Brookesía

Documentation

version 1.5.4, 2020/08/08

Contents

1 Installation

Brookesia is developed under the informatic language Python 3. It can be used on Linux, macOS, and Windows.

The necessary computations for the reduction procedure are done with the open-source Cantera software toolkit. The first step is thus to install it.

1.1 Cantera

Every necessary information for the installation of Cantera are provided on Cantera website :

https://www.cantera.org/install/conda-install.html

1.2 Brookesia

Once cantera has been installed, the procedure for installing Brookesia is as follows:

1. Activate the environment in which Cantera is installed:

```
conda activate <name_of_your_environment>
```

2. Via pip, install Brookesia and other necessary modules:

```
conda install brookesia
conda install pandas
conda install scipy
conda install matplotlib
conda install numpy
```

1.3 Very first steps with Brookesia

In python (possibly under Spyder), to import the Brookesia module, type :

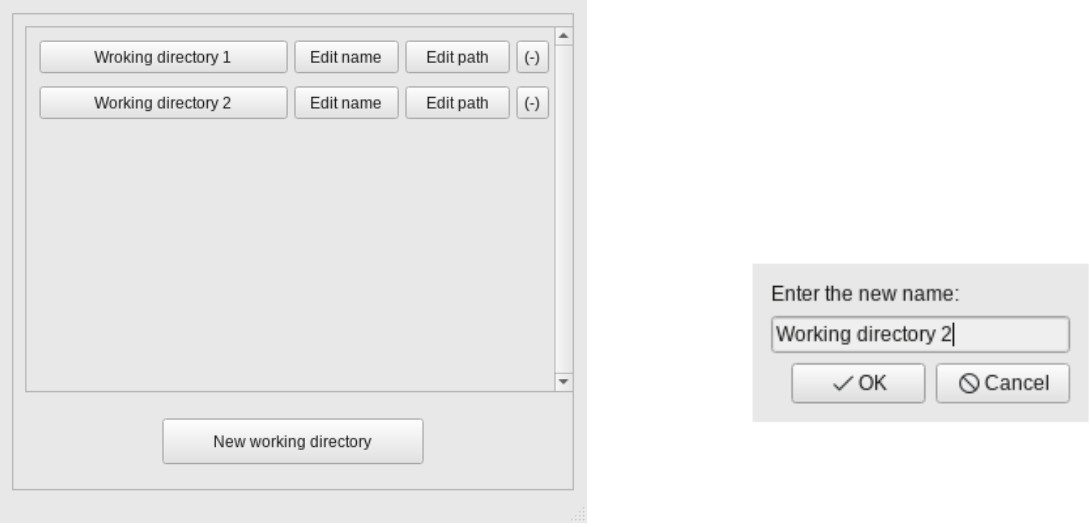
```
import brookesia as bk
```

Then, a window opens to define a folder that will be the working environment. In this folder, 3 sub-folders will be created:

- **_conditions_input**, containing 6 reduction/optimization example files(1_reactor.inp, 2_JSR.inp, etc.)
- **_kinetic_mech**, containing 4 kinetic mechanisms (C0_Konnov.cti [1], C1_GRI30.cti [2], C3_Gong.cti [3] and C7_Mehl.cti [4])
- **_results_input**, containing 5 .csv experimental data files and a sub-folder _flame_results containing old simulation results that can be reused as input data thanks to cantera's restore function (see, for instance, *3_Freeflame.inp* condition file) and *restore_flame_folder* option in the keyword dictionary.

The next times, the working directory choice window will automatically open only if several working directories are possible. To manage the working directories, click on the Working directory button on the graphic user interface (framed in figure 8 of the table 1) or via the Brookesia select_wd function :

```
bk.select_wd()
```



(a) Working directory selection

(b) Working directory name

Figure 1: Definition of the working directory

2 Basics for reduction and subsequent optimization of detailed mechanisms

Before using Brookesia, make sure your Cantera environment is activated.
On Anaconda: click on Environments tab then on your environment name.
On a terminal, type:

```
conda activate <path_to_cantera_environment>
```

2.1 Input file creation through the Graphic User Interface

On python or **Spyder** (normally installed with the Anaconda suite), load the Brookesia module:

```
import brookesia as bk
```

If needed, select your working directory, then type:

```
bk.gui()
```

Then, proceed to the following steps to operate a mechanism reduction (see the screen shots in Table 1):

On the **Main parameters** tab:

1. Load the kinetic mechanism
   Table 1 example: C1_GRI30.cti
   Note: It is possible to skip all the following steps by loading a previous input file by clicking on ***load file*** *button*
2. Select the target species
   Table 1 example: CH_4/CO/CO_2
3. Select the reduction methods you want to apply
   Table 1 example: DRGEP_sp + SA_r
   → then new tabs must appear
4. Select the errors options (pts/QoI ; max/mean)
5. On the **Conditions** tab:
   define the canonical configurations you want to run
   Table 1 example: Reactor (H,p) CH_4/air $\Phi = 0{,}5/1/1{,}5$ $p = 10^5$ Pa; $T = 1600$ K
6. On the ***reduction*** tab:
   define all the options you want to specify (see section 2.4 for details)
7. On the **GA** tab (if so):
   define all the options you want to specify (see section 2.4 for details)
8. On the **Main parameters** tab:
   If necessary:

- change the condition name and save it by clicking on the button Save current conditions
  Table 1 example: First_example.inp → then saved in the _condition_input folder
- change the folder name
  Table 1 example: First_example_folder
  Note: to this will be added the date and the hour of reduction (i.e. here 20191225_1200_First_example_folder)
- change the amount of outputs during the reduction process (verbose /show plot during reductions)

Run the reduction by clicking on **RUN** button

The reduction shall now take place. If the calculations are operated on Spyder and the option *Show plots during reduction* has been selected, you can follow the accuracy of the reduced models on the graphs displayed on the Spyder console.

2.2 Input file modification on a text editor

Saved input files (presented in appendices) are stored in an ASCII file, in the _condition_input folder. It may sometimes be more convenient to modify this text file directly than resorting to the GUI. Those input files can be then run by command line from a terminal (see next section).

They are composed in three parts:

- ```
  #=================================================
  #                Main parameters
  #=================================================
  ```

  Corresponds to the information displayed in "*Main parameters*" tab in GUI.
  Note: several species can be associated to the global data T, igt, Sl, and K. To do that just separate the associated species by a coma, i.e.:

  ```
  sp_T                    = CO2, CO
  ```

  ```
  #=================================================
  #                Simulation cases
  #=================================================
  ```

  Note: instead of entering fuel, oxidant, diluent, options, you can enter the mixture in the Cantera format using "mixt" option (and "mixt2" for counterflow burner configurations), i.e.:

  ```
  mixt                    = CH4:0.1, O2:0.1, N2:0.8
  ```

- ```
  #=================================================
  #                Operators
  #=================================================
  ```

  Corresponds to the information displayed in "*Reduction operator*" and "*GA*" tabs in GUI.
  Note: if the number of values provided for the tolerance limit of target species is lower than the number of target species, only the first value will be transcribed and applied to target species. Thus, if you want to define a tolerance limit of 30% for all target species, you only need to enter the value 30 after the max_error_sp keyword, i.e.:

  ```
  max_error_sp      = 30
  ```

An example of input file and the corresponding keyword dictionary are presented in appendices

Table 1: Successive usual steps for the reduction and optimization of a kinetic mechanism

2.3 Running Brookesia from command line

From the head of the working folder, type:

```
python main_reduction.py _conditions_input/<name_of_the_input_file>
```

The argument "_conditions_input/<name_of_the_input_file>" is the path toward the reduction condition input file.

2.4 Reduction and optimization options

- **Reduction options :**

Seven reduction methods are implemented in Brookesia :

- DRG - oriented species (DRG_sp)
- DRGEP - oriented species (DRGEP_sp)
- Sensitivity analysis - oriented species (SAR_sp)
- Sensitivity analysis with graph search - oriented species (SARGEP_sp)
- DRG - oriented reactions (DRG_r)
- Sensitivity analysis - oriented reactions (SAR_r)

For all reduction methods, you can specify (Figure 7 a.):

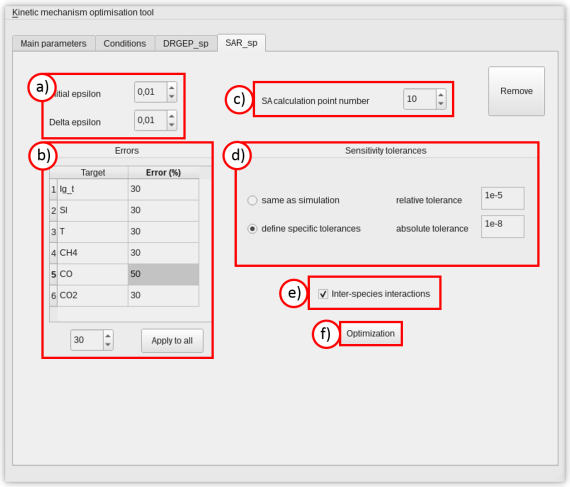
a) The starting epsilon,
b) The starting delta epsilon that will be applied between each iteration (subject to change during the reduction process)
c) The number of calculation points distributed in each simulation
d) The target tolerances
e) The Inter Species Interactions loop
f) The optimization after the reduction, if so, a new tab will appear

To facilitate the sensitivity analyses, it is possible to modify the tolerances (in the present example, relative/absolute tolerances are fixed at $10^{-5}/10^{-8}$, respectively)

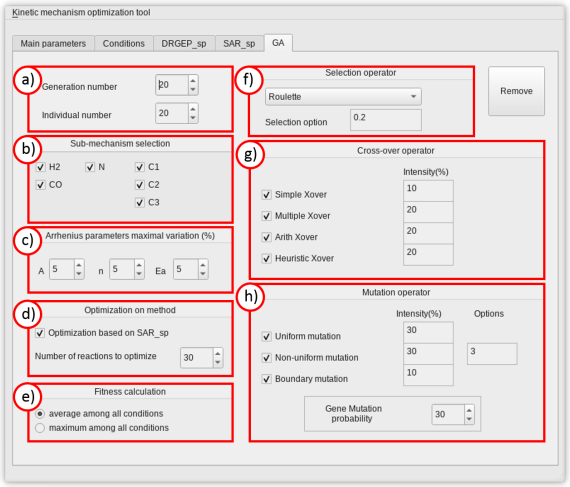
- **Optimization options:**

On the genetic algorithm tab, you can specify (Figure 7 b.):

a) the number of generation, of individuals,
b) the sub-mechanism to optimize
c) the variation range of reaction rate coefficients B, n and C
d) to constrain the reduction method to the important reactions identified during the reduction
e) to define the fitness computation depending on the average error or the maximal error (among all targets and conditions)
f) the selection method
g) the cross-over method and corresponding intensity
h) the mutation method and corresponding intensity. The gene mutation probability define the reaction probability to get its reaction rate coefficients modified



(a) Reduction tab, example of Sensitivity Analysis based Reduction



(b) Genetic algorithm tab

Figure 2: Composition of the operator tabs (description in section 2.4)

2.5 Output files

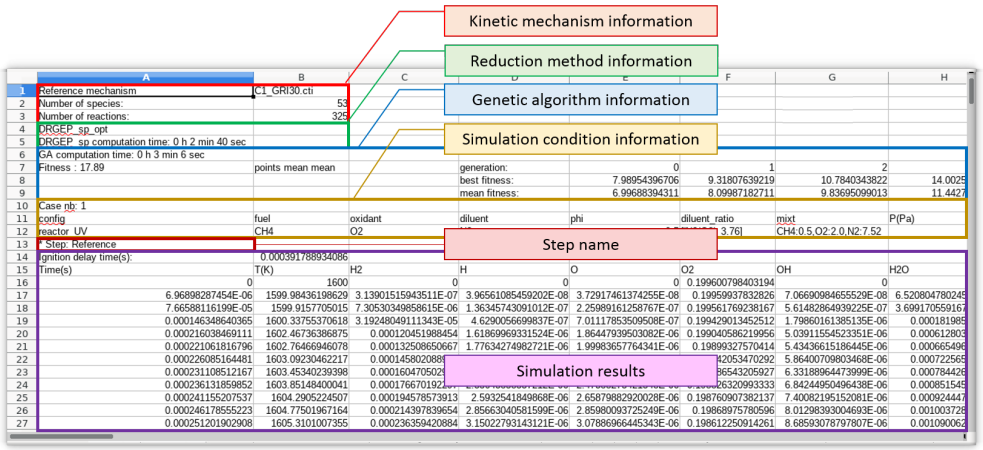
Ones the reduction is finished, you can check the results on the new folder [Date_Time_<name of the folder>], (here: 191225_1200_First_example_folder). It contains the following elements:

- **File.** Conditions_redopt.inp: input file of the reduction
- **File.** *.cti: reference mechanism (here: C1_GRI30.cti)
- **File.** red_info.txt: contains information of the reduction process
- **File.** X_reduction_results.csv: contains the simulation results of the X reduction step (here, 0_reduction_result.csv for the DRGEP_sp step and 1_reduction_result.csv for the SA_r step). See details, hereafter.
- **Folder.** Flame_ref_results (if so): store the *.xml reference flame results used for *restore* Cantera option to accelerate the new mechanism simulations
- **Folder.** Red_mech: contains the reduced and optimized step named as follow:
  X(opt)_B.cti
  where X represent the step number, (opt) is a character string added for the optimized kinetic mechanism, B is the reference mechanism name.
  In the present example, the kinetic mechanism created are:
  - 1_C1_GRI30.cti: reduced mechanism after DRGEP_sp step
  - 2_C1_GRI30.cti: reduced mechanism after SAR_sp step
  - 2opt_C1_GRI30.cti: optimized mechanism after genetic algorithm step

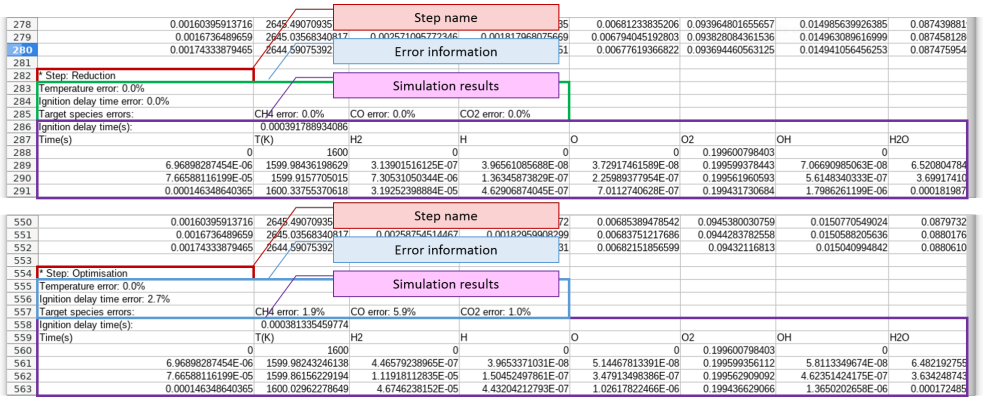
  Chemkin files can also be written by Brookesia, as far as the option "Write Chemkin file" in the Main Parameters tab is selected. Note that, at the moment (release 1.3.), only the kinetic mechanism file is converted (i.e., thermodynamical and transport data files are not). Conversion can also be done throught the command :

```
python main_reduction.py convert <name_of_the_cti_file>
```

Simulation results are written in a *.csv file as presented in Figures 3. The first part of the file (a) indicate information concerning the reference mechanism, the reduction method and, if so, the genetic algorithm. The main information concerning the conditions of the simulation cases (canonical configuration, mixture composition, pressure, temperature, simulation tolerances, etc.) are reminded and the results of the reference simulation are provided. As indicated, in the Figure 3 b), after the reference results, the simulation results obtained with the reduced mechanism and, if so, the optimized mechanism are given. Error information are also provided.



(a) Reference result part



(b) Reduction/optimization results part

Figure 3: Results file main sections composition

3 Data importation

3.1 Start from a previously reduced kinetic mechanism

It is possible to continue a reduction/optimization process previously started while keeping the initial data (i.e. i.e. simulation reference results and reaction rate constants) of the detailed baseline mechanism. To do

this, you must import the reference mechanism and specify the reduced mechanism from which you wish to start.

Procedure for specifying the reduced mechanism:

- from the GUI: click on the **Reduced mechanism** button on **Main parameter** tab.
- in an input file: in the `Main parameters` section, use the keyword `mech_prev_red`, e.g.:
  `mech_prev_red = red_mech.cti`

3.2 Import results file

In section 2.1, the reference results are calculated at the beginning of the reduction process from the detailed mechanism. It is also possible to conduct the reduction/optimization process from imported reference data. It allows, for instance, to use data simulated by another kinetic mechanisms than the one to be reduced/optimized as references, or also to define experimental data as references.

The input file for external results has to be filled in the same way than the output file provided by Brookesia (see figure 3). Thus, the easiest way to fill a new input result file is probably to start from a former output result file. If the value of a target for optimization or reduction is not provided in the input file, it is simply skipped for the evaluation of the simulation quality. For instance, it is possible to include in the experimental data panel experiments containing only ignition delay times while the optimization targets include ignition delay time as well as the evolution of CH_4, CO, and NO species. In the same way, flame structure experimental measurements containing only the profile of NO can be included even if the targets are CH_4, CH, and NO.

Results input files must be csv semicolon-separated file. Comments must be preceded by the hash symbol (#). The key-word "*Case*" is used to separate the different input conditions. This line is followed by the conditions specification, constituted, themselves in two lines. The first is a headline and the second contains the condition information. The column order must respect the arrangement presented in Table 2.

Important: when reference data are imported, the simulation conditions are only defined by those provided in the data file. In GUI, the **Conditions** tab is removed.

Procedure for importing a results file:

- from the GUI: after defining the reference kinetic mechanism, click on the buttaon `import external results`. Then, a window will appear to specify the units of the data (molar concentrations / molar fractions).
- in an input file: in the `Main parameters` section, use the keyword `ext_results_file`, e.g.:
  `ext_results_file = red_mech.cti`
  then specify the nature of the imported data with the keywords `conc_units`, e.g.:
  `conc_units = Molar_fraction`

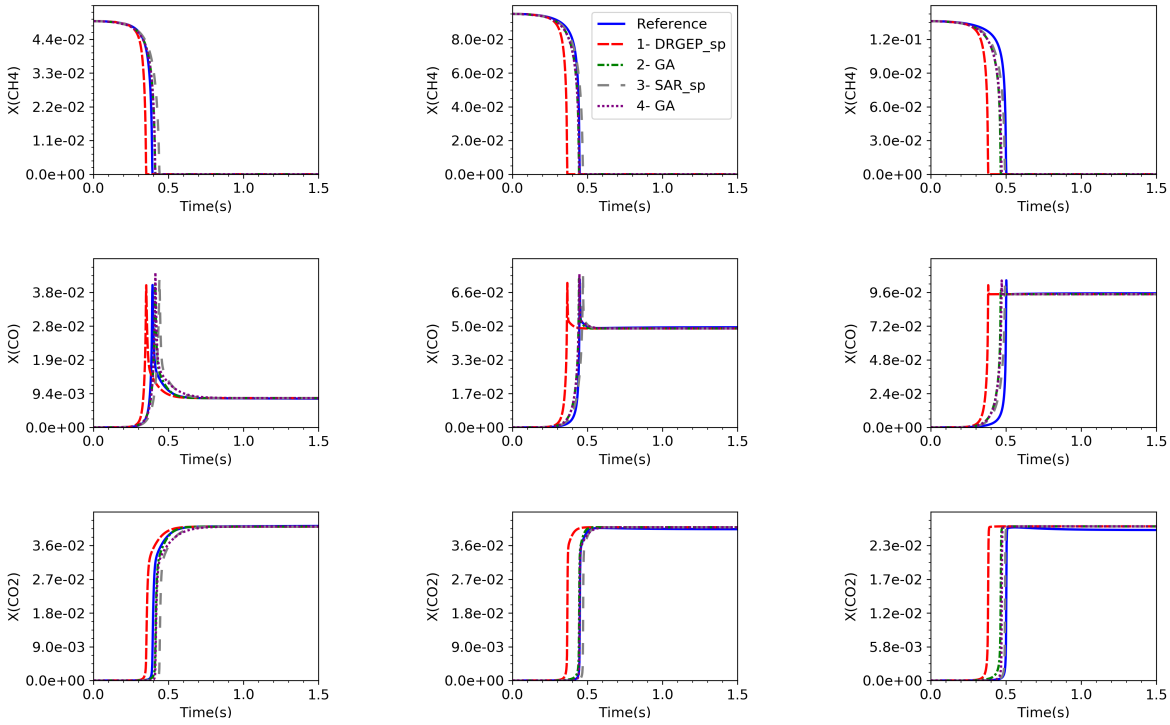
4 First examples

Five input files are provided for a gentle start with Brookesia software. Input conditions are summarized in Table 3.

As an example, the results obtained with the successive kinetic mechanisms simulation of the reduction procedure given in 1_reactor.inp are presented in Figure 4.

column	**reactor_HP**	**reactor_UV**	**JSR**	**free_flame**	**burner_flame**	**diff_flame**	**pp_flame**
1	config						
2	fuel (*)						
3	oxidant (*)						
4	diluent (*)						
5	phi (*)						
6	diluent_ratio (*)						
7	mixt (*)						
8	P (Pa)						
9	Ti (K)		time (s)	Ti (K)		fuel2 (*)	
10	rtol_ts					oxidant2 (*)	
11	atol_ts					diluent2 (*)	
12	ig_time (s)			rtol_ss		phi2 (*)	
13				atol_ss		diluent_ratio2 (*)	
14				transport_model		mixt2	
15				Sl (m/s)	mdot (kg/m2/s)	mdot2	
16						Ti (K)	
17						rtol_ts	
18						atol_ts	
19						rtol_ss	
20						atol_ss	
21						transport_model	
22						K_max(1/s)	

Table 2: Simulation condition information layout in the imported result files



(a) $\Phi = 0.5$ (b) $\Phi = 1$ (c) $\Phi = 1.5$

Figure 4: Simulation results obtained during the reduction process defined in the example 1_reactor.inp

Input file name	1_reactor.inp	2_JSR.inp	3_Free_Flame.inp	4_diff.inp	5_pp.inp	6_import_Sl
kinetic mechanism	C1_GRI30	C1_GRI30	C1_GRI30	C0_Konnov	C1_GRI30	C1_GRI30
Species (reaction)	53 (325)	53 (325)	53 (325)	15 (75)	53 (325)	28 (118)
conditions	Reactor (U,V) CH_4/air Φ=0.5, 1, 1.5 $p = 10^5$ Pa $T = 1600$ K	JSR CH_4/air $\Phi = 0.5, 1, 1.5$ $p = 10^5$ Pa $T = 600 - 1200$ K	Free flame CH_4/air $\Phi = 0.5, 1, 1.5$ $p = 10^5$ Pa $T = 300$ K	Diffusion flame H_2/air $p = 10^5$ Pa $T = 300$ K	Partially-premixed flame CH_4/air $\Phi = 1/0.5, 1.5/0.5$ $p = 10^5$ Pa $T = 300$ K	Free flame C_2H_6/air $\Phi = 0.6$ - 1.8 $p = 10^5$ Pa $T = 300$ K
Target(error)	CH_4(30) CO(30) CO_4(30) T(30) Igt(30)	CH_4(30) OH(30) CO_2(30) T(30)	CH_4(30) CO(30) CO_2(30) T(30) Sl(30)	H_2(30) H_2O(30) H(30) T(30) K(30)	CH_4(30) CO(30) CO_2(30) T(30)	Sl
Error method	points	points	QoI	QoI	points	points
Reduction method	DRGEP_sp + GA SAR_sp + GA	DRGEP_sp DRG_r + GA	DRGEP_sp SARGEP_sp + GA	DRGEP_sp DRG_r + GA	DRG_sp SAR_r + GA	GA
Final Species (reaction)	15 (57)	23 (114)	20 (100)	8 (15)	26 (87)	53 (325)
Computation time	7 min	5 min	2 h 08 min	30 min	40 min	2 h 54 min

Table 3: Summary presentation of the five input files provided as examples
targets symbols: *T: temperature, igt: ignition delay time, Sl: laminar flame speed, K: Extinction stretch rate*

5 Theory

The Brookesia software has been developed to allow the reduction and optimization of kinetic mechanisms. It is based on the open-source tool Cantera to perform simulations of combustion processes. The objective of this section is to recall some theoretical elements about the implemented algorithms.

5.1 DRG & DRGEP algorithms

The DRG method and its derived approaches are based on the analysis of reaction flows to identify couplings between species. Initially developed by Lu and Law [5], the DRG method rely on inter-species couplings to differentiate *important* species to *redundant* species which can be removed from the initial mechanism without major impact on a simulation set.

The success of these methods is strongly linked to their low numerical cost. Substantial reductions can be achieved in a short time. It is therefore recommended to use these algorithms as a priority for the first steps of a reduction process.

5.1.1 Direct interaction coefficients

Two species are said to be directly coupled when they are involved in the same elementary reaction. Due to the presence of interactions with other species, the degree of coupling can be strongly asymmetric between two species. It can be estimated in various ways (see refs. [5–7]). The formulation proposed by Pepiot-Desjardins and Pitsch, used in the Brookesia software, is be presented here. It compares the absolute net production/consumption rate of A when species B is involved in the reaction to the maximum of the production and consumption of A:

$$r_{AB} = \frac{|\sum_{i=1}^{n_R} \nu_{A,i}\omega_i\delta_{Bi}|}{max(P_A, C_A)} \tag{1}$$

with:

$$\begin{aligned} P_A &= \sum_{i=1}^{n_{\text{reactions}}} max(0, \nu_{A,i}\omega_i) \\ C_A &= \sum_{i=1}^{n_{\text{reactions}}} max(0, -\nu_{A,i}\omega_i) \end{aligned} \tag{2}$$

The interaction graph is then built based on these coefficients.

5.1.2 Graph search analysis and cutoff threshold

When two species A and B are not involved in the same elementary reactions, their direct interaction coefficients r_{AB} and r_{BA} are equal to zero. However, the potential existence of indirect couplings means that the presence of one of these species may be necessary for the reasonable estimation of the second. The distinction between necessary and redundant species is therefore done by means of a directed graph to take into account the reaction pathways of the whole chemical process and the indirect interactions. Each node of the graph represents a chemical species and each edge represents a direct coupling. These edges are all characterized by an orientation and a weight corresponding to the interaction coefficient between the nodes. This oriented graph is translated numerically in the form of a non-symmetrical (nS,nS) matrix containing the direct interaction coefficients (see Figure 5(a) and 5(b)).

Once the graph is constructed, a graph search algorithm is operated from the nodes representing the target species. With the DRG method, the selection of the critical species is performed from the values of the direct interactions between the species composing the path starting from A and the cut-off threshold ϵ.

To estimate the coupling between A and B, a first interaction coefficient between A and B, considering a path p, is defined as:

$$r_{AB,p}^{DRG} = \min_{i=1}^{n_{sp}-1} r_{S_i S_{i+1}} \tag{3}$$

with n_{sp}, the number of species, $r_{S_i S_{i+1}}$, the direct coupling coefficient between the species S_i and S_{i+1} (see eq. 1). A graph search is employed to evaluate all the possible path from one species to another. Dijkstra algorithm is used in Brookesia (you can refer to [8] to get more details on this algorithm).

The overall local DRG interaction coefficient is computed from the relation:

$$R_{AB}^{DRG} = \max_{\text{all path } p} r_{AB,p}^{DRG} \tag{4}$$

At this stage, a discrimination can be established between the *target* species, the *important* species and the *redundant* species (see Figure 5(c)). This discrimination is valid under the conditions under which the chemistry is evaluated.

Direct interaction coefficients are local quantities. They have to be regularly recalculated during the simulation in order to properly integrate the couplings on the overall combustion process.

In the present tool, the DRG is applied for each species over a number of time steps/mesh points initially set by the operator, distributed along the discretization in time (0D reactor) or distance (1D flame). As shown in Figure 6, if the coefficients computation is performed in 4 points, important combustion steps can be missed. However, provided that the iteration is correctly set up, doing coefficients computation on 10 points is usually sufficient to catch every important stage of the combustion process and to get good reduction performances.

Finally, the interaction coefficient R, defined over all the important points during the combustion is calculated as follow:

$$R_{AB,\text{global}}^{\text{DRG}} = \max_{\text{all points}} \left[\max_{\text{all path } p} r_{AB,p}^{DRG} \right] \tag{5}$$

Then, a cut-off threshold can be used to eliminate species whose interactions with the target species are not significant (i.e., with a low $R_{AB,\text{global}}^{\text{DRG}}$ coefficient).

In the DRGEP approach, the length of the path between a species (B) and the target (A) is taken into account. A new definition of the interaction coefficient is proposed:

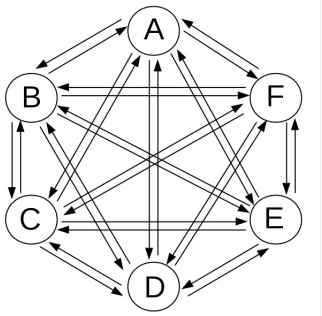
$$r_{AB,p}^{\text{DRGEP}} = \prod_{i=1}^{n-1} r_{S_i S_{i+1}} \tag{6}$$

The local and global interaction coefficient remains defined as in eq. 4 and 5, respectively. The graph search is performed using the Dijkstra algorithm.

5.2 SAR algorithms

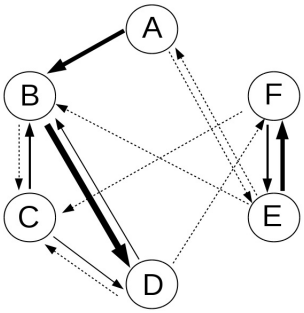
Sensitivity analysis methods quantify the influence of input parameters on the outputs of a model. They can be applied to chemical model analysis to identify the main parameters controlling kinetics, uncertainty analysis, or model reduction [9]. Brookesia thus relies on this analysis to identify the parameters having the most influence on the quantities that we want to reproduce faithfully (*targets*). As in the case of reduction by DRGs, sensitivity analysis will be applied for species and reaction withdrawal.

Reduction methods based on sensitivity analysis are generally very effective. However, the associated numerical cost is high. It is therefore recommended that they be used after an initial reduction by reaction flow analysis (DRG and/or DRGEP).



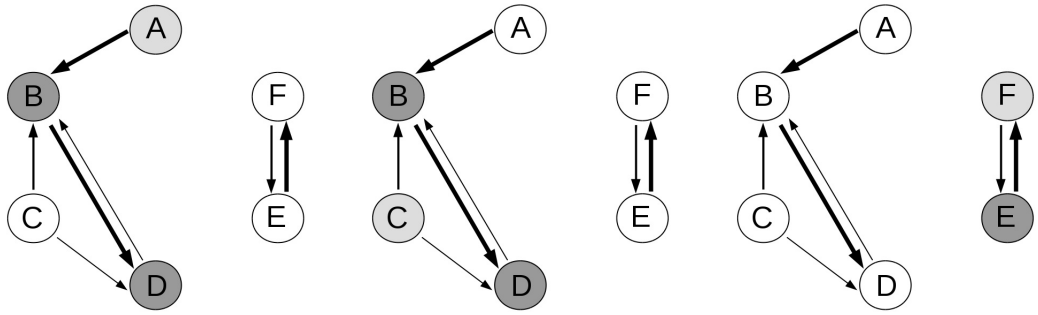
$$\begin{bmatrix} r_{AA} & r_{AB} & r_{AC} & r_{AD} & r_{AE} & r_{AF} \\ r_{BA} & r_{BB} & r_{BC} & r_{BD} & r_{BE} & r_{BF} \\ r_{CA} & r_{CB} & r_{CC} & r_{CD} & r_{CE} & r_{CF} \\ r_{DA} & r_{DB} & r_{DC} & r_{DD} & r_{DE} & r_{DF} \\ r_{EA} & r_{EB} & r_{EC} & r_{ED} & r_{EE} & r_{EF} \\ r_{FA} & r_{FB} & r_{FC} & r_{FD} & r_{FE} & r_{FF} \end{bmatrix}$$

(a) Complete fictitious mechanism: potential interactions



$$\begin{bmatrix} r_{AA} & 0.4 & 0 & 0 & 0.02 & 0 \\ 0 & r_{BB} & 0.02 & 0.6 & 0 & 0 \\ 0 & 0.3 & r_{CC} & 0.2 & 0 & 0 \\ 0 & 0.1 & 0.03 & r_{DD} & 0 & 0.05 \\ 0.02 & 0.03 & 0 & 0 & r_{EE} & 0.4 \\ 0 & 0 & 0.03 & 0 & 0.2 & r_{FF} \end{bmatrix}$$

(b) Complete fictitious mechanism: calculated interactions. Solid line arrows: non-negligible interactions, dashed arrows: interactions below ε_s



(c) Example of reduced mechanisms for target species A, C and E (from left to right). Light grey: target species, dark grey: important species, white: redundant species.

Figure 5: Procedure of reduction by Directed Relation Graph: from the complete mechanism to the skeletal mechanisms depending on the target species. Illustration inspired by Lu and Law [5].

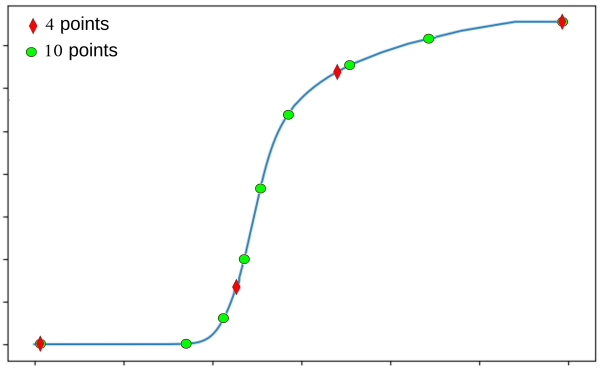


Figure 6: Schematic representation of the localization of the interaction / sensitivity coefficients calculation points on an increasing profile.

5.3 Reaction withdrawal using a Sensitivity Analysis based Reduction method

To identify redundant reactions, the Brookesia tool is based on the matrix of normalized sensitivity coefficients:

$$\tilde{S}_{T,i} = \frac{\partial \ln([T])}{\partial \ln(k_j)} = \frac{k_i}{[T]} \frac{\partial [T]}{\partial k_i}$$

In Cantera, the *sensitivities* function of the *ReactorNet* class provides easy access to the sensitivity vectors $\tilde{S}_{T,i}$ of the targets T (species, temperature) to the i^{th} reaction rates. For 1D flames, the sensitivity of the targets T (species, temperature or laminar flame speed) $\tilde{S}_{T,i}$ are calculated using an adjoint method implemented in Cantera via the *solve_ adjoint* function. In order to evaluate the important reactions regardin ignition delay times, and extinction strain rate, associated target species are to be defined by the operator (it can be the H radical, for example).

As in the case of direct interaction coefficients, calculations will therefore have to performed at multiple points in the simulation. The maximum sensitivity values encountered will be retained for the reduction process:

$$\tilde{S}_{T,i,\text{global}} = \max_{\text{all points}} \left[\tilde{S}_{T,i} \right] \tag{7}$$

Reactions for which a sensitivity lower than the ε_r cutoff is obtained are be removed from the mechanism. If all reactions involving a chemical species are redundant, that species is also be removed.

5.4 Species withdrawal using a Sensitivity Analysis based Reduction method (SAR_sp)

The greatest reductions in computational cost time are achieved by reducing the number of species. The sensitivity analysis was therefore extended to be more directly devoted to species withdrawal. The brute force method of perturbing concentrations would involve performing multiple simulations at each time step and would be too tedious and computationally cumbersome. Consequently, the formulation of Løvås [10] is adopted in Brookesia and the local sensitivity of the target T to a second species B is estimated as the sum of the normalized sensitivities of this species to the reactions involving species B, weighted by the stoichiometric coefficients of B:

$$\tilde{S}_{TB} = \sum_{i=1}^{n_{\text{reactions}}} \left| \tilde{S}_{Ti} \nu_{i,k_B} \right| \tag{8}$$

The global sensitivity becomes:

$$\tilde{S}_{TB,\text{global}} = \max_{\text{all points}} \left[\tilde{S}_{TB} \right] \tag{9}$$

Inter-species sensitivities are calculated for several time steps of the simulation. The inter-species interaction coefficient calculated by the sensitivity analysis is expressed as:

$$\tilde{S}_{TB,\text{global}} = \max_{\text{all points}} \left[\tilde{S}_{TB} \right] \tag{10}$$

Based on the calculation of these values, the reduction algorithm will be able to identify, according to a cut-off threshold, the redundant species.

5.5 Sensitivity Analysis based Reduction assisted by a Graph search with Error Propagation (SARGEP_sp)

The assessment of indirect interations by graph search was associated with a reduction algorithm based on sensitivity analysis. The idea is to strengthen the consideration of indirect interactions in the reduction process. The graph search exploited in Brookesia integrates error propagation (see DRGEP description in section 5.1). The local interaction coefficient between the target T and a species B is defined as:

$$r_{TB}^{\text{SARGEP}} = \max_{\text{all path } p} \left[\tilde{S}_{TS_i} \prod_{i=1}^{n_{\text{sp}}-1} \tilde{S}_{S_i S_{i+1}} \right] \tag{11}$$

with n_{sp}, the number of species, $\tilde{S}_{S_i S_{i+1}}$, the sensitivity of the species S_i to the species S_{i+1} (see eq. 8).

The overall global interaction is:

$$R_{TB,\text{global}}^{\text{SARGEP}} = \max_{\text{all points}} \left[r_{TB}^{\text{SARGEP}} \right] \tag{12}$$

if $R_{TB,\text{global}}^{\text{SARGEP}} < \epsilon$, the species B is removed from the kinetic mechanism.

5.6 Optimization by Genetic Algorithms

Brookesia comes with Genetic algorithm to allow optimization of kinetic mechanisms. This optimization can be considered in order to compensate for errors introduced by the reduction of the kinetic model, or simply to optimize all or part of the kinetic constants of the model in order to improve the predictivity of the simulations with respect to an experimental data set.

According to Darwin's theory of evolution, the characteristics of individuals best adapted to their environment are transmitted over generations. In this way, populations develop in a way that is optimised with respect to the constraints of their environment. Optimisation, known as genetic algorithm (GA) is based on this finding to search for an optimal solution based on a population of potential solutions. By analogy with biology, each individual will be described by a sequence of genes representing the different variables that can be optimized. For kinetic mechanism optimization, the variables to be optimized are the kinetic constants of the extended Arrhenius Law A, n, and Ea, specific to each of the elementary reactions. Each reaction is then associated with a triplet of genes and the individuals will be constructed with a number of triplets corresponding to the number of elementary reactions of the mechanism to be optimized. This structure is illustrated in Figure 7.

After evaluation of each individual (5.6.1), a selection procedure will make it possible to keep the individuals with the best characteristics (5.6.2). At each new stage of the algorithm, the genetic mixing takes place through cross-over (5.6.3), and of mutations (5.6.4) will allow the creation of new, hopefully better, individuals.

The theory of genetic algorithms is well documented in Holland et al [11] and Elliot et al. [12] papers. The operators introduced in Brookesia are presented below.

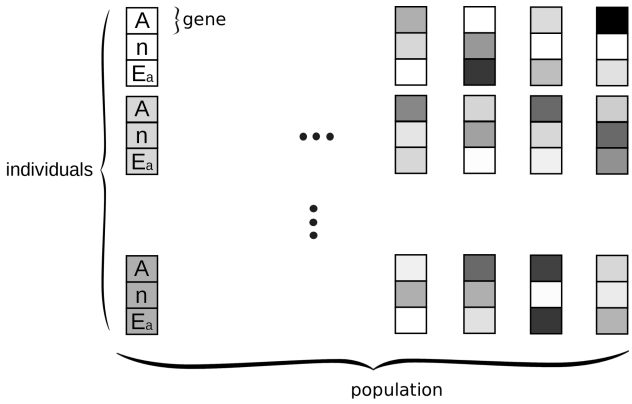


Figure 7: Schematic illustration of genes, individuals and population in the context of optimization of kinetic mechanisms with genetic algorithm

5.6.1 Kinetic mechanism quality assessment

A key step in the genetic algorithm is the definition of the individual assessment function, commonly referred to as the fitness function. It will help to distinguish between individuals who are close to an optimal solution and those whose genes are unsuitable for solving the problem. It will therefore make it possible to classify the individuals in a population. To do this, it must cover all aspects of the optimization problem in order to identify the most suitable individual. The classification of individuals will allow the selection of those with the best characteristics in order to direct the research towards an optimal solution.

The error committed on each target species will be evaluated using the point by point error estimators or QoI. These measures will make it possible to characterize the individual's ability to reproduce the different quantities of interest. As in the case of error evaluation, two definitions are proposed to the user (eq. 13). The first defines fitness as the inverse of the largest error measured among the target species and global data. The second uses the average of the errors encountered on each of the reduction targets. Therefore, the lower the measured errors, the higher the fitness value of the individual.

$$\text{fitness} = \frac{1}{\max(\text{erreur})} \qquad \text{ou} \qquad \text{fitness} = \frac{1}{\overline{\text{erreur}}} \tag{13}$$

The choice of the average or maximum value will influence the choices made by individuals during the selection process. The use of the maximum will favour mechanisms that improve the estimation of the least well reproduced quantity. The mean value, on the other hand, will favour the overall improvement in the quality of the reduced mechanism even if some estimates have deteriorated. These differences should be taken into account when the genetic algorithm is applied over a small number of generations. However, they will diminish over the generations.

When the reduction, and thus the optimization, is done over several configurations, the maximum and average error values are evaluated over all the cases studied. Thus the fitness values characterize the capacity of individuals to model well all configurations.

It can happen that a potential solution does not allow to carry out a simulation (lack of reactivity, convergence problem, ...). In this case, a null fitness will be associated to the individual. Thus, this one will not be able to be selected to be part of the next generation.

5.6.2 Selection

The implemented selection methods are as follows:

- Rank method: where the probability of selection P_i is weighted by the individual ranking, among the entire population (with N, the population size):

$$P_i = \frac{\text{rank}_i}{\sum_{j=1}^{N} j} \tag{14}$$

- Roulette wheel: where the probability of selection is weighted by the individual fitness:

$$P_i = \frac{F_i}{\sum_{j=1}^{N} F_j} \tag{15}$$

- Normalized geometric ranking: based on the geometric probability law:

$$P_i = \frac{q}{1-(1-q)^N}\,(1-q)^{\text{rank}-1} \tag{16}$$

  where the parameter q allows to adapt the selective character of the method.

- Elitist method: which consists in keeping only a fraction of the best individuals.

Whatever method is chosen, the program is adapted to ensure keeping the best individual for the future generation.

5.6.3 Cross-over

The implemented crossing methods are as follows:

- Simple cross-over and Multiple cross-over: which involves the exchange of gene sections between the two parents according to one (Simple cross-over) or multiple (Multiple cross-over) cut-off sites.

- Arithmetic cross-over: whose method consists in generating two children from a linear combination between two individuals (parents). For two parents X and Y, the x' and y' genes creation of the two-child individuals X' and Y' is calculated as follows:

$$\begin{cases} x'_i = rx_i + (1-r)y_i \\ y'_i = (1-r)x_i + rx_i \end{cases} \quad \forall i \in [1, n_{gene}] \tag{17}$$

  with r a random variable between 0 and 1 redefined for each reaction.

- Heuristic cross-over: who creates a first child by linear extrapolation from the parent individuals as well as information about their respective fitness. The second child is a copy of the parent with the best fitness. Assuming that the X individual has higher fitness than Y, the genes of the children's individuals are computed as:

$$\begin{cases} x'_i = x_i + r(x_i - y_i) \\ y'_i = x_i \end{cases} \quad \forall i \in [1, n_{gene}] \tag{18}$$

  with r a random variable between 0 and 1 redefined for each reaction.

  One of the characteristics of the heuristic crossing is that linear extrapolation can generate parameter values outside the validity interval. A feasibility condition f_{heur} must therefore be integrated into the operator. The value of the gene x_i must be included in the validity range $\left[b_1^i, b_2^i\right]$:

$$f_{heur} = \begin{cases} 1 & \text{if } b_1^i < x'_i < b_2^i \\ 0 & \text{otherwise} \end{cases} \quad \forall i \in [1, n_{gene}] \tag{19}$$

  As long as this condition is not satisfied, the interpolation is recalculated with a new random value of r. In the present tool, a modified form of the heuristic crossing is applied after three unsuccessful attempts: for each gene where the feasibility condition is not satisfied, the gene of the best individual will be preserved for the two child individuals.

5.6.4 Mutation

Mutation consist of randomly modifying a gene. The intensity of the mutation among all the genes is defined by the operator. The default amount of genes affected by a mutation operator is set at 30%. The methods introduced in the code are:

- Simple mutation: The simplest form of the mutation operator is to randomly select one or more genes from a parent individual and modify their values. In the present code, a value in the validity range $\left[b_1^i, b_2^i\right]$ is generated randomly for a gene x_i and applied to the individual child.

$$x_i' = r_i \qquad \text{with} \quad r_i \in [b_1^i, b_2^i] \tag{20}$$

- Non-uniform mutation: As optimization progresses, individuals are theoretically closer to the desired optimum. Mutations of too important amplitudes can deviate from the desired solution. The non-uniform mutation operator is defined such as the intensity of the mutation is decreasing during generations. For a gene associated with the value x_i, the value after a non-uniform mutation is calculated as:

$$x_i' = \begin{cases} x_i + \left(r_1^i - x_i\right) * f(G) & \text{if } r_1 < 0.5 \\ x_i - \left(x_i + b_2^i\right) * f(G) & \text{if } r_1 > 0.5 \end{cases} \tag{21}$$

  with:

$$f(G) = \left(r_2 \left(1 - \frac{G}{G_{max}}\right)\right)^s \tag{22}$$

  The importance of the variations is weighted by the number of generations, G already carried out through a function $f(G)$. A shape parameter s defined by the operator controls the intensity of this weighting. Two random variables r_1 and r_2 belonging to the interval $[0, 1]$ are generated. The first one defines the direction of the variation (in the validity range $\left[b_1^i, b_2^i\right]$), the second one controls the intensity.

- Boundary mutation: This operator allows to explore the limit values of the validity intervals:

$$x_i' = \begin{cases} b_1^i & \text{if } r_1 < 0.5 \\ b_2^i & \text{if } r_1 > 0.5 \end{cases} \tag{23}$$

  This mutation operator is generally used in combination with other mutation operators to introduce diversity into the population.

References

[1] A. A. Konnov, Yet another kinetic mechanism for hydrogen combustion, Combust. Flame 203 (2019) 14–22.

[2] G. P. Smith, D. M. Golden, M. Frenklach, N. W. Moriarty, B. Eiteneer, M. Goldenberg, C. T. Bowman, R. K. Hanson, S. Song, W. C. Gardiner, V. V. Lissianski, GRI-Mech 3.0 (1999).
URL http://combustion.berkeley.edu/gri-mech/version30/text30.html

[3] J. Gong, S. Zhang, Y. Cheng, Z. Huang, C. Tang, J. Zhang, A comparative study of n-propanol, propanal, acetone, and propane combustion in laminar flames, Proc. Combust. Inst. 35 (1) (2015) 795–801.

[4] M. Mehl, W. J. Pitz, C. K. Westbrook, H. J. Curran, Kinetic modeling of gasoline surrogate components and mixtures under engine conditions, Proc. Combust. Inst. 33 (1) (2011) 193–200.

[5] T. Lu, C. K. Law, A directed relation graph method for mechanism reduction, Proc. Combust. Inst. 30 (1) (2005) 1333–1341.

[6] T. Lu, C. K. Law, On the applicability of directed relation graphs to the reduction of reaction mechanisms, Combust. Flame 146 (3) (2006) 472–483.

[7] P. Pepiot-Desjardins, H. Pitsch, An efficient error-propagation-based reduction method for large chemical kinetic mechanisms, Combust. Flame 154 (1–2) (2008) 67–81.

[8] K. E. Niemeyer, C.-J. Sung, On the importance of graph search algorithms for DRGEP-based mechanism reduction methods, Combust. Flame 158 (8) (2011) 1439–1443.

[9] A. S. Tomlin, T. Turányi, M. J. Pilling, Mathematical tools for the construction, investigation and reduction of combustion mechanisms, Compr. Chem. Kinet. 35 (1997) 293–437.

[10] T. Lovas, Automatic generation of skeletal mechanisms for ignition combustion based on level of importance analysis, Combust. Flame 156 (7) (2009) 1348–1358.

[11] J. H. Holland, Genetic Algorithms, Cient. Am. 267 (1) (1992) 66–73.

[12] L. Elliott, D. Ingham, A. Kyne, N. Mera, M. Pourkashanian, C. Wilson, Genetic algorithms for optimisation of chemical kinetics reaction mechanisms, Prog. Energy Combust. Sci. 30 (3) (2004) 297–328.

6 Appendices

Input file example

```
#=================================================
#            Main parameters
#=================================================
main_path           = TEST_1_reactor
mech                = C1_GRI30.cti
verbose             = 4
show_plots          = False
write_ck            = True
tspc                = CH4, CO, CO2
T_check             = True
sp_T                = CO2
Sl_check            = False
sp_Sl               = H
ig_check            = True
sp_ig               = CH3
K_check             = False
sp_K                = H
error_calculation   = points
error_coupling      = mean


#=================================================
#            Simulation cases
#=================================================

#=======> Case 1
config              = reactor_UV
Ps                  = 100000.0
fuel                = CH4
oxidant             = O2
diluent             = N2
diluent_ratio       = N2/O2 3.76
Ts                  = 1600.0
phis                = 0.5, 1.0, 1.5
n_pts               = 250.0
delta_npts          = 20.0
t_max_coeff         = 5.0
Scal_ref            = H2O
grad_curv_ratio     = 0.5
tign_nPoints        = 450.0
tign_dt             = 1e-09
tol_ts              = 1e-06, 1e-12



#=================================================
#            Operators
#=================================================


#=========> Op: DRGEP_sp
operator            = DRGEP_sp
eps                 = 0.02
delta_eps           = 0.01
n_points            = 10.0
max_error_sp        = 30, 30, 30
max_error_T         = 30
max_error_ig        = 30
inter_sp_inter      = True
optim               = GA
#=======> Genetic Algorithm Optimization
n_gen                   = 5
n_indiv                 = 5
```

```
error_fitness        = mean
Arrh_max_variation   = 5, 5, 5
optim_on_meth        = False
sub_mech_sel         = H2, CO, C1, C2, C3, N
selection_operator   = Roulette
selection_options    = 0.2
Xover_operator       = simple_Xover, multiple_Xover, arith_Xover, heuristic_Xover
Xover_pct            = 10, 20, 20, 20
mut_operator         = uniform_mutation, non_uniform_mutation, boundary_mutation
mut_pct              = 30, 30, 10
mut_opt              = , 3,
mut_intensity        = 20


#=============> Op: SAR_sp
operator          = SAR_sp
eps               = 0.02
delta_eps         = 0.01
n_points          = 10.0
max_error_sp      = 30.0, 30.0, 30.0
max_error_T       = 30.0
max_error_ig      = 30.0
inter_sp_inter    = True
ttol_sensi        = 1e-05, 1e-08
optim             = GA
#=====> Genetic Algorithm Optimization
n_gen                = 5
n_indiv              = 5
error_fitness        = mean
Arrh_max_variation   = 5, 5, 5
optim_on_meth        = False
sub_mech_sel         = H2, CO, C1, C2, C3, N
selection_operator   = Roulette
selection_options    = 0.2
Xover_operator       = simple_Xover, multiple_Xover, arith_Xover, heuristic_Xover
Xover_pct            = 10, 20, 20, 20
mut_operator         = uniform_mutation, non_uniform_mutation, boundary_mutation
mut_pct              = 30, 30, 10
mut_opt              = , 3,
mut_intensity        = 20
```

Input file keyword dictionary

Main parameters keywords

main_path	name of the folder containing all results (to this will be added the date and the hour of reduction)
mech	reference mechanism name
mech_prev_red	starting reduced mechanism name
ext_results_file	name of the external file containing reference conditions and corresponding results
conc_units	concentration unit of the imported external data (Molar_fraction/mol_m3)
show_plots	for Spyder essentially, select to display or not the simulation results during the reduction process
write_ck	write reduced / optimized mechanism for chemkin
verbose	rules the amoung of information displayed during the reduction process (range between 0 and 10)
tspc	name of target species
ig_check	add (True) or not (False) the ignition delay time as target
K_check	add (True) or not (False) the extinction stretch rate as target
Sl_check	add (True) or not (False) the flame speed as target
T_check	add (True) or not (False) the temperature as target
sp_ig	ignition delay time associated species to manage reduction with reduction methods
sp_K	extinction stretch rate associated species to manage reduction with reductions methods
sp_Sl	flame speed associated species to manage reduction with DRG methods
sp_T	temperature associated species to manage reduction with DRG methods
error_calculation	error method (point/QoI)
error_coupling	error interpretation (max/mean)

Configuration keywords

config	configuration (reactor_UV, reactor_HP, free_flame, diff_flame, pp_flame, tp_flame)
reactor_UV	adiabatic, constant volume reactor
JSR	Jet-stirred reactor
PFR	Plug-flow reactor (!) *beta version*
reactor_HP	adiabatic, constant pressure reactor
free_flame	freely propagative adiabatic flame
diff_flame	counterflow diffusion flame
pp_flame	partially-premixed flame
tp_flame	twin-premixed flames
mixt	initial mixture composition (e.g. mixt = CH4:0.1, O2:0.1, N2:0.8)
fuel	fuel species
oxidant	oxidant species
diluent	diluent species
diluent_ratio	dilution ratio (%)
Ts	initial temperature(s) (K)
Ps	initial pressure(s) (Pa)
phis	equivalent ratio(s)
tol_ts	simulation transient tolerances
n_pts	*for reactors:* simulation point number (for reactors)
delta_npts	*for reactors:* tolerance in simulation point number

Configuration keywords

t_max_coeff	*for reactors:* final time of the simulations (multiple of ignition delay time)
Scal_ref	*for reactors:* reference scalar for grad and curve interpretation
grad_curv_ratio	*for reactors:* grad/curv options for time discretization
t_max	*for JSR:* residence time on the reactor (s)
xmax	*for free_flames:* dimension of the computational domain (m)
tol_ss	*for flames:* simulation steady-state tolerances
transport_model	*for flames:* transport model (Mix/Mult)
pts_scatter	*for flames:* initial grid
slope	*for flames:* Cantera refining criteria normalized maximum difference in value between two adjacent points
curve	*for flames:* Cantera refining criteria normalized maximum difference in slope between two adjacent intervals
ratio	*for flames:* Cantera refining criteria add points if the ratio of the spacing on either side of a grid point exceeds this value
prune	*for flames:* Cantera refining criteria remove point if the slope or curve criteria are satisfied to the level of "prune"
restore_flame_folder	*for flames:* name of the folder containing cantera former flame results (*.xml). Note that these files must start with the number of the simulation in Brookesia, i.e. if, in one case, three simulations have to be computed at equivalence ratio $\Phi = 0.5$, then $\Phi = 1$, then $\Phi = 1.5$, the *.xml files must be named **1**_whatever.xml, **2**_whatever.xml, **3**_whatever.xml and contain the results of flames at $\Phi = 0.5$, $\Phi = 1$, and $\Phi = 1.5$, respectively.
xmax	*for free_flames:* dimension of the computational domain (m)
fuel_1	*for counterflow flames:* bottom burner fuel species
oxidant_1	*for counterflow flames:* bottom burner oxidant species
diluent_1	*for counterflow flames:* bottom burner diluent species
diluent_ratio_1	*for counterflow flames:* bottom burner diluent species
Ts_1	*for counterflow flames:* bottom burner initial temperature(s) (K)
phis_1	*for counterflow flames:* bottom burner equivalent ratio(s)
mdots_1	*for counterflow flames:* bottom burner mass flux ($kg/m^2/s$)
fuel_2	*for counterflow flames:* top burner fuel species
oxidant_2	*for counterflow flames:* top burner oxidant species
diluent_2	*for counterflow flames:* top burner diluent species
diluent_ratio_2	*for counterflow flames:* top burner diluent ratio
Ts_2	*for counterflow flames:* top burner temperature(s) (K)
phis_1	*for counterflow flames:* top burner equivalent ratio(s)
mdots_2	*for counterflow flames:* top burner mass flux ($kg/m^2/s$)
width	*for counterflow flames:* interval between the bottom and the top burner (m)

Reduction method keywords

operator	reduction operator (DRG_sp, DRGEP_sp, DRG_r, SAR_sp, SAR_r, SARGEP_sp)
eps	starting cut-off threshold ϵ
delta_eps	starting delta epsilon (subject to change during the reduction process)
n_points	number of calculation points distributed in each simulation
max_error_sp	target tolerances Note: if the number of values provided for the tolerance limit of target species is lower to the number of target species, only the first value will be transcribed and applied to target species
max_error_T	temperature tolerances

Reduction method keywords

max_error_ig	ignition delay time tolerances
max_error_Sl	flame speed tolerances
max_error_K	extinction stretch rate tolerances
inter_sp_inter	application of the Target Species Interaction loop (True/False)
ttol_sensi	sensitivity analyses relative/absolute tolerances

Genetic Algorithm keywords

optim	genetic algorithm optimization (GA/False)
n_gen	generation number
n_indiv	individual number
error_fitness	fitness computation option mean: computation based on the average error / max: computation based on the maximal error among all targets and conditions on all /max
Arrh_max_variation	variation range of reaction rate coefficients B, n and C
optim_on_meth	optimization method constrained to the important reactions identified by reduction methods (True/False/DRG/SA)
nb_r2opt	number of reaction to optimize
selection_operator	selection operator
selection_options	selection operator option
Xover_operator	cross-over operator
Xover_pct	cross-over operator use rate (% of the number of individuals)
mut_operator	mutation operator
mut_pct	mutation operator use rate (% of the number of individuals)
mut_opt	mutation options 'leave empty if not necessary
mut_intensity	probability of a reaction to get its reaction rate coefficients modified by mutation operator
sub_mech_sel	sub-mechanism to optimize