MATLAB ESATAN POST-PROCESSING TOOL

README

Contact: olesjakubb@gmail.com

Last update: 28.08.2018

1. Overview

MATLAB ESATAN POST-PROCESSING TOOL is a software dedicated to fast post-processing of Esatanoutput binary analysis files *.tmd.

It is fully written in Matlab and its goal is to provide a user with a set of methods that speed up obtaining valid thermal engineering data on graphs and reports. It also allows for further data investigation and addition of new valuable routines.

The main purpose for development of the tool is to deliver a fast substitute for time-consuming processing of most common tasks in ThermNV software.

2. Software

The software is written in object-oriented style in order to hide implementation details from the user and to provide an interface to interact with internal logic without the need for understanding underlying calculus.

All functions are described within their source files, including inputs, outputs and usage.

The functions that are available for the user:

-->tmd = ESATAN_TMD_PostProcessing(path)

constructor of the object (in this case named "tmd"). The "path" leads to the *.tmd file we want to process.

-->tmd.displayInfo()

shows the structure of the *.tmd file with fields and properties.

--> attribute = tmd.getAttribute(attributeNameString)

returns a specified attribute (or a list of attributes possible to obtain).

ex. 1.

>>tmd.getAttribute()

% returns a list of attributes:

List of attributes to get:

- times
- conductorsGL
- conductorsGR
- thermalNodes
- thermalNodesRealAttributes
- conductorDataGL
- conductorDataGR
- thermalNodesRealData
- thermalNodesStringData

Quick explanation:

Attribute	**Description**
times	Time array of *n* time instants in [s].
thermalNodes	Array of nodes, sorted by node number
conductorsGL	2D array of pairs of nodes creating a GL conductors. The nodes are referred to byindices in thermalNodes array, ex.: thermalNodes = [2000, 2001, 4000, 4001] conductorsGL = [1, 2; 3, 4], which means, that two conductors are: GL(2000, 4000) and GL(2001, 4001).
conductorsGR	2D array of pairs of nodes creating a GR conductors. The nodes are referred to by indices from thermalNodes array.
conductorDataGL	2D array of size conductivity between all GL conductors, referred to by indices in conductorsGL. The size is: conductorDataGL (length_of_conductorsGL, length_of_time_instants)
conductorDataGR	2D array of size conductivity between all GR conductors,

	referred to by indices in conductorsGR. The size is: conductorDataGR (length_of_conductorsGR, length_of_time_instants)
thermalNodesRealAttributes	1D array of node attributes, that appear in thermalNodesRealData.
thermalNodesRealData	3D array of values of node attributes in time. The size is: thermalNodesRealData (length_of_thermalNodesRealAttributes, length_of_thermalNodes, length_of_times)
thermalNodesStringData	2D array of node type and node label for each node from thermalNodes.

ex. 2.

```
>>tmd.getAttribute('thermalNodesRealAttributes')

        % returns the meaning of attributes in thermalNodesRealData:

        ans =
           'Temperature                '
           'Capacitance               '
           'Total_Albedo_Heat_Source      '
           'Total_Earth_Heat_Source       '
           'Total_Internal_Heat_Source    '
           'Total_Rest_Heat_Source        '
           'Total_Solar_Heat_Source       '
           'Area                      '
           'Solar_Absorptivity            '
           'Infra-Red_Emissivity          '
           'Incident_Albedo_Heat_Source   '
           'Incident_Earth_Heat_Source    '
           'Incident_Solar_Heat_Source    '
           'X_Coordinate              '
           'Y_Coordinate              '
           'Z_Coordinate              '
```

-->heatFlow = getHeatFlowForNodes(conductorType, node1, node2)

returns heat flow 'GL' or 'GL' for two specified nodes.

-->drawTemperatureForNodes(nodes, saveFlag)

creates n charts with temperatures of all nodes belonging to the groups specified in 'nodes'.

'nodes' are specified in this form:

```
nodes = {
   'PX_PANEL',          [2100:2149, 3100:3149], [];
   'PY_PANEL',          [2002, 3002], [3000];
   'PZ_PANEL',          [2000, 3000], [];
   'MY_PANEL',          [2003, 3003], [];
   'MZ_PANEL',          [2001, 3001], [];
   'SOLAR_PANEL',       [5000:5005, 6000:6005], [6002, 6005];
   'RADIATOR',          [2500:2549, 3500:3549], [];
};
```

-->groupHeatFlow = getHeatFlowForGroupsOfNodes(conductorType, nodeset1, nodeset2, timestep)

returns heat flow between two sets of nodes.

-->drawGroupHeatFlow(conductorType, nodes1, nodes2, saveFlag)

creates a chart of heat flow between two sets of nodes.

--> [QS, QA, QE] = getEnvironmentalFluxesSumForOneSetOfNodes(nodeset)

returns environmental fluxes (Sun, Albedo, Planet(Earth)) for one set of nodes.

-->drawEnvironmentalFluxesSum(nodes, saveFlag)

creates one chart with environmental fluxes for each node set in 'nodes'.

--> [QI, QR] = getUnitHeaterFluxesSumForOneSetOfNodes(nodeset)

returns unit and heater fluxes for one set of nodes.

-->drawUnitHeaterFluxesSum(nodes, saveFlag)

creates one chart with unit and heater fluxes for each node set in 'nodes'.

--> logical = isConductorDataConstant(conductorType)

returns True if conductivity of all conductors of type 'conductorType' is constant.

-->runSelfDiagnostics()

function for further developers - see how methods were validated during development and do not repeat whole work!

-->showEsatanConstants()

returns Stefan Boltzmann and Absolute Temperature constants used in calculations.

--> [temperatureRanges, details] = getTemperatureRanges(nodeset)

returns temperature ranges of specified node set with some additional info.

-->generateTemperatureRangesReport(nodes, limit, saveFlag)

generates Excel file with temperature ranges, safe limits and details for all specified node sets in 'nodes'.

--> [nodesNotAssigned] = getNotAssignedNodes(nodes)

returns nodes that are not specified in 'nodes' but exist in ESATAN *.tmd file

--> [nodesRepeated] = getRepeatedNodes(nodes)

returns nodes that repeat in 'nodes'

-->drawHeatFlowBalanceForOneSetOfNodes(nodes, nodesRawNumber, saveFlag)

creates charts with heat flows between all defined nodesets and chosen nodeset

For more help with names, calculations and meaning of data being returned from functions, please, refer to Esatan and ThermNV manuals.

3. Enjoy and feel free to add new features!