
Lamprop manual

Roland F. Smith

Eindhoven 2018

Contents

1	Introduction	3
2	Building and installing the program	4
2.1	Requirements	4
2.2	Installation	4
3	Using the program	5
3.1	The lamprop file format	5
3.2	Material data	6
3.3	Using the command-line front-end	6
3.4	Using the GUI front-end	7
3.5	Using the lamprop module from Python 3	7
	Colofon	9

1 Introduction

The purpose of this program is to calculate some properties of fiber-reinforced composite laminates. It calculates:

- engineering properties like E_x , E_y and G_{xy}
- thermal properties like α_x and α_y
- physical properties like density and laminate thickness
- stiffness and compliance matrices (ABD and abd)

Although these properties are not very difficult to calculate, (the relevant equations and formulas can be readily found in the available composite literature) the calculation is time-consuming and error-prone when done by hand.

This program can *not* calculate the strength of composite laminates; because there are many different failure modes, strengths of composite laminates cannot readily be calculated from the strengths of the separate materials that form the laminate. These strengths have to be determined from tests.

The original version of this program was written in C, since implementing it in a spreadsheet proved cumbersome, inflexible and even produced incorrect results. The C version ran up to 1.3.x.

As an exercise in learning the language, the author ported the program to the Python programming language. This proved to be a much cleaner, more maintainable and shorter implementation.

In the meantime, the program was ported from python version 2 to python version 3 and the core objects (in the `types.py` file) were made immutable. Also the output method was made generic to enable output in different formats, such as \LaTeX , HTML and RTF.

Additionally, the generally hard to obtain transverse fiber properties were replaced with properties derived from the matrix.

2 Building and installing the program

2.1 Requirements

The main requirements are `python` (version 3.4 or later) and the `numpy` library (version 1.6 or later). Currently the development is done using `python 3.7` and `numpy 1.15`.

For developers: You will need `py.test`¹ to run the provided tests. Code checks are done using `pylama`². Both should be invoked from the root directory of the repository.

There are basically two versions of this program; a console version (installed as `lamprop`) primarily meant for POSIX operating systems and a GUI version (installed as `lamprop-gui`) primarily meant for ms-windows.

You can try both versions without installing them first, with the following invocations in a shell from the root directory of the repository.

Use `python3 -m lamprop.console -h` for the console version, and `python3 -m lamprop.gui` for the GUI version.

Note that if `lamprop` is already installed, using the above commands will use the *installed* version of the `lamprop` module.

2.2 Installation

- Unpack the tarball or zipfile, or clone the github repository.
- Open a terminal window or (on ms-windows a `cmd` window).
- Change into the `lamprop` directory.
- Run `python3 setup.py install`. This will install both the module and the scripts that use it.

¹<https://docs.pytest.org/>

²<http://pylama.readthedocs.io/en/latest/>

3 Using the program

There are basically three ways to use lamprop.

1. Use the command-line front-end `lamprop`.
2. Use the GUI-frontend `lamprop-gui`.
3. Use the lamprop module directly from Python 3.

The first and second method depend on files written in a domain-specific language.

3.1 The lamprop file format

The file format is very simple. Functional lines have either `f`, `r`, `t`, `m`, `l` or `s` as the first non-whitespace character. This character must immediately be followed by a colon `:`. All other lines are seen as comments and disregarded.

This program assumes specific metric units. The units used below are important because the program internally calculates the thickness of layers (in mm) based on the volume fractions and densities of the fibers and resins.

The `f` : -line line contains a definition of a fiber. The parser converts this into an instance of a `Fiber` object. The line must contain the following values, separated by whitespace:

E_1 Young's modulus in the fiber direction in MPa.
 ν_{12} Poisson's constant (dimensionless).
 α_1 Coefficient of Thermal Expansion in the fiber direction in K^{-1} .
 ρ Density of the fiber in g/cm^3 .
name The identifier for the resin. This should be unique in all the files read. Contrary to the previous values, this may contain whitespace.

Usually, E_1 and other properties in the fibre length direction are easily obtained from a fiber supplier. Previous versions of this program also required some properties perpendicular to the fiber to calculate transverse properties of the lamina. Since these values are

very hard to obtain, they have been replaced by values derived from the matrix, according to Tsai(1992).

In the `tools` subdirectory of the source distribution you will find a script called `convert-lamprop.py` to convert old-style fiber lines to the new format.

The `r` : -line line contains a definition of a resin. Like with the fibers, this becomes an instance of a `Resin` object in the code. The resin line must contain the following values, separated by whitespace.

E Young's modulus in MPa.
 ν Poisson's constant (dimensionless).
 α Coefficient of Thermal Expansion in K^{-1} .
 ρ Density of the resin in g/cm^3 .
name The identifier for the resin. This should be unique in all the files read. Contrary to the previous values, this may contain whitespace.

The `t` : line starts a new laminate. It only contains the name which identifies the laminate. This name must be unique within the current input files. It may contain spaces.

The `m` : line chooses a resin for the laminate. It must appear after a `t` : line, and before the `l` : lines. It must contain the following values, separated by whitespace:

v_f The fiber volume fraction. This should be a number between 0 and 1 or between 1 up to and including 100. In the latter case it is interpreted as a percentage.
name The name of the resin to use. This must have been previously declared with an `r` : -line.

The `l` : line defines a single layer (lamina) in the laminate. It must be preceded by a `t` : and a `m` : line. It must contain the following values, separated by whitespace (optional items in brackets):

weight The area weight in g/m^2 of the dry fibers.
angle The angle upwards from the x-axis under which the fibers are oriented.
(vf) Optionally the layer can have a different fiber volume fraction.
name The name of the fiber used in this layer. This fiber must have been declared previously with an `f :` line.

The last line in a laminate definition can be an `s :` line, which stands for "symmetry". This signifies that all the lamina before it are to be added again in reverse order, making a symmetric laminate stack. An `s :` line in any other position is an error.

```
Fiber definition
E1      v12  alphas  rho  naam
f: 233000 0.2  -0.54e-6 1.76 Hyer's carbon fiber

Matrix definition
Em      v      alpha  rho name
r: 4620 0.36 41.4e-6 1.1  Hyer's resin

t: [0/90]s laminate
This is a standard symmetric cross-ply laminate. It has fine extensional
moduli in the fiber directions, but a very low shear modulus.
m: 0.5 Hyer's resin
l: 100 0 Hyer's carbon fiber
l: 100 90 Hyer's carbon fiber
s:
```

3.2 Material data

Over the years, the author has gathered a lot of data for different fibers from datasheets provided by the manufacturers. Data for different carbon fibers is given in Table 3.1. In case the ν_{12} is not known for a carbon fiber, it is estimated at 0.25. Similarly, if the α_1 is not known, it is estimated at $-0.12 \times 10^{-6} \text{ K}^{-1}$. For glass fibers, ν_{12} is estimated 0.33 unless known and α_1 is estimated $5 \times 10^{-6} \text{ K}^{-1}$ unless known.

Several resins are shown in Table 3.2. For resins, ν is estimated 0.36 unless known and α is estimated $40 \times 10^{-6} \text{ K}^{-1}$ unless known.

3.3 Using the command-line front-end

The command `lamprop -h` produces the following overview of the options.

```
usage: lamprop [-h] [-l | -H | -r] [-e | -m] [-L | -v]
              [--log {debug,info,warning,error}]
              [file [file ...]]

Calculate the elastic properties of a fibrous composite laminate.

positional arguments:
  file                one or more files to process

optional arguments:
  -h, --help          show this help message and exit
  -l, --latex          generate LaTeX output (the default is plain text)
  -H, --html          generate HTML output
```

Table 3.1: fibers

Name	E_1 [MPa]	ν_{12} [-]	α_1 [K ⁻¹]	ρ [g/cm ³]	Type
Tenax HTA	238000	0.25	-0.1e-6	1.76	carbon
Tenax HTS	240000	0.25	-0.1e-6	1.77	carbon
Tenax STS40	240000	0.25	-0.12e-6	1.78	carbon
Toracya T300	230000	0.27	-0.41e-6	1.76	carbon
Torayca T700SC	230000	0.27	-0.38e-6	1.80	carbon
pyrofil TR30S	235000	0.25	-0.5e-6	1.79	carbon
sigrafil CT24-5.0-270/E100	270000	0.25	-0.12e-6	1.79	carbon
K63712	640000	0.234	-1.47e-6	2.12	carbon
K63A12	790000	0.23	-1.2e-6	2.15	carbon
Torayca T800S	294000	0.27	-0.60e-6	1.76	carbon
K13C2U	900000	0.234	-1.47e-6	2.20	carbon
M35J	339000	0.27	-0.73e-6	1.75	carbon
M46J	436000	0.234	-0.9e-6	1.84	carbon
PX35UD	242000	0.27	-0.6e-6	1.81	carbon
Granoc XN-80-60S	780000	0.27	-1.5e-6	2.17	carbon
Granoc XN-90-60S	860000	0.27	-1.5e-6	2.19	carbon
e-glass	73000	0.33	5.3e-6	2.60	glass
ecr-glass	81000	0.33	5e-6	2.62	glass

Table 3.2: Resins

Name	E [MPa]	ν [-]	α [K ⁻¹]	ρ [g/cm ³]	Type
Epikote EPRO4908	2900	0.25	40e-6	1.15	epoxy
Palatal P4-01	4300	0.36	40e-6	1.19	polyester
Synolite 2155-N-1	4000	0.36	40e-6	1.22	polyester
Distitron 3501LS1	4100	0.36	40e-6	1.2	polyester
Synolite 1967-G-6	3800	0.36	40e-6	1.165	DCPD
atlac 430	3600	0.36	55e-6	1.145	vynilester

```

-r, --rtf           generate Rich Text Format output
-e, --eng           output only the layers and engineering properties
-m, --mat           output only the ABD and abd matrices
-L, --license       print the license
-v, --version       show program's version number and exit
--log {debug,info,warning,error}
                    logging level (defaults to 'warning')

```

3.4 Using the GUI front-end

The GUI front-end was written (using `tkinter`) primarily for users of ms-windows, since they are generally not used to the command-line interface. The contents of its window are shown in Figure 3.1. The image shows the looks of the widgets on UNIX-like operating systems. On ms-windows follow the native look.

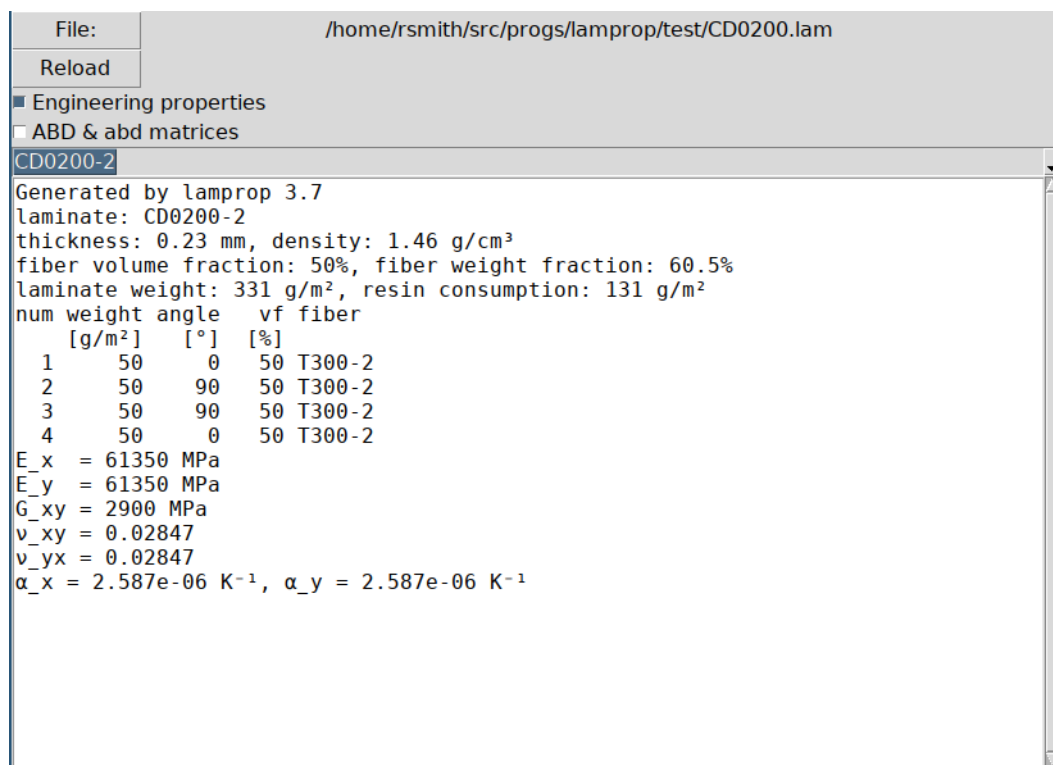


Figure 3.1: lamprop GUI

The **File** button allows you to load a lamprop file. If a file is loaded its name is shown right of the button. The **Reload** button re-loads a file. The checkboxes below determine which

results are shown. If a file contains different laminates, the dropdown allows you to select a laminate to display. The textbox at the bottom shows the lamprop output as text.

Pressing the `q`-key terminates the program.

3.5 Using the lamprop module from Python 3

An example reproducing the results from Figure 3.1 is shown below.

```
import lamprop as la
```

```
t300 = la.Fiber(230000, 0.3, -0.41e-6, 1.76, 'T300-2')
epro4908 = la.Resin(2900, 0.36, 41.4e-6, 1.15, 'Epikote 04908')
```

```
Lo = la.Lamina(t300, epro4908, 100, 0, 0.50)
L90 = la.Lamina(t300, epro4908, 100, 90, 0.50)
```

```
cdo200 = la.Laminate('CD0200-2', (Lo, L90, L90, Lo))
```

```
print(la.text.engprop(cdo200))
```

This is probably the most flexible way to use it, since you could use Python to generate large and complex laminates. Below, a symmetric and balanced quasi-isotropic laminate with layers every 15° is generated. There is no artificial limit to the amount of layers that can be defined. The author has used laminates with up to 250 layers. Calculating the properties of that laminate took approximately 0.5 s on a machine with an Intel Core2 Q9300 running FreeBSD.

```
import lamprop as la
```

```
t300 = la.Fiber(230000, 0.3, -0.41e-6, 1.76, 'T300-2')
epro4908 = la.Resin(2900, 0.36, 41.4e-6, 1.15, 'Epikote 04908')
```

```
layers = [la.Lamina(t300, epro4908, 100, a, 0.50) for a in range(-90, 95, 15)]
layers += layers[::-1]
```

```
qi = la.Laminate('quasi-isotropic', layers)
```

```
print(la.latex.out(qi, eng=True, mat=False))
print(la.latex.out(qi, eng=False, mat=True))
```

The resulting `LaTeX` code basically produces Table 3.3 and Table 3.4; the content was separated into two tables to fit on the page.

Table 3.3: Layers and engineering properties of quasi-isotropic

calculated by lamprop 3.7

Laminate stacking					Engineering properties		
Layer	Weight [g/m ²]	Angle [°]	vf [%]	Fiber type	Property	Value	Dimension
1	100	-90	50	T300-2	v_f	50	%
2	100	-75	50	T300-2	w_f	60.5	%
3	100	-60	50	T300-2	thickness	2.95	mm
4	100	-45	50	T300-2	density	1.46	g/cm ³
5	100	-30	50	T300-2	weight	4299	g/m ²
6	100	-15	50	T300-2	resin	1699	g/m ²
7	100	0	50	T300-2	E_x	40924	MPa
8	100	15	50	T300-2	E_y	48752	MPa
9	100	30	50	T300-2	G_{xy}	15327	MPa
10	100	45	50	T300-2	ν_{xy}	0.266215	-
11	100	60	50	T300-2	ν_{yx}	0.317136	-
12	100	75	50	T300-2	α_x	3.11462e-06	K ⁻¹
13	100	90	50	T300-2	α_y	2.12576e-06	K ⁻¹
14	100	90	50	T300-2			
15	100	75	50	T300-2			
16	100	60	50	T300-2			
17	100	45	50	T300-2			
18	100	30	50	T300-2			
19	100	15	50	T300-2			
20	100	0	50	T300-2			
21	100	-15	50	T300-2			
22	100	-30	50	T300-2			
23	100	-45	50	T300-2			
24	100	-60	50	T300-2			
25	100	-75	50	T300-2			
26	100	-90	50	T300-2			

Table 3.4: Matrices of quasi-isotropic

calculated by lamprop 3.7

$$\begin{aligned}
 &\text{Stiffness (ABD) matrix} \\
 &\begin{pmatrix} N_x \\ N_y \\ N_{xy} \\ M_x \\ M_y \\ M_{xy} \end{pmatrix} = \begin{pmatrix} 1.3206 \times 10^5 & 4.1882 \times 10^4 & 0 & 0 & 0 & 0 \\ 4.1882 \times 10^4 & 1.5732 \times 10^5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 4.5286 \times 10^4 & 0 & 0 & 0 \\ 0 & 0 & 0 & 7.9688 \times 10^4 & 2.8649 \times 10^4 & -1.8401 \times 10^4 \\ 0 & 0 & 0 & 2.8649 \times 10^4 & 1.3446 \times 10^5 & -2.9077 \times 10^4 \\ 0 & 0 & 0 & -1.8401 \times 10^4 & -2.9077 \times 10^4 & 3.1125 \times 10^4 \end{pmatrix} \times \begin{pmatrix} \epsilon_x \\ \epsilon_y \\ \gamma_{xy} \\ \kappa_x \\ \kappa_y \\ \kappa_{xy} \end{pmatrix} \\
 &\text{Compliance (abd) matrix} \\
 &\begin{pmatrix} \epsilon_x \\ \epsilon_y \\ \gamma_{xy} \\ \kappa_x \\ \kappa_y \\ \kappa_{xy} \end{pmatrix} = \begin{pmatrix} 8.2705 \times 10^{-6} & -2.2017 \times 10^{-6} & 0 & 0 & 0 & 0 \\ -2.2017 \times 10^{-6} & 6.9425 \times 10^{-6} & 0 & 0 & 0 & 0 \\ 0 & 0 & 2.2082 \times 10^{-5} & 0 & 0 & 0 \\ 0 & 0 & 0 & 1.4796 \times 10^{-5} & -1.5802 \times 10^{-6} & 7.2712 \times 10^{-6} \\ 0 & 0 & 0 & -1.5802 \times 10^{-6} & 9.4889 \times 10^{-6} & 7.9304 \times 10^{-6} \\ 0 & 0 & 0 & 7.2712 \times 10^{-6} & 7.9304 \times 10^{-6} & 4.3835 \times 10^{-5} \end{pmatrix} \times \begin{pmatrix} N_x \\ N_y \\ N_{xy} \\ M_x \\ M_y \\ M_{xy} \end{pmatrix} \\
 &ABD = \begin{pmatrix} A_{11} & A_{12} & A_{16} & B_{11} & B_{12} & B_{16} \\ A_{12} & A_{22} & A_{26} & B_{12} & B_{22} & B_{26} \\ A_{16} & A_{26} & A_{66} & B_{16} & B_{26} & B_{66} \\ B_{11} & B_{12} & B_{16} & D_{11} & D_{12} & D_{16} \\ B_{12} & B_{22} & B_{26} & D_{12} & D_{22} & D_{26} \\ B_{16} & B_{26} & B_{66} & D_{16} & D_{26} & D_{66} \end{pmatrix} \quad abd = \begin{pmatrix} a_{11} & a_{12} & a_{16} & b_{11} & b_{12} & b_{16} \\ a_{12} & a_{22} & a_{26} & b_{12} & b_{22} & b_{26} \\ a_{16} & a_{26} & a_{66} & b_{16} & b_{26} & b_{66} \\ b_{11} & b_{12} & b_{16} & d_{11} & d_{12} & d_{16} \\ b_{12} & b_{22} & b_{26} & d_{12} & d_{22} & d_{26} \\ b_{16} & b_{26} & b_{66} & d_{16} & d_{26} & d_{66} \end{pmatrix}
 \end{aligned}$$

The units of the parts of the ABD and abd matrix are as follows (where i and j are 1, 2 or 6): A_{ij} is in N/mm. B_{ij} is in N. D_{ij} is in N mm. a_{ij} is in mm/N. b_{ij} is in 1/N. d_{ij} is in 1/Nmm.

The stress resultants N are units of force per unit of length (N/mm). Moment resultants m are in units of torque per unit of length (Nmm/mm = N). Both strains ϵ and κ are dimensionless.

The stiffness or ABD matrix and compliance or abd matrix are shown with the relevant strains and forces in Table 3.4. Both are 6×6 matrices that can be divided into three 3×3 matrices; A, B and D or a, b and d.

The expansions below reveal the symmetries in these matrices.

Colofon

This document has been typeset with the \TeX ¹ software, using the \LaTeX ² macros and specifically the `MEMOIR`³ style.

¹<http://nl.wikipedia.org/wiki/TeX>

²<http://nl.wikipedia.org/wiki/LaTeX>

³<http://www.ctan.org/tex-archive/macros/latex/contrib/memoir/>