Worked Example for Micromechanics and CLPT

SESG6039

Prof Stephen Boyd

In [1]:
```
# import the CLPT functions so that they can be executed
from CLPT import *
from sympy import *
import numpy as np
init_printing(use_latex=True)
```

Introduction

This is a worked example to be used as part of assignment 1 for the verification of your CLPT code. The composite laminate is based on carbon fibres and an epoxy resin. The properties of which are provided in Table 1. The fibre weight fraction, W_f is 0.6 and the layup of the laminate is $[0, 90, 90, 0]$ or sometimes presented as $[0, 90]_s$ where each layer is 1mm thick.

Table 1 Constituent material properties

Property	Symbol	Value	units
Fibre Young's modulus	Ef	310	GN/m^2
Fibre density	Rhof	1870	kg/m^3
Fibre Poisson's ratio	vf	0.23	-
Matrix Young's modulus	Em	2.4	GN/m^2
Matrix density	Rhom	1870	kg/m^3
Matrix Poisson's ratio	vm	0.23	-

Step 1 - Define the Fibre and Matrix properties and calculate lamina properties

For this example the weight fraction of the fibres is provided, so we need to convert to Voume fraction first

In [2]:
```
Wf = 0.6
Rhof = 1870 # kg/m^3
Rhom = 1200 # kg/m^3
Vf = Weight2Volfrac(Wf, Rhof, Rhom)
Vm = 1- Vf
print(f'The fibre volume fraction, Vf, of the lamina is {round(Vf,3)} and therefore the 
```

```
The fibre volume fraction, Vf, of the lamina is 0.49 and therefore the matrix volume fra
ction, Vm, is 0.51
```

The properties of the individual lamina can be calculated in two ways, the Rule of Mixtures or the Halpin-Tsai equations.

Rule of Mixtures

$$E_{11} = E_f V_f + E_m V_m$$

$$E_{22} = \frac{E_f E_m}{E_f V_m + E_m V_f}$$

$$v_{12} = V_f v_f + V_m v_m$$

$$G = \frac{E}{2(1+v)} \therefore G_f = \frac{E_f}{2(1+v_f)} \; and \; G_m = \frac{E_m}{2(1+v_m)}$$

$$G_{12} = \frac{G_f G_m}{G_f V_m + G + m V_f}$$

Halpin-Tsai

$$E_{11} = E_f V_f + E_m V_m$$

$$E_{22} = E_m \left(\frac{1+\xi\eta V_f}{1-\eta V_f}\right) \; where \; \xi = 0.2 \; and \; \eta = \frac{\frac{E_f}{E_m}-1}{\frac{E_f}{E_m}+\xi}$$

$$v_{12} = V_f v_f + V_m v_m$$

$$G = \frac{E}{2(1+v)} \therefore G_f = \frac{E_f}{2(1+v_f)} \; and \; G_m = \frac{E_m}{2(1+v_m)}$$

$$G_{12} = G_m \left(\frac{1+\xi\eta V_f}{1-\eta V_f}\right) \; where \; \xi = 1.0 \; and \; \eta = \frac{\frac{G_f}{G_m}-1}{\frac{G_f}{G_m}+\xi}$$

In [3]:

```
Ef = 310.0e9 # N/m^2
vf = 0.23
Vf = 0.49
Em = 2.4e9 #N/m^2
vm = 0.33
Vm = 1-Vf
Gf = -1 # Value makes the calculation revert to the classic calculation
Gm = -1
props = [Ef, vf, Em, vm, Vf, Vm, Gf, Gm]
E11, E22, v12, v21, G12 = Calc_lamina_props(props, 1)
print_lam_props(props)
```

```
----- Halpin-Tsai  -----------            ------- Rule of Mixtures ---------
E11 = 153124000000.0 N/m^2                E11 = 153124000000.0 N/m^2
E22 = 5117175692.932 N/m^2                E22 = 4671136894.447 N/m^2
v12 = 0.281                               v12 = 0.281
v21 = 0.009                               v21 = 0.009
G12 = 2588319435.399 N/m^2                G12 = 1757041912.251 N/m^2
------------------------------            -------------------------------
```

For the remainder of this worked example the **Halpin-Tsai** values are used.

Step 2 - Define the Q matrix for the lamina

As per the notes we can obtain the Q matrix for the lamina in one of two ways

Method 1 - Construct the compliance matrix $\left[\,S\,\right]$ and invert

Using the relationship $\epsilon = [\,S\,]\,\sigma$, where

$$[\,S\,] = \begin{bmatrix} S_{11} & S_{12} & 0 \\ S_{12} & S_{22} & 0 \\ 0 & 0 & S_{66} \end{bmatrix}$$

where

$$S_{11} = \frac{1}{E_{11}},\ S_{12} = \frac{v_{12}}{E_{11}} = \frac{v_{21}}{E_{22}},\ S_{22} = \frac{1}{E_{22}}, S_{66} = \frac{1}{G_{12}}$$

To get stress as a function of strain

$$\sigma = [\,S\,]^{-1}\epsilon \qquad or \qquad \sigma = [\,Q\,]\,\epsilon \qquad as \qquad [\,Q\,] = [\,S\,]^{-1}$$

Method 2 - Construct the $[\,Q\,]$ matrix directly

$$[\,Q\,] = \begin{bmatrix} Q_{11} & Q_{12} & 0 \\ Q_{12} & Q_{22} & 0 \\ 0 & 0 & Q_{66} \end{bmatrix}$$

where

$$Q_{11} = \frac{E_{11}}{1-v_{12}v_{21}},\ Q_{12} = \frac{v_{21}E_{11}}{1-v_{12}v_{21}}, Q_{22} = \frac{E_{22}}{1-v_{12}v_{21}},\ Q_{66} = G_{12}$$

where v_{21} can be obtained from the equations for S_{12}

In [4]:
```
Q = Q_matrix(E11, E22, v12, v21, G12)
print('Q (GN/m^2)')
Matrix(np.round(Q/1e9,3))
```

```
Q (GN/m^2)
```

Out[4]:

$$\begin{bmatrix} 153.529 & 1.442 & 0 \\ 1.442 & 5.131 & 0 \\ 0 & 0 & 2.588 \end{bmatrix}$$

In order to have properties in any fibre direction we need to transform the $[\,Q\,]$ matrix.

$$\sigma_p = [\,T\,]\,\sigma_c$$

Where $[\,T\,]$ is the transfomation matrix, σ_p is the stress in the principal fibre direction and σ_c is the composite laminate direction.

$$[\,T\,] = \begin{bmatrix} \cos^2\theta & \sin^2\theta & 2\sin\theta\cos\theta \\ \sin^2\theta & \cos^2\theta & -2\sin\theta\cos\theta \\ -\sin\theta\cos\theta & \sin\theta\cos\theta & \cos^2\theta - \sin^2\theta \end{bmatrix}$$

If $m = \cos\theta$ and $n = \sin\theta$ then:

$$[\,T\,] = \begin{bmatrix} m^2 & n^2 & 2mn \\ n^2 & m^2 & -2mn \\ -nm & nm & m^2 - n^2 \end{bmatrix}$$

Expanding the stress transformation equation above

$$\begin{bmatrix}\sigma_1\\\sigma_2\\\tau_{12}\end{bmatrix} = [T]\begin{bmatrix}\sigma_x\\\sigma_y\\\tau_{xy}\end{bmatrix} \Rightarrow \begin{bmatrix}\sigma_x\\\sigma_y\\\tau_{xy}\end{bmatrix} = [T]^{-1}\begin{bmatrix}\sigma_1\\\sigma_2\\\tau_{12}\end{bmatrix}$$

The stress in the principal fibre direction σ_p is given by: $\sigma_p = [Q]\epsilon_p$ and the stress in the composite material direction σ_c is given by: $\sigma_c = [T]^{-1}\sigma_p$ giving the following relationship:

$$\sigma_c = [T]^{-1}[Q]\epsilon_p$$

But this has principal stress on its right hand side.

Going from tensoral strain to engineering strain the transformation equation doesn't have the factor 2 in the shear terms. So we account for this by introducing an additional martix $[R]$

$$[R] = \begin{bmatrix}1 & 0 & 0\\0 & 1 & 0\\0 & 0 & 2\end{bmatrix}$$

so

$$\begin{bmatrix}\epsilon_1\\\epsilon_2\\\frac{\gamma_{12}}{2}\end{bmatrix} = [T]\begin{bmatrix}\epsilon_x\\\epsilon_y\\\frac{\gamma_{xy}}{2}\end{bmatrix}$$

multiplying both sides by $[R]$ gives

$$\begin{bmatrix}\epsilon_1\\\epsilon_2\\\gamma_{12}\end{bmatrix} = [R][T]\begin{bmatrix}\epsilon_x\\\epsilon_y\\\frac{\gamma_{xy}}{2}\end{bmatrix}$$

multiplying both sides by the inverse of $[R]$ gives

$$\begin{bmatrix}\epsilon_1\\\epsilon_2\\\gamma_{12}\end{bmatrix} = [R][T][R]^{-1}\begin{bmatrix}\epsilon_x\\\epsilon_y\\\gamma_{xy}\end{bmatrix}$$

The term $[R][T][R]^{-1}$ is equivalent to the inverse of the transformation matrix transposed, $[T]^{-1^T}$, and so

$[\bar{Q}] = [T]^{-1}[Q][T]^{-1^T}$$

This gives:

$$\begin{bmatrix}\sigma_x\\\sigma_y\\\tau_{xy}\end{bmatrix} = [\bar{Q}]\begin{bmatrix}\epsilon_x\\\epsilon_y\\\gamma_{xy}\end{bmatrix}$$

Given the compliance matrix Q, we can calculate the matrix for any given ply angle.

Step 3 - Define the laminate architecture

For this example the laminate has 4 layers, of equal thickness, with a layup of $[0, 90, 90, 0]$ or sometimes presented as $[0, 90]_s$ where each layer is 1mm thick.

We therefore need to create two $[\bar{Q}]$ matrices, $[\bar{Q}]_0$ and $[\bar{Q}]_{90}$

In [5]:
```
print('Qbar_0 (GN/m^2)')
Matrix(np.round(Qbar_matrix(Q,0)/1e9,3))
```

```
Qbar_0 (GN/m^2)
```

Out[5]:

$$\begin{bmatrix} 153.529 & 1.442 & 0 \\ 1.442 & 5.131 & 0 \\ 0 & 0 & 2.588 \end{bmatrix}$$

In [6]:
```
print('Qbar_90 (GN/m^2)')
Matrix(np.round(Qbar_matrix(Q,90)/1e9,3))
```

```
Qbar_90 (GN/m^2)
```

Out[6]:

$$\begin{bmatrix} 5.131 & 1.442 & 0 \\ 1.442 & 153.529 & 0 \\ 0 & 0 & 2.588 \end{bmatrix}$$

Working through the laminate layer by layer, there are 5 layer surfaces, each surface is defined by it's distance from the laminate mid-plane, giving the lamina thickness $Z_k - Z_{k-1}$

$$A_{ij} = \sum_{k=1}^{N} (Q_{ij})_k (Z_k - Z_{k-1})$$

$$B_{ij} = \frac{1}{2} \sum_{k=1}^{N} (Q_{ij})_k ({Z_k}^2 - {Z_{k-1}}^2)$$

$$D_{ij} = \frac{1}{3} \sum_{k=1}^{N} (Q_{ij})_k ({Z_k}^3 - {Z_{k-1}}^3)$$

As a worked example the $[A]$ matrix is constructed by:

$$A_1 = [\bar{Q}_0] \left(-0.001 - (-0.002)\right)$$
$$A_2 = [\bar{Q}_{90}] \left(0 - (-0.001)\right)$$
$$A_3 = [\bar{Q}_{90}] \left(0.001 - (0)\right)$$
$$A_4 = [\bar{Q}_0] \left(0.002 - (0.001)\right)$$

A similar approach is employed for the $[B]$ and $[R]$ matrices as shown in the results presented below.

In [7]:
```
nLayers = 4 # Number of Layers
Tlayers = 1 # Thickness per layer
LayerT = [Tlayers for i in range(nLayers)] # List of layer thicknesses
stack = [0, 90, 90, 0] # Lamina angles
A, B, D = A_B_D(Q, nLayers, LayerT, stack) # Construct the A, B and D matrices
print('A (N/m)')
Matrix(A)
```

```
A (N/m)
```

Out[7]:

$$\begin{bmatrix} 317319681.503175 & 5766922.99794364 & 1.82185219030555 \cdot 10^{-10} \\ 5766922.99794364 & 317319681.503175 & 1.79913787977858 \cdot 10^{-8} \\ 1.82185219030555 \cdot 10^{-10} & 1.79913787977858 \cdot 10^{-8} & 10353277.7415943 \end{bmatrix}$$

In [8]:
```
print('B (N)')
Matrix(B)
```

B (N)

Out[8]:

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 4.54747350886464 \cdot 10^{-12} & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

In [9]:
```
print('D (Nm)')
Matrix(D)
```

D (Nm)

Out[9]:

$$\begin{bmatrix} 719.889732530147 & 7.68923066392485 & 6.07284063435182 \cdot 10^{-17} \\ 7.68923066392485 & 126.296084811653 & 5.99712626592862 \cdot 10^{-15} \\ 6.07284063435182 \cdot 10^{-17} & 5.99712626592862 \cdot 10^{-15} & 13.8043703221257 \end{bmatrix}$$

Step 4 - Create the 9x9 ABD matrix

Combine the three matricies into the format: $\begin{bmatrix} A & B \\ B & D \end{bmatrix}$ a 6x6 matrix

$$\left[\begin{array}{ccc:ccc} A_{11} & A_{12} & A_{16} & B_{11} & B_{12} & B_{16} \\ A_{12} & A_{22} & A_{26} & B_{12} & B_{22} & B_{26} \\ A_{16} & A_{26} & A_{66} & B_{16} & B_{26} & B_{66} \\ \hdashline B_{11} & B_{12} & B_{16} & D_{11} & D_{12} & D_{16} \\ B_{12} & B_{22} & B_{26} & D_{12} & D_{22} & D_{26} \\ B_{16} & B_{26} & B_{66} & D_{16} & D_{26} & D_{66} \end{array}\right]$$

In [10]:
```
ABD = ABD_matrix(A,B,D)
print('ABD')
Matrix(np.around(ABD, 3))
```

ABD

Out[10]:

$$\begin{bmatrix} 317319681.503 & 5766922.998 & 0 & 0 & 0 & 0 \\ 5766922.998 & 317319681.503 & 0 & 0 & 0 & 0 \\ 0 & 0 & 10353277.742 & 0 & 0 & 0 \\ 0 & 0 & 0 & 719.89 & 7.689 & 0 \\ 0 & 0 & 0 & 7.689 & 126.296 & 0 \\ 0 & 0 & 0 & 0 & 0 & 13.804 \end{bmatrix}$$

We know that stress is given by: $\sigma_x = \frac{N_x}{A}$, where A is the cross sectional area

As we are dealing here with unit width the equation for stress becomes $\sigma_x = \frac{N_x}{h}$, where h is the laminate stack thickness.

$$\sigma_x = \frac{A_{11}\epsilon_x + A_{12}\epsilon_y + A_{16}\gamma_{xy} + B_{11}\kappa_x + B_{12}\kappa_y + B_{16}\kappa_{xy}}{h}$$

This allows one to calculate the Young's modulus which is given by:

$$E_x = \frac{\sigma_x}{\epsilon_x}$$

Inverting the $\begin{bmatrix} A & B \\ B & D \end{bmatrix}$ matrix allows one to solve for the strains and curvatures.

Where a value of N_x is provided, ϵ_x is obtained from:

$$\begin{bmatrix} \epsilon_x \\ \epsilon_y \\ \gamma_{xy} \\ \kappa_x \\ \kappa_y \\ \kappa_{xy} \end{bmatrix} = \begin{bmatrix} A & B \\ B & D \end{bmatrix}^{-1} \begin{bmatrix} N_x \\ N_y \\ N_{xy} \\ M_x \\ M_y \\ M_{xy} \end{bmatrix}$$

The same can be done for E_y and G_{xy} by defining a value for N_y and N_{xy}, respectively.

In [11]:
```
Exlam, Eylam, Gxylam = Lam_props(ABD, LayerT)
print_laminate_props(ABD, LayerT, stack)
```

```
******Engineering properties of laminate stack [0, 90, 90, 0]   ******

      Young's modulus in the x-direction is  79303718565.13  N/m^2
      Young's modulus in the y-direction is  79303718565.13  N/m^2
      Shear modulus in the xy-plane is       2588319435.4    N/m^2

***********************************************************************
```

In [ ]:
```

```