The OpenSMOKE++ kinetic pre-processor

In order to preprocess a kinetic scheme, the `OpenSMOKE++ Suite` provides the `OpenSMOKEpp_CHEMKIN_PreProcessor` utility. The user has to supply the files containing the thermodynamic data, the kinetic mechanism, and (optionally) the transport data. For the simulation of ideal reactors, usually the transport data are not needed and therefore the user could choose to preprocess only the thermodynamic and the kinetic data. This is useful, since in many cases the transport data are not available. In order to run the `OpenSMOKEpp_CHEMKIN_PreProcessor` utility, the user has to write an input file containing the instructions (i.e. the dictionary) to perform the preprocessing and/or additional useful operations (checking of the thermodynamic properties, post-processing of reaction rates, etc.). The rules to write the dictionary are reported in Section below. Please look at the tutorials to have more details. The `OpenSMOKEpp_CHEMKIN_PreProcessor` utility can be run from the command line using the following instructions, where it is assumed that the file containing the dictionary is called `myinput.dic` and the dictionary `mydictionary`:

1. in Linux

```
OpenSMOKEpp_CHEMKIN_PreProcessor.sh --input myinput.dic --dictionary mydictionary
```

2. in Microsoft Windows

```
%OPENSMOKEPP_EXE_FOLDER%\OpenSMOKEpp_CHEMKIN_PreProcessor.exe --input myinput.dic
  --dictionary mydictionary
```

Please, consider that the instruction reported above must be written in a single line. You can use multiple lines using the / symbol (Linux) or the ^ symbol (Microsoft Windows).

Notes

1. The `--input` option can be omitted. In this case the `OpenSMOKEpp_CHEMKIN_PreProcessor` assumes that the input file is called `input.dic`
2. The `--dictionary` option can be omitted. In this case the `OpenSMOKEpp_CHEMKIN_PreProcessor` assumes that the dictionary is called `CHEMKIN_PreProcessor`
3. In addition to several ASCII files (depending on the options specified in the dictionary), a `kinetics.xml` file will be generated by the `OpenSMOKEpp_CHEMKIN_PreProcessor`. This is the only file needed by the `OpenSMOKE++ Suite` solvers.

Dictionary: CHEMKIN-PreProcessor

This dictionary is used by the `OpenSMOKEpp_CHEMKIN_Preprocessor` solver with the aim to preprocess (i.e. interpret) a kinetic mechanism written in the standard `CHEMKIN` format. Only the thermodynamic file is strictly necessary, but, obviously, a kinetic mechanism must be provided if the aim is to perform simulations of reacting systems. For most of the ideal reactors no transport data are required and therefore the user could choose to pre-process only the thermodynamic and the kinetic files. According to the specified options, additional analysis can be easily and rapidly performed. Please consider that the kinetic and thermodynamic data must be always provided in two different files (i.e. the thermodynamic data cannot be added to the same file where the kinetic scheme is written).

Option	Type	Meaning
@Thermodynamics	PATH	Name of the file (ASCII) containing the thermodynamic data (CHEMKIN format)
@Output	PATH	Name of the folder where the pre-processed data are written
@Transport	PATH	Name of the file (ASCII) containing the transport data (CHEMKIN format)
@Kinetics	PATH	Name of the file (ASCII) containing the kinetic mechanism (CHEMKIN format)
@CheckThermodynamics	BOOL	The thermodynamic data are checked and additional files are written in output (together with a consistent reformulation of thermodynamic data)
@OutputOldStyle	BOOL	The output file are written also using the old format used by the OpenSMOKE framework (before 2013)
@TransportFittingCoefficients	BOOL	The fitting coefficients for the transport properties are written on a file. Please consider that this operation is very slow for large kinetic mechanisms (more than 1000 species) and produces huge files
@ReactionTables	BOOL	For each reaction detailed information is reported on a file (kinetic constants, change of moles, etc.)
@ReactionTablesListOfTemperatures	TODO	List of temperatures at which the detailed information is reported on a file for each reaction (kinetic constants, change of moles, etc.)
@ReverseFitting	BOOL	For each reversible reaction the reverse kinetic constants are estimated assuming the Arrhenius' law
@Comments	SUBDICTIONARY [Comments]	Additional data (author name, comments, etc.) can be added to the pre-processed kinetic mechanism
@SparsityPatternAnalysis	BOOL	Additional analyses about the sparsity pattern of stoichiometric matrix and associated Jacobian matrix
@RewriteCHEMKIN	BOOL	The kinetic mechanism is rewritten in CHEMKIN format by accounting for the corrections on atomic balances carried out during the pre-processing phase (default: true)
@SpeciesBundling	BOOL	Enables the species bundling technique for the fast estimation of mass diffusion coefficients (default: false)

Table 0.1: CHEMKIN-PreProcessor dictionary

0.0.1 Additional comments

@Thermodynamics

This option requires the name of the file (ASCII) containing the thermodynamic data (in CHEMKIN format). Both local and global paths can be accepted. Several useful files are automatically generated in the `Output` folder. In particular, the `Thermodynamics_Coefficients.out` file reports the correlation coefficients for the thermodynamic properties in a readable format. The `Thermodynamics_Tables.out` file reports the thermodynamic properties (specific heat, enthalpy, entropy, etc.) for each species as function of temperature.

@Kinetics

This option requires the name of the file (ASCII) containing the kinetic mechanism (in CHEMKIN format). Both local and global paths can be accepted. The `Kinetics_Summary.out` file reports the whole kinetic mechanism written in a more readable format.

@Transport

This option requires the name of the file (ASCII) containing the transport data (in CHEMKIN format). Both local and global paths can be accepted.

@Output

This can specify the name of the folder where the results of the simulation will be written. Both local and global paths can be accepted.

@CheckThermodynamics

This option allows to perform a detailed check of the thermodynamic data, to find possible inconsistent or unphysical data. The `Thermodynamics_Status.out` file reports a summary of the analysis of the thermodynamic data. In addition, a new file, containing new thermodynamic data, called `thermo.CHEMKIN.CKT`, will be provided. This new thermodynamic file is generated on the basis of the original thermodynamic data, which are made perfectly consistent at the transition temperature between low and high temperature correlations.

@TransportFittingCoefficients

The fitting coefficients for the transport properties (viscosity, thermal conductivity, mixture-averaged mass diffusion, and mixture-averaged thermal diffusion) for each species in the kinetic file are written on a file. Please consider that this operation is very slow for large kinetic mechanisms (more than 1000 species) and produces huge files (dimensions of Gb).

@ReactionTables

For each reaction detailed information is reported on the `Reaction_Tables.out` file (kinetic constants, change of moles, etc.). The `Reaction_Tables.out` file is self-explanatory.

@ReactionTablesListOfTemperatures

List of temperatures at which the detailed information is reported on a file for each reaction (kinetic constants, change of moles, etc.).

@ReverseFitting

For each reversible reaction the reverse kinetic constants are estimated assuming the Arrhenius' law. The kinetic parameters of the reverse reactions are then written on the `Reaction_FittedKinetics.out` file.

@SparsityPatternAnalysis

Additional analyses about the sparsity pattern of stoichiometric matrix and associated Jacobian matrix

@RewriteCHEMKIN

The kinetic mechanism is rewritten in CHEMKIN format by accounting for the corrections on atomic balances carried out during the pre-processing phase (default: true). At the end of this operation a new kinetic file (`kinetics.CHEMKIN.CKI`) is available in the Output folder.

@SpeciesBundling

Enables the species bundling technique for the fast estimation of mass diffusion coefficients (default: false). The technique is described in the following paper: **Lu and Law**, *Diffusion coefficient reduction through species bundling*, Combustion and Flame, 148(3), p. 117-126 (2007)