Brookesia

Tutorial for starting with Brookesia

version, 11/05/2020

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

At the moment, Brookesia do not need installation. The only thing you need to do is to download the source files, install the required modules and use it properly!

1.2.1 Necessary python modules for proper use of Brookesia

To use Brokesia, the modules Pandas (https://pandas.pydata.org/) and Scipy (https://www.scipy.org/) must be installed. On a terminal [1] , go to your future reduction working folder then type:

```
conda activate <name_of_your_environment>
conda install pandas
conda install scipy
conda install matplotlib
conda install numpy
```

1.2.2 Through command line on Linux and Mac

On a terminal, go to your future reduction working folder then type:

```
git clone https://github.com/Brookesia-py/Brookesia
```

1.2.3 By downloading the software on Github

1. Go to the following webpage
   https://github.com/Brookesia-py/Brookesia
2. click on **Clone or download** button then **Download zip**
3. unzip the downloaded file on your future working folder

After that, all source files shall be on your working folder and you can directly start to work!

[1] On Windows, start Anaconda then click on:
Environment → Name_of_your_environment → ▶ → Open Terminal

2 Basic 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 **Spyder** (normally installed with the Anaconda suite), load the file GUI_red_mech.py and run it.
By command line, on Brookesia folder, just type:

```
python GUI_red_mech.py
```

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)
- Computational Singular Perturbation - QSSA candidate identification and oriented reactions reduction (CSP)

For all reduction methods, you can specify (Figure 1 a.):

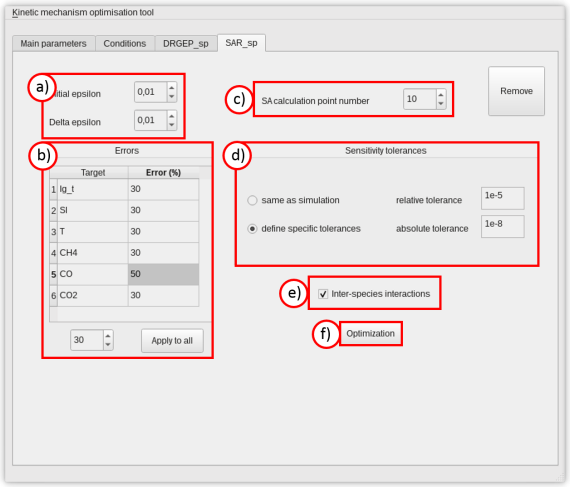
a) The starting epsilon,
b) The starting delta epsilon that will be applied between each iteration (subject to change during the reduction process)
c) The number of calculation points distributed in each simulation
d) The target tolerances
e) The Inter Species Interactions loop
f) The optimization after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

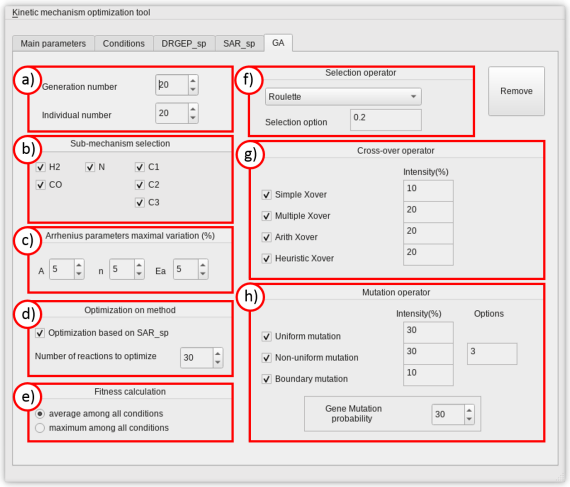
- **Optimization options:**

On the genetic algorithm tab, you can specify (Figure 1 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 1: Composition of the operator tabs (description in section 2.4)

2.5 Output files

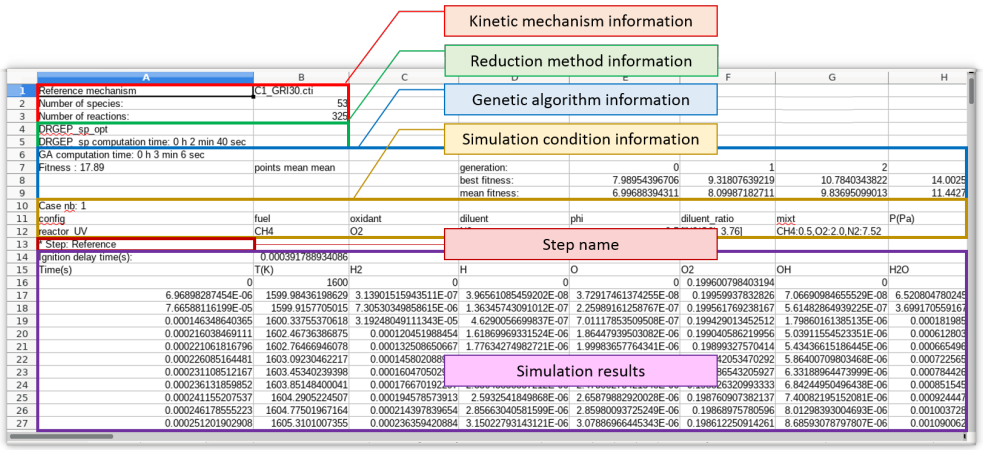
Ones the reduction is finished, you can check the results on the new folder [Date_Time_<name of the folder>], (here: 191225_1200_First_example_folder). It contains the following elements:

- **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

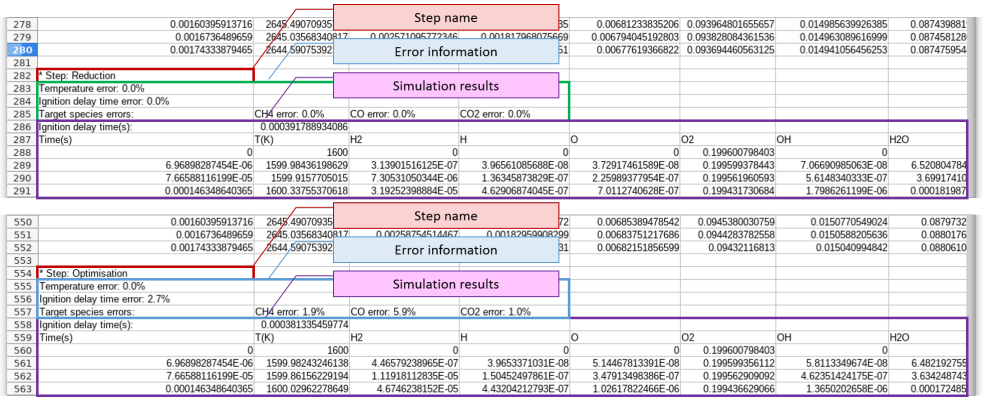
  Chemkin files can also be written by Brookesia, as far as the option "Write Chemkin file" in the Main Parameters tab is selected. Note that, at the moment (release 1.3.), only the kinetic mechanism file is converted (i.e., thermodynamical and transport data files are not). Conversion can also be done 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 2. 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 2 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 2: 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 2). 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`

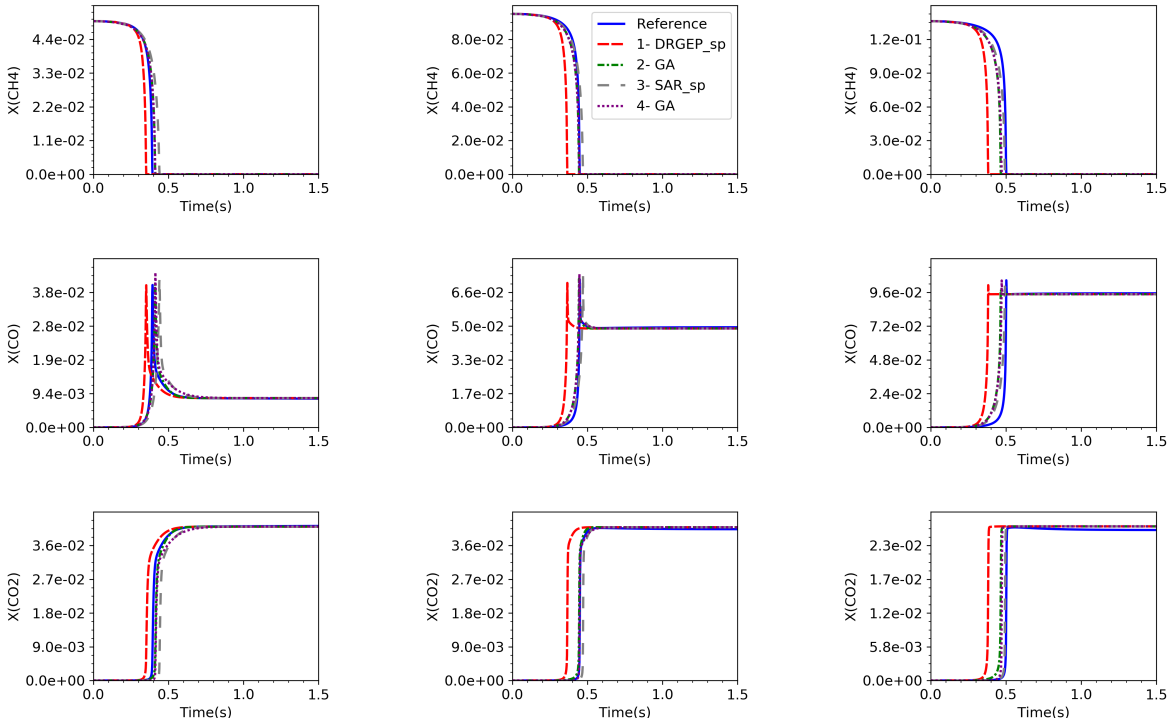
4 First examples

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 3.

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 3: 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_H2_Law	C1_GRI30	C3_Gong_red_a
Species (reaction)	53 (325)	53 (325)	53 (325)	11 (32)	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 Φ = 0.5, 1, 1.5 $p = 10^5$ Pa $T = 600 - 1200$ K	Free flame H_2/air Φ = 0.5, 1, 1.5 $p = 10^5$ Pa $T = 300$ K	Diffusion flame CH_4/air $p = 10^5$ Pa $T = 300$ K	Partially-premixed flame CH_4/air Φ = 1/0.5, 1.5/0.5 $p = 10^5$ Pa $T = 300$ K	Free flame C_2H_6/air Φ = 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 (11)	26 (87)	28 (118)
Computation time	7 min	5 min	2 h 08 min	35 min	40 min	23 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 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             = True
#=======> Optimization
n_gen               = 5
n_indiv             = 5
```

```
error_fitness       = mean
Arrh_max_variation = 5, 5, 5
optim_on_meth       = False
nb_r2opt            = 30
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
sub_mech_sel        = H2, CO, C1, C2, C3, N


#=========> 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
optim             = True
ttol_sensi        = [1e-05, 1e-08]
#=====> Optimization
n_gen               = 5
n_indiv             = 5
error_fitness       = mean
Arrh_max_variation = 5, 5, 5
optim_on_meth       = False
nb_r2opt            = 30
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
sub_mech_sel        = H2, CO, C1, C2, C3, N
```

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"
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
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 (True/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