Processing Unit for Laser flaSh Experiments (PULsE)

User Manual

Version v1.91FM

Artem Lunev & Robert Heymer

June 4, 2021

Contents

Introduction

PULsE is an open-source, cross-platform and hardware-independent software suite described in Lunev and Heymer [2020], Lunev et al. [2020, 2021] that may be used as a substitute to commercial software packaged with the laser flash analysis (LFA) instruments. The idea behind releasing this software is to grant a wide audience of users easy access to the state-of-the-art numerical solvers and parameter estimation techniques specifically adopted to address experimental conditions of varying complexity. Although aimed mainly at expert users with prior knowledge of the method, thanks to the intuitive interface, PULsE can also be used by novices in the field. The purpose of this guide is to give a comprehensive overview of several aspects of working with PULsE, including installation, running, and basic functionality.

1 Installation

PULsE is distributed in `.zip` and `.tar.gz` archives. Current version can be downloaded by clicking at the respective buttons shown in the web-page `https://kotik-coder.github.io/`. All PULsE versions are archived on `git`, so it's possible to download an older version if required for any reason.

The basic installation steps are listed below.

(a) Download and extract the contents of the PULsE package, which should include the runnable `.jar` file and a folder with external libraries. Do not rename these files.

(b) Please note that PULsE requires Java 11 to run.

If your are not familiar with Java, either you won't have it installed on your machine or will only have Java 8. Java 8 and Java 11 are very different and you can't run PULsE if you only have Java 8, it simply won't start. Unfortunately, it is Oracle's policy to prevent Java 9+ being installed on corporate machines without paying for an expensive license first. **However, individual users do not have to pay to download Java 11**. Therefore, we ask you to download it manually.

You can use PULsE on any operating system that supports JDK 11, including 64-bit `Windows`, `Linux`, `MacOS`, and potentially `Solaris`. Note that because Oracle does not offer support to 32-bit architecture starting from Java 9, you cannot run PULsE currently on a 32-bit operating system.

On 64-bit Windows, Java 11 needs to be downloaded from the Oracle website.

On Linux, the installation package is downloaded directly from the OpenJDK repo. The process is straightforward. Open the Terminal and type in the following commands:

```
sudo add-apt-repository ppa:openjdk-r/ppa
sudo apt-get update
sudo apt-get install openjdk-11-jre
```

(c) Ensure the PULsE.jar is runnable. On Linux, to allow the program to execute, open a terminal in the same folder as PULSE.jar, and into this type:

```
chmod +x <name-of-the-jar-file>.jar
```

(d) Run the `.jar` file by double-clicking on it.

If it doesn't start

If you are experiencing problems, please open the Terminal or the Command prompt and type:

```
java -jar <name-of-the-jar-file>.jar
```

Please send the generated output to the developer.

2 Version Control

Starting from v1.88, PULsE automatically checks if current version is the latest. This requires an Internet connection on your PC in order to download meta-information from the server. If a newer version is available, you will be prompted to visit the web-site at program start.

3 Runtime errors

PULsE catches most of runtime exceptions. Upon encountering an error, PULsE will display a message dialog on the screen, signalling that something went wrong during execution. When an exception occurs, a detailed stack trace is written to a log file located in the directory where the `.jar` file is installed. A shutdown hook is associated with empty logs that will be automatically deleted when PULsE exits. Non-empty logs should be sent to the developer by e-mail [1].

[1] E-mail: alounev<at>list.ru (replace <at> with @)

4 Input file formats

4.1 Netzsch Proteus Export (`.csv` format)

On Netzsch devices, the Proteus software is used for measurement and analysis. LFA Analysis is able to export the detector signal and pulse directly to individual `.csv` files. These files can then be imported in PULsE. Because Proteus uses locale-specific decimal separators and delimiters, the user needs to switch the language of the software to English or customize the export format, otherwise the input files will not be recognised by PULsE.

In Proteus, click `File` → `Output` → `Export Detector Signal` → `Export Data...` Select all shots that you wish to export. Proceed with exporting the files by clicking "OK" in the main window. Repeat this for the pulse data you wish to export, which may be accessed via `File` → `Output` → `Export Laser Pulse` → `Export Data...`

For more information on how importing of Proteus ASCII files is managed in PULsE, please refer to this section of the Javadoc

Check the locale-specific separators

In the "`Export Detector Signal`" window, click the "`Format...`" button. To change the decimal and column separators, select the "`Custom`" radiobutton in the "`Export Format`" pop-up window. Please make sure that "`Column separator`" is set to "`comma`" and the decimal symbol is set to "`point`".

4.2 Linseis `.LFR` format

PULsE accepts the formats of files output by Linseis LFA systems. Note that by default the measurement results are stored in a binary (.lfi) format, but it is possible to export those curves in ASCII following the procedure below.

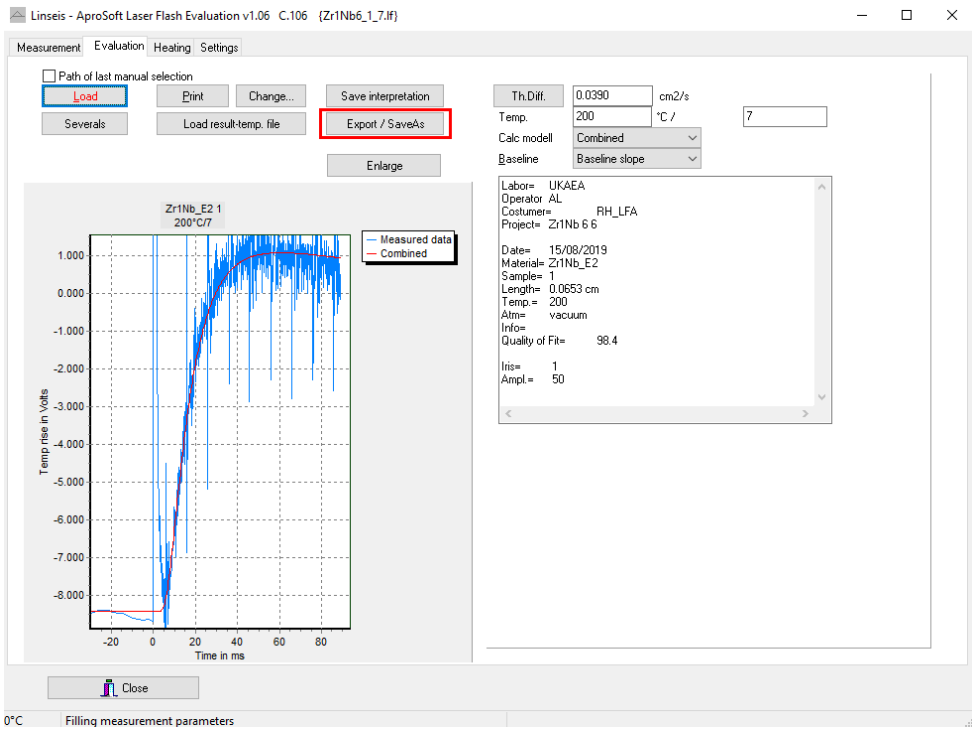
(a) After performing a shot, clicking "Export/Save As" in the Linseis AproSoft program (fig. 1a) will export the data for the heating curve recorded as a .txt file (fig. 1b). This should be done for any shot or curve you wish to analyse in PULsE.

(b) After all data has been recorded, click "Severals" in the Linseis analysis window (fig. 1a) and select all exported heating curve .txt files for the experiment.

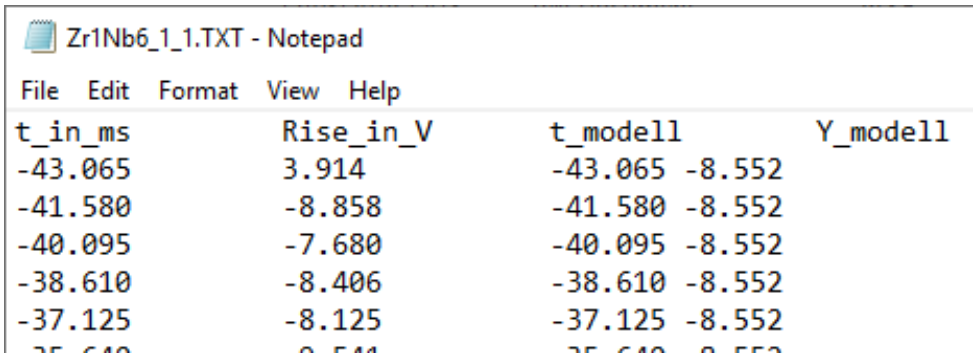
(c) Clicking "Ok" and "Save" on the following windows will create a .lfr file with file locations and data for all the heating curves (fig. 2). Save this in the same folder as the .txt files.

Unexpected zero values in the .lfr file

If the heating curve recorded by Linseis is erroneous, it may record a zero in the .lfr file for a crucial parameter. This can cause "divide by zero" errors in the program, often resulting in an infinity handling error. Remove this heating curve from the file and try again.



(a) AproSoft Evaluation screen with the export button highlighted in red



(b) Exported .txt file example

Figure 1: Exporting in Linseis Aprosoft

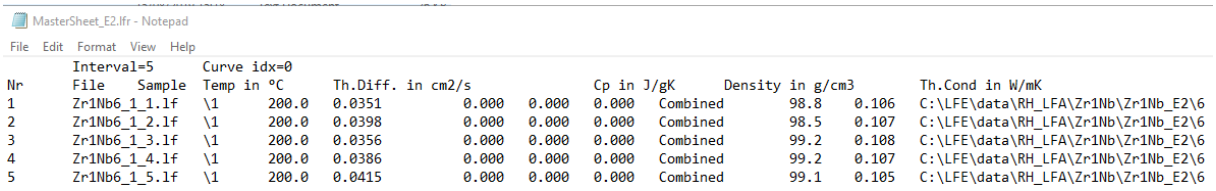


Figure 2: Example .lfr file generated from "Severals"

For more information, please refer to this section of the Javadoc

4.3 DAT format

Importing of individual shots is possible for input files with a .dat extension. The required data structure is described below. The first row gives the temperature (in degrees Celsius). Starting from the second row, the values are arranged in a table, where the first two columns give the values of time (in seconds) and temperature response (arbitrary units), respectively. The third and subsequent columns are ignored.

Example `.dat` file

```
587.863
0.0000000   -0.01227   5.99260742187500E+0002
0.0002720   -0.04987   5.99349670410156E+0002
0.0005440    0.25281   5.99201477050781E+0002
0.0008160    0.24573   5.99171752929688E+0002
0.0010880    0.17072   5.99260742187500E+0002
0.0013600    0.13086   5.99245910644531E+0002
0.0016320    0.12146   5.99216247558594E+0002
0.0019040    0.12384   5.99334838867188E+0002
0.0021760    0.11682   5.99320007324219E+0002
0.0024480    0.12854   5.99260742187500E+0002
...              ...                ...
```

For more information, please refer to this section of the Javadoc

4.4 Metadata

Ideally, metadata for each shot should be recorded during the experiment in a tab delimited ASCII format, with a .met file suffix (fig. 3). The header of the `.met` file contains constant data recorded in tab-separated pairs. The full list of keywords is given in this section of the Javadoc. Two line breaks below, a tab-delimited table with headers for variables should contain variable data for each shot. If any of the parameters related to the shot are variable, then they should be included in the variable table.

4.5 Thermal property file format (`.tbl`)

Specific heat and density at different temperatures can be read as ASCII files with a .tbl suffix (fig. 4), where the first column is temperature (in °C) and the second column is the specific heat capacity (in $J \times kg^{-1} \times K^{-1}$) or density (in $kg \times m^{-3}$).

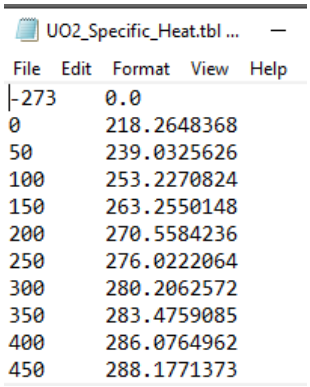
Zr1Nb_E2_200_to_850_Metadata.met - Notepad

File Edit Format View Help

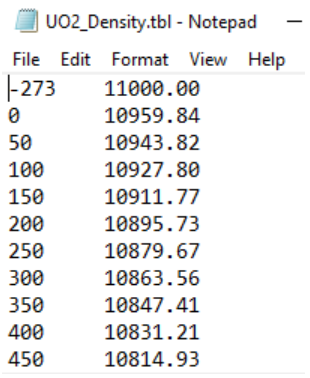
```
Sample_Name      Zr1Nb_E2
Thickness        0.653
Diameter         10
Spot_Diameter    2
PulseShape       TRAPEZOIDAL
Detector_Iris    1

ID      Test_Temperature        Pulse_Width     Absorbed_Energy Detector_Gain
1       200     1.2     9.83    50
2       200     1.2     9.83    50
3       200     2       15.68   50
4       200     1.8     14.22   50
5       200     1.8     14.22   50
```

Figure 3: Example metadata .met file



(a) Example specific heat capacity .tbl file



(b) Example density .tbl file

Figure 4: Example .tbl input files

5 First run

After launching PULsE, a splash screen should appear for a few seconds displaying the current software version. When the software is fully loaded, a graphical user interface (GUI) will appear as shown in fig. 5. The main view consists of four internal frames that can be resized and dragged:

(a) the Task Manager;

(b) the plotting window for the Time-Temperature Profile(s);

(c) the Log

(d) and the Results.

The default layout will restore itself after the user navigates to secondary screens and then back to the main screen. Each internal frame hosts a control panel with a number of buttons that are used to manage its contents.

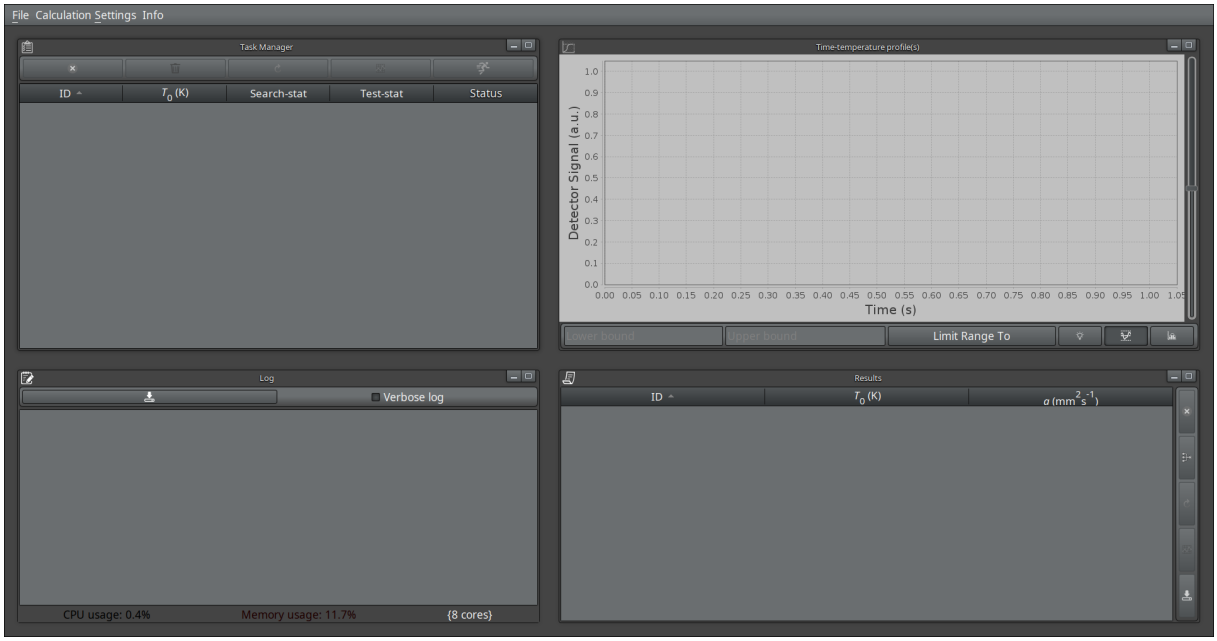


Figure 5: Main view.

Hint

If you are not sure what a certain button does, hover the mouse cursor over the button icon. A helper tooltip appears with a hint to what this button does.

The main menu contains three sub-menus. The `File` sub-menu is used exclusively for importing and exporting data; the `Calculation Settings` can be used to customise the problem statement, set up the optimizer, specify required statistical analysis and change the result format.

5.1 Loading data

Four types of data can be imported in PULsE:

- the experimental detector signal stored separately for each pulse;
- the corresponding numerical pulse data;
- the metadata file
- and the thermal property files

. Thermal properties are loaded when customizing the problem statement in a secondary window, whereas operations with the first three items are conducted solely via the "`File`" menu. After clicking on the loader menu items, a file chooser window will prompt to select one or more files. Multiple files can be selected by holding the `shift` key. The file chooser has an extension filter, which shows only those extensions that are recognized by PULsE. For example, when selecting to load heating curves, the accepted extensions

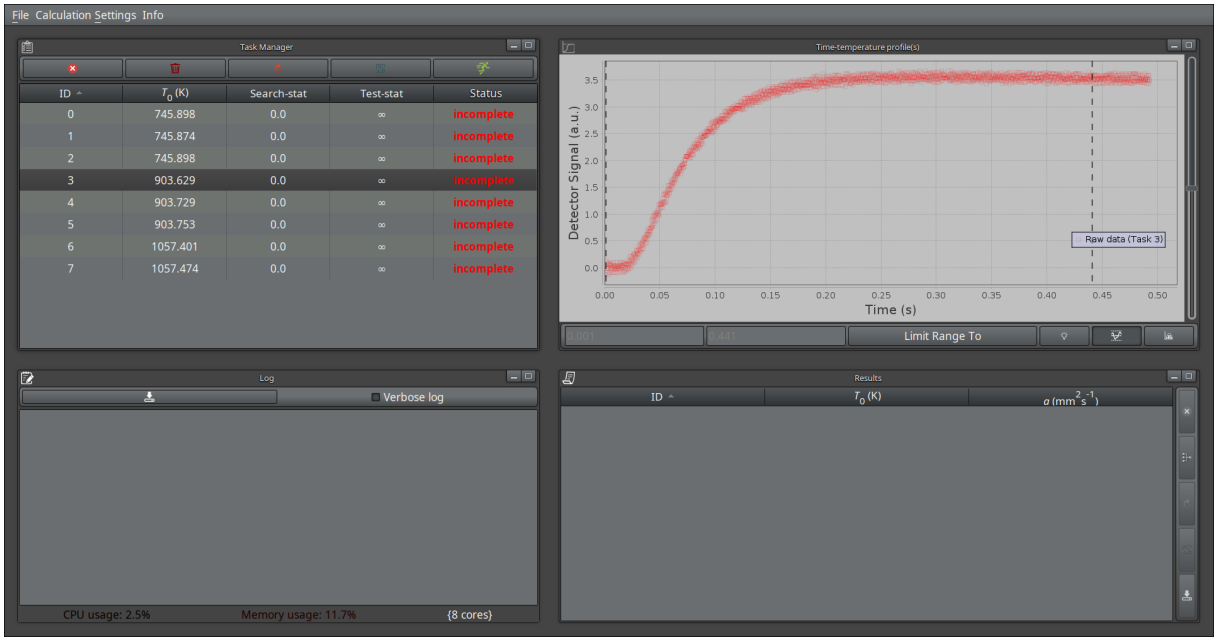


Figure 6: After loading the heating curves.

are `.csv`, `.dat` and `.lfr`. Programmatically, all "read" operations are managed by this part of the code.

Upon successful loading of the heating curves, a list of tasks will appear in the "Task Manager" internal frame. A separate task with its own internal identifier is always assigned to each new dataset imported by PULsE. The external identifier, if present, will also be stored in a metadata object. Multiple tasks will be generated if the user either selects multiple files in the file chooser dialog, or if the user loads an `.lfr` file with links to multiple individual files. The selected task will be highlighted in the "Task Manager, as shown in fig. 6. Navigating through the task list can either be done by scrolling and clicking on the task of interest or using the "down" and "up" keys on the keyboard. When a task is selected, the associated data (solution curve, experimental dataset, residuals, etc.) will be displayed in the "Time-Temperature Profile(s)" internal frame. In addition, the log entries will be shown in the "Log" frame and the result(s) will be highlighted in the "Results" frame.

Calculation range

When generating tasks from an imported dataset, PULsE calculates the half-time and initial guess for each individual profile, as described here. Upon completion, the calculation range will be adjusted according to the truncation procedure. In some extremely rare cases, this may lead to an incorrect calculation time limit, which may be adjusted manually.

Clicking "`File`" → "`Load Metadata...`" and selecting your metadata file for this dataset will fill in the experimental parameters for each task. Pulse files need to be loaded after all detector signals are imported. Each pulse will be assigned to the task

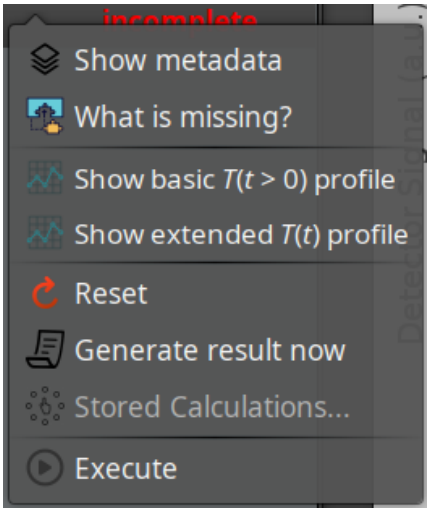


Figure 7: Task menu

that matches its external ID found in the imported file.

Pulse import

It is important to load the numeric pulses prior to performing any actions with the tasks, otherwise, by default, the calculation will use a standard rectangular pulse with a pulse width t_p = 1.5 ms.

5.2 Managing tasks

The task table shows the task ID for each (an internal value, not necessarily related to the ID of the shot's metadata), the ambient temperature at the time of shot T_0 (in Kelvin), the value of the objective function (such as the residual sum of squares, which dynamically changes during execution of a task), the test statistic (filled only after the task completion), and the Status of each task. These columns can be sorted by clicking the corresponding header of the table. Click the same header again to reverse the sorting direction. Right-clicking on a selected task will bring up a menu fig. 7.

- "Show metadata" displays a pop-up window with non-editable metadata for this task, consisting of "name" - "value" pairs. The units here are SI (distances in metres, energy in Joules, temperature in Kelvin) so may differ from the values in the metadata file;
- "What is missing?" shows the specific reason why a task cannot run when its status is highlighted as red;
- "Show basic $T(t > 0)$ profile" shows a graph excluding the baseline data prior to the laser being shot;
- "Show extended $T(t)$ profile" shows the complete time-temperature profile;

- “Reset” will reset the task to its original state when loaded;
- “Generate result now” will store the current non-converged values for the selected heating curve, either using the Parker’s solution (if the task has not yet been executed) or using the most recent parameter values (if it has been executed). The result will be found in the results table;
- “Execute” will run a single-thread calculation for the selected task only. The result will be found in the results table.

The toolbar above the task table lets you manage all tasks at once, by removing or resetting all of them, removing only a specific tasks, and submitting all tasks to execution. The bottom of the frame shows the CPU and RAM usage and the available CPU cores. The corresponding labels will turn red if the CPU or RAM usage gets too high (above 75%) or amber (if above 50%).

Hint

Please pay attention to the CPU and memory consumption. Memory consumption may rise when quickly scrolling through the task list, as each new selection triggers re-plotting of the chart. When a task is running, it will use one of the available CPU threads. A maximum of $n - 1$ calculation threads may run simultaneously, where n is the number of hardware threads. When CPU or memory consumption reaches the bright red level, the program may become unstable and freeze.

6 Calculation Settings

To select or change the settings for the calculation, select the “Calculation Settings” menu in the top of the window.

6.1 Heat Problem: Statement & Solution

Selecting this item opens an internal frame, allowing the user to choose the heat transfer model and specify any parameters that will be used for calculation (fig. 8). To complete the setup, both the problem statement (left column) and the finite-difference solver (right column) must be specified.

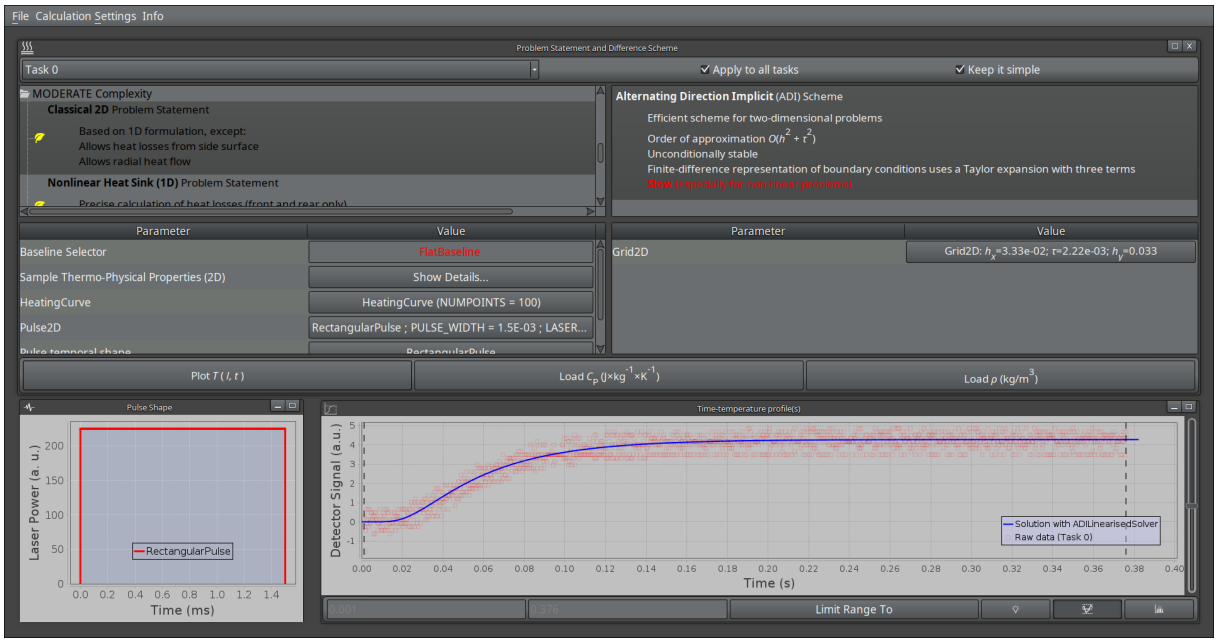


Figure 8: Choosing the heat transfer model and parameters.

The “Problem Statement and Difference Scheme” frame (note it can be maximized by clicking on the icon in the upper right corner) contains a list of available problem statements (**left column**) and list of numerical solvers suitable for the selected problem (**right column**). Available statements and solvers are loaded at runtime using the `Reflection` API.

Problem statements are ranged by complexity, which can either be low, moderate or high. The complexity is defined in terms of the required computational time. Complexity is highlighted by the colour of the leaf icon that is displayed to the left of the main description. Search tasks using problem statements marked with a green leaf icon will normally take less than one second to finish. Red leaves symbolize high complexity calculations, which may take up to a few minutes to complete on a modern laptop. When a problem is selected, a default solver will automatically be selected for this statement. This may be changed by the user. The tables below the lists allow specifying individual settings, such as the type of the baseline, the pulse shape, and the calculation parameters. When modifying the value of a numeric property contained therein, such as

sample thicknees, the program will check if the value entered by the user is sensible, i.e. if the input is a valid number that falls into the range of that specific property. The boundary limits and the default value that a property takes, as well as its description strings, are loaded from an `.xml` file packaged in the `.jar`. All values and settings can be edited by clicking on the text fields with their values.

Editing properties

Note that, by default, a property changed to the current task (see the combobox on the top left of the frame) is also changed to all other tasks. The entered value will be assigned to the same property of any other task by the Task Manager. If only the current task needs to have this property modified, the user should untick the "**Apply to all tasks**" checkbox.

Hidden properties

Some properties are hidden from the user, since they do no typically require adjustment. To enable editing those hidden properties, the user needs to untick the "**Keep it simple**" checkbox.

C_p and ρ data tables can be loaded using the "Load C_p" and "Load ρ" buttons at the bottom of this window, allowing calculation of thermal conductivity or using advanced problem statements, such as the Participating Medium (1D). Click the cross in the top right of the menu to return to the task view.

6.2 Parameter Estimation: Methods & Settings

This will show up an internal frame where the user is required to select the search algorithms, the dimensionality of the optimisation vector, and may modify other settings used in the parameter estimation algorithm (fig. 9).

The user needs to choose one option from the top panel. The following optimisers are currently supported:

- **Levenberg-Marquardt (LM) optimiser**. An optimiser specifically developed to minimise the ordinary least-squares objective function. This version of the LM optimiser uses a mixed damping strategy and a delayed gratification scheme. It is highly efficient, but may fail if the starting values of the parameters are poorly selected;
- **BFGS Quasi-Newton optimiser**. A robust optimiser from the approximate Hessian family that was initially developed for large-scale problems and arbitrary objective functions. Due to its exceptional stability when used in conjunction with the Wolfe linear search (default), it is often useful even for small-scale problems. The implementation of this optimiser is described in this Javadoc section and in the journal paper Lunev and Heymer [2020];

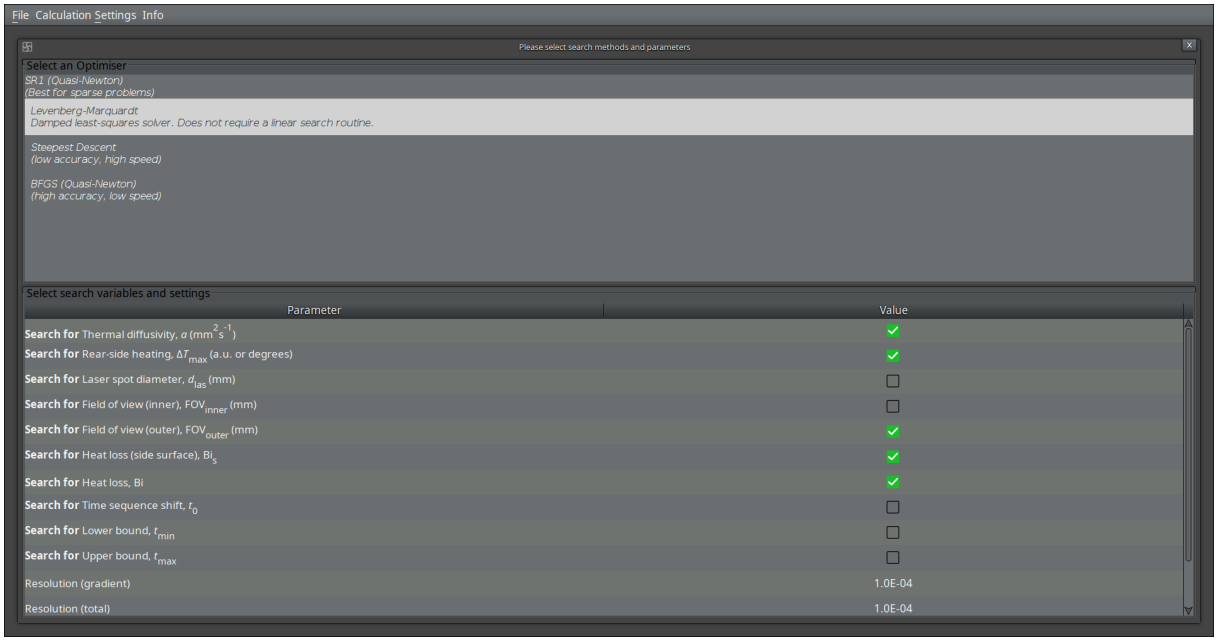


Figure 9: Search settings view

- **SR1 Quasi-Newton optimiser**. This is similar to BFGS, but uses a different (rank one) formula for calculating the search direction;
- **Steepest Descent**, aka the Gradient Descent method. This is a simple optimiser that uses the antigradient of the objective function as the search direction at each step. A linear search routine is used, such as the Golden section.

Objective Function (Optimiser Statistic)

When using LM Optimiser, only the Ordinary Least Squares (OLS) optimiser statistic can be selected. To use a different statistic, switch to another optimiser.

If an optimiser requiring a linear search routine is selected, the latter will be set to default for this method and may be changed from the bottom table. By scrolling the table to the bottom, other settings (resolutions) can be edited.

Search vector

Before proceeding, make sure that the search vector contains the correct parameters. You can change the search vector by ticking or unticking the checkboxes next to the parameter names, in the bottom table of this scren. The number of checked parameters is not limited.

6.3 Statistical Analysis

There are three sub-menus in this section:

- Normality tests. Default value: *Don't Test Please*;
- Optimiser statistic. Default value: *Ordinary Least Squares* and
- Correlation tests. Default value: *Spearman's Rank Correlation*.

If a normality test is selected, it will run when a task finishes and will be used to assess whether a search task completed successfully or failed. The optimiser statistic is the objective function used by the Optimiser. Both of these statistics rely on calculating the model residuals. A correlation test deals with the data that is stored in the opimisation buffer for each task. The history of changes to the search vector parameters over the course of the optimisation run may be used to assess whetherparameter pairs change independently or if the changes are correlated. In case of high correlation (typically, a threshold of 0.8 – note this can be changed from the menu), the result of the task will not be accepted, since this usually indicates the problem is over-parameterised.

Below is a list of available non-parametric normality tests:

- *Anderson-Darling*. This follows the SSJ implementation of the test;
- *Kolmogorov-Smirnov*. This is less strict than the Anderson-Darling. This follows the `ApacheCommonsMath` implementation of the test
- R^2 test. This is a legacy option, which does not really test normality but may be used as a rough acceptance criterion. If selected, all shots with $R^2 < 0.2$ will be ignored.

The threshold of the normality tests can be changed by clicking on the respective menu item.

Correlation tests include

- *Spearman's Rank Correlation* test for nonlinear correlations and
- *Pearson's Product-Moment Correlation* test, which measures only linear correlation between two sets of data.

Finally, available optimiser statistics are:

- *Ordinary Least Squares* is the simple sum of squared residuals divided by the number of data points. Compatible with any optimiser;
- *Absolute Deviations* is an alternative statistic equal to the sum of absolute values of the residuals. Incompatible with LM Optimiser;
- L_2 *Regularised Least Squares* is an OLS statistic with a penalising term equal to the squared length of the search vector. Used to penalise over-parametrisation.

6.4 Change Result Format

Clicking on this menu item allows the selection and ordering of entries in the result table. The user can select any combination of allowed parameters by moving them from left to right in the pop-up window (fig. 10). Select an item from the left list to add a new output parameter, then click on “»” button in the middle. To remove an output parameter, select the item from the right list and click on the “«” button. To commit changes, press “Commit” on the lower toolbar. The output will only contain selected parameters.

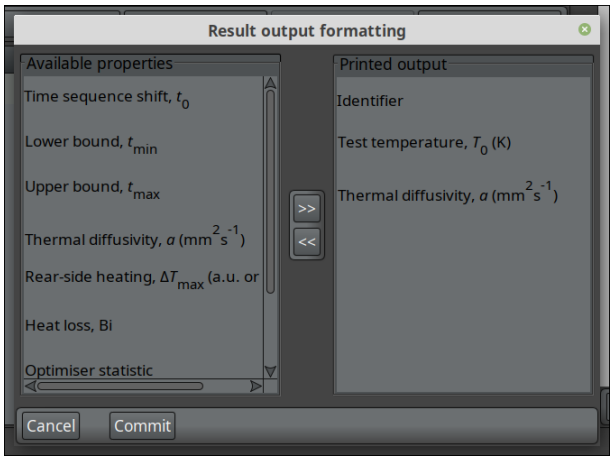


Figure 10: Changing output parameters

Hover over the headers in the result table for a tooltip definition. The results can be sorted by clicking the header similar to the task table. The columns can also be rearranged by clicking and dragging the column headers into the desired order.

6.5 Buffer size

Buffers are filled during task execution with meta-information from consecutive iterations, storing the parameter values and value of the objective function. The Buffer object contains methods used to determine convergence of a set of parameters. By clicking on “Buffer size...” menu item, the user opens a pop-up window prompting to enter a new size for the buffer, which should be an integer. The higher the buffer size, the longer it will take for a task to finish. A default buffer size of 8 is used.

7 Heat Problem Statements

7.1 Classical Problem (1D)

The description below is copied from Lunev and Heymer [2020].

A one-dimensional heat conduction problem for the laser flash experiment must include:

(i) a heat source term $QP(t)$, where Q is the heat current and $P(t)$ is a time-dependent laser pulse function distributed over an area πd_{las}^2, where d_{las} is the diameter of the laser spot at the front surface of the sample;

(ii) heat sink terms at both the front $x = 0$ and the rear $x = l$ surfaces due to the radiation heat transfer $\varepsilon(T)\sigma_0(T^4(x,t) - T_0^4(x,t))$ from the sample to its surroundings in the furnace (assuming the latter are kept at a stable temperature T_0), where ε is the material emissivity.

If $\Delta T/T_0 \ll 1$, it may be additionally assumed that $\varepsilon(T) \approx \varepsilon(T_0)$, so that the boundary problem is written as:

$$C_{\text{p}}\rho\frac{\partial T}{\partial t} = \frac{\partial}{\partial x}\left[\lambda\frac{\partial T}{\partial x}\right], \quad 0 < x < l, \quad t > 0, \tag{1a}$$

$$\lambda\frac{\partial T}{\partial x}\bigg|_{x=0} = -\frac{Q}{\pi d_{\text{las}}^2}P(t) + \varepsilon(T_0)\sigma_0\left[T^4(0,t) - T_0^4\right], \tag{1b}$$

$$\lambda\frac{\partial T}{\partial(-x)}\bigg|_{x=l} = \varepsilon(T_0)\sigma_0\left[T^4(l,t) - T_0^4\right], \tag{1c}$$

$$T(0,x) = T_0, \tag{1d}$$

where C_{p} is the specific heat at constant pressure and ρ is the material density.

The dimensionless quantities are then introduced as follows:

$$\text{Fo} := at/l^2, \tag{2a}$$

$$\text{Bi} := 4\sigma_0\varepsilon T_0^3 l/\lambda, \tag{2b}$$

$$\theta := (T - T_0)/\delta T_{\text{m}}, \tag{2c}$$

$$\delta T_{\text{m}} := Q/\left(C\rho\pi d_{\text{las}}^2 l\right), \tag{2d}$$

$$y := x/l. \tag{2e}$$

Assuming that $\lambda \neq \lambda(x)$ and $(T - T_0)/T_0 \ll 1$, linearisation of Eqs. (1) results in an alternative boundary problem:

$$\frac{\partial\theta}{\partial\text{Fo}} = \frac{\partial^2\theta}{\partial y^2}, \quad 0 < y < 1, \quad \text{Fo} > 0, \tag{3a}$$

$$\frac{\partial\theta}{\partial y}\bigg|_{y=0} = \text{Bi}\cdot\theta - \Phi\left(\text{Fo}\right), \tag{3b}$$

$$\frac{\partial\theta}{\partial(-y)}\bigg|_{y=1} = \text{Bi}\cdot\theta, \tag{3c}$$

$$\theta(0,y) = 0. \tag{3d}$$

And this problem is solved in PULsE. The following parameters can be included in the search vector:

Model Parameters	
Parameter	Meaning
a	Thermal diffusivity
Bi	Heat losses (front/rear)
T_{max}	Maximum heating

7.2 Classical Problem (2D)

To account for non-uniform heating of a sample's surface and the partial detection area, it is necessary to include the radial heat fluxes in the mathematial formulation. A two-dimensional problem is visualised in fig. 11. A mask at the top creates an annular detection area limited by two diameters, FOV_{inner} and FOV_{outer}. The illuminated area is limited by the outer diameter d_{las}, and the sample diameter is d.

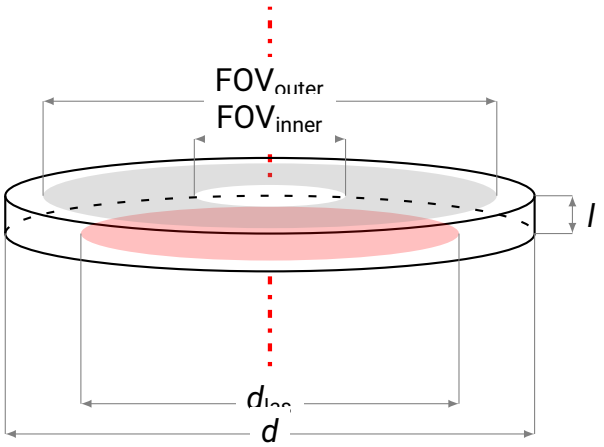


Figure 11: Scheme of the two-dimensional heat conduction problem.

The two-dimensional boundary problem with nonlinear heat losses is written as:

$$\frac{\partial T}{\partial t} = \frac{\lambda}{C_{\mathrm{p}}\rho}\left\{\frac{1}{r}\frac{\partial}{\partial r}\left[r\frac{\partial T}{\partial r}\right] + \frac{\partial^2 T}{\partial z^2}\right\}, \quad 0 < r < d/2,\ 0 < z < l,\ t > 0, \tag{4a}$$

$$\lambda\frac{\partial T}{\partial z}\bigg|_{z=0} = -\frac{4QP(t)}{\pi d^2}\Theta\left(d_{\mathrm{las}}/2 - r\right) + \varepsilon_1(T_0)\sigma_0\left[T^4(0,r,t) - T_0^4\right], \tag{4b}$$

$$\lambda\frac{\partial T}{\partial(-z)}\bigg|_{z=l} = \varepsilon_1(T_0)\sigma_0\left[T^4(l,r,t) - T_0^4\right], \tag{4c}$$

$$\lambda\frac{\partial T}{\partial r}\bigg|_{r=0} = 0, \tag{4d}$$

$$\lambda\frac{\partial T}{\partial(-r)}\bigg|_{r=R} = \varepsilon_2(T_0)\sigma_0\left[T^4(z,R,t) - T_0^4\right], \tag{4e}$$

$$T(0,r,z) = T_0, \tag{4f}$$

where $Q = AQ_0$ is the heat absorbed by the thin surface layer (Q_0 is the energy of the laser pulse and A is the laser-wave absorption coefficient of the opaque material under study), ε_1 and ε_2 are the emissivities of the flat sample's surfaces and the side surface.

Linearising this problem and switching to dimensionless variables leads to:

$$\frac{\partial\theta}{\partial \mathrm{Fo}} = \omega^2 \frac{1}{x}\frac{\partial}{\partial x}\left[x\frac{\partial\theta}{\partial x}\right] + \frac{\partial^2\theta}{\partial y^2}, \quad 0 < x < 1,\ 0 < y < 1,\ \mathrm{Fo} > 0, \tag{5a}$$

$$\left.\frac{\partial\theta}{\partial y}\right|_{y=0} = -\Phi(\mathrm{Fo})\Theta\left(d_{\mathrm{las}}/d - x\right) + \mathrm{Bi}_1 \cdot \theta(y = 0), \tag{5b}$$

$$\left.\frac{\partial\theta}{\partial(-y)}\right|_{y=1} = \mathrm{Bi}_1 \cdot \theta(y = 1), \tag{5c}$$

$$\left.\frac{\partial\theta}{\partial x}\right|_{x=0} = 0, \tag{5d}$$

$$\left.\frac{\partial\theta}{\partial(-x)}\right|_{x=1} = \omega\mathrm{Bi}_2 \cdot \theta(x = 1), \tag{5e}$$

$$\theta(0, x, y) = 0, \tag{5f}$$

where the following notations are introduced: $\omega := 2l/d, \theta := (T - T_0)/T_{\mathrm{m}}, T_{\mathrm{m}} = 4Q/(\pi d^2 C_{\mathrm{p}}\rho l)$, $\mathrm{Bi} := 4\sigma_0\varepsilon T_0^3 l/\lambda$, $\mathrm{Fo} := at/l^2$, $a = \lambda C_{\mathrm{p}}^{-1}\rho^{-1}$, $y = z/l$, $x = 2r/d$. When $\omega \ll 1$, the terms governing the radial heat fluxes in Eq. (5a) and the side heat losses in Eq. (5e) are negligible, and hence the problem is effectively one-dimensional.

The following parameters can be included in the search vector:

Model Parameters	
Parameter	Meaning
a	Thermal diffusivity
Bi_1	Heat losses (front/rear)
Bi_2	Heat losses (sides)
$T_{\max}$	Maximum heating
d_{las}	Laser spot diameter
$\mathrm{FOV}_{\mathrm{outer}}$	Outer FOV diameter
$\mathrm{FOV}_{\mathrm{inner}}$	Inner FOV diameter

7.3 Penetration (1D)

Consider a heat source term expressed through the spatial and time distribution functions $\Psi(y)$ and $\Phi(\mathrm{Fo})$.

The problem statement is then:

$$\frac{\partial\theta}{\partial \mathrm{Fo}} = \frac{\partial^2\theta}{\partial y^2} + \Phi(\mathrm{Fo})\Psi(y), \tag{6a}$$

$$\left.\frac{\partial\theta}{\partial y}\right|_{y=0} = \mathrm{Bi}\cdot\theta, \tag{6b}$$

$$\left.\frac{\partial\theta}{\partial(-y)}\right|_{y=1} = \mathrm{Bi}\cdot\theta, \tag{6c}$$

$$\theta(0,y) = 0. \tag{6d}$$

Two absorption models are implemented in PULsE and either can be selected when setting up the problem statement:

- *Beer-Lambert Absorption* law, which leads to an exponential decay in $\Psi(y) = \gamma/l \times \exp(-\gamma y)$. Applicable to samples where the absorption length is lower than the sample thickness.
- *Insulator Absorption law* for fully translucent samples. The expression for the spatial heat distibution is given in Salazar et al. [2014].

Finally, the detector does not take readings from the rear surface of the sample ($T(x = l)$) as in the case of an opaque body, but rather collects data from within a finite layer at the rear. The expression for the detector signal is derived assuming the thermal radiation from the rear surface is approximated with a grey-body spectrum:

$$\Delta V_{\mathrm{det}} \approx 4GT_0^3\varepsilon\sigma_0 \int_0^1 \theta(y)e^{-\gamma_{\mathrm{IR}}(1-y)}dy, \tag{7}$$

where γ_{IR} is an effective absorption coefficient of thermal radiation, G is a dimensional and geometric factor describing the relative positioning of the sample's rear and the detector's active surface, the efficiency of radiation absorption within a hardware-limited range of wavelengths, and the conversion from absorbed energy to generated voltage, assumed linearly-dependent.

The following parameters can be included in the search vector:

Model Parameters	
Parameter	Meaning
a	Thermal diffusivity
Bi	Heat losses (front/rear)
γ	Flash absorptivity
γ_{IR}	Thermal absorptivity

7.4 Diathermic Samples with Gray Walls (1D)

The description below is copied from Lunev et al. [2021].

The diathermic model is based on the following propositions:

(a) A sample is completely transparent to the incident laser and thermal radiation;

(b) The front and rear faces of the sample are coated by a grey absorber;

(c) The side surface of the sample is completely free from any coating;

(d) The contact resistance between the sample and the coatings is zero, thus no temperature gradients are formed;

(e) The coatings are grey bodies, i.e. have a flat absorption spectrum.

Consequently, the monochromatic laser radiation is partially absorbed at the front face of the sample ($y = 0$), causing immediate heating. A portion of thermal radiation causes the rear face ($y = 1$) to start heating precisely at the same time (ahead of thermal conduction). The remainder energy dissipates in the ambient. It is thus sufficient to consider three radiative heat fluxes. The first two correspond to heat dissipation within the furnace chamber. The third flux acts to thermalise the parallel boundaries by radiative transfer only:

$$q_{0\to\infty} \approx \varepsilon\sigma_0 \left(T^4(0,t) - T_0^4\right) H, \tag{8a}$$

$$q_{1\to\infty} \approx -\varepsilon\sigma_0 \left(T^4(l,t) - T_0^4\right) H, \tag{8b}$$

$$q_{1\to 2} = \frac{\varepsilon}{2-\varepsilon}\sigma_0 \left(T^4(0,t) - T^4(l,t)\right) H, \tag{8c}$$

where the emissivities of both faces are assumed to be equal ($\varepsilon_1 = \varepsilon_2 = \varepsilon$).

Let $\eta = \varepsilon/(2-\varepsilon)$, so that $0 < \eta \leq 1$. Since nonlinear heat losses can be neglected [see Appendix A], the boundary problem can be written as:

$$\frac{\partial\theta}{\partial \mathrm{Fo}} = \frac{\partial^2\theta}{\partial y^2}, \quad 0 < y < 1, \quad \mathrm{Fo} > 0, \tag{9a}$$

$$\left.\frac{\partial\theta}{\partial y}\right|_{y=0} = \mathrm{Bi}\cdot\theta_{y=0} + \eta\mathrm{Bi}\cdot(\theta_{y=0} - \theta_{y=1}) - \Phi\,(\mathrm{Fo})\,, \tag{9b}$$

$$\left.\frac{\partial\theta}{\partial(-y)}\right|_{y=1} = \mathrm{Bi}\cdot\theta_{y=1} + \eta\mathrm{Bi}\cdot(\theta_{y=1} - \theta_{y=0}), \tag{9c}$$

$$\theta(0,y) = 0, \tag{9d}$$

The model defines the following parameters:

Model Parameters	
Parameter	Meaning
a	Thermal diffusivity
Bi	Heat losses (front/rear)
$T_{\max}$	Maximum heating
η	Diathermic coefficient

7.5 Nonlinear Heat Losses (1D)

This is based on solving eq. (1) in the non-linearised form. The difference with the linearised form eq. (3) lies within the definition of boundary conditions:

$$\left.\frac{\partial\theta}{\partial y}\right|_{y=0} = -\Phi(\mathrm{Fo}) + \mathrm{Bi}\cdot T_0\left[\left(\theta_{y=0}\delta T_\mathrm{m}/T_0+1\right)^4-1\right]/(4\delta T_\mathrm{m}), \tag{10a}$$

$$\left.\frac{\partial\theta}{\partial y}\right|_{y=1} = -\mathrm{Bi}\cdot T_0\left[\left(\theta_{y=1}\delta T_\mathrm{m}/T_0+1\right)^4-1\right]/(4\delta T_\mathrm{m}), \tag{10b}$$

An absolute temperature scale is assumed. This problem statement is only compatible with LFA instruments where the detector has been pre-calibrated to give absolute temperature readings.

7.6 Participating Medium (1D)

The description below is copied from Lunev et al. [2021]. Consider a coupled conductive-radiative heat transfer problem, where the radiation fluxes $f(y)$ are calculated separately (for details, please see the reference paper). For a semi-transparent sample with opaque coatings on both front and rear surfaces, the problem statement is:

$$\frac{\partial\theta}{\partial\mathrm{Fo}} = \frac{\partial^2\theta}{\partial y^2} + \frac{\tau_0}{N_\mathrm{P}}\times\left(-\frac{df}{d\tau}\right), \tag{11a}$$

$$\left.\frac{\partial\theta}{\partial y}\right|_{y=0} = \mathrm{Bi}\cdot\theta - \Phi\left(\mathrm{Fo}\right) + \frac{1}{N_\mathrm{P}}f(0), \tag{11b}$$

$$\left.\frac{\partial\theta}{\partial y}\right|_{y=1} = -\mathrm{Bi}\cdot\theta + \frac{1}{N_\mathrm{P}}f(1), \tag{11c}$$

$$\theta(0,y) = 0. \tag{11d}$$

where $f = F/(n^2\sigma T_0^3)$ is the dimensionless radiative flux; in addition, the Planck number is introduced: $N_\mathrm{P} = \lambda/\left(4\sigma_0 n^2 T_0^3 l\right)$, and the optical thickness is τ_0.

The radiation field can be calculated using:

- **The analytical solution for the non-scattering case**. An integration scheme must be selected, which can either use Newton-Cotes rules or the more accurate Chandrasekhar's quadratures. Flux derivatives can be calculated either analytically or discretely using central differences or with
- **The discrete ordinates method (DOM)**. In the latter case, a scattering phase function is introduced, which by default is the Henyey-Greenstein function. The user can switch to a linear-anisotropic function. An integrator needs to be selected for the radiative transfer equation. By default, this uses the highly-optimised TR-BDF2 integrator (please see the reference paper). However, the user can change this to other Runge-Kutta integrators.

Calculations may throw a `SolverException` in case when the RTE takes too long to integrate or when the adaptive grid size is too large. Typically, this can be avoided by a proper choice of calculation parameters.

The model requires C_p and ρ data to be pre-loaded for the full temperature range of calculations and defines the following parameters:

Model Parameters	
Parameter	Meaning
a	Thermal diffusivity
Bi	Heat losses (front/rear)
T_{max}	Maximum heating
τ_0	Optical thickness
N_P	Planck number
ω_0	Scattering albedo
A	Scattering anisotropy

7.7 Baseline types

To customise the baseline, use the baseline selector from the problem statement table (fig. 8). Select from one of the options:

- Flat baseline, a horizontal line;
- Linear baseline, a sloped line.
- Sinusoidal baseline, an oscillating baseline.

Parameters of the flat and linear baseline are automatically calculated via a linear regression using data points at $t < 0$. To use the sinusoidal baseline, manual input of the initial guess is required to fill in the estimates of the frequency, phase shift, and amplitude of the oscillations. These can be customized from the same table, by clicking on the “SinusoidalBaseline” button at the bottom.

All baseline parameters can be included in the search vector.

Note

Flat baseline is not currently adjustable via the search procedure. If desired, the user should select the linear baseline from the list, manually set slope to 0.0 and include only the intercept in the search vector to reproduce this behaviour.

The following parameters can be included in the search vector:

Baseline Parameters	
Parameter	Meaning
Intercept	Value at $t = 0$
Slope	Linear slope
f	Frequency
A	Amplitude
ϕ	Phase shift
Intercept	Value at $t = 0$

7.8 Flash shapes

Several flash shapes are available using the “Pulse Shape Selector” from the “Pulse Properties” (accessible via the table in fig. 8):

- *Rectangular pulse*. This is simply a rectangular signal (the height is always normalised). Fully defined by the pulse width parameter t_p from the “Pulse properties”;
- *Trapezoidal pulse*, a trapez with adjustable fall t_{fall} and rise t_{rise} segments. Note that t_{fall} and t_{rise} may be changed individually, thus it is possible to shift the center of gravity for this baseline;
- *Exponentially modified Gaussian pulse*. A skewed Gaussian curve with adjustable center of gravity and width. Represented by the mean μ, variance σ and skewness λ;
- *Numerical pulse*. Measured with a pulse diode simultaneously with the detector signal using pulse mapping and loaded as a separate file.

Numerical pulse

If the pulse width is very small compared to the measurement duration, it is recommended to proceed with a simple pulse shape, such as a rectangular pulse. A numerical pulse correction is only required when the pulse shape is in the range of 0.05–0.75 half-times.

Pulse parameters cannot be included in the search vector. A pulse may be visualised by clicking the “Plot” button at the bottom left of the problem statement setup screen (fig. 8).

8 Execution

Clicking the “Execute” action (running man icon at the top of the Task Manager internal frame) will submit a batch of tasks to a dynamic queue for parallel processing using multiple threads, according to the settings previously selected. When execution is in progress, the running man green icon is replaced by a red stop sign. Pressing this button during execution will cause tasks to terminate prematurely, which changes their status to “terminated”. The calculation can be resumed at any time.

8.1 Statuses

The statuses of tasks will change as a result of this:

(a) “ready” means that the settings have been chosen for the task, but calculation has not yet started;

(b) "queued” means that the task will execute once other tasks have finishing executing;

(c) “done” means that the calculation for the task has completed and all tests passed. You can find the results for this task in the results pane. The calculation can be re-started if a new problem statement is selected. In this case, the old results can be accessed via the “Stored Calculations...” menu item (fig. 7);

(d) “timeout” means that the calculation took too long to complete. This may be due to an error in calculation or due to the search algorithm being unable to find the optimal values fast enough. The results will not be shown in the results table. Right-clicking the task and clicking “Execute” for this task only will continue calculation, and may be able to complete the calculation. Right-clicking the task and choosing “Generate result now” will generate a result immediately based on the most recently output values;

(e) “incomplete” means that no execution has taken place yet as the settings (heat problem and least squares methods) have not yet been chosen. Additional status description may be shown on request;

(f) “terminated” means that the execution was manually stopped before completion. Similar to tasks with a “timeout” status, re-executing may allow completion of the calculation;

(g) “failed” means either a solver error has occurred or the parameter estimates are outside the allowed range.

8.2 Charting

Switching between tasks in the task table or in the settings view, pressing the “Graph” button in the task toolbar, or selecting the corresponding option from the pop-up menu for individual tasks will result in the calculated solution to be shown in the top-right

internal frame, along with the raw data (fig. 12). Raw data is plotted as a scatter chart, while the solution is plotted as a line series. It may happen that the solution line is not visible because of dense raw data – in this case try moving the slider on the right down, this will decrease the opacity of raw data.

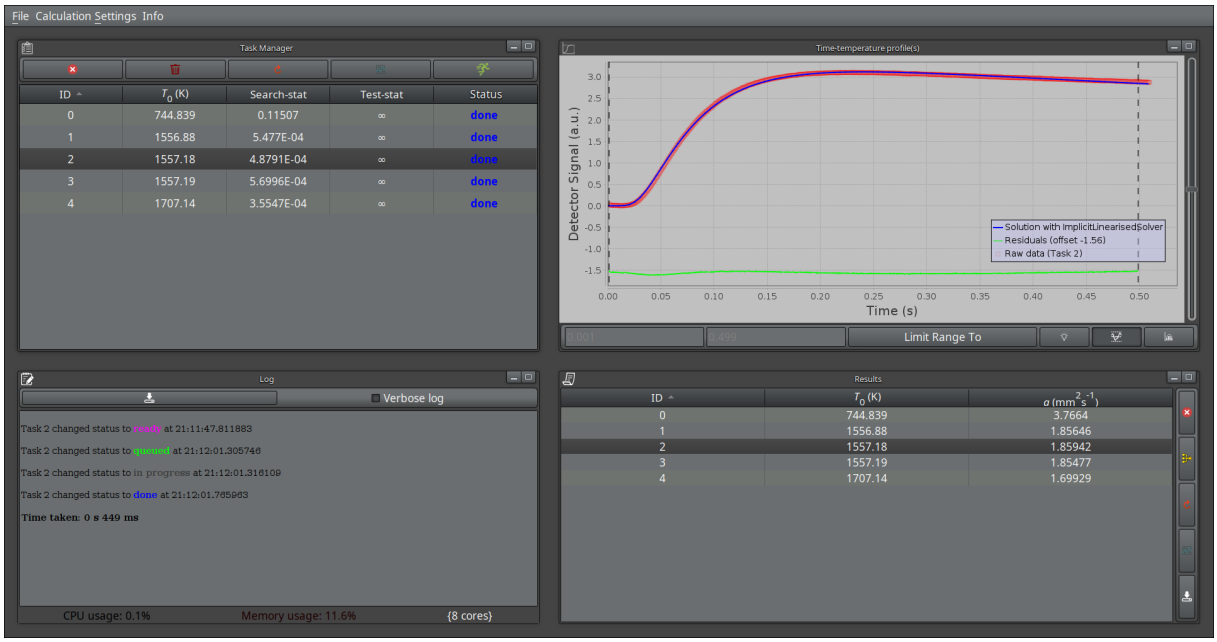


Figure 12: All tasks completed successfully.

The graph toolbar combines the following elements:

(a) Search range selection tools. The text fields show the current search domain for the selected task. Both the lower and upper bounds may be changed by entering their respective values in the text fields and pressing the “Limit Range To” button. This will show a confirmation dialog where the user will be prompted to confirm their choice. Note that the bounds will not be changed if the button is not pressed. Selecting and confirming a new search range will result in the vertical dashed lines in the graph to shift according to the selection. Only the raw data contained between these vertical lines will be used during optimisation;

(b) A “Sanity Check” toggle button. When activated, this will show how the calculated solution deviates from the analytical solution of the adiabatic problem given by Parker *et al*. Note that the model and the Parker’s solution will be different in most of the cases;

(c) “Plot Residuals” toggle button. When activated, this will show a graph of residuals. The residuals are calculated by comparing the raw data values to the interpolated values from the solution curve. If the parameters are chosen appropriately and if no systematic error is present in the calculations or in the experimental data, the graph of residuals will show white noise;

(d) A “Residuals Histogram” button. This will plot the residuals in a frequency chart, where the user is able to adjust the bin width.

9 Best Model Selection

The use may select to run a task multiple times with different problem statements or search vector parameters. In this case, each successful optimisation is stored. Clicking on the “Stored calculations” from the task menu (fig. 7) will open a new screen where previously completed calculations are listed (fig. 13).

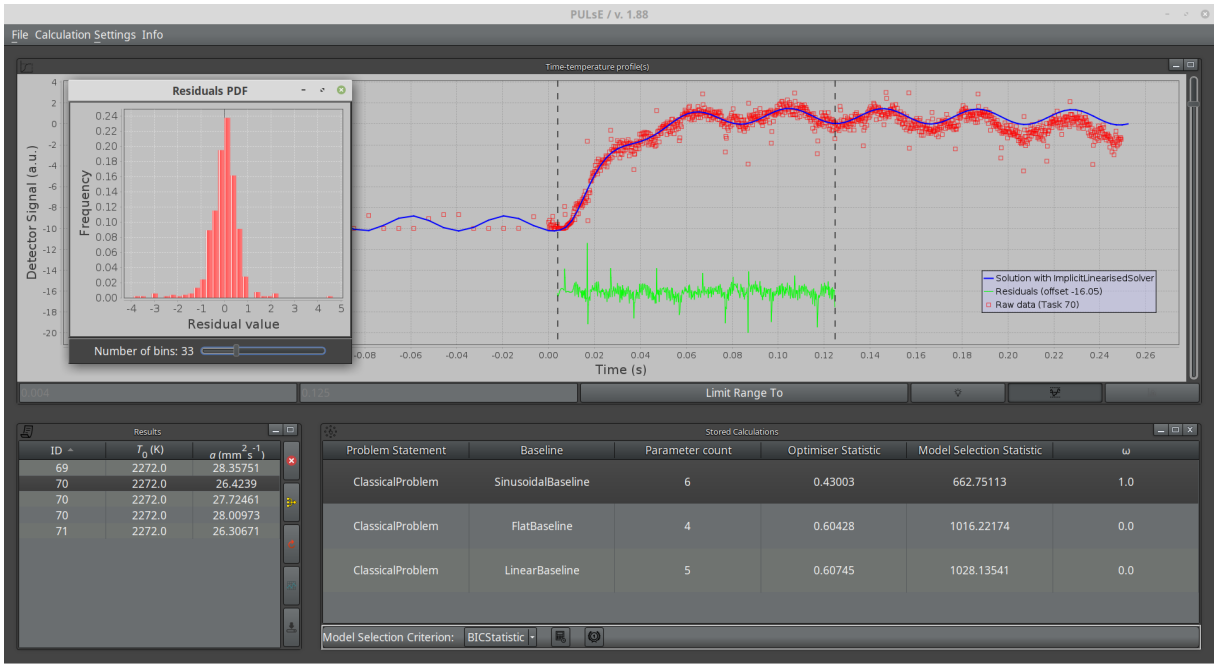


Figure 13: Best Model Selection screen

The current calculation is highlighted in the “Stored Calculations” table at bottom right, which lists the following properties:

- Name of the model;
- Baseline type;
- Number of parameters in the search vector;
- Optimiser statistic value (the objective function value);
- Model selection statistic (either AIC or BIC);
- Calculated weight of the model ω.

Model selection statistic can be changed by clicking on the drop-down menu at the toolbar below. Clicking the calculator button next to it will re-calculate the statistics and weights. The prize button (second to the right) is used to select the best model from the list with the highest ω score. When a new model is selected, its results are shown to the user for inspection. Leaving the screen will result in substituting the current calculation with the selection.

10 Results

Once execution has completed, the result table at the bottom right will look similar to fig. 12.

Actions that can be performed on these results are as follows:

(a) “Delete Entry” removes the selected result from the table.

(b) “Merge (auto)” allows calculating averages (and averaging errors) for tasks that are within a small interval of ambient temperatures. Clicking this will bring up a context window. The results will be grouped according to the selection;

(c) Clicking “Undo” returns the results to their individual values, ungrouping them;

(d) “Preview” allows to plot the results on a chart. Clicking it brings up the preview mode, which includes a large chart panel on top with the results plotted against user-selected axes and the results table in the bottom. Chart axes are selected by clicking the combo boxes. Any column of the results table can be selected as an axis. Clicking the "Preview" button will update the chart, e.g. after merging or removing specific results (fig. 14). The graph can be saved as a .png image. Cleaner graphs can be plotted using the exported data and plotting software, such as `gnuplot` or LaTeX. Along with the axis selection, the preview panel has a toggle button enabling spline fitting of data. This can be turned off or on (on by default);

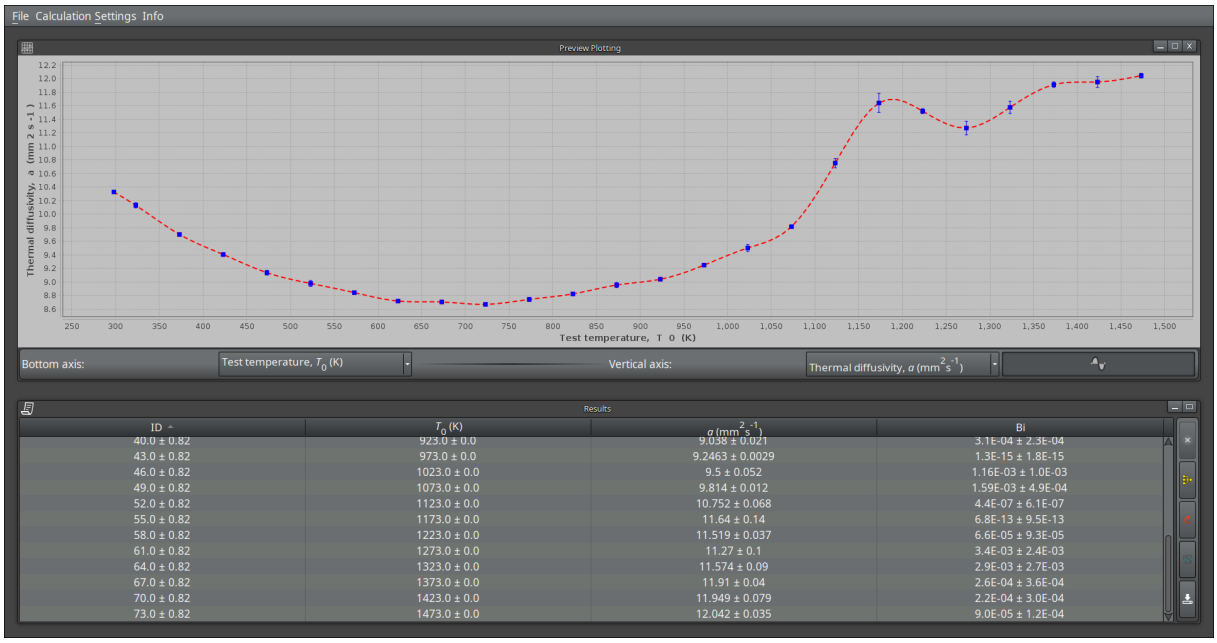
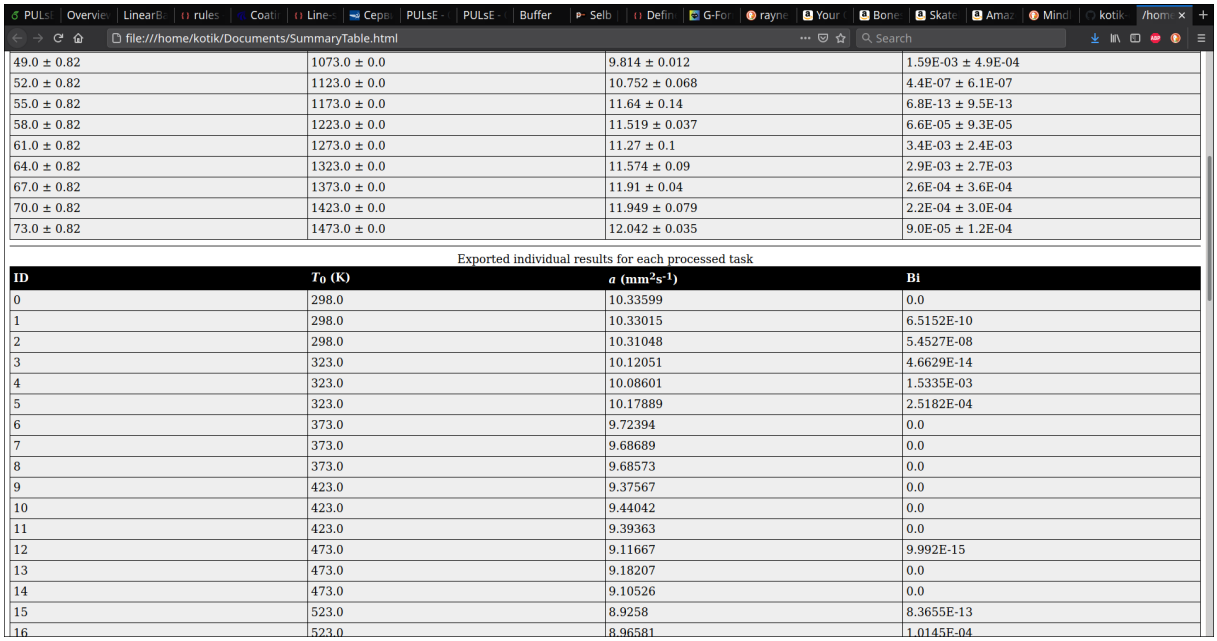


Figure 14: Preview mode.

(e) “Save” button allows to export the results in an .html or .csv format. Saving while the results are merged will save both the merged results and the individual results (fig. 15).



file:///home/kotik/Documents/SummaryTable.html

49.0 ± 0.82	1073.0 ± 0.0	9.814 ± 0.012	1.59E-03 ± 4.9E-04
52.0 ± 0.82	1123.0 ± 0.0	10.752 ± 0.068	4.4E-07 ± 6.1E-07
55.0 ± 0.82	1173.0 ± 0.0	11.64 ± 0.14	6.8E-13 ± 9.5E-13
58.0 ± 0.82	1223.0 ± 0.0	11.519 ± 0.037	6.6E-05 ± 9.3E-05
61.0 ± 0.82	1273.0 ± 0.0	11.27 ± 0.1	3.4E-03 ± 2.4E-03
64.0 ± 0.82	1323.0 ± 0.0	11.574 ± 0.09	2.9E-03 ± 2.7E-03
67.0 ± 0.82	1373.0 ± 0.0	11.91 ± 0.04	2.6E-04 ± 3.6E-04
70.0 ± 0.82	1423.0 ± 0.0	11.949 ± 0.079	2.2E-04 ± 3.0E-04
73.0 ± 0.82	1473.0 ± 0.0	12.042 ± 0.035	9.0E-05 ± 1.2E-04

Exported individual results for each processed task

ID	T_0 (K)	a (mm^{2}s^{-1})	Bi
0	298.0	10.33599	0.0
1	298.0	10.33015	6.5152E-10
2	298.0	10.31048	5.4527E-08
3	323.0	10.12051	4.6629E-14
4	323.0	10.08601	1.5335E-03
5	323.0	10.17889	2.5182E-04
6	373.0	9.72394	0.0
7	373.0	9.68689	0.0
8	373.0	9.68573	0.0
9	423.0	9.37567	0.0
10	423.0	9.44042	0.0
11	423.0	9.39363	0.0
12	473.0	9.11667	9.992E-15
13	473.0	9.18207	0.0
14	473.0	9.10526	0.0
15	523.0	8.9258	8.3655E-13
16	523.0	8.96581	1.0145E-04

Figure 15: Summary table exported in `html`

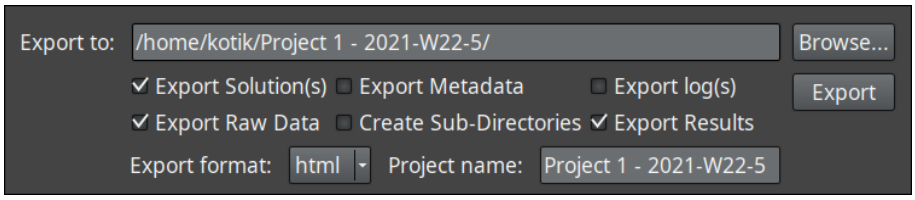


Figure 16: Export dialog.

10.1 Export

Clicking “File - Export...” in the main menu brings up the export dialog (fig. 16). The destination directory is shown in the non-editable text field at the top, which is updated by clicking the “Browse...” button. A file chooser dialog is then initiated, which allows choosing a parent directory for the exported data. By default, files are saved in a sub-directory in the user’s home. Editing the project name (bottom right) or selecting a new parent directory will be reflected in the text field on the top. The exported format is selected with the combo box in the bottom left. If the selected format is unsupported for this specific data type, the export will revert to the default format (html), but only for this specific category of property holders. Names of all exported files are formed of the prefix, which indicated the type of the exported data, and the ID of the task, to which this property holder belongs to, and which matches the ID in the task table and the results table. Finally, after configuring the desired output, the “Export” button may be pressed creating a new directory (if necessary) and populating it accordingly.

11 Log

The log pane is found in the bottom left of the main window, and displays the status of currently executing tasks. Ticking the “Verbose log” checkbox forces more information to be written in the log pane to show the changes in various search parameters. By default, this option is not selected, and only time events, such as change of statuses and task completion, will be recorded in the log. Clicking “Save Log” saves the current contents of the log to a location of choice.

12 Advice to maximise accuracy

12.1 Identifying systematic errors

The Preview graph is useful for identifying which data points may arise from erroneous heating curves or searches, if any points are far outside the expected range. Once the corresponding task is identified in the results table, it is mandatory to review the heating curve graph for that task to see if the calculation has been done correctly. Incorrect calculations are usually easy to identify, since they reveal an ordered arrangement of the residuals, e.g. slopped data sample. If this is the case, the user needs to:

(a) manually change the search range, as explained in section 8.2;

(b) change the individual heat problem parameters (section 6.1);

(c) or revise the optimisation vector by including additional or excluding redundant variables (section 6.2)

The result of the task needs to be removed from the table of results, and the task needs to be re-executed by right clicking on it the task table and pressing the Execute menu item again. Some advices on fine tuning are given in Lunev and Heymer [2020].

12.2 Changing individual settings

Open the problem view from the main menu, use the combo box on the top-left to select the task that needs to be adjusted. Ensure that “Apply to all tasks” and “Keep it simple” are unticked. Changing settings such as the initial thermal diffusivity a, baseline slope and baseline intercept to expected values and re-executing only that task (through the right-click menu) may also improve results. Comparing the fitted baseline to the T($t < 0$) data seen in the “show T(t) profile” in the right-click menu may be of use when choosing values for the baseline.

12.3 Parameter evaporation

In some rare cases, when the number of parameters is high and a correlation exists between them, the optimiser might enter into a state of parameter evaporation. This corresponds to a plateau on the objective function landscape where large changes in parameters lead to tiny changes of the objective function. The optimiser becomes lost on this plateau as parameters will not converge. To avoid this behaviour, try imposing restrictions on the model parameters and restarting the search.

13 Standard Usage

Install PULsE and create the .lfr and .met files as detailed at the start of this manual. To quickly analyse a dataset:

(a) Open PULsE;

(b) "File" $\rightarrow$ "Load Heating Curve(s)..." and load the .lfr file containing your curve details;

(c) "File" $\rightarrow$ "Load Metadata..." and load your .met metadata file;

(d) "Calculation Settings" $\rightarrow$ "Heat Problem: Statement and Solution" to open the problem view. Select "Classical 1D Problem" and "Fully Implicit", ensure the sample thickness is correct, and close the view (cross at the top-right corner – be careful not to confuse this with quitting the program!);

(e) "Calculation Settings" $\rightarrow$ "Parameter Estimation: Method and Settings" to load the search view. Select "Approximated Hessian Method" and "Wolfe Conditions", and tick the checkboxes for rear-side heating, thermal diffusivity, baseline intercept (and baseline slope if you suspect the baseline is sloped). Close the view (again. ensure you are not quitting the program).;

(f) Click "Execute" and wait until all tasks have the "done" status. Right-click $\rightarrow$ "Execute" any tasks with the "timeout" status;

(g) "Merge (auto)" $\rightarrow$ "10" $\rightarrow$ "Apply";

(h) "Preview" $\rightarrow$ choose "T_0" as horizontal axis, "a ($\mathrm{mm^2 s^{-1}}$)" as the vertical axis $\rightarrow$ "plot" $\rightarrow$ Save or Export graph. Close the preview mode;

(i) "Save results" and choose the directory, file name, and extension (you may want to select .csv);

(j) "File" $\rightarrow$ "Export...", choose the location to export and the project name.

References

A. Lunev and R. Heymer. Decreasing the uncertainty of classical laser flash analysis using numerical algorithms robust to noise and systematic errors. *Review of Scientific Instruments*, 91(6):064902, June 2020. doi: 10.1063/1.5132786. URL https://doi.org/10.1063/1.5132786.

A. Lunev, V. Zborovskii, T. Aliev, R. Heymer, and O. Vilkhivskaya. PULsE: An open-source software for laser flash analysis. *Software Impacts*, 6:100044, Nov. 2020. doi: 10.1016/j.simpa.2020.100044. URL https://doi.org/10.1016/j.simpa.2020.100044.

A. Lunev, V. Zborovskii, and T. Aliev. Complexity matters: Highly-accurate numerical models of coupled radiative–conductive heat transfer in a laser flash experiment. *International Journal of Thermal Sciences*, 160:106695, Feb. 2021. doi: 10.1016/j.ijthermalsci.2020.106695. URL https://doi.org/10.1016/j.ijthermalsci.2020.106695.

A. Salazar, A. Mendioroz, E. Apiñaniz, C. Pradere, F. Noël, and J.-C. Batsale. Extending the flash method to measure the thermal diffusivity of semitransparent solids. *Measurement Science and Technology*, 25(3):035604, 2014.