default, this filtering is done whenever a ray has lost at least 99% of its energy (1% of remaining energy). In such cases, the remaining energy is considered as absorbed by the last impacted node.
- **Maximum ray bounces:** This option is used to stop a ray propagation in the case of too many bounces. It allows in fact to detect a model failure, such as an active closed cavity with an epsilon or alpha value of 0 (theoritically leading to infinite ray bounces). Whenever rays are being filtered due to maximum bounces reached, the total filtered energy from a node is given in the log file for both IR and Visible contributions so to check if energy levels are significant or not.

Critical / Normal / Low

- **Ray building method:** Select "Per node" / "Per mesh" / "Density". The default "Per node" value indicates that each node will use the given number of rays. In the case of condensed nodes, this number of rays is distributed to the node's mesh according to their respective area (i.e. providing a uniform density of rays into the node). The "Per mesh" value may be used to specify the number of rays per radiative mesh. A condensed node will then use as many more rays as there are condensed radiative mesh into the node. For computation performances, the total number of rays that shall be used for a node is re-distributed on the mesh so to keep a uniform density of rays. This option shall be avoided when using CAD geometries because of those are build from a geometrical tesselation (and so with lots of mesh).

Outputs

- **Network format:** Set the format of the nwk output (only the THERMISOL format is available for this module).
- **Parametric fluxes:** If set to "Yes", fluxes may be exported in the nwk using emitted flux variable names instead of values (variables are generated by the Emission Source module). This option lets users update the emitted flux values of the lamps conveniently (by hand in the .dck file), avoiding the necessity to rerun the coupling calculations.
- **Ray display length:** Set the length of ray display to space. By default 2x the size of the model.

STEP-TAS Interface

STEP-TAS Interface

Introduction

STEP-TAS* is a neutral format for exchanging thermal analysis models and results. The development of this format has been coordinated by ESA.
STEP-TAS is a protocol (a formal data model) containing four modules:*

- MGM:Meshed Geometric Model
- NRF:Network model & Results Format
- SKM:Space Kinematics Model
- SMA:Space Mission Aspects

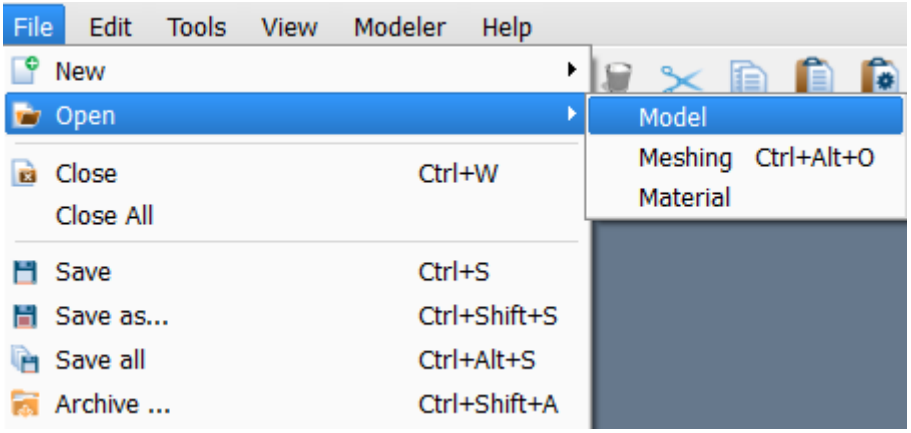
SYSTEMA-THERMICA contains a direct import/export to STEP-TAS (MGM/NRF) through an interface developed under an ESA contract. This new interface allows exchanging the geometrical model with its physical properties and nodal breakdown.
Moreover an export of the THERMISOL results from its native H5 format to STEP-TAS (NRF) will be soon available. This will also allow the exchange of results and the possibility to use post-processing tools based on the STEP-TAS format such as ESATAP (more details on this tool can be found on the ESA website).

A complete description of STEP-TAS can be found on ESA web site http://www.esa.int/

Meshed Geometric & Network Model Interface

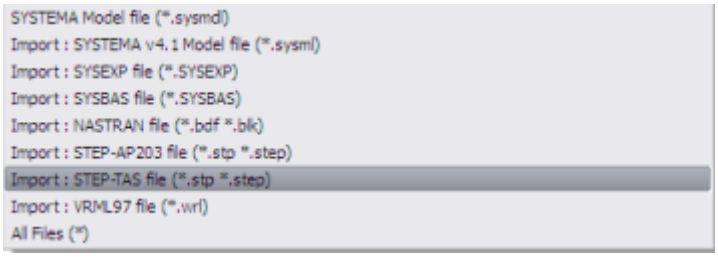
STEP-TAS Import

From the Systema main window, under the condition that THERMICA is installed, go to the *File* menu then choose the option *Open*.
You can also open the STEP-TAS file from the *Open-Meshing* submenu. In both cases, the import will produce a SYSTEMA model (sysmdl format) and its associated meshing (sysmsh format).



Set the format to the STEP-TAS and choose your file (that should have the appropriate extension).

Note that SYSTEMA also has an STEP-AP 203 import which is dedicated to CAD geometry (Compouted Assisted Design). Make sure to select the correct format.



Note that SYSTEMA has an inheritance management which allows setting the properties on a group of object only once. In the case of a STEP-TAS import, all properties are local settings assigned on the shapes. If inherited definitions are preferred, it is the responsibility to the user to make a "delete properties" in a subpart of the model in order to assign the properties only once to the father object.
At the end of the import process, the user will be asked for displaying the import log report. This file contains general warnings about not fully compatible features and specific information on shapes or objects concerned. The log file is by default not saved but you can choose to save under any name.

When importing a STEP-TAS file you can uncounter some meshing errors on numbering parameters in the log. The **Authorize condensation** option allows you to authorize that two meshes have the same number. Meshes with the same number are considered like one, so this option must be used carefully.

Incompatibilities between STEP-TAS and SYSTEMA

- Lengths and angles are converted to meter and degrees. The other quantities are valued as in SI units without conversion.
- In SYSTEMA, colors can only be set on objects. As a consequence, all color definition on shapes are transferred to their father object.
- Sub-model name (in numbering parameters) are not supported
- Cutters : only finite cylinder, box & triangle prism are supported. Infinite solid by plane cutter is managed by extruded polygon cutter that automatically « grows » if shapes are added to the model. This « growing » property is not displayed in the graphical user interface. If a shape and cutter are at the same level in the tree with different transformations, the 3D display of the cut can be glitched. Resolving transformations on one of them solves the problem.

Remark on the meshing (for version 4.3.3 and above):
By default, SYSTEMA does not authorize node condensation unless it is specified by the user (this behavior is to prevent from having undesired overlapping numbering). When importing a STEP-TAS file (as for a SYSEXP import), the mesh structure is notified with errors. To suppress those, simply add *Numbering Parameters* on the *Geometric meshing* object then check the box *Authorize condensation*. This will add default numbering rules to the entire tree, modifying intermediate object and shapes numbers but not the mesh ones (node numbers are locally set on the shapes, overloading any previously defined rule).

STEP-TAS Export

To export a model with its nodal breakdown under the STEP-TAS format, choose the option *Save As* from the *File / Meshing* list.
As for the import, a log file can be displayed.

Model can be exported from the modelling tab and the meshing tab. If exported from the modelling tab only the model and the material will be exported (as several meshings can be created from a single

model), a warning will be raised to inform you that the meshing was not exported . To export the model, the material and the meshing you must export it from the meshing tab.

Note: If you have checked the "reverse" option on some shapes only the following shapes will have the correct orientation after the export : triangles, rectangles, discs, boxes and quadrangles. Other shapes will be exported as not reversed.

Incompatibilities between SYSTEMA and STEP-TAS

- All physical properties and colors are transferred to the shapes (no inheritance in the STEP-TAS file)
- Transformations are translated to cosines matrix
- Transverse properties of material are not supported
- STEP-TAS is very strict when defining quadrangles, an option in the settings can be enabled/disabled to correct the coplanarity automatically before exporting the model in STEP-TAS format. It will apply the same correction as the script “correctQuadrangleDNVersion.py” available in the installation directory of Systema “python\publicScripts\”.
- Cutters : only finite cylinder, box & triangle prism shapes are supported. If an infinite solid by plane cutter was imported it will be exported as an infinite solid by plane cutter as well.
- STEP-TAS does not support different material for UV and IR thermo-optical properties, a warning is raised if different materials are used : "Step-TAS does not support different material for UV and IR thermo-optical properties: a new material has been created."

Batch commands

SYSTEMA provides a batch mode on which it is possible to perform several operations like executing a process, loading or converting a file.
Here are presented some examples of batch command related to the STEP-TAS format. More details on SYSTEMA batch commands can be found on the SYSTEMA User's Manual.

- **Loading directly a STEP-TAS file**

Windows: [path]\SystemaWIN.exe –load *path\file.stp*
Linux: [path]/SystemaLNX –load *path/file.stp*

- **Converting a STEP-TAS file to SYSTEMA files**

Windows: [path]\SystemaWIN.exe –convert *path\file.stp* –to *path\file.sysmsh*
–type MESHING [-log *path\file.log*]
Linux: [path]/SystemaLNX –convert *path/file.stp* –to *path/file.sysmsh*
–type MESHING [-log *path/file.log*]

- **Converting SYSTEMA files to a STEP-TAS file**

Windows: [path]\SystemaWIN.exe –convert *path\file.sysmsh* –to *path\file.stp*
–type MESHING [-log *path\file.log*]
Linux: [path]/SystemaLNX –convert *path/file.sysmsh* –to *path/file.stp*
–type MESHING [-log *path/file.log*]