# zlambda v. 0.73 - User manual (not yet completed - we are working on it)

April 27, 2023

## 1 The software overview

The `zlambda` is a numerical simulation software for modeling $Z$-scan experiment and analysis of the heat transfer caused by laser radiation.

In order to speed up the calculations, the program supports multithreading of modern processors with support from 1 to 32 threads. The number of threads used during the calculations is defined by the user. Therefore, in order to parallelize the calculations, for example, on a 32-thread unit, it will be possible to run 4 simulations running 8-thread simultaneously or 2 simulations running 32, etc.

## 2 Software operating and files structure

### 2.1 Start the program

The `zlambda` is a console application that can be run in the terminals of operating systems such as Linux, macOS or Windows. To start a program, open a terminal and go to the directory where the program is located. Then simplify write the command:

```
./zlambda
```

After pressing the enter button, the program will start calculations. Depends on the mode selected in `ConfigExperiment.txt` the application will perform the full or single shot calculation.

### 2.2 File structure

The structure of the program files and directories is as follows:

```
zlambda <Parent Folder>
    |-Results <Folder>
    |   |-Surfaces <Folder>
|   |-HeatMaps <Folder>
|-zlambda <main application>
|-ConfigExperiment.txt <ascii txt file>
|-ConfigSample.txt <ascii txt file>
```

The `zlambda` is the main application, that can be run in the system console. The main application uses the `ConfigExperiment.txt` and `ConfigSample.txt` files for calculation. First config file contains the physical parameters of the $Z$-scan experiment. The second file configures the thermal, physical and numerical parameters of the calculation. The wider description of the `ConfigExperiment.txt` and `ConfigSample.txt` files are presented in the section 2.3 and 2.4 respectively.

## 2.3 Setting up the ConfigExperiment.txt file

The `ConfigExperiment.txt` file contains the set of variables which control the $Z$-scan experiment, simulated by the `zlambda` software.

# The hash symbol comments the line. If the line starts with a #, anything after that symbol will not be interpreted by the program.

Syntax

```
# This is a comment, and this line is not interpreted
# by the program.
```

**SingleShot** The value of this variable indicates the calculation mode of the `zlambda` software. There is a two available mode implemented in the software. If the value of **SingleShot** variable is equal to **1** the calculation will start with single mode and only 1000 iteration will be perform according to other variables set in both ConfigExperiment.txt and `ConfigSample.txt` files. In this mode the sample is irradiated by one pulse of the laser at the sample is located at the $z_0$.

Syntax:

```
SingleShot = <index>
```

Where the `<index>` is equal 0 or 1.

**ZScanRange** indicates the distance between $-z$ and $+z$. This is a distance which the sample moves in the $Z$-scan experiment with the constant velocity.

Syntax:

```
ZScanRange = <real_value>
```

Where the `<real_value>` is a real number representing the distance. Example:

```
# Z-scan range. Distance between -z and +z [mm]
# Below the range is set for 100mm.
ZScanRange = 100
```

**SampleVelocity** This variable sets the sample's velocity at the range from $-z$ to $+z$.

Syntax:

```
SampleVelocity = <real_value>
```

Where the `<real_value>` is a real number of the sample's speed. The unit of the variable `SampleVelocity` is (mm/s). Below example sets the sample speed at 5.5 (mm/s):

```
#Sample velocity [mm/s].
SampleVelocity = 5.5
```

**SimulationTime** This variable controls the time that is simulated during numerical calculations. This value should be selected appropriately to the modeled experiment, taking into account the time needed to simulate the $Z$-scan experiment. For example, if the sample speed is set to 1mm/s and the variable `ZScanRange` is set to 30mm, the time needed to simulate the full $Z$-scan experiment (and Simultaneously the heat transfer) will take 30s. If this variable is set to less than 30 seconds, the program will finish the calculations earlier, before the $Z$-scan experiment is modeled. Therefore, it is good to set simulation time for longer than the modeled $Z$-scan experiment takes. Then it will be possible to observe the cooling processes of the material after the end of the laser irradiation resulting from the simulated $Z$-scan experiment.

Syntax:

```
SimulationTime = <integer_value>
```

Where the `<integer_value>` is a integer number. The unit of this variable is one second - (s). Example:

```
#Simulation time set for 100 seconds
SimulationTime = 100
```

**NumericalTimeStep** This variable specifies the time step that is used in the numerical calculations. If we divide the values of `SimulationTime` and `NumericalTimeStep` by ourselves, we get the number of iterations that the program must perform to complete the calculations. If the `SimulationTime` is set on 100s and the `NumericalTimeStep` is set on 0.2ms the number of iteration will be 500000.

Syntax:

```
NumericalTimeStep = <real_value>
```

Where the `<real_value>` is a real number of the numerical differential time step. The unit of this variable is one second (s). Below example sets the time step of numerical calculation to 0.2ms:

```
#Duration of the numerical time step. Real value in [s]
NumericalTimeStep = 0.2e-3
```

**LambdaOfLaser** This variable specifies the wavelength of the laser used in simulation. The type of variable is real number and the unit is nanometers (nm). Syntax:

```
LambdaOfLaser = <real_value>
```

Below example sets the laser wavelength to 532nm:

```
#Laser wavelength [nm]
LambdaOfLaser = 532.0
```

**omegaZero** is the beam waist $\omega_0$ at $z = z_0 = 0$. Syntax:

```
omegaZero = <real_value>
```

Where the `<real_value>` is a real number of the beam waist expressed in meters (m). The following example sets $\omega_0$ to 25$\mu$m

```
#Beam radius at z0 [m] sets to 25 micrometers.
omegaZero = 25e-6
```

**LaserPulseDuration** is a real variable which specifies the pulse duration of the laser used ind simulation. Syntax:

```
LaserPulseDuration = <real_value>
```

The `<real_value>` is the pulse duration expressed in seconds (s). The example sets pulse duration to 100ps:

```
#Laser pulse duration [s] (i.e. 5ns -> 5e-9, 100ps -> 100e-12)
LaserPulseDuration = 100e-12
```

```
#Laser Pulse Energy (if the LaserPulseDuration > 0) [J]
LaserPulseEnergy = 9.81748e-4

#LaserPower (if the LaserPulseDuration = 0) [W]
LaserCWPower = 10e-3

#Pulse Repetition frequency [Hz] for pulsed laser. Please notice that the value must
LaserPulseRepetition = 10

#Save Zero Planes interval [s] If the value is 0, then data will be saved after each
SaveZPRInterval = 0.01

#Save Zero Planes (1 = Yes, 0 = No) - indicate that the program will be saved Zero P
DoesSaveZPRInterval = 1

#Save all sample Interval [s] - If the value is 0, then data will be saved after eac
SaveASInterval = 25.0

#Save all sample (1 = Yes, 0 = No) - indicate that the program will be saved All sam
DoesSaveASInterval  = 1

Threads = 4
```

If you want to comment some phrase in the ConfigExperiment.txt file, please use # at the beginning of the line # Below you can set the simulation condition. # Please notice that you have to use the units pointed in [ ] brackets.

## 2.4 Setting up the ConfigSample.txt file

## Alphabetical Index

## References