CCFE
CULHAM CENTRE FOR FUSION ENERGY

UK Atomic Energy Authority

UKAEA-CCFE-PR(23)89

Artem Lunev, Vadim Zborovskii, Teymur Aliev,
Robert Heymer, Olga Vilkhivskaya

PULsE: an open-source software for laser flash analysis

PULsE: an open-source software for laser flash analysis

Artem Lunev, Vadim Zborovskii, Teymur Aliev, Robert Heymer,
Olga Vilkhivskaya

This is a preprint of a paper submitted for publication in
Software Impacts

PULsE: an open-source software for laser flash analysis

Artem Lunev[a], Vadim Zborovskii[b], Teymur Aliev[b], Robert Heymer[a], Olga Vilkhivskaya[a]

[a] *United Kingdom Atomic Energy Authority (UKAEA), Culham Science Centre, Abingdon, Oxfordshire OX14 3DB, United Kingdom*
[b] *P.N. Lebedev Physical Institute, Russian Academy of Sciences, Leninsky Prospect 53, Moscow 119991, Russia*

Abstract

PULsE (Processing Unit for Laser Flash Experiments) is a software suite for processing data generated by a variety of laser flash analysis instruments. It is highly customisable and allows adjusting to the finest experimental detail, including but not limited to electromagnetic interferences, non-uniform laser heating and detection, and coupled radiative-conductive transfer. It offers noise-resilient mathematical optimisation capabilities for processing raw experimental data, where the search vector can include an arbitrary number of variables allowed by the model, and features a statistical toolkit for result validation. The most important software features are briefly described. Example data processing is shown for different experimental setups.

Keywords: laser flash analysis, thermal properties, thermophysical

1. Introduction

The laser flash method, first introduced by Parker et al. [1], is an experimental procedure for the determination of material thermal properties. Thermal diffusivity, a quantity inferred in this method, is closely related to thermal conductivity λ via the relation $a = \lambda(C_\mathrm{p}\rho)^{-1}$, where C_p is the specific heat capacity and ρ - the density. As such, this transport property cannot be measured directly, but can be reverse-engineered from the data acquired with an infrared detector monitoring the sample heating caused by the laser pulse. The conversion of time-temperature profiles into thermal diffusivity values requires calculations based on a formal model, which reflects on the specific experimental conditions [2, 3] and material parameters: radiative cooling at high temperatures; finite-pulse effects; non-uniform laser illumination of the sample surface; limited field-of-view of the detector; porous

structure of the material under study; its semi-transparency to thermal radiation, etc. More complications arise when other uncertainty sources [4, 5] are considered, including electromagnetic interferences (caused by a lack of shielding, excessive cable length, poor soldering of electric contacts, etc.), poor detector performance (e.g. due to a low specific detectivity inherent to its design), or problematic temperature regulation distorting the signal and increasing noise level. Once the heat transfer model is selected, synthetic time-temperature profiles may be produced, directly comparable to experimental data. By running a mathematical optimisation algorithm, model parameters are found reproducing the shape of registered signal curve. An optimiser robust to noise and systematic signal drift is required to ensure accurate results – not always available in commercial software packaged with the laser flash analysis (LFA) instruments. The open-source PULsE software is now here to help engineers and scientists in this field, featuring advanced thermal transfer models, robust and reliable algorithms, with limited (but growing) support to data formats used in commercial LFA instruments.

2. Software functionality

PULsE, written in pure `Java`, is theoretically compatible with any platform and practically tested on `Windows`, `Linux` and `MacOS`. The software runs from a single `.jar` file. which activates the `Swing`-based lightweight graphical user interface (GUI), where plotting is done with `JFreeChart`. Data can be exported either in `html` or `csv` file formats, or as images in `png` format. The general flowchart is presented in fig. 1. At start, the user loads experimental data and metadata files, thus generating a set of tasks, requiring a problem statement and solver. Available statements are loaded in runtime using dynamic class-loading integrated in the `Java Reflection API`. Solvers represent a variety of finite-difference schemes (fully implicit, semi-implicit, forward-time centred-space, alternating direction implicit), for the numeric calculation of time-temperature profiles $T(t)$. Each problem statement listed below features a dedicated set of solvers. The simplest model applicable to thin, wide samples includes one-dimensional heat flow and radiative cooling from both faces. In case of thick samples, the front surface partially exposed to the laser, or a limited detector field-of-view – a two-dimensional problem may be used instead. Alternatively, when the laser heating is significant, a model accounting for nonlinear radiative cooling is implemented based on a fixed-point iteration algorithm embedded into the finite-difference schemes. Two models are applicable to semi-transparent samples. For diathermic graphite-coated samples, a special solution has been formulated using a modified Thomas algorithm and the Sherman-Morrison-

Woodbury identity. When this simpler model fails, a fully-coupled radiative-conductive problem may be used, with the radiative part treated either using the discrete ordinates method (DOM) with specialised solvers or analytically for the absorbing-emitting case. Finally, a model with distributed heat absorption works on porous samples experiencing bulk laser penetration. Once the problem statement is selected and the initial parameters (e.g., sample thickness l, sample diameter d, etc.) are set, the user is asked to specify the laser pulse and baseline. Currently, rectangular, trapezoidal and exponentially-modified Gaussian pulse shapes are supported with fine tuning of the parameters possible; baselines can either be constant, linear, or sinusoidal. The latter can be adjusted both by fitting to the initial data at $t < 0$ and in the general optimisation procedure. If the laser and detector are out of sync, resulting in a shifted time sequence, this can be corrected for by an optional time shift parameter. Some problem statement require loading additional data on heat capacity C_p and density ρ, also used for thermal conductivity calculation. The parameters are loaded as a set of discrete temperature-value pairs and then interpolated using natural cubic splines with the `Apache Commons Mathematics Library`. The discrete set of data points generated by the solvers is interpolated likewise to enable the calculation of residuals. Task execution requires selecting an optimiser and a linear search algorithm. The user has an ability to include any number of search variables allowed by the model. By default, an approximated Hessian BFGS direction search with a central-difference gradient calculation is employed and the step size determined based on Wolfe conditions. A fallback option consists in using a gradient-descent algorithm with the golden section rule for linear search. During the optimisation cycle, a fixed-size buffer is filled storing parameter values at each iteration, the buffer is used to determine if the procedure converged to a minimum of the objective function. The latter is usually the sum of squared residuals (SSR) divided by the number of data points, with the alternatives being a range-penalised SSR and the AIC statistic. If convergence is reached, a preliminary result is generated and assessed with the help of a statistical toolkit, which includes testing the residuals for normality (Kolmogorov or Anderson-Darling tests) and verifying the search variables are not correlated (Pearson correlation or Spearman's rank correlation test). If the tests pass and the parameter values are deemed sensible (i.e., within the limits specified in a configuration `xml` file), the result is stored in a table with a customisable format. Tasks can be processed individually or in a batch-mode utilising multithreading – in this case, a `ForkJoinPool` is created. Once all tasks are processed, results at similar test temperature can be auto-merged and the standard error calculated.

3. Impact overview

The need for a comprehensive software suite for the LFA was recognised in 2008 after noticing that the standard half-time and logarithmic techniques occasionally resulted in low-accuracy results for data produced by a bespoke LFA instrument designed at the Moscow Engineering Physics Institute. A new Delphi code utilising finite-difference schemes and least-squares minimisation [6] was presented and validated on a set of data recorded from select nuclear materials at high temperatures. Using finite-difference schemes seemed promising as the semi-analytical solutions could incorrectly describe high radiation losses [7]. Full-scale development of a user-friendly Java code began in 2014 relying on the previously collected validation database. In 2018 - 2019, the software was tailored for use in the Materials Research Facility at UKAEA where it was shown to decrease the uncertainty in thermal diffusivity evaluation associated with instrumental problems [8]; this new approach proved to be more effective compared to the standard software packaged with the commercial instrument. The authors contributed to the model development for semi-transparent samples, resulting in a unique set of models recently implemented in PULsE [9]. The fully-coupled radiative-conductive model helped understand the origin of the sharp initial peak observed on coated alumina samples at high temperatures. The various applications are summarised in fig. 2 and table 1, where example conditions and different data treatment capabilities are included. The software is updated on a regular basis, with future work planned on model development for layered samples.

Table 1: Experimental parameters for heating curves plotted in fig. 2

Parameter		Test case		
		(a)	(b)	(c)
Test temperature	(K)	676	2273	1680
Detector		MCT	PbSe	InGaAs
Peak wavelength	(μm)	5.0	4.6	1.5
Laser		Nd:YAG	Nd:YAG	Ruby
Pulse width	(ms)	2.0	5.0	1.0
Pulse energy	(J)	30	30	5
Computation model		2-D with detector sync correction and variable FOV	1-D with sinusoidal baseline	Participating medium (strong scattering)

Acknowledgements

This work was partially funded by the RCUK Energy Programme (Grant No. EP/T012250/1). Special thanks to to Thomas Stainer (`https://github.`

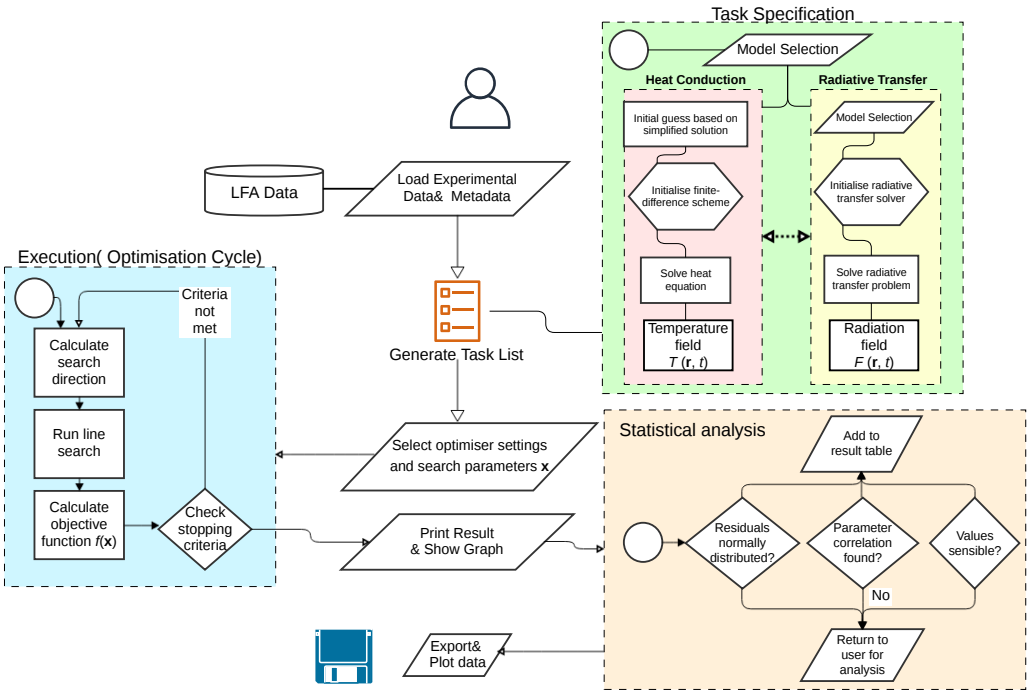

Figure 1: A flowchart of the PULsE software

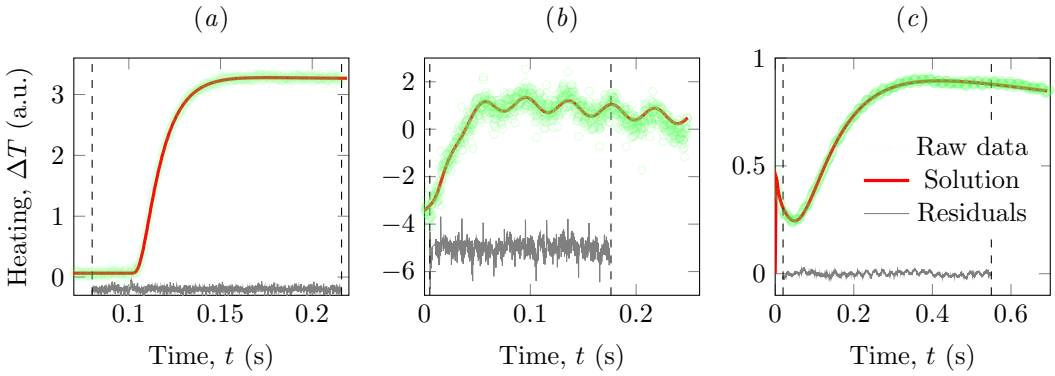

Figure 2: Example tasks: (a) Isotropic AIST graphite sample; (b) Tungsten sample (graphite-coated); (c) Synthetic alumina sample (graphite-coated)

`com/thomasms`) for providing many useful advices on improvement of the code structure and on continuous integration and development aspects.

Required Metadata

Current code version

Ancillary data table required for subversion of the codebase. Kindly replace examples in right column with the correct information about your current code, and leave the left column as it is.

Nr.	Code metadata description	Please fill in this column
C1	Current code version	v1.85_03 (a6e8eaf)
C2	Permanent link to code/repository used for this code version	`https://github.com/kotik-coder/PULsE`
C3	Permanent link to Reproducible Capsule	leave empty
C4	Legal Code License	Apache-2.0 License
C5	Code versioning system used	git
C6	Software code languages, tools, and services used	Java
C7	Compilation requirements, operating environments & dependencies	Apache Maven, Jave SE Development Kit 11
C8	If available Link to developer documentation/manual	`https://kotik-coder.github.io/apidocs/index.html`
C9	Support email for questions	artem.lunev@ukaea.uk, alounev@list.ru

Table 2: Code metadata (mandatory)

Current executable software version

Ancillary data table required for sub version of the executable software: (x.1, x.2 etc.) kindly replace examples in right column with the correct information about your executables, and leave the left column as it is.

Nr.	(Executable) software metadata description	Please fill in this column
S1	Current software version	1.85
S2	Permanent link to executables of this version	`https://github.com/kotik-coder/PULsE/releases/tag/v1.85`
S3	Permanent link to Reproducible Capsule	leave empty
S4	Legal Software License	Apache-2.0 License
S5	Computing platforms/Operating Systems	Any (tested on Windows, Linux, MacOS)
S6	Installation requirements & dependencies	Java 11
S7	If available, link to user manual - if formally published include a reference to the publication in the reference list	`https://kotik-coder.github.io/PULsE_Quickstart_Guide.pdf`
S8	Support email for questions	artem.lunev@ukaea.uk, alounev@list.ru

Table 3: Software metadata (optional)

References

[1] W. J. Parker, R. J. Jenkins, C. P. Butler, G. L. Abbott, Flash method of determining thermal diffusivity, heat capacity, and thermal conductivity, Journal of Applied Physics 32 (1961) 1679–1684. doi:10.1063/1.1728417.

[2] T. Baba, A. Ono, Improvement of the laser flash method to reduce uncertainty in thermal diffusivity measurements, Measurement Science and Technology 12 (2001) 2046.

[3] M. Sheindlin, D. Halton, M. Musella, C. Ronchi, Advances in the use of laser-flash techniques for thermal diffusivity measurement, Review of Scientific Instruments 69 (1998) 1426–1436. doi:10.1063/1.1148776. arXiv:https://doi.org/10.1063/1.1148776.

[4] L. Vozár, W. Hohenauer, Uncertainty of thermal diffusivity measurements using the laser flash method, International Journal of Thermophysics 26 (2005) 1899–1915.

[5] B. Hay, J. Filtz, J. Hameury, L. Rongione, Uncertainty of thermal diffusivity measurements by laser flash method, International Journal of Thermophysics 26 (2005) 1883–1898.

[6] V. Baranov, A. Tenishev, A. Lunev, S. Pokrovskii, A. Khlunov, High-temperature measurements of the thermal conductivity of reactor materials by the laser flash method, Yad. Fiz. Inzhin 2 (2011) 291–302.

[7] A. Philipp, J. F. Eichinger, R. C. Aydin, A. Georgiadis, C. J. Cyron, M. Retsch, The accuracy of laser flash analysis explored by finite element method and numerical fitting, Heat and Mass Transfer 56 (2020) 811–823.

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

[9] A. Lunev, V. Zborovskii, T. Aliev, Complexity matters: highly-accurate numerical models of coupled radiative-conductive heat transfer in a laser flash experiment, 2020. arXiv:2008.04079.