Matlab Laser Toolbox
User Manual

Gert-willem Römer

UNIVERSITY OF TWENTE.
Faculty of Engineering Technology
Chair of Applied Laser Technology
P.O. box 217, NL7500AE, Enschede, the Netherlands
Email g.r.b.e.romer@wb.utwente.nl
Phone +31 (53) 4892519

September 23, 2010

Version 0.1 beta

Preface

The Matlab® Laser Toolbox provides several functions and scripts for analysis and visualization of laser beam properties, as well as, functions to calculate the interaction (e.g. induced temperature by absorbed laser energy in a solid) in a material.

Contents

1 Introduction

1.1 Matlab Laser Toolbox

Matlab® is high-level interpreted language and interactive environment for algorithm development, data visualization, data analysis, and numeric computation, Developed by MathWorks Inc. (http://www.mathworks.com) Add-on toolboxes, which are collections of special-purpose Matlab functions and scripts, extend the Matlab environment to solve particular classes of problems.

The Laser Toolbox provides several functions and scripts for analysis and visualization of laser beam properties, as well as, functions to calculate the interaction (e.g. induced temperature by absorbed laser energy in a solid) in a material. This User Manual addresses the usage of the Laser Toolbox including examples.

The author appreciates that publications describing work using the Laser Toolbox quote one of the references given below:

- G.R.B.E. Römer and A.J. Huis in 't Veld. Matlab Laser Toolbox. Physics Procedia, **5**(0), Pages 413-419.
- G.R.B.E. Römer and A.J. Huis in 't Veld. Matlab Laser Toolbox. Proceedings of the 29th International Congress on Applications of Lasers & ElectroOptics (ICALEO), September 26-30, 2010, Anaheim, CA, USA.

Critical evaluation of the Laser Toolbox is welcomed.

1.2 System requirements

The Laser Toolbox was developed for Matlab version 7.5, and should run on any operating system supported by Matlab.

1.3 Installation instructions

1. Download the Matlab Laser Toolbox, as a compressed ZIP file from the website: `http://www.wa.ctw.utwente.nl/software/laser/`
2. Decompress the ZIP file into a directory on your hard disk, e.g. in `c:/laser`
3. Start Matlab
4. Run Matlab's graphical path set tool by executing the command `pathtool` from the command line, and add the directory of the Laser Toolbox (here `c:/laser`) to the Matlab search path, by clicking the "Add folder" button. Or execute the command `addpath 'c:/laser' -end` from the command line
5. Execute the command `help laser` from the command line, which should give to following output

```
Laser Toolbox
Version 0.1 (R2007b) September 20th, 2010

Power density distribution.
   gauss      - Gaussian power density profile.
   templ      - Gauss-Laguerre mode.
   temmn      - Gauss-Hermite mode.
   tophat     - Top hat power density profile.
   rectunif   - Rectangular uniform power density profile.
   t2surfsrc  - Surface power density profile based on desired temperature profile.
   mdfread    - Read MDF file from disk.
   uffread    - Read UFF file from disk.
   plotpdd    - Power density distribution plot.
   disppdd    - Display power density distribution.

Laser beam.
   iso11146   - Beam propagation ratios according to ISO 11146-1.
   caustic    - Caustic plot.
   dispbeam   - Display beam characteristics.

Temperature profile.
   tpntsrc    - Temperature profile of point surface heat source.
   tlinesrc   - Temperature profile of line heat source.
   tsurfsrc   - Temperature profile of surface heat source.
   plottemp   - Temperature plot.


General.
   overlap    - Overlap plot of pulses.
   materials  - Sets and saves materials parameters.
   momentxy   - First and second order moments.
   num2pstr   - Convert numbers to a prefixed string.
   hermite    - Hermite polynomial.
   laguerre   - Laguerre polynomial.

(c) 2008-2010 G.R.B.E. Romer, University of Twente.
This Matlab Laser Toolbox is for private, non-commercial, single home
computer, or educational use only. The use of Matlab Laser Toolbox for
commercial purposes is strictly prohibited. Please read the detailed
license agreements in the Matlab "Laser Toolbox User Manual".
```

1.4 Licence and disclaimer

Licence

The Matlab Laser Toolbox is free for private, non-commercial, single home computer, or educational use only. The use of Matlab Laser Toolbox for commercial purposes is strictly prohibited. By downloading the Laser Toolbox you express your approval of the conditions and terms hereof and you are bound thereby. Should you disagree with the conditions and terms hereof, promptly terminate the use of this software and destroy any copies thereof or delete the program already downloaded. The Laser Toolbox subject thereto is and continues to be a property of the University of Twente. The Laser Toolbox is also subject to copyrights, and as such it is subject to a full protection thereof.

Disclaimer

The Matlab Laser Toolbox is provided "as is", without warranty of any kind, express or implied, including but not limited to the warranties of merchantability, fitness for a particular purpose and noninfringement. In no event shall the University of Twente, nor the authors or copyright holders be liable for any claim, damages or other liability, whether in an action of contract, tort or otherwise, arising from, out of or in connection with the software or the use or other dealings in the software.

2 Data structures

Four data structures hold and organize related data to be handled and processed by the Laser Toolbox.

2.1 Power density distribution (pdd)

Table 1 shows the fields of the data structure, referred to here as `pdd` (an abbreviation of *power density distribution*). This structure holds data related to a power density distribution $I(x, y)$ [W/m^2] (measured or calculated) at a specified plane along the axis z of propagation of the laser beam.

Table 1: Data structure pdd holding power density distribution(s).

FIELD	DESCRIPTION
name	Identifier
fileid	ID of measurement file (if any)
comment	Comments (if any)
px	No. of values in x direction of I(x,y)
py	No. of values in y direction of I(x,y)
rangexy	Length [m] and width [m] of xy plane
x	x coordinates at which I(x,y) is defined
y	y coordinates at which I(x,y) is defined
z	z coordinate at which I(x,y) is defined
amp	Amplification [dB] of sensor (if any)
average	No. of measurements of I(x,y)
offset	Offset (noise level) of sensor
lambda	Laser wavelength [m]
foclen	Focal length of lens used (if any)
datestr	Date (e.g. of measurement)
timestr	Time (e.g. of measurement)

ixy	Array of values of I(x,y)

It should be noted that some data fields are optional. In the case several power density distributions are defined (or measured) at specified locations z along the optical axis, an array of these data structures is constructed. Then, each entry of the array holds a `pdd` structure defined in Table 1, one for each specified location z. Although arrays of structures in Matlab are less computationally efficient, than structures of arrays, the readability and usability of the code of the Laser Toolbox requires the use of arrays of structures. Below some functions which create a `pdd` structure, or create it from measured data.

2.2 Propagation properties of a laser beam (beam)

Table 2 shows the fields of the data structure, referred to here as `beam`, holding propagation properties of a laser beam as defined in the ISO11146 standard dealing with *Lasers and laser related equipment - Test methods for laser beam widths, divergence angles and beam propagation ratios.*

Table 2: Data structure beam holding propagation properties of a laser beam.

FIELD	DESCRIPTION
name	Identifier, string
lambda	Laser wavelength [m], double
power	Power [W] of the laser radiation, double
z	Locations z along axis of propagation, vector of doubles
dr	Diameters of the beam at locations z, vector of doubles
dx	Width of the beam at locations z, vector of doubles
dy	Length of the beam at locations z, vector of doubles
eta	Ellipticity of the beam at locations z, vector of doubles
d0r	Diameter of waist/focus of beam, double
z0r	Location at optical axis of waist/focus, double
zRr	Rayleigh length of beam,, double
divr	Far-field divergence angle [rad] of the beam, double
M2	Beam propagation ratio (quality) M2, double
Cr	Coefficients of 2nd order polynomial describing beam diameter propagation, vector of 3 doubles
d0x	Diameter of waist/focus in xz-plane, double
z0x	Location of waist/focus in xz-plane, double
zRx	Rayleigh length of beam in xz-plane, double
divx	Far-field divergence angle [rad] in xz-plane, double
M2x	Beam propagation ratio M2 in xz-plane, double
Cx	Coeff. of 2nd order polynomial describing diameter propagation in xz-plane, vector of 3 doubles
d0y	Diameter of waist/focus in yz-plane, double

z0y	Location of waist/focus in yz-plane, double
zRy	Rayleigh length of beam in yz-plane, double
divy	Far-field divergence angle [rad] in xz-plane, double
M2y	Beam propagation ratio M2 in yz-plane, double
Cy	Coeff. of 2nd order polynomial describing diameter propagation in yz-plane, vector of 3 doubles

This datastructure allows also the characterization of simple astigmatic laser beams. As the propagation of the beam diameter along the axis of propagation is described by a second order polynomial, the fields containing the vectors `z`, `dr`, `dx`, `dy` and `eta` should, at least, hold three elements. Only then the other fields of the data structure are defined. The function `iso11146` of the Laser Toolbox can be used to calculate the data fields of the beam structure from the data fields of the `pdd` structure.

2.3 Temperature profile (temp)

Table 3 shows the fields of the data structure, referred to here as `temp` (short for *temperature*), holding data related to a 3D temperature profilei.e temperature rise $T(x, y, z)$ [K], e.g. induced by absorbed laser energy. In the case a transient temperature distribution is defined, at specified time instances t [s], an array of this data structure is constructed. Then, each entry of the array holds a `temp` structure defined in Table 3, one for each every instant t specified. Note the similarity of this data structure to the structure `pdd` of the power density distribution(s).

Table 3 Data structure temp holding 3D temperature profile(s).

FIELD	DESCRIPTION
name	Identifier
x	x coordinates of T(x,y,z)
y	y coordinates of T(x,y,z)
z	z coordinates of T(x,y,z)
Txy	Values T(x,y,z)
t	Time t>=0 [s]

2.4 Material

Table 4 shows the fields of the data structure, referred to here as `mat`, holding material properties. It should be noted that, some data fields are optional. In the case the material properties are temperature dependent, an array of this data structure is constructed. Then, each entry of the array holds a material structure defined in Table 4, one for each temperature. Note that, regarding this aspect, the similarity of this data structure to the data structures `pdd` and `temp`.

Table 4 Data structure material holding material properties

FIELD	DESCRIPTION
A	Absorption*

```
lambda  Laser wavelength(s), scalar or vector of doubles*
n       Refractive index/indices*
k       Extinction coefficient(s)*
rho     Density [kg/m3]
K       Heat conductivity [W/(mK)]
Cp      Heat capacity [J/K]
Tm      Melt temperature [K]
Tv      Vaporisation temperature [K]
* defined at wavelengths specified by lambda
```

3 Functions by category

The functions of the Laser Toolbox are easy to analyze and easily modified, in the case the user needs to extend or adapt the functionality of the scripts. In the remainder of this section, the usage of these functions are illustrated by examples.

3.1 Power density distribution

3.1.1 GAUSS

A Gaussian power density distribution in an xy-plane is defined by

$$I(x,y) = \frac{8P}{\pi d^2} \exp\left[-\left(\frac{2\sqrt{2}}{d}\right)^2 \cdot (x^2+y^2)\right] \tag{1}$$

where P [W] is the laser power and d [m] the diameter of the power density distribution.

`PDD=GAUSS(P,d)` returns a structure (struct) `PDD` with a Gaussian power density distribution of power `P` [W] and diameter `d` [m].

`PDD=GAUSS(P,d,N)` returns the struct `PDD` of which the Gaussian power density `PDD.ixy` [W/m^2] is a N-by-N matrix, and the corresponding x and y vectors have length `N`. If not specified `N` equals 128.

`PDD=GAUSS(P,d,X,Y)` returns the struct `PDD` of a Gaussian power density distribution at locations defined by the vectors `X` and `Y`.

3.1.2 TEMPL

The power density distribution of a Gauss-Laguerre Transverse Electro Magnetic (TEM) mode, designated by the integer mode numbers p and l, in an xy-plane is defined by

$$I_{pl}(r,\varphi) = I_0 \left(\frac{8r^2M^2}{d^2}\right)^l \left[L_l^p\left(\frac{8r^2M^2}{d^2}\right)\right]^2 \cos^2(l\varphi) \exp\left(-\frac{8r^2M^2}{d^2}\right) \tag{2}$$

where I_0 [W·m^{-2}] denotes the intensity scale factor, d the diameter of the profile and $M^2 \geq 1$ the *times-diffraction-limited-factor*, or short beam quality number. The beam diameter and the beam quality number are defined according to the ISO11146 standard. Finally, $L_l^p(\cdot)$ denotes the generalized Laguerre polynomial of order p and index l. The intensity scale factor I_0, expressed in the modes numbers and the total laser power P, reads

$$I_0 = \begin{cases} \dfrac{8M^2}{\pi d^2} P\,, & l = 0 \\ \dfrac{16M^2 p!}{\pi d^2 (p+l)!} P\,, & l = 1, 2, 3, \ldots \end{cases} \tag{3}$$

For Gauß-Laguerre modes the times-diffraction-limited-factor equals $M^2 = 2p + l + 1$. In stead of polar coordinates (r, φ) carthesian coordinates are returned by the function `TEMPL`.

`PDD=TEMPL(p,l,P,d)` returns a structure (struct) `PDD` with a laser power density distribution of a passive laser resonator in Gauss- Laguerre mode (TEMpl) with scalar mode numbers `p` and `l`, power `P` [W] and diameter `d` [m].

`PDD=TEMPL(p,l,P,d,N)` returns the struct `PDD` of which the power density `PDD.ixy` [W/m^2] is a N-by-N matrix, and the corresponding x- and y-vectors have length `N`. If not specified `N` equals 128.

`PDD=TEMPL(p,l,P,d,X,Y)` returns the struct `PDD` of the power density distribution at locations defined by the vectors `X` and `Y`.

3.1.3 TEMNN

The power density distribution of a Gauss-Herimte Transverse Electro Magnetic (TEM) mode, designated by the integer mode numbers m and n, in an xy-plane is defined by

$$I_{mn}(x,y) = I_0 \left[H^m \left(\frac{2x\sqrt{2}\sqrt{2m+1}}{d_x} \right) \exp\left(\frac{-4x^2(2m+1)}{d_x^2} \right) \right]^2 \times \left[H^n \left(\frac{2y\sqrt{2}\sqrt{2n+1}}{d_y} \right) \exp\left(\frac{-4y^2(2n+1)}{d_y^2} \right) \right]^2$$

where $H^m(\cdot)$ denotes the m^{th} order Hermite polynomial, d_x [m] and d_y [m] the beam length and width respectively. The width and length are defined according to the ISO11146. The intensity scale factor I_0, expressed in the modes numbers and the total laser power P, reads

$$I_0 = \frac{2^{3-m-n}\sqrt{(2m+1)(2n+1)}}{n!m!\pi d_x d_y} P \tag{4}$$

The times-diffraction-limited-factor for Gauß-Hermite modes equals $M^2 = m + n + 1$.

`PDD=TEMmn(m,n,P,d)` returns a structure (struct) `PDD` with a laser power density distribution of a passive laser resonator in Gauss-Hermite mode (TEM_{mn}) with scalar mode numbers `m` and `n`, power `P` [W] and where diameter `d` [m] is a 2-element vector with dimensions of the distribution in x and y direction.

`PDD=TEMMN(m,n,P,d,N)` returns the struct `PDD` of which the power density `PDD.ixy` [W/m^2] is a N-by-N matrix, and the corresponding x- and y-vectors have length `N`. If not specified `N` equals 128.

`PDD=TEMMN(m,n,P,d,X,Y)` returns the struct `PDD` of the power density distribution at locations defined by the vectors `X` and `Y`.

3.1.4 TOPHAT

A Top Hat power density distribution in an xy-plane is defined by

$$I(x,y) = \begin{cases} P/\left(\pi \frac{d^2}{4}\right), & \sqrt{x^2+y^2} \leq \frac{d}{2} \\ 0, & \sqrt{x^2+y^2} > \frac{d}{2} \end{cases} \tag{5}$$

where d [m] is the diameter of the power density profile, and P [W] the total laser power.

`PDD=TOPHAT(P,d)` returns a structure (struct) `PDD` with a Top Hot power density distribution of power `P` [W] and diameter `d` [m].

`PDD=TOPHAT(P,d,N)` returns the struct `PDD` of which the Top Hat power density `PDD.ixy` [W/m^2] is a N-by-N matrix, and the corresponding x and y vectors have length `N`. If not specified `N` equals 128.

`PDD=TOPHAT(P,d,X,Y)` returns the struct `PDD` of the power density distribution at locations defined by the vectors `X` and `Y`.

3.1.5 RECTUNIF

A Rectangular Uniform power density distribution in an xy-plane is defined by

$$I(x,y) = \begin{cases} \frac{P}{d_x d_y}, & \{(x,y) \in \mathbb{R}, |x| \leq d_x \ \& \ |y| \leq d_y\} \\ 0 & \text{elsewhere} \end{cases} \tag{6}$$

where d_x [m] and d_y [m] are the dimensions of the power density profile, and P [W] the total laser power.

`PDD=RECTUNIF(P,d)` returns a structure (struct) `PDD` with a rectangular uniform power density distribution of power `P` [W] and where diameter `d` [m] is a 2-element vector with dimensions of the distribution in x and y direction.

`PDD=RECTUNIF(P,d,N)` returns the struct `PDD` of which the power density `PDD.ixy` [W/m^2] is a N-by-N matrix, and the corresponding x- and y-vectors have length `N`. If not specified `N` equals 128.

`PDD=RECTUNIF(P,d,X,Y)` returns the struct `PDD` of the power density distribution at locations defined by the vectors `X` and `Y`.

3.1.6 T2SURFSRC

Besides the calculation of the temperature distribution, given the intensity profile I and the function W, see equation (11) also the inverse problem can be solved by applying the FFT method, see section 3.3.3. That is, given a desired temperature distribution T and the function W, the required intensity profile I inducing the desired temperature field can be determined, by rewriting equation (20) as

$$\mathcal{F}_2\{I\} = \frac{1}{A}\mathcal{F}_2\{T\}\left[\mathcal{F}_2\{W\}\right]^{-1} \tag{7}$$

and by applying the inverse Fourier transform. Then, the intensity profile reads,

$$I = \frac{1}{A}\mathcal{F}_2^{-1}\{\mathcal{F}_2\{T\}\left[\mathcal{F}_2\{W\}\right]^{-1}\} \tag{8}$$

This is a very powerful method as it allows the calculation of the required intensity profile directly from any desired temperature distribution. It is evident that this method will only yield a feasible intensity profile ($I(x,y) \geq 0 \; \forall \; (x,y) \in \mathbb{R}^2$), if the desired temperature field is physically feasible.

`PDD=T2SURFSRC(MAT,T,V)` returns a `PDD` struct containing a power density distribution, which, when absorbed at the surface (z=0) generates the 2D temperature profile struct `T` in a semi-inifite material `MAT`, when the `PDD` moves at velocity `V` [m/s] relative to the material. The x and y dimensions of `PDD` are defined by the x and y coordinates of `PDD`.

3.1.7 MDFREAD

`PDD=MDFREAD(FILENAME)` reads the data from a file (extension `*.mdf`), created by the "LaserDiagnoseSoftware" (version 2.81) by Primes GmbH representing the power density profile(s) (caustic) of a laser beam measured by the FocusMonitor of Primes GmbH (`http://www.primes.de`) and returns an array of `PDD` stucts (one for each plane), with the following fields

```
name     (FocusMonitor, string)
filename (Filename, string)
fileid   (File ID, string)
comment  (Comments, string)
px       (No of pixels along x-axis, integer)
py       (No of pixels along y-axis, integer)
rangexy  (Row vector with measuring range in x [m] and y [m] direction)
z        (z-position [m] along the axis of propagation)
posxy    (Row vector with x [m] and y [m] position of measuring range)
amp      (Amplification of the sensor signal [dB], double)
```

```
average   (Number of averages, integer)
offset    (Offset (noise level) of data, integer)
lambda    (Wavelegth laser [m])
power     (Laser power [W])
foclen    (Focal length [m])
datestr   (Date, string)
timestr   (Time, string)
dataxy    (Data, px*py matrix, integers)
x         (x coordinates, vector of px elements)
y         (y coordinates, vector of py elements)
dataxy    (Data, px*py matrix, integers)
ixy       (px*py matrix power density [W/m^2])
```

If `FILENAME` is ommitted, a dialog box for the user to select a filename is displayed.

3.1.8 UFFREAD

`PDD=UFFREAD(FILENAME)` reads the data from a file (`D*.*`) created by the "Laserscope" by Prometec GmbH representing the power density profile (caustic) of a laser beam measured by the Laserscope UFF100 of Prometec GmbH (`http://www.prometec.de`) and returns a `PDD` stuct, with the following fields

```
name          (Prometec UFF100 Laserscope, string)
filename      (File ID, string)
comment       (Comments, string)
px            (No of pixels along x-axis, integer)
py            (No of pixels along y-axis, integer)
focus         (Integer)
amp           (Amplification of the sensor signal, integer)
windowsize    (Window size, integer)
windowcenterx (x coordinate of window center, 48x1 double)
windowcentery (y coordinate of window center, 48x1 double)
versionsensor (Version sensor, integer)
versionuff    (Version UFF, integer)
power         (Laser power [W])
datestr       (Date, string)
timestr       (Time, string)
x             (x coordinates, vector of px elements)
y             (y coordinates, vector of py elements)
ixy           (px*py matrix power density [W/m^2])
```

If `FILENAME` is ommitted, a dialog box for the user to select a filename is displayed.

3.1.9 PLOTPDD

PLOTPDD(PDD) plots a contour plot, two cross section as well as a parametric mesh of the power density profile(s) in the struct PDD; one figure for each plane.

3.1.10 DISPPDD

DISPPDD(PDD) displays the characteristics of each plane of a power density profile struct PDD.

S=DISPPDD(PDD) returns a cell array S of strings of the characteristics of each plane.

3.2 Laser beam propagation parameters

The ISO11146 standard governs test methods for laser beam widths, divergence angles and beam propagation ratios.

3.2.1 ISO11146

BEAM=ISO11146(PDD) returns a strcut BEAM containing the propagation parameters (or ratio's) according to the international standard ISO11146-1 *Lasers and laser-related equipment - test methods for laser beam widths, divergence angles and beam propagation ratios - part 1: Stigmatic and simple astigmatic beams* (2005), with the following (optional) fields

```
name       (Identifier, string)
filename   (Filename, string)
wavelength (Laser wavelength [m], double)
z          (z-coordinates along optical axis, vector of doubles)
mx         (1st moment at all planes in x direction, vector of doubles)
my         (1st moment at all planes in y direction, vector of doubles)
vx         (2nd moment (variance) at all planes in x direction, vector of doubles)
vy         (2nd moment (variance) at all planes in y direction, vector of doubles)
dx         (Width of PDD at all planes in x direction, vector of doubles)
dy         (Length of PDD at all planes in x direction, vector of doubles)
dr         (Diameter of PDD at all planes in x direction, vector of doubles)
eta        (Elipticity dx/dy of PDD at all planes in x direction, vector of doubles)
```

If PDD holds three planes or more the following fields are also returned

```
Cx         (Coefficients of 2nd order polynomial fit of PDD widths, vector of 3 doubles)
z0x        (Location of focus/waist [m] along optical axis in XZ plane, double)
d0x        (Width of focus/waist [m] in XZ plane, double)
divx       (Full fair-field divergence angle [mrad] in XZ plane, double)
zRx        (Rayleigh length [m] in XZ plane, double)
M2x        (Times-limited-diffraction number (beam quality) in XZ plane, double)
Cy         (Coefficients of 2nd order polynomial fit of PDD widths, vector of 3 doubles)
z0y        (Location of focus/waist [m] along optical axis in YZ plane, double)
```

```
d0y        (Length of focus/waist [m] in YZ plane, double)
divy       (Full fair-field divergence angle [mrad] in YZ plane, double)
zRy        (Rayleigh length [m] in YZ plane, double)
M2y        (Times-limited-diffraction number (beam quality) in YZ plane, double)
Cr         (Coefficients of 2nd order polynomial fit of PDD widths, vector of 3 doubles)
z0r        (Location of focus/waist [m] along optical axis, double)
d0r        (Diameter of focus/waist [m], double)
divr       (Full fair-field divergence angle [mrad], double)
zRr        (Rayleigh length [m], double)
M2r        (Times-limited-diffraction number (beam quality), double)
```

3.2.2 CAUSTIC

`CAUSTIC(BEAM)` plots the beam dimensions (width, length, diameter) of `BEAM` struct along the optical axis z. If three or more planes are defined in BEAM also the corresponding 2n order polynomial fits are plotted.

3.2.3 DISPBEAM

`DISPBEAM(BEAM)` displays the characteristics of laser beam struct `BEAM`.
`S=DISPPDD(BEAM)` returns a cell array of strings of the characteristics of the laser beam.

3.3 Temperature models

3.3.1 TPNTSRC

Assume a semi-infinite substrate with constant material parameters, and a point source of heat of P [W] on the substrate's surface. This line heat source will induce the following 2D temperature (rise) in the substrate when it is moving at a constant velocity of v [m/s] in the x-direction relative to the surface

$$T(x, y, z, t) = \frac{1}{2\pi K\sqrt{x^2 + y^2 + z^2}} \exp\left[-\frac{v}{2\kappa}\left(x + \sqrt{x^2 + y^2 + z^2}\right)\right] \tag{9}$$

where K denotes the thermal conductivity [W/(m· K)] and κ the thermal diffusitivity [m^2/s] of the material.

`T=TPNTSRC(MAT,P,V,X,Y)` returns the 3D temperature profile `T` at x, y and z coordinates defined by vectors `X`, `Y`, and `Z` in a semi-inifite material MAT due to a point heat source of `P` [W] moving at velocity `V` [m/s] in the positive x-direction relative to the material.

3.3.2 TLINESRC

Assume a semi-infinite substrate with constant material parameters, and a semi-infinite line source of heat of Q [W/m] perpendicular to the substrate's surface. This line heat source will induce the following 2D temperature (rise) in the substrate when it is moving at a constant velocity of v [m/s] in the x-direction relative to the surface*

$$T(x,y) = \frac{Q}{2\pi K} \exp\left(\frac{-vx}{2\kappa}\right) K_0 \left(\frac{v\sqrt{x^2+y^2}}{2\kappa}\right) \tag{10}$$

where K denotes the thermal conductivity [W/(m· K)] and κ the thermal diffusitivity [m^2/s] of the material. And K_0 is the modified Bessel function of order 0.

`T=TLINESRC(MAT,Q,V,X,Y)` returns the 2D temperature profile `T` at x and y coordinates defined by vectors `X` and `Y`, in a semi-inifite material `MAT` due to a line heat source of `Q` [W/m] moving at velocity `V` [m/s] in the positive x-direction relative to the material.

3.3.3 TSURFSRC

Assume a semi-infinite substrate with constant material parameters, and a surface heat source (absorbed laser energy) defined by the power density $I(x,y)$ [W/m^2] moving over the substrate's surface. This surface heat source will induce the following 3D temperature (rise) in the substrate when it is moving at a constant velocity of v [m/s] in the x-direction relative to the surface of

$$T(x,y,z,t) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} AI(x',y')W(x,y,z,x',y',v)U(R,t,v)\,dx'dy' \tag{11}$$

where $R = \sqrt{x^2+y^2+z^2}$ and

$$W(x,y,z,x',y',v) = \frac{1}{2\pi KR} \exp\left[-\frac{v}{2\kappa}\left(x - x' + R\right)\right] \tag{12}$$

and

$$\begin{aligned} U(R,t,v) =& \frac{\vartheta}{\sqrt{\pi}} \int_{1/\sqrt{\kappa t}}^{\infty} \exp\left[-\frac{\left(\vartheta\xi^2 - v/\kappa\right)^2}{4\xi^2}\right] d\xi \\ =& \frac{1}{2}\left[1 - \operatorname{erf}\left(\frac{R - vt}{2\sqrt{\kappa t}}\right) + e^{\vartheta v/\kappa}\left(1 - \operatorname{erf}\left(\frac{R+vt}{2\sqrt{\kappa t}}\right)\right)\right] \end{aligned} \tag{13}$$

in which K denotes the thermal conductivity [W/(m· K)] and κ the thermal diffusitivity [m^2/s] of the material.

This expression can be evaluated numerically in two ways: multi-integration and using the Fast Fourier Transform (FFT). For simplicity the steady state situation $t \to \infty$, so $U = 1$, is considered only below.

*J.M. Dowden. The Mathematics of Thermal Modeling. An Introduction to the theory of laser material processing. Chapman & Hall, 2001, equation (3.21)

Multi-integration The laser intensity profile $I(x, y)$ is approximated as piecewise constant with the value $I(x_i, y_j) = I(x, y)$ in the region

$$\left\{ (x_i, y_j) \in \mathbb{R}^2 \middle| \, x_i - \frac{L}{2N} \leq x \leq x_i + \frac{L}{2N} \wedge y_j - \frac{L}{2N} \leq \; y \leq y_j + \frac{L}{2N} \right\} \tag{14}$$

representing a square grid of $N \times N$ equidistant points in the xy-plane,

$$(x, y) \in \left[-\frac{L}{2} \leq x \leq \frac{L}{2}, -\frac{L}{2} \leq y \leq \frac{L}{2} \right] \tag{15}$$

The temperature distribution (11) with $U' = 1$, can be numerically evaluated as

$$T(x, y, z, t) = T_0 + \frac{AL^2}{2\pi K N^2} \sum_{i=1}^{N} \sum_{j=1}^{N} \frac{I(x_i, y_j)}{\vartheta'} \exp\left[-\frac{v(x - x_i) + \vartheta'}{2\kappa} \right] \tag{16}$$

where $\vartheta' = \sqrt{(x - x_i)^2 + (y - y_j)^2 + z^2}$. As ϑ' is a factor in the denominator of the function W it represents a singularity, which causes numerical problems if $\vartheta' \ll 1$. In that case the function W should be replaced by W_0, which is defined by

$$\begin{aligned} W_0(x, y, z, x_i, y_j, v) = {} & \frac{1}{2\pi K \Delta x \Delta y} \exp\left(-\frac{vz}{2\kappa} \right) \times \\ & \left[\Delta x \ln\left(\frac{w_1 + \Delta x}{w_1 - \Delta x} \right) + \Delta y \ln\left(\frac{w_2 + \Delta y}{w_2 - \Delta y} \right) \right. \\ & \left. -4z \left(\tan^{-1}\left(\frac{w_1}{2z} \right) + \tan^{-1}\left(\frac{w_2}{2z} \right) \right) + 2z\pi \right] \end{aligned} \tag{17}$$

where $\Delta x = \Delta y = L/N$ and

$$\begin{aligned} & w_1 = \sqrt{\frac{\Delta x^2 + 4z^2 \cos^2 \beta}{\sin^2 \beta}}, \; w_2 = \sqrt{\frac{\Delta y^2 + 4z^2 \sin^2 \beta}{\cos^2 \beta}}, \\ & \beta = \tan^{-1}\left(\frac{\Delta y}{\Delta x} \right) \end{aligned} \tag{18}$$

In the case $z = 0$ and $\vartheta' \ll 1$, function W should be replaced by $\lim_{z \to 0} W_0$. The number of operations, which are required to evaluate (16) for a xy-plane of $N \times N$ nodal points, is proportional to N^4. This is denoted by $O(N^4)$. For accuracy reasons, N must be large (typically $N > 250$).

Two-dimensional Fast Fourier Transform Even fewer operations are required when the two-dimensional Fast Fourier Transform (FFT) algorithm is applied to evaluate (11) numerically. For that purpose it is noted that the steady state version ($U = 1$) of equation (11) can be rewritten as

$$T(x, y, z) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} AI(x', y', t') W(x - x', y - y', z, v) \, dx' dy' \tag{19}$$

which is a convolution of the intensity profile AI and the function W, with respect to the coordinates x and y, — i.e. $T = AI * W$, where $*$ denotes the convolution operator. Hence, by applying the two-dimensional Fourier transform $\mathcal{F}_2\{\cdot\}$, equation (19) can be rewritten as

$$\mathcal{F}_2\{T\} = \mathcal{F}_2\{AI * W\} = A\mathcal{F}_2\{I\}\mathcal{F}_2\{W\} \tag{20}$$

where the well known equality $\mathcal{F}_2\{f * g\} = \mathcal{F}_2\{f\}\mathcal{F}_2\{g\}$ has been applied. Finally, by applying the inverse Fourier transform to this expression yields,

$$T = A\mathcal{F}_2^{-1}\left\{\mathcal{F}_2\left\{I\right\}\mathcal{F}_2\left\{W\right\}\right\} \tag{21}$$

By replacing the continuous convolution by discrete convolution, and using the two-dimensional FFT method, the temperature distribution (21) can be calculated numerically in only $O(N^2)$ operations. This method was used to calculate the temperature profile in a substrate.

A basic condition for the two-dimensional Fourier transformation is that the function to be transformed is periodic in the xy-plane. This is not the case for the intensity profile I and the function W, and will introduce errors in the evaluation of the temperature distribution. These errors are introduced by the truncation of I and W—i.e. if L (grid size) is chosen too small, then for $|x| > L$ and $|y| > L$, $W(x, y) \neq 0$ and $I(x, y) \neq 0$. The errors are significant for low temperatures only. To reduce these errors, the region of calculation should be chosen sufficiently large. As a rule of thumb, one should choose $L > 3d$, where d is the diameter of the laser beam.

`T=TSURFSRC(MAT,PDD,V)` returns steady state (time is infinity) 3D temperature profile `T` of a struct `TEMP` in a semi-inifite material `MAT` due to a surface heat source defined by a single plane in `PDD` moving at velocity `V` [m/s] relative to the material. The x and y dimensions of `T` are defined by the x and y coordinates of `PDD`. The z coordinates (depth) are 0, $d_h/2$ and d_h [m] by default where d_h is the heat penetration depth.

`T=TSURFSRC(MAT,PDD,V,TIME)` returns an array of temperature structs `T`; one struct per time instance defined by vector `TIME`. If not defined `TIME=Inf`.

`T=TSURFSRC(MAT,PDD,V,TIME,Z)` returns an array of temperature structs `T`; one struct per time instance defined by vector `TIME`, at z coordinates [m] defined by vector `Z`.

`T=TSURFSRC(MAT,PDD,V,TIME,Z,METHOD)` returns an array of temperature structs `T` at time instances, at z coordinates [m] `Z`, using the calculation method `METHOD`. `METHOD` is a string equalling `'mulitint'` (default), which slow(er) but accurate, or `'fft'` which is fast but inacurrate.

3.3.4 PLOTTEMP

`PLOTTEMP(TEMP)` plots a contour plot and a parametric mesh of the 3D temperature profile at the minimum z-corrdinate specified in the struct `TEMP`, as well as corresponding cross sections in the xz plane and the yz plane.

3.4 General functions

3.4.1 OVERLAP

`OVERLAP(D,F,V)` plots 10 circular laser pulses of diameter `D` [m], at pulse frequency `F` [Hz], at velocity `V` [m/s] of the laser beam relative to the substrate.

3.4.2 MATERIALS

`MATERIALS` defines structs with materials parameters and saves these workspace variable(s) to disk. The structs contain the following fields

```
name    (Indentifier, string)
A       (Absorptivity, ((vector of) double(s))
lambda  (Laser wavelength [m] for which A is defined, (vector of) double(s))
K       (Thermal conductivity [W/(m*K), double]
rho     (Density [kg/m^3], double)
Cp      (Thermal heat capcity [J/(kg*K)], double)
kappa   (Thermal diffusitivity [m^2/s], double)
T       (Temperature [K] for which the above parameters are defined, double)
Tm      (Melt temperature [K], double)
Tv      (Vaporization temperature [K], double)
```

3.4.3 MOMENTXY

`[MX,MY,VX,VY,VXY]=MOMENTSXY(X,Y,Ixy)` returns the first order moments `MX` [m] and `MY` [m] in x and y direction, of the matrix `Ixy` [W/m^2] holding values of power density distribution, at the coordinates specified by vectors X and Y, as well as the corresponding second order moments (variance) `VX` [m^2] and `VY` [m^2], and cross variance `VXY`.

This function is used by `ISO11146`.

3.4.4 NUM2PSTR

`T = NUM2PSTR(x)` converts the scalar `x` into a string representation `T` with about 4 digits and a prefix of the metric system. This is useful for labeling plots with the `TITLE`, `XLABEL`, `YLABEL`, and `TEXT` commands.

3.4.5 HERMITE

The n$^{\text{th}}$ degree Hermite polynomial $H^n(\cdot)$ reads

$$H^n(x) = \begin{cases} 1 & n = 0 \\ 2x & n = 1 \\ 2xH^{n-1}(x) - 2(n-1)H^{n-2}(x) & n > 1 \end{cases} \tag{22}$$

`Y=HERMITE(N,X)` returns the `N`th degree Hermite polynomial of `X`, where `N` is a non-negative integer and `X` a real scalar.

This function is used by `TEMMN`.

3.4.6 LAGUERRE

The Laguerre polynomial $L_l^n(\cdot)$ of order $n \geq 0$ and index $l \geq 0$ reads

$$L_l^n(x) = \begin{cases} 1 & n = 0 \\ 1 - x + l & n = 1 \\ \frac{1}{n}(2n + l - 1 - x)L_l^{n-1}(x) - \frac{1}{n}(n + l - 1)L_l^{n-2}(x) & n > 1 \end{cases} \tag{23}$$

`Y=LAGUERRE(N,L,X)` returns the generalized Laguerre polynomial of order `N` and index `L` of `X`, where `N` and `L` are a non-negative integers and is `X` a real scalar.

This function is used by `TEMPL`.

4 Known issues

1. Some scripts can/will be optimized for speed even more.