Lamprop manual

Roland F. Smith

Eindhoven 2020

Contents

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_x
- 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 really have to be determined from tests. However, the author has found Hart-Smith [1992] useful for initial estimation of the strengths of multi-layer laminates.

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 were replaced by functions. (now in `core.py`) Also the output method was made generic to enable output in different formats, such as LaTeXand HTML.

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

The books from Hyer [1998] and Tsai [1992] and the report from Nettles [1994] were instrumental in writing the code.

All the important code is covered by tests using pytest, and pylama is used to check the code when it is saved.

2 Building and installing the program

2.1 Requirements

The only requirement is Python (version 3.6 or later). Currently the development is done using Python 3.7.

For developers: You will need pytest[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 console.py -h` for the console version, and `python3 gui.py` for the GUI version.

2.2 Installation

First, you need to install Python 3. For UNIX-like operating systems, use the packages or build scripts that your operating system provides.

There are Python binaries for ms-windows available from python.org[3], and those should work fine for lamprop.

Once the requirements are met, you can proceed with the installation of lamprop.

- Unpack the tarball or zipfile, or clone the github repository.
- Open a terminal window or (on ms-windows a `cmd` window).
- Change into the directory where lamprop was unpacked or cloned.
- Run `python3 build.py`. This will create both self-contained scripts. If you get an error message that the command `python3` is not recognized, try `python` instead.
- On POSIX operating systems, run `make install` as root to install the programs. By default they are installed in `\usr\local\bin`.
- On ms-windows, copy `lamprop.pyz` and `lamprop-gui.pyz` to the `Scripts` subdirectory of your Python installation.

[1] https://docs.pytest.org/
[2] http://pylama.readthedocs.io/en/latest/
[3] https://www.python.org/downloads/

3 Using the program

There are basically two ways to use lamprop.

1. Use the command-line program `lamprop`.
2. Use the GUI-program `lamprop-gui`.

They 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 white space:

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 white space.

Below an example for standard e-glass fiber.
`73000 0.33 5.3e-6 2.60 e-glass`

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 the Young's modulus perpendicular to the fiber to calculate transverse properties of the lamina. Since this value is generally not given in the manufacturer documentation, it has been replaced by the modulus of the matrix multiplied by a factor, according to Tsai [1992]. However, the author has found that the factor provided by Tsai overestimates E_2, especially when using glass fibers. So from lamprop version 3.6 onwards, this factor has been reduced. All users of previous versions are encouraged to upgrade.

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 white space.

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.

An example of a generic thermoset resin is shown below.
`3800 0.36 40e-6 1.165 generic`

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 white space:

vf 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 als 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 white space (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.

An example is given below.

```
Fiber definition
   E1     v12  alpha1   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:
```

There is no artificial limit to the amount of layers that you can use other than Python running out of memory. 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.

3.2 Material data

Over the years, the author has gathered a lot of data for different fibers from data sheets provided by the manufacturers. Data for different 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 -0.12×10^{-6} K^{-1}. For glass fibers, ν_{12} is estimated 0.33 unless known and α_1 is estimated 5×10^{-6} K^{-1} unless known.

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

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

Table 3.2: Resins

Name	E [MPa]	ν [-]	α [K^{-1}]	ρ [g/cm^3]	Type
Epikote EPR04908	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	vinylester

3.3 Using the command-line program

The command `lamrop -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
  -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')
```

Running `lamprop` from the command line produces text output by default. Output in LaTeX or HTML format can be requested with the appropriate arguments. As of 4.0, RTF output (for inclusion in word processor documents) has been removed. Since most word processors can read HTML, use that instead.

3.4 Using the GUI program

The GUI program 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.

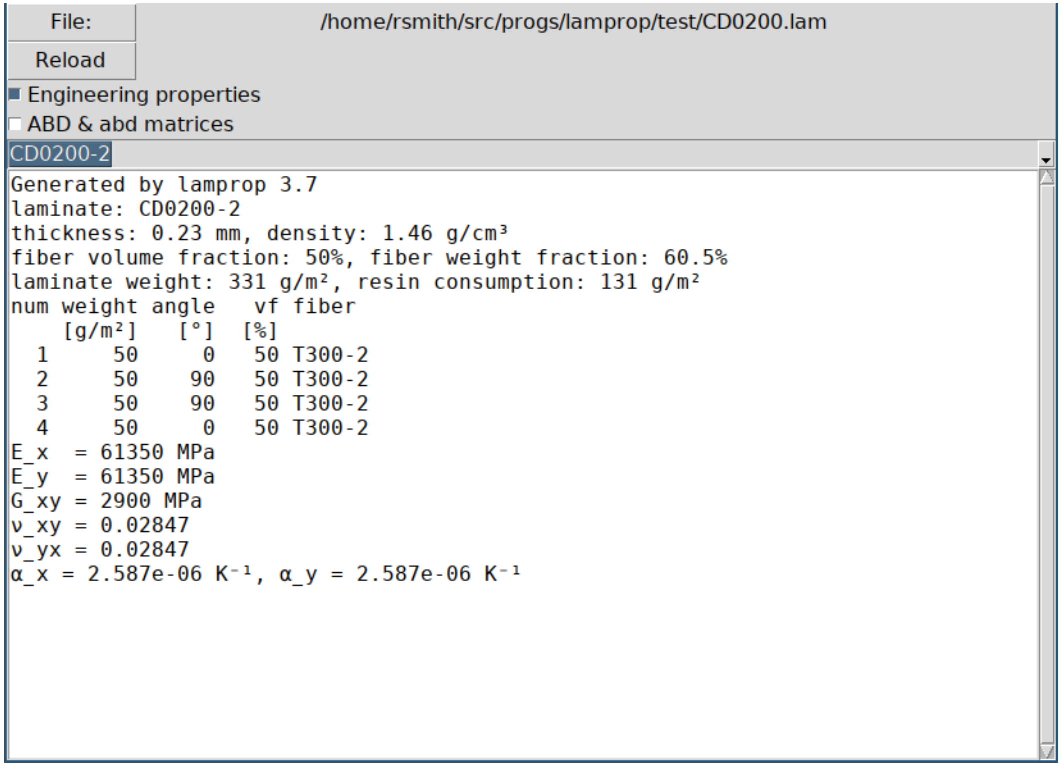

Figure 3.1: lamprop GUI

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 dropbox 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 Meaning of the ABD and abd matrices

The stiffness or ABD matrix and compliance or abd matrix are what converts strains into forces and the other way around, see Table 3.3. 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.

$$ABD = \begin{vmatrix} A_{11} & A_{12} & A_{16} & B_{11} & B_{12} & B_{16} \\ A_{12} & A_{22} & A_{26} & B_{12} & B_{22} & B_{26} \\ A_{16} & A_{26} & A_{66} & B_{16} & B_{26} & B_{66} \\ B_{11} & B_{12} & B_{16} & D_{11} & D_{12} & D_{16} \\ B_{12} & B_{22} & B_{26} & D_{12} & D_{22} & D_{26} \\ B_{16} & B_{26} & B_{66} & D_{16} & D_{26} & D_{66} \end{vmatrix} \qquad abd = \begin{vmatrix} a_{11} & a_{12} & a_{16} & b_{11} & b_{12} & b_{16} \\ a_{12} & a_{22} & a_{26} & b_{12} & b_{22} & b_{26} \\ a_{16} & a_{26} & a_{66} & b_{16} & b_{26} & b_{66} \\ b_{11} & b_{12} & b_{16} & d_{11} & d_{12} & d_{16} \\ b_{12} & b_{22} & b_{26} & d_{12} & d_{22} & d_{26} \\ b_{16} & b_{26} & b_{66} & d_{16} & d_{26} & d_{66} \end{vmatrix}$$

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 Nmm/mm = 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. This clearly shows that abd is the inverse of ABD.

The stress resultants N are units of force per unit of width (N/mm). Moment resultants

m are in units of torque per unit of width (Nmm/mm = N). Both strains ϵ and κ are dimensionless.

The matrix equations in Table 3.3 basically show the behavior of a square piece of laminate small enough that the stress and strain resultants can be considered constant over its dimensions.

Table 3.3: Matrices of quasi-isotropic

calculated by lamprop 3.7

Stiffness (ABD) matrix

$$\left\{\begin{matrix} N_x \\ N_y \\ N_{xy} \\ M_x \\ M_y \\ M_{xy} \end{matrix}\right\} = \left|\begin{matrix} 1.3206\times10^{5} & 4.1882\times10^{4} & 0 & 0 & 0 & 0 \\ 4.1882\times10^{4} & 1.5732\times10^{5} & 0 & 0 & 0 & 0 \\ 0 & 0 & 4.5286\times10^{4} & 0 & 0 & 0 \\ 0 & 0 & 0 & 7.9688\times10^{4} & 2.8649\times10^{4} & -1.8401\times10^{4} \\ 0 & 0 & 0 & 2.8649\times10^{4} & 1.3446\times10^{5} & -2.9077\times10^{4} \\ 0 & 0 & 0 & -1.8401\times10^{4} & -2.9077\times10^{4} & 3.1125\times10^{4} \end{matrix}\right| \times \left\{\begin{matrix} \epsilon_x \\ \epsilon_y \\ \gamma_{xy} \\ \kappa_x \\ \kappa_y \\ \kappa_{xy} \end{matrix}\right\}$$

Compliance (abd) matrix

$$\left\{\begin{matrix} \epsilon_x \\ \epsilon_y \\ \gamma_{xy} \\ \kappa_x \\ \kappa_y \\ \kappa_{xy} \end{matrix}\right\} = \left|\begin{matrix} 8.2705\times10^{-6} & -2.2017\times10^{-6} & 0 & 0 & 0 & 0 \\ -2.2017\times10^{-6} & 6.9425\times10^{-6} & 0 & 0 & 0 & 0 \\ 0 & 0 & 2.2082\times10^{-5} & 0 & 0 & 0 \\ 0 & 0 & 0 & 1.4796\times10^{-5} & -1.5802\times10^{-6} & 7.2712\times10^{-6} \\ 0 & 0 & 0 & -1.5802\times10^{-6} & 9.4889\times10^{-6} & 7.9304\times10^{-6} \\ 0 & 0 & 0 & 7.2712\times10^{-6} & 7.9304\times10^{-6} & 4.3835\times10^{-5} \end{matrix}\right| \times \left\{\begin{matrix} N_x \\ N_y \\ N_{xy} \\ M_x \\ M_y \\ M_{xy} \end{matrix}\right\}$$

4 Tips and tricks

The 0° direction is generally in the length of the part or in the direction of the largest load.

The following section should be considered a *general guideline*. Sometimes there can be good reason to deviate from it.

4.1 Keep your laminates symmetric and balanced

Looking at the stacking of the layers, it should be symmetric w.r.t. the middle of the stack. So the following laminate is symmetric:

1. 0°
2. 45°
3. 90°
4. -45°
5. -45°
6. 90°
7. 45°
8. 0°

This is often shortened to "[0/45/90/-45]s". The area weights of the layers should also be symmetric.

A balanced laminate is a laminate where for every layer at an angle on *n*° there is also a layer at -*n*°. It is often added that for every 0° layer there should also be an equally sized 90° layer, but the author disagrees. For beam-like parts it is often desirable to have the majority of the fibers in the 0° direction.

Classical laminate theory strictly speaking is only valid for stackings of unidirectional layers. For woven fabrics and random oriented fiber products approximations are used.

4.2 Representing woven fabrics

A woven fabric is approximated as a [0°/90°]s stack, where the weight of each layer is 1/4 of the total weight of the woven fabric. If warp and weft of the weave are not of equal weight, you should adjust the layers accordingly. Symmetry is important because a lone [0°/90°] would exhibit tension/bending coupling that doesn't occur in a woven fabric. If the woven fabric is a small part of a larger stacking, you can use [0°/90°] to represent a weave.

4.3 Representing non-wovens

Things like chopped strand mat ("CSM"), continuous filament mat ("CFM") or other non-wovens can be approximated as a [0°/60°/-60°]s stack with the area weight evenly divided over the directions. Do keep in mind that the fiber volume fraction for such materials is significantly lower than for unidirectional or woven materials. For CSM it is unlikely to exceed 25% and for CFM 10–15% are typical values.

4.4 Align your fibers with the expected load

This is a no-brainer for tensile loads, but there is a twist. To counter torsion and shear loads, there should be layers of fibers in the ±45° direction. For bending loads the 0° layers should be at the outside of the part.

4.5 Laminate strength

As mentioned before, this program cannot predict the strength of laminates from the properties of the fibers and resin used in the layers; it is outside the scope of classical laminate theory.

Even stronger, the author does not believe that a general theory of laminate strength based on constituent properties is feasible due to the many different possible failure modes and the factors outside of the fiber and resin properties that influence the laminate. Examples of the latter are the void content, the degree of cure of the resin and errors in cutting or placing the fibers. These are determined by type of production process used and the craftsmanship of the people involved.

However, the following guidelines have served the author well over the years.

For unidirectional layers loaded in the fiber direction, the strain at which either the fibers or the matrix fail in tension multiplied by the laminate's Young's modulus is the maximum allowed tensile stress.

The allowed compression stress for such layers is deemed to be 1/2 of the allowed tensile stress

The strength of unidirectional layers in the ±45° or 90° directions is estimated as 10% of the strength in the 0° direction. This is the 10%-rule according to Hart-Smith [1992].

Bibliography

L. J. Hart-Smith. The ten-percent rule for preliminary sizing of fibrous composite structures. *Weight Engineering*, 52:29–45, 1992.

Micheal W. Hyer. *Stress analysis of fiber-reinforced composite materials*. McGraw–Hill, 1998. ISBN 0 07 115983 5.

A.T. Nettles. Basic mechanics of laminated plates. Technical Report Reference Publication 1351, NASA, 1994.

Stephen W. Tsai. *Theory of composites design*. Think Composites, 1992. ISBN 0 9618090 3 5.

Colofon

This document has been set with the "TeX Live"[1] implementation of the TeX[2] typesetting software, using the LaTeX[3] macros and specifically the MEMOIR[4] style.

The main font used for the text is Alegreya[5]. The TeX Gyre Heros[6] font is used for sans-serif text, while `TeX Gyre Cursor` is used for program names and program output.

[1] https://www.tug.org/texlive/
[2] http://nl.wikipedia.org/wiki/TeX
[3] http://nl.wikipedia.org/wiki/LaTeX
[4] http://www.ctan.org/tex-archive/macros/latex/contrib/memoir/
[5] https://github.com/huertatipografica/Alegreya
[6] http://www.gust.org.pl/projects/e-foundry/tex-gyre