

NIST Technical Note 2148

Data Reduction Tool for Spherical Constant Volume Flame Experiments

Michael J. Hegetschweiler
Gregory T. Linteris

This publication is available free of charge from:
<https://doi.org/10.6028/NIST.TN.2148>

NIST
**National Institute of
Standards and Technology**
U.S. Department of Commerce

NIST Technical Note 2148

Data Reduction Tool for Spherical Constant Volume Flame Experiments

Michael J. Hegetschweiler
Gregory T. Linteris
*Building Energy and Environment Division
Engineering Laboratory*

This publication is available free of charge from:
<https://doi.org/10.6028/NIST.TN.2148>

April 2021



U.S. Department of Commerce
Gina M. Raimondo, Secretary

National Institute of Standards and Technology
*James K. Olthoff, Performing the Non-Exclusive Functions and Duties of the Under Secretary of Commerce
for Standards and Technology & Director, National Institute of Standards and Technology*

Certain commercial entities, equipment, or materials may be identified in this document in order to describe an experimental procedure or concept adequately. Such identification is not intended to imply recommendation or endorsement by the National Institute of Standards and Technology, nor is it intended to imply that the entities, materials, or equipment are necessarily the best available for the purpose.

National Institute of Standards and Technology Technical Note 2148
Natl. Inst. Stand. Technol. Tech. Note 2148, 30 pages (April 2021)
CODEN: NTNOEF

This publication is available free of charge from:
<https://doi.org/10.6028/NIST.TN.2148>

Abstract

A data reduction tool was developed to conveniently post-process spherical constant volume flame experiments. Such experiments are employed to obtain laminar flame velocities in a premixed gas mixture. The flame velocity is an important combustion parameter and is used, for example, in safety standards and chemical kinetic model validation. The experimental setup is relatively simple, and the only recorded physical quantities are pressure and time. Under various model assumption this so called pressure trace can be used to deduce the flame velocity. Many such models with different levels of complexity are available in the literature. One category of models are the shell models where the vessel content is divided in grid shells and the flame history is reconstructed employing physical models. Three shell models are implemented in the present tool. This technical note describes the software installation and provides a user guide with additional details about the models and the algorithms. Its purpose is to help users efficiently use the tool and obtain physically meaningful results.

Key words

Burning Velocity; Constant Volume Combustion; Data Reduction; Laminar Flame Velocity; Post Processing; Premixed Flames; Spherical Flame

Table of Contents

1. Introduction	1
2. Installation Guide	1
2.1. Overview	1
2.2. Step-By-Step Installation Guide.....	2
2.2.1. Linux (tested on Ubuntu 18.04)	2
3. User Guide.....	5
3.1. Getting Started.....	5
3.1.1. Single Case	6
3.1.2. Multi-Case	18
4. Theory and Implementation	19
4.1. Two-Zone Model.....	19
4.2. Multi-Zone Model MECT	21
4.3. Multi-Zone Model HTDR	22
References.....	24

List of Figures

Figure 1: Main window of the CVDART tool. It is divided into two main sections, the "blue" section with tabs containing single experiments (here called cases), and the "green" section at the bottom where multi case evaluation can be performed.....	6
Figure 2: The reactant species selection window after pressing the “Set Reactant Composition” button when no reactants are selected yet.	7
Figure 3: Reactant species selection window with reactant species and mole fractions. For internal use the mole fractions will be normalized, so in this example there would be 1/3 mole of each species in one mole reactants.	8
Figure 4: Section in the main window to define the pressure trace cropping limits.....	8
Figure 5: Example of a typical pressure trace (left) and its first derivative (right). The indicated cropping limits are obtained by applying the settings shown in Figure 4.....	9
Figure 6: Smoothing parameter selection window for the dp/dt -inflection point search algorithm. Derivative smoothing and a Savitzky-Golay filter acting on the pressure trace can be used.	10
Figure 7: Smoothing scheme and parameter selection for the data reduction model.	11
Figure 8: Data reduction model and parameter selection in the evaluation section.	11
Figure 9: Two-Zone model parameter selection window.	12
Figure 10: Multi-Zone model parameter selection window. On the left side the mask for model type MECT and on the right for model type HTDR.	13
Figure 11: Illustration of the equal radius (left) and equal volume (right) method.	14
Figure 12: Run-time information window showing when running a Multi-Zone model.	15
Figure 13: Extrapolation section of main window for a single case.....	16
Figure 14: Single-case section for opening data visualization figures.	16
Figure 15: Setup window for the boundary dependence plot. The data reduction model type and the lower and upper cropping method and values have to be selected.	17
Figure 16: Example of a plot generated by the "Su Bound Depend" button. The dependency on the lower pressure limit is shown by x-axis value and the different upper limits, by the different colored curves.	17
Figure 17: Multi-Case section of the main window.....	19

1. Introduction

The spherical constant volume (SCV) experiment is a relatively simple way of determining the premixed laminar flame velocity S_u [1], which is a fundamental combustion parameter. It is related to the gas mixture's diffusion rate of heat and species and the rates of chemical reaction and heat release. Additionally, it is used as a scaling parameter for turbulent combustion models, as a flammability metric in safety codes, and to validate chemical kinetic models. Deriving the laminar flame velocity of a SCV experiment is a non-trivial task, particularly for the case of low burning velocities. It would be very helpful for industry to have a convenient tool for this task.

For a generic SCV experiment a spherical vessel is filled with a fuel-oxidizer gas mixture, ignited in the center of the vessel, typically by an electrically generated spark, and subsequently a spherical outwardly propagating flame develops. During the combustion process the vessel pressure increases steadily due to the temperature increase of the burning gas and the gas is compressed. The only recorded physical parameter is the static pressure as a function of time. This pressure trace $p(t)$ can then be used to derive the flame propagation velocity as a function of the unburned reactant temperature and pressure $S_u(p, T_u)$. Often, one is interested in the flame velocity at a near-ambient pressure and temperature (e.g. normal conditions), and therefore extrapolation of S_u is needed.

The present tool is named CVDART (Constant Volume Data Reduction Tool) and provides a convenient way to post-process spherical constant volume experiments and offers data analysis functionality. A typical workflow to evaluate a single experiment is as follows: Enter the initial conditions and the pressure trace, define the raw data smoothing scheme and select a data reduction model with the respective input parameters. After running the model, an extrapolation of the flame velocity can be done and the model results can be visualized in figures. The whole data reduction process can be performed on a single experiment or on a group of similar experiments.

Though the experimental and data reduction processes seem quite simple, effects like buoyancy, radiation, flame stretch, or ignition disturbances can affect the experiments and hence must be considered carefully in order to obtain reliable results. For those topics, the reader is referred to the relevant literature as they are not part of this technical note.

In Section 2 the installation of the program is described, followed in Section 3 by a user guide. Section 4 contains the theory, equations and background information on the models implemented into the tool.

2. Installation Guide

2.1. Overview

The tool is written in Python 3. Additionally, the following native and third party (so called site-packages) packages are needed:

- pip3: Installation tool for python packages (<https://pip.pypa.io>)
- tkinter: Python interface to Tk/Tcl graphic toolkit
- Cantera: Tool suite for solving chemical kinetics, thermodynamic, and transport process problems (<https://cantera.org>)
- matplotlib: Visualization package for Python
- scipy: Scientific calculation package
- numpy: Numerical array based calculation package (matlab style)

Python 3 (Python 2 won't work) must be installed first since all the other packages rely on it. Usually pip and tkinter are included in the standard Python installation, otherwise see the step-by-step instructions (next section) for how they are installed. The third party modules matplotlib, scipy, and numpy are readily installed using pip; e.g., \$ pip3 install numpy, \$ pip3 install scipy, etc. However, the installation package of CVDART does check for these packages and if they are not found it installs them.

Cantera is a thermo-, kinetic-, and transport processes suite and must be installed separately. It is written in C++ but provides interfaces to Python, Matlab, and Fortran; here we only use the Python interface. It is loaded and used by CVDART as a module.

Environmental variables are needed to point to library and other files; however, their definition differs depending on the operating system. Note that the same variable can include more than one directory path. If this is the case the search order is the same as the order of the path definition. Therefore, if different versions of the same package are installed it must be ensured that the correct path precedes the others. The following variables have to be defined:

- PYTHONPATH: Locations of site packages as Cantera, numpy, matplotlib, scipy.
- CANTERA_DATA: Cantera searches these path locations for mechanism files.

See below the step-by-step instruction for further specific details.

2.2. Step-By-Step Installation Guide

2.2.1. Linux (tested on Ubuntu 18.04)

Terminal commands are indicated by a starting dollar sign \$, and commands to be entered in a Python shell by >>>. A Python shell is opened by typing "\$ python" (or "\$ python3" to be sure it is version 3) in a terminal. Setting environmental variables can be done locally in a terminal or by including the command in the ~/.bashrc file. If defined locally in a terminal, they are only valid in this shell and not system wide. To check the content of a variable type

```
$ echo $<variable-name>
```

in a terminal. The arrow bracket notation (as e.g. <variable-name>) is a place holder and must be replaced with the respective variable name.

Some required software packages can be installed (or checked if they are installed) by the Ubuntu package manager “apt-get”. Note, this requires administrator (super-user, root) privileges. The following steps have to be carried out:

1. Check if Python 3 is installed (version ≥ 3.6):
\$ sudo apt-get install python3
(Here you are asked for the superuser/root/administrator password)
2. Make sure pip is installed:
\$ sudo apt-get install python3-pip
3. Make sure tkinter is installed :
\$ sudo apt-get install python3-tk
\$ pip install tk
4. Install Cantera (version ≥ 2.4) (not yet installable via pip):
\$ sudo apt-get install cantera cantera-python3

If a newer version is installed by the package manager and this version does not work together with CVDART, install a downgraded version.

5. Further instructions about installing Cantera are available on <https://cantera.org/install>, for example on how to build Cantera from source or how to install it without administrator permissions.

After installing Cantera check if the module can be loaded in a python shell:

```
$ python
>>> import cantera
```

If this works without an error message then it should be fine.

If there is an error message like “ImportError: No module named cantera” or “ModuleNotFoundError: No module named ‘cantera’”, then source the “cantera_setup” script:

```
$ source setup_cantera
```

If this command gives an error then the “setup_cantera” script is not in your “PATH”. The best option is to search your home directory for the script:

```
$ find ~/. -name setup_cantera
```

and provide the whole file-path as an argument for the source command:

```
$ source <file-path-to-setup_cantera>
```

After that try again:

```
$ python
>>> import cantera
```

In case it still doesn't work, read the instruction on the Cantera site carefully and repeat the installation. Additionally, check that the "PYTHONPATH" environmental variable contains the path to the Cantera site-package. For example if the site package path is `~/.local/lib/python3.6/site-packages` the bashrc entry must be:

```
export PYTHONPATH=$HOME/.local/lib/python3.6/site-  
packages:$PYTHONPATH
```

6. When working with Cantera, a mechanism file is needed as an input. The environmental variable "CANTERA_DATA" can be defined in order to let Cantera know where to look for such files:

```
$ export CANTERA_DATA=<path-1-to-search-directory>:<path-2-to-search-  
directory>:...:<path-N-to-search-directory>
```

Note as indicated above more than one path can be provided.

7. Get the installation file for the CVDART package and install it via pip:

```
$ pip install <path-to-the-cvdart-installation-file.whl>
```

The installer will check for compatible versions of numpy, scipy, and matplotlib and if not found they are installed along with CVDART. In the case where numpy, scipy, or matplotlib are installed along with CVDART, it must be ensured that the environmental variable PYTHONPATH points to the installation location where these modules are installed. If pip was invoked without administrator rights this will be somewhere under the users home directory, e.g. in `~/.local/lib/python3/site-packages`.

```
$ export PYTHONPATH=<path-to-local-site-packages>:$PYTHONPATH
```

8. Now the tool is ready to use. To start, type in a terminal:

```
$ cvdart
```

If the command is not found then that may be because the path to the executable script "cvdart" is not found. Search in the home directory for the name "cvdart":

```
$ find ~/. -name cvdart
```

If found, add the directory path to the environmental variable PATH:

```
$ export PATH=$PATH:<path-to-directory-where-cvdart-resides>
```

3. User Guide

3.1. Getting Started

After the program is installed, it can be started by typing “\$ cvdart” in a terminal. Optionally, a configuration file can be loaded by providing a file path command line argument, i.e. “\$ cvdart <config-file>”. To explain the workflow from scratch we start without a configuration file.

Figure 1 shows the graphical user interface (GUI) opening after executing \$ cvdart.

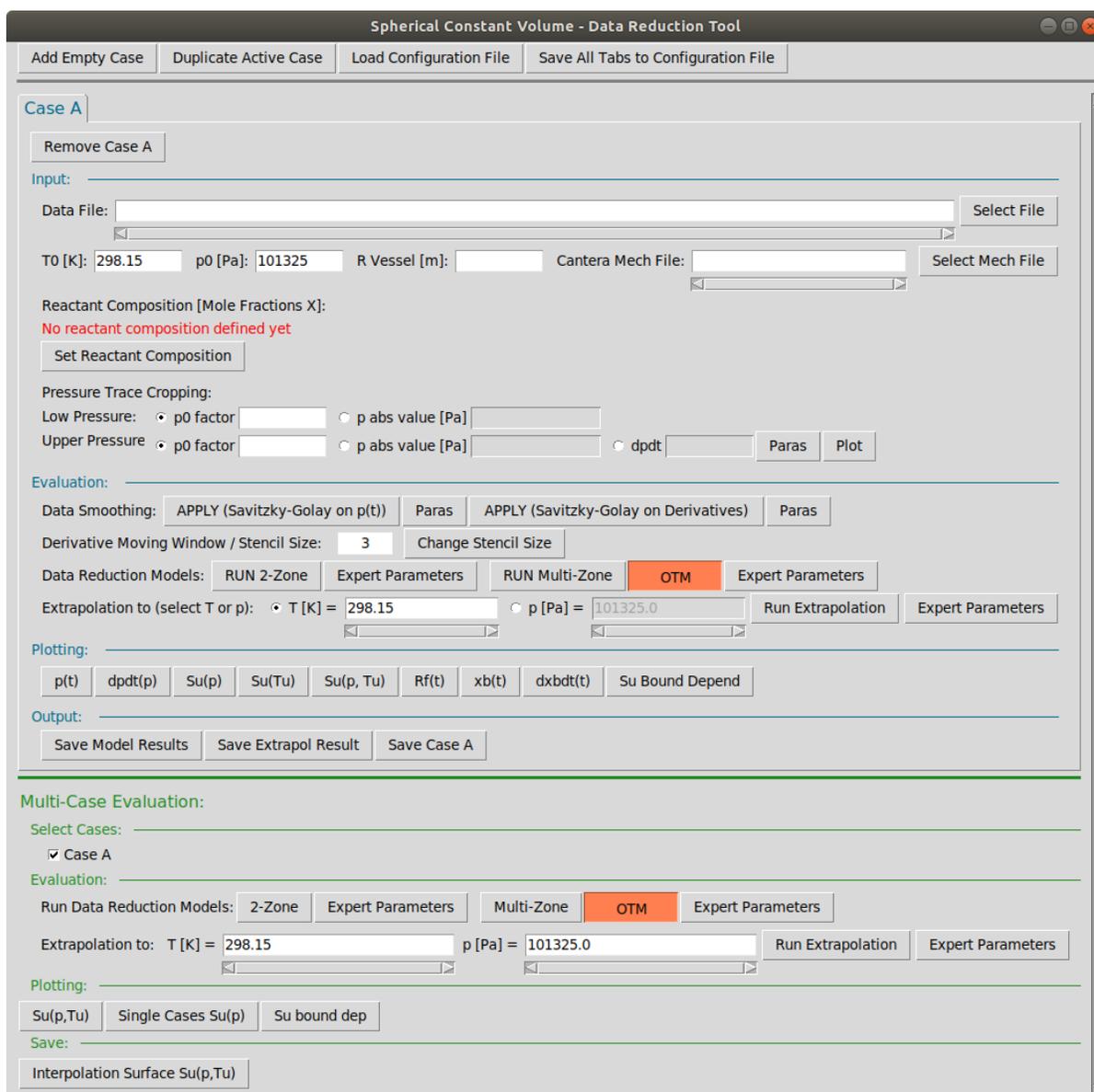


Figure 1: Main window of the CVDART tool. It is divided into two main sections, the "blue" section with tabs containing single experiments (here called cases), and the "green" section at the bottom where multi case evaluation can be performed.

The GUI is designed so that most tasks can be performed by pressing buttons or entering values directly in the main window without clicking through menu trees. The main window is partitioned into the "blue" and the "green" sections. The blue section holds single experiments (here also called cases). Each case is hosted in a tab, and there can be an arbitrary number of single cases. A new tab (Case) can be added by pressing the "Add Empty Case" or "Duplicate Active Case" button at the top of the window. To remove a tab permanently press the "Remove <case-name>" button at the top of the tab. In the blue zone the experimental information is entered, and the entire data reduction process can be performed for a single experiment.

The green multi-case section is designed such that several single experiments can be processed and the results of these experiments are combined to perform the flame velocity extrapolation. Combining similar experiments is a way to increase the reliability and accuracy of the results, in particular the extrapolation process.

In the following two sections a detailed description is provided for building a single case and using the multi case evaluation.

3.1.1. Single Case

There are four sub sections in the blue, single case part of the main window:

- i. Input
- ii. Evaluation
- iii. Plotting
- iv. Output

i. Input

A file with pressure data has to be loaded first. The file can be selected by pressing the "Select File" button or by entering the file path directly in the respective field. The data file must have two columns, one with time and the other with pressure values. Optionally, a header line can be included to indicate the column quantities. Generic names as e.g. time, Time, TIME, p, Pressure, pressure, are recognized. Otherwise the user will be prompted to specify the type of the column data. A typical and accepted data type is the "csv" format with comma separated fields. But it can also be a text file with space or tab separated fields, in this case the file extension doesn't matter.

Note, the units must be SI units, second for time and Pascal for the pressure. Even if different units are specified in the header line they are not recognized.

On the next line of the main window the initial temperature "T0" (Kelvin), the initial pressure "p0" (Pascal), and the vessel radius "R Vessel" (meter) of the experiment must be

entered. Also, an appropriate Cantera mechanism file must be entered (note, only thermodynamic data are required, no reaction rate data). Either the whole file path can be directly entered in the respective field or selected in a file browser by pressing the “Select Mech File” button, or just the mechanism file name can be entered in case the “CANTERA_DATA” environmental variable is set accordingly (meaning it points to a directory where this file resides). The mechanism file must be in Cantera format *.cti. The instruction to convert Chemkin files to Cantera formatted files can be found on the Cantera home page (a conversion script comes along with the Cantera installation called “ck2cti”).

The reactant composition can be specified by pressing the “Set Reactant Composition” button and a window opens as depicted in Figure 2. By pressing the “Add Entry” button a new line for entering a species name and mole fraction is added. Alternatively, the “Add Entry from Species List” can be used to select species from a list of available species and add them to the reactant composition. Note, a valid Cantera file must be already selected in order for this option to work. Adding species from an existing mechanism is particularly useful since the species name syntax can vary for a given chemical, and only the name used in the mechanism will be recognized. The mole fractions of the final species list are not required to sum to one; however, internally, they are normalized so the sum is unity. After pressing the “OK” button the list of reactants is transferred to the main window, where they can also be manipulated.

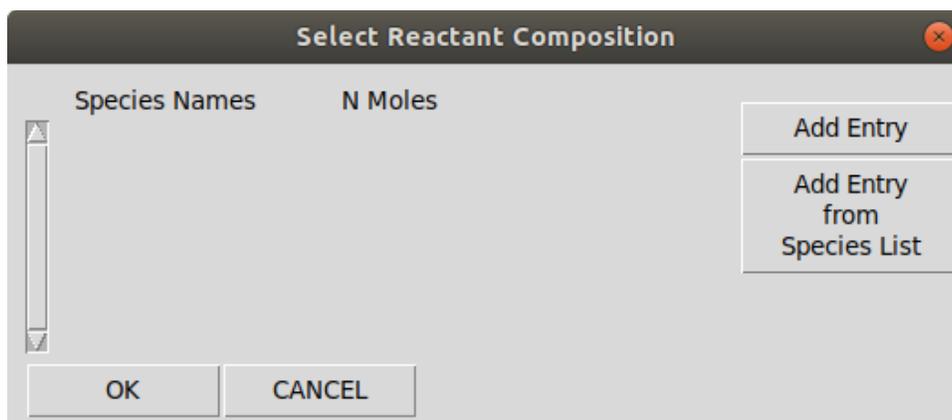


Figure 2: The reactant species selection window after pressing the “Set Reactant Composition” button when no reactants are selected yet.

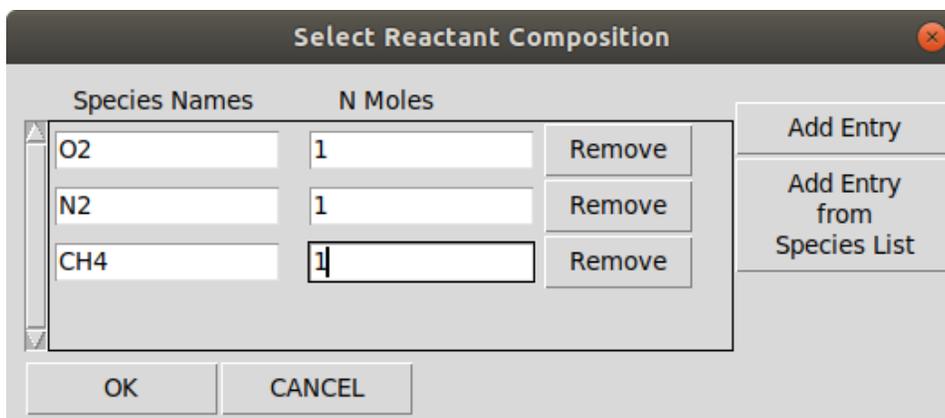


Figure 3: Reactant species selection window with reactant species and mole fractions. For internal use the mole fractions will be normalized, so in this example there would be 1/3 mole of each species in one mole reactants.

Only part of the entire pressure trace is used, and it must therefore be cropped. Defining the cropping boundaries for the pressure trace is the next task. Three different methods are available, two of which are appropriate for both lower and upper limits and one method that can only be invoked for the upper pressure limit. The methods are activated by the respective radio-buttons, as indicated in Figure 4, which shows the relevant “Pressure Trace Cropping” section of the main window. Using the cropping settings of Figure 4, Figure 5 shows a typical pressure trace and its derivative with indicated cropping limits (see the “Plotting” section below for how such figures can be opened).

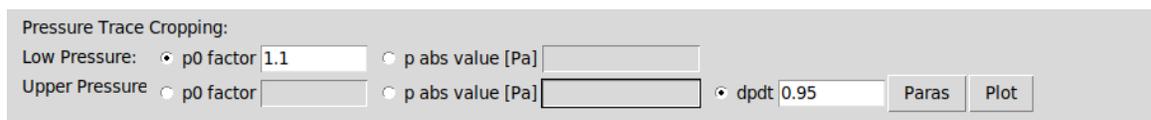


Figure 4: Section in the main window to define the pressure trace cropping limits.

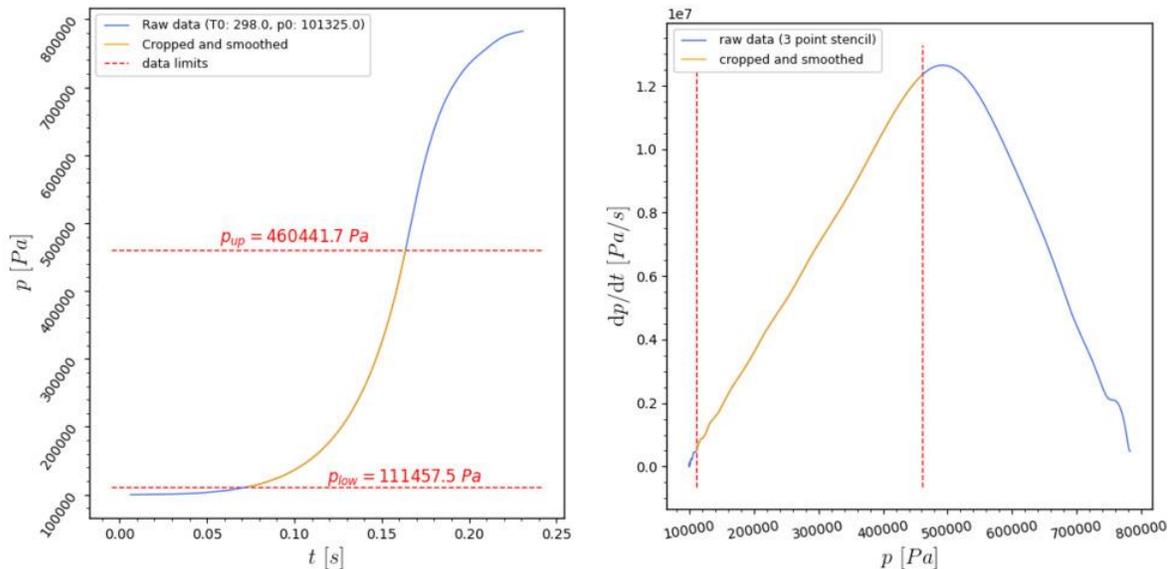


Figure 5: Example of a typical pressure trace (left) and its first derivative (right). The indicated cropping limits are obtained by applying the settings shown in Figure 4.

- “p0 factor”: The factor entered by the user is multiplied by the initial pressure p_0 to obtain the absolute pressure limit. For example, entering 1.25 would set the lower bound below which the data are truncated to 1.25 times the initial pressure p_0 .
- “p abs value [Pa]”: The value entered by the user directly is the pressure limit.
- “dpdt”: This method is only suited for application to the upper pressure limit. The inflection point of the first derivative of the pressure trace is determined and is multiplied by the factor entered by the user. For example, if the user enters 0.9, the upper pressure limit is determined as 0.9 times the pressure at the inflection point. The inflection point is determined automatically; however, depending on the nature of the pressure trace it can be that the respective algorithm fails or returns an inappropriate value. Therefore, a visual check is available and the corresponding figure can be opened by pressing the “Plot” button on the right of the “dpdt” entry field. The figure shows the first and second derivative of $p(t)$ as a function of p . Usually, some smoothing is required, therefore, both raw and the smoothed curves are drawn. The determined inflection point is graphically marked so its validity can be verified.

Depending on the quality of the experimentally measured pressure trace, smoothing is usually required for further data processing that uses the dp/dt criterion for upper limit data cropping, i.e., if derivatives are taken of poor data the errors are amplified. Note that smoothing applied in the inflection point search algorithm is independent of that used in the data reduction model (see below under “Evaluation”). The following two smoothing methods are available:

Savitzky-Golay filter:

The raw data is approximated by a spline which consists of many small spline pieces. A window length and a spline order have to be defined, then least square splines of the specified order and considering the specified number of points are fitted to the raw spline data. The spline pieces are stitched together by fulfilling continuity constraints.

Derivative smoothing:

The derivative at a certain spline point can be calculating by using a 3 point stencil and applying a central differencing scheme (no smoothing). Smoothing can be introduced by considering a larger number of points and using the slope of the linear regression through those points as an approximation of the derivative. The only parameter to choose for this method is the number of points (here also called derivative stencil size) considered for the regression straight.

The smoothing parameters can be adjusted by pressing the “Paras” button right of the “dpdt” entry field. See Figure 6 for the corresponding pop-up window. If no smoothing is required, then 3 should be entered for the derivative stencil size and 3 and 2 for the Savitzky-Golay filter window length and spline order, respectively.

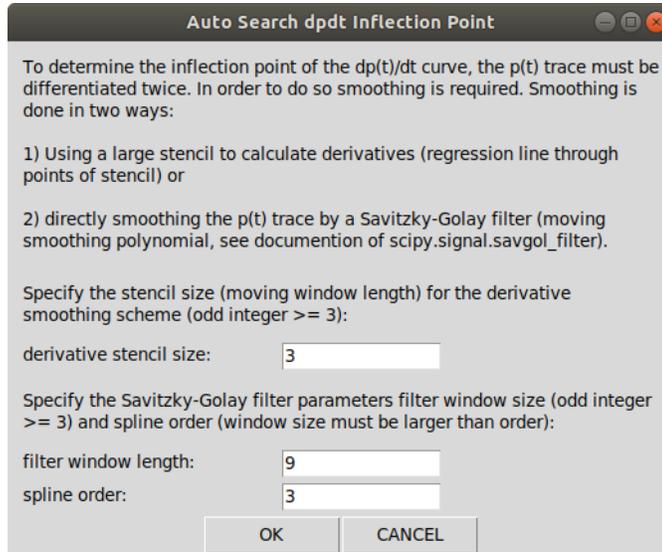


Figure 6: Smoothing parameter selection window for the dp/dt -inflection point search algorithm. Derivative smoothing and a Savitzky-Golay filter acting on the pressure trace can be used.

ii. Evaluation

The evaluation section is the key part of the program. The data reduction model with the respective parameter settings can be defined and extrapolation (or interpolation) of the flame velocity results to a specific pressure and temperature performed.

Data smoothing was already addressed above at the end of the “Input” section. Here in the evaluation section the data smoothing used for the data reduction model must be defined. Three specific smoothing schemes can be selected (see Figure 7 for illustration):

- Savitzky-Golay filtering of the raw pressure traced trace,
- Savitzky-Golay filtering of derivatives, i.e. dx/dt , or dp/dt , and
- Derivative smoothing by using a large stencil size and linear regression.

The first two schemes can be activated by the respective buttons in the main window. If a smoothing scheme is active the button turns red. The parameters for a certain scheme are adjusted by pressing the respective “Paras” button. For the third method (derivative smoothing) only one parameter is needed, the stencil size or moving window length. A value of 3 means no smoothing and with increasing numbers more smoothing is introduced. Note: The moving window / stencil size for both the Savitzky-Golay filter and for the derivative smoothing must be an odd number.

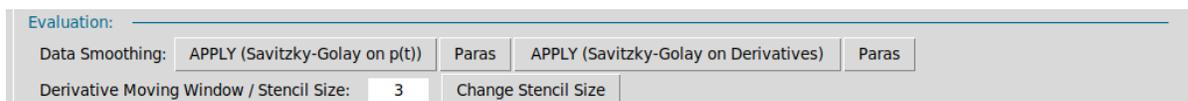


Figure 7: Smoothing scheme and parameter selection for the data reduction model.

Now a data reduction model can be selected and executed. A Two-Zone model and 2 Multi-Zone models are available. For both model families there is a run and an expert parameters button (see Figure 8). Pushing the parameter button opens the parameter selection window. There are physical model parameters and algorithmic/numerical parameters. More details about the models and the algorithmic parameters can be found in section 4. Here only the physical model parameters are explained.

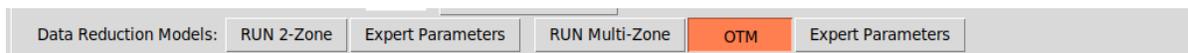


Figure 8: Data reduction model and parameter selection in the evaluation section.

Figure 9 shows the Two-Zone model parameter window opened by pressing the “Expert Parameters” button right of the “RUN 2-Zone” button. Only two physical model parameter can be adjusted, the equilibrium method and the number of pressure steps (see Figure 9 first three lines). The equilibrium method parameter defines under which condition the equilibrium calculation for estimating the burned gas conditions is performed. Available are

“UV” and “HP”, which means either keeping internal energy and specific volume constant, or keeping enthalpy and pressure constant. Experience show that usually “HP” is the better choice. The number of pressure steps is internally used to calculate a pressure increment as $dp = (p_{end} - p_0)/N$, where p_{end} is an estimated final pressure value after all reactants in the vessel have been burned. The larger the number of steps the smaller the pressure increment and the better resolved is the model simulation. For each mixture the minimum number of steps required for a well resolved solution may vary. A good way to check if the resolution is appropriate is to consult the $Su(p)$ or $Su(Tu)$ plot (see the “Plotting” section below). When these functions are smooth then the number of pressure steps is large enough. A good value to start is around 500. Note, the Two-Zone model runs quite fast therefore a rather large number of steps can be used.

The remaining parameters in the 2-Zone model solver parameters window are explained in section 4.1.

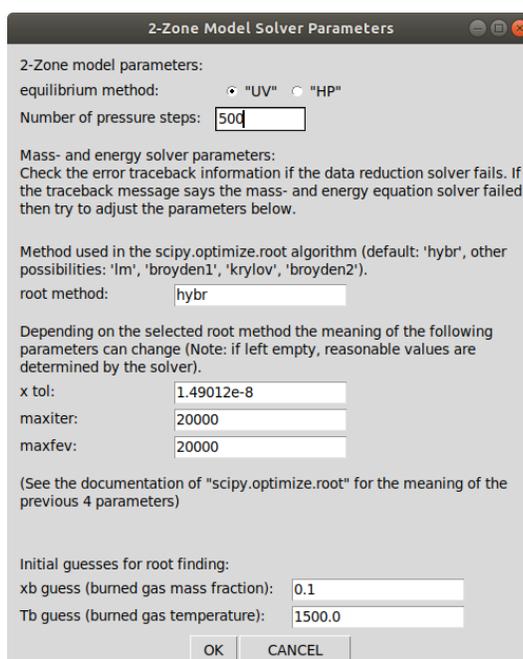


Figure 9: Two-Zone model parameter selection window.

Figure 10 shows the parameter setting window of the two Multi-Zone models. At the top the model type can be selected. Available are the Mass- and Energy Conserving Thermo (MECT) and the Hybrid ThermoDynamic-Radiation (HTDR) models (for details see section 4). Since a slightly different set of parameters is needed for the two models, the window mask changes when switching the model type. First the parameters that are similar for the two models are explained.

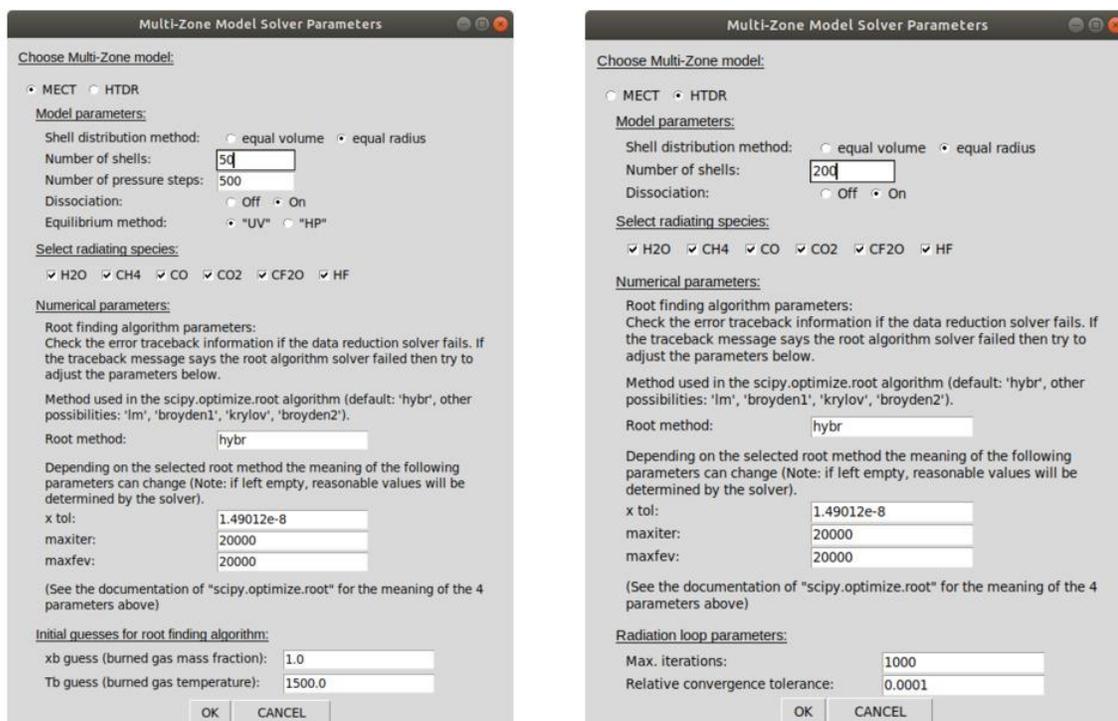


Figure 10: Multi-Zone model parameter selection window. On the left side the mask for model type MECT and on the right for model type HTDR.

The shell distribution method defines how the vessel is divided into a one-dimensional radial computational grid (or shells). The selection is between two methods named “equal volume” and “equal radius”. As the name suggests the “equal volume” methods leads to shells with similar volumes. Consequently, the radial shell thickness is decreasing with increasing vessel radius. The “equal radius” method distributes the shells radially even. An illustration of resulting grids is depicted in Figure 11.

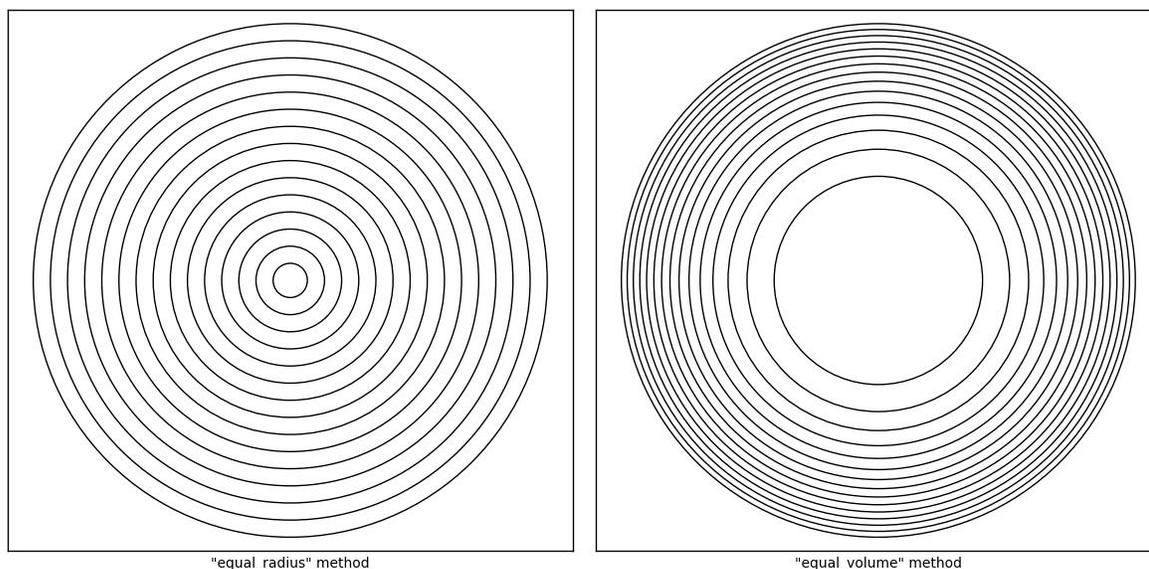


Figure 11: Illustration of the equal radius (left) and equal volume (right) method.

The next parameter to choose is the number of shells. Depending on the model type, the shell distribution method and the reactant mixture, appropriate values may vary a lot. Best practice is to make a grid convergence study. Usually the MECT model needs fewer shells than the HTDR model. Reasonable values are 50 and 200 for the MECT and the HTDR model, respectively.

On the next line the model for dissociation can be switched on and off. Depending on the chemical kinetics, dissociation can have a considerable effect on the burning velocity.

For the HTDR model, no more parameters are needed, whereas for the MECT model, the number of pressure steps and an equilibrium method can be adjusted. The meaning and use of the number of pressure steps is similar as described above for the Two-Zone model. Here the ratio of the number of pressure steps and number of shells is important. Usually a ratio of 10 to 20 is appropriate (i.e., number of pressure steps is 10 to 20 times the number of shells). But again it depends strongly on the particular case (for further detail see section 4.2).

A last parameter to select for the MECT model is the "Equilibrium method". This parameter defines which physical quantities to hold constant during mixture equilibration. Reactant mixture equilibration is used, e.g., to model the burned gas composition. As already explained for the Two-Zone model either constant internal energy and specific volume ("UV") or constant enthalpy and pressure ("HP") can be selected. Based on theory it is not entirely obvious which method is more appropriate, but experience show that "HP" is usually the better choice.

Both Multi-Zone models have the capability to account for the limiting case of optically thin radiation. Not every species exhibits radiation properties strong enough to have an influence on the flame velocity, or it can simply be that the radiation properties for a species is not known yet. Therefore, only a limited list of species is given in the parameter selection window and each of these species can be activated or deactivated. An activated species

means that it is considered for radiation. Typically, one wants all of them active, but for sensitivity studies it is convenient to have control over individual species.

As mentioned earlier, the remaining settings are for algorithmic parameters and are not discussed here; more information is given in sections 4.2 and 4.3.

The parameter setting window can now be closed. Return to the main window before running a Multi-Zone model, where radiation can be switched on or off by the “OTM”/”ADI” button. OTM means the optically thin radiation model is active, and ADI is the abbreviation for an adiabatic treatment (no radiation). Note, the button turns red if radiation is active.

Now the program is ready to start a data reduction model by either pressing the “RUN 2-Zone” or “RUN Multi-Zone” button (see Figure 8). The Two-Zone model typically runs very fast, whereas the Multi-Zone models can consume more time. Therefore, when running Multi-Zone models a window pops up with some run-time information and it is possible to abort a calculation. Note, the results of a canceled model simulation are still available for further processing, saving, or plotting. See Figure 12 for an example of a runtime information window.

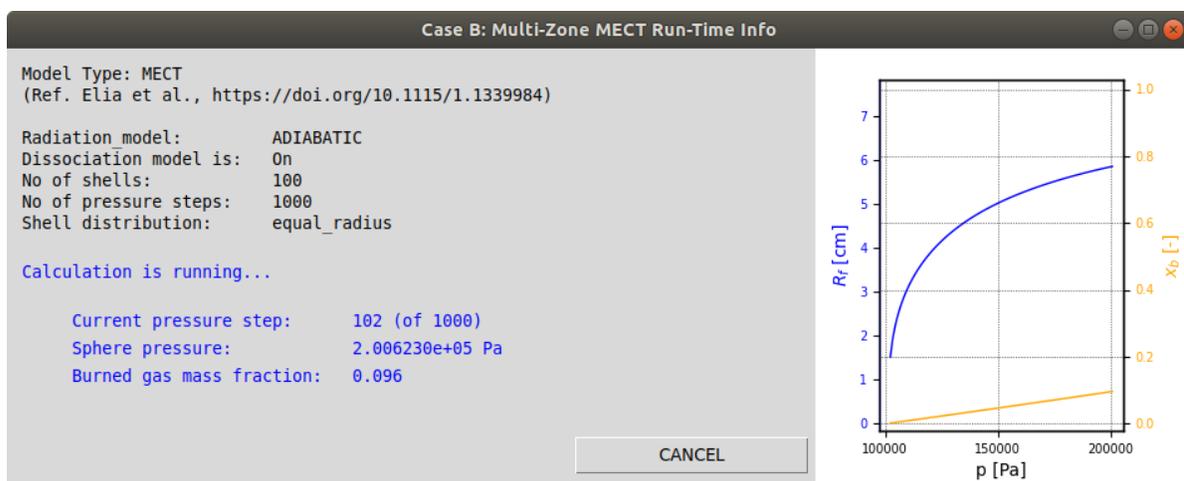


Figure 12: Example of a run-time information window showing when running a Multi-Zone model.

If a simulation fails another window opens with some error information. A typical reason for a failed simulation is an inappropriate grid, e.g., too coarse or a wrong ratio between shell numbers and pressure steps for the MECT model.

After a simulation has successfully finished, extrapolation (or interpolation) of the flame velocity to a certain pressure/temperature can be conducted. See Figure 13 for the respective section in the main window. Not all combinations of pressure and temperature are physically meaningful. Only points on an isentrop through the initial pressure p_0 and temperature T_0 are appropriate. Therefore, either the pressure or the temperature can be entered and the other quantity is calculated internally. By pushing the “Run Extrapolation” button the extrapolation

process is executed. If the extrapolation was successful, a window with results appears; in the case of failure, an error message window appears. In the latter case it is worth checking the $S_u(p, T_u)$ function (see the “Plotting” section how to do that). There is also the possibility to adjust parameters of the extrapolation solution algorithm. By pressing the “Expert Parameter” button on right of the “Run Extrapolation” button the corresponding window appears. Information about these parameters are given in this window.

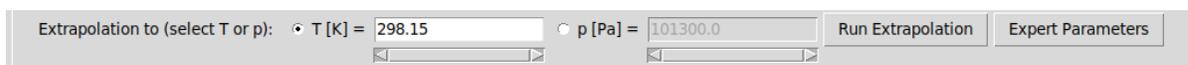


Figure 13: Extrapolation section of main window for a single case.

iii. Plotting

In order to check and visualize both input and resulting data, several parameter combinations can be plotted as indicated in Figure 14. The first two buttons “p(t)” and “dpdt(p)” show the raw data of the pressure vs. time and its first derivative vs. pressure together with their respective smoothed splines (using the user specified smoothing methods and parameters). These two plots use the input and smoothing settings only; no data reduction model needs to be run.

The button names to open the other figures “Su(p)”, “Su(Tu)”, “Su(p, Tu)”, “Rf(t)”, and “dxdbdt(t)”, in which Su is the burning velocity and Rf is the flame radius, are self-explanatory.

Note: Data reduction models can be run as many times as desired but only the most recent run remains in the memory and can be visualized or written to a file.



Figure 14: Single-case section for opening data visualization figures.

A special figure is created on pressing the “Su Bound Depend” button. It is an analysis of the dependency of the extrapolated flame velocity S_u^0 on the choice of lower and upper pressure crop limits. Such plots are very convenient to assess the sensitivity of S_u^0 with respect to the selected pressure range. Pressing the “Su Bound Depend” button first opens a window where the type of data reduction model and the methods and values to determine the lower and upper pressure range bounds can be selected (see Figure 15). The same cropping methods are available as explained earlier in the “Evaluation” section. The data reduction parameters are the ones specified in the respective parameter selection window. An example of such a plot is shown in Figure 16. The x-axis indicates the lower pressure limit as a multiple of the initial pressure p_0 , and each of the colored curves represents a different upper limit, expressed as a

fraction of the pressure at the inflection point of the dP/dt curve; e.g., the red line is for a fixed upper limit (here 1.0 times the pressure at the dP/dt inflection point) and a range of lower pressure limits.

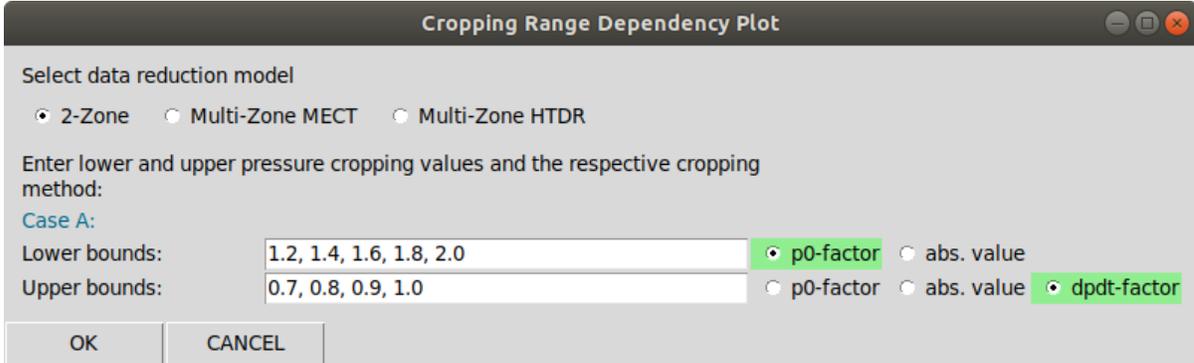


Figure 15: Setup window for the boundary dependence plot. The data reduction model type and the lower and upper cropping method and values have to be selected.

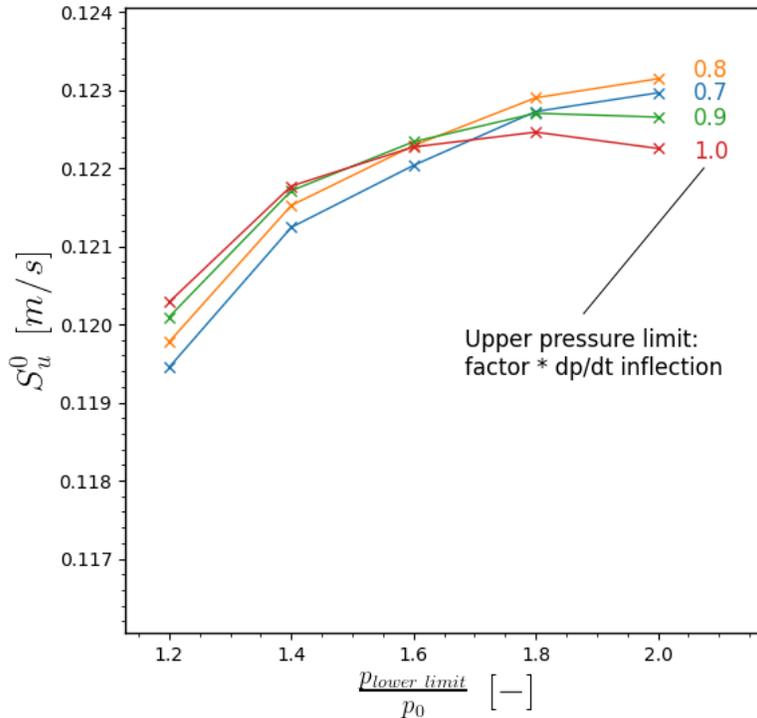


Figure 16: Example of a plot generated by the "Su Bound Depend" button. The dependency on the lower pressure limit is shown by x-axis value and the different upper limits, by the different colored curves.

iv. Output

Pressing the “Save Model Results” button writes the following quantities to a file: time (s), pressure (Pa), time derivative of pressure dP/dt (Pa/s), temperature of the unburned mixture T_u (K), ratio of specific heats of the unburned mixture γ_u (-), burned gas mass fraction x_b (-), mass averaged burned gas temperature T_b (K), time derivative of burned gas mass fraction dx_b/dt (1/s), flame front radius R_f (m), and flame velocity S_u (m/s).

Results of the flame velocity extrapolation are saved via the “Save Extrapol Result” button.

To save the current single-case configuration the button “Save <case-name>” must be pressed. This file can then be used to reopen the tool in the exact same state.

Note: Saving the configuration here will save only this tab or case! For saving all the tabs/cases use the button “Save All Tabs to Configuration File” on the top of the window.

3.1.2. Multi-Case

It is a common practice to repeat the same experiment several times with slightly different initial pressure and/or temperature, and then using the combined flame velocity results to apply the extrapolation fit equation. This increases the accuracy of the flame velocity extrapolation. The multi-case section is designed for this task.

Those single cases to be included in the multi-case evaluation are selected by the radio button list. Note, each time a single case is added or removed this list is automatically updated. A data reduction model for each activated single case is run by pressing one of RUN buttons. The model parameters are the ones set in each single-case tab. To synchronize these parameters, the “Expert Parameter” button should be used and after accepting and closing the parameter window all active cases have the same set of parameters.

Note: After running the data reduction model, the results of all active cases will also be available in the single case section and figures of individual results can be opened or saved and extrapolation performed.

Performing the extrapolation by pressing the “Run Extrapolation” button collects the results of all selected single cases and applies the least square fit to the combined results, which can then be used to extrapolate to conditions of interest (typically ambient conditions); however, before running extrapolation, the extrapolation point must be selected. In contrast to a single case, the extrapolation can be run to arbitrary p and T ; hence, it is the user’s responsibility to select reasonable points for extrapolation.

There are two visualization plots available for the burning velocities calculated from the pressure rise data of multiple cases. The “ $S_u(p, T_u)$ ” figure shows a 3D plot with the included single-case flame velocities, the least square surface fit, and the extrapolated flame velocity. The other figure “Single Cases $S_u(p)$ ” is a 2D cut along each of the individual $S_u(p)$ curves and shows $S_u(p)$ itself (data reduction model result) and the respective cut of the surface fit. This figure helps to assess visually the quality of the surface fit at the positions of the single case results. The same analysis plot as for the single cases is available showing the

dependency of the extrapolated flame velocity on the lower and upper cropping limits. Pressing the “Su bound dep” button opens a similar window as in the single case where data reduction model and crop limits can be selected.

The surface fit data is saved by pressing the “Interpolation Surface Su(p,Tu)” button.

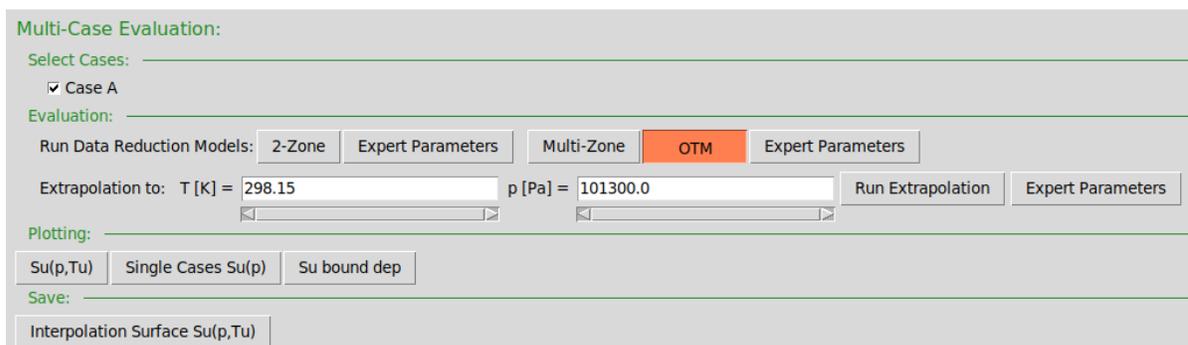


Figure 17: Multi-Case section of the main window.

4. Theory and Implementation

More detailed information about the data reduction models is given here. This should help the user to select the appropriate data reduction model and parameters.

4.1. Two-Zone Model

The model is based on the work of Metghalchi [1]

The following model assumption are made:

- The fluid in the sphere is divided into a burned and an unburned region, separated by a smooth, infinitely thin, and spherical flame front
- Both regions have uniform properties.
- No chemical reactions occur in the unburned gas and its composition remains the same at the composition of the first unburned region.
- Only pressure and temperature increase due to isentropic compression driven by the larger volume of the burned gas.
- The pressure is always uniform in the entire fixed volume combustion vessel.
- Spark energy and heat conduction to the vessel wall are not considered
- Radiation is not considered

A mass- and energy conservation equation can be derived for the gas in the vessel as follows:

$$v = \int_0^{x_b} v_b dx' + \int_{x_b}^1 v_u dx'$$

and

$$e = \int_0^{x_b} e_b dx' + \int_{x_b}^1 e_u dx'.$$

v and e are the total specific volume and internal energy in the vessel and subscript b denotes values in the burned gas and u in the unburned gas. The integration is over the burned gas mass fraction x_b . Note that v_b , v_u , e_b , and e_u are all constant here and the evaluation of the integrals is trivial. Furthermore, these are static equations which hold for each pressure level. The unknowns in these equations are the burned gas mass fraction x_b and the temperature of the burned gas T_b . The quantities v_b and e_b both depend on the unknown T_b and therefore, an iterative process is needed to solve the system of equations. A root finding algorithm of the package `scipy.optimize` is employed for this task. Different parameters can be adjusted for convergence improvement:

- Root method
- Relative convergence criteria
- Maximal number of iterations

Depending on the root method there are more parameters. For more details, consult the help page of the `scipy.optimize.root` function. Besides the numerical parameters, initial guesses of the unknowns x_b and T_b can be provided. This can help when the algorithm does not converge or a wrong solution is found. Usually it is a good guess to start with x_b between 0 and 1 and T_b depending on the case between 1500 and 3000 Kelvin.

To solve the conservation equations an estimation of the burned gas composition is required. This is modeled by using the chemically equilibrated reactant gas mixture.

In a pressure loop the conservation equations are solved and the burned gas mass fraction x_b and the burned gas temperature T_b for each pressure level is obtained. With this information the flame radius depending on the pressure $R_f(p)$ can be calculated and finally the flame velocity by

$$S_u(p, T_u) = \frac{R_v^3}{3R_f^2} \left(\frac{p}{p_0} \right)^{(-1/\gamma_u)} \frac{dx_b}{dt} \quad \text{Eq. 1}$$

The quantities R_f , γ_u , and x_b are taken from the simulation and p is the experimentally measured pressure.

4.2. Multi-Zone Model MECT

The basics of this model are explained in Elia et al. [2]

The same requisites as for the Two-Zone model are valid here. However, instead of uniform burned gas conditions gradients are allowed. The spherical vessel is divided into a number of grid shells and shells in the burned region can have different states. In each shell the conditions are uniform and no mass or energy is exchanged between shells.

One distinguishes between three types of shells:

- Unburned shells: All these shells are in the same state (similar as in the Two-Zone model). During the simulation this state changes only by isentropic compression and no chemical reactions occur.
- Burning shell: The location of the flame front defines this shell. It consists of a burned and an unburned region separated by the flame front. The unburned part is treated similar as the gas in the unburned shells. Mass- and energy conservation equations are solved with the burning shell as control volume (compared to the Two-Zone model where these equations are solved for the entire vessel). The burned gas mass fraction and temperature are solutions of the conservation equations. The composition of the burned gas is estimated by equilibrating the reactants, starting with the current temperature and pressure of the unburned mixture.
- Burned shells: As soon as the flame front crosses an outer shell boundary this shell belongs to the third category of (previously) burned shells. As in the case of the unburned gas these shells are compressed isentropically. However, the earlier in the simulation a shell is fully burned the more it is compressed which leads to different temperatures in the different burned shells. Additionally, models for radiation and dissociation can be applied. Depending on the radiation and thermodynamic properties of the mixture, those effects can have a large influence on the flame velocity. For example, for slow burning mixtures with strongly radiating species in the products, the temperature distribution in the burned gas can differ by several hundred K.

The following process is applied to obtain the flame history. First a final vessel pressure is estimated for the state when all reactants are consumed. This is done by using the equilibrium state of the reactants. Then a pressure increment is calculated by dividing the difference of the estimated final pressure and the initial pressure by the number of pressure steps. The number of pressure steps is a user defined value and should be in a reasonable ratio to the number of shells. For example, if a large number of shells is combined with a low number of pressure steps, then during each pressure step the flame crosses several shells which makes no sense. A value of 10 is reasonable for the ratio of pressure steps to the number of shells; however, the appropriate value for this ratio can vary since it depends, among other on the flame velocity.

Next a pressure loop is started from pressure p_0 to the final pressure p_{end} with the increment as explained above.

The quantities: burned mass fraction x_b , flame radius R_f , and the unburned gas properties are derived for each discrete pressure level and then used to obtain the flame velocity $S_u(p, T_u)$. A pressure loop involves the following steps:

- Decision on which shell is currently burning (in which shell the flame front resides).
- Isentropic compression of all unburned shells by the pressure increment.
- Evaluation of combined effect of isentropic compression, radiation heat loss, and dissociation in all burned shells.

Calculation of radiation heat loss needs time information which is only available in the experimental pressure trace $p(t)$, therefore the pressure of the pressure loop must be aligned with the experimental $p(t)$ and thus a time increment dt corresponding to pressure increment dp can be derived.

The effect of dissociation is modeled by equilibrating the mixture (keeping enthalpy and pressure constant) of each burned shell using the updated pressure and temperature.

- The mass- and energy equations are solved for the burning shell. A similar algorithm as explained for the Two-Zone model is applied and also similar parameters can be adjusted.
- The new flame radius is calculated by

$$R_f = \left(\frac{3}{4\pi} M_{vessel} \sum_{shells} x_i v_i \right)^{1/3}, \quad \text{Eq. 2}$$

where M_{vessel} is the total gas mass in the vessel and x_i and v_i are the burned gas mass fraction and the specific volume in shell i . The summation $\sum x_i v_i$ is over all shells and describes the total burned gas mass fraction.

After finishing the pressure loop, the flame velocity can be calculating similarly as for the Two-Zone model.

4.3. Multi-Zone Model HTDR

The implementation of this model is based on the original work by Xiouris et al. [3]

The requisites and the shell treatment are the same as in the MECT Multi-Zone model. However, instead of defining a pressure step, in each loop an entire shell is burned and no conservation equations are solved. The model ensures by design mass conservation and with increasing shell numbers the energy defect tends to zero. Compared to the MECT model a larger number of shells is required for a grid converged solution since an entire shell is burned in one step and a coarse shell distribution leads to large pressure jumps. Additionally, as mentioned above the energy defect is larger with fewer shells.

The following loop is repeated until all shells are burned:

- In each loop, a new shell is burned starting with the innermost shell. The burned gas conditions are modelled by the equilibrium state of the reactants. Compared to the MECT model not only the equilibrium gas composition is used but also the equilibrium temperature. The increased temperature and keeping the shell mass constant leads to a larger shell volume.
- The increased volume of the burning shell is compensated by compressing all other shells. This is done in an iterative process where all shells are compressed isentropically by the same pressure increment until the summed volumes of all shells is equal the vessel volume. The size of the pressure increment is a result of this process and that explains why a larger number of shells is needed for a converged solution compared to the MECT model. The pressure increment size is directly linked to the volume of the burning shell.
In case the effects of radiation and/or dissociation are taken into account the respective models are applied to the burned shells before compressing the shells. For estimating the radiation energy loss, a time scale is needed and a similar approach as described for the MECT model is used (aligning of simulated and real pressure trace). The problem is that the pressure increment is not known before compressing the shells. Therefore, an additional iterative loop is needed, in a first iteration step the shell compression is calculated without accounting for radiation. With the resulting pressure increment, the radiation loss energy can be estimated and the compression step is repeated. This is done until the pressure increment convergences.
- Burned gas mass fraction x_b and specific volume v of all shells are now determined and by using Eq. 2 the new flame radius R_f can be calculated.

Similarly to the other models, the flame velocities $S_u(p, T_u)$ are calculated via Eq. 1 after the pressure loop is finished. Note that as mentioned above, x_b and R_f are taken from the simulation and the pressure $p(t)$ are the experimentally measured pressure values.

Acknowledgments

The authors would like to thank Dr. John Pagliaro and Dr. Jagannath Jayachandran for fruitful discussions and input concerning the physical modeling and treatment of experimental results. This work was supported by the Buildings Technologies Office of the U.S. Department of Energy, Office of Energy Efficiency and Renewable Energy under contract no. DE-EE0007615 with Antonio Bouza serving as Project Manager.

References

- [1] Metghalchi M , Keck JC (1980) Laminar burning velocity of propane-air mixtures at high temperature and pressure. *Combust Flame* 38:143-154.
- [2] Elia M, Ulinski M, Metghalchi M (2001) Laminar Burning Velocity of Methane–Air–Diluent Mixtures. *J Eng Gas Turbines Power* 123(1):190.
<https://doi.org/10.1115/1.1339984>
- [3] Xiouris C, Ye T, Jayachandran J, Egolfopoulos FN (2016) Laminar flame speeds under engine-relevant conditions: Uncertainty quantification and minimization in spherically expanding flame experiments. *Combust Flame* 163:270-283.
<https://doi.org/http://dx.doi.org/10.1016/j.combustflame.2015.10.003>