PROYECTO INTEGRADOR DE MATERIALES

Ingeniería Mecatrónica

Facultad de Ingeniería
Universidad Nacional de Cuyo
2020

Carlos Alberto Bustillo López
Legajo 11586

Desarrollo del proyecto

Para el desarrollo de este proyecto integrador utilizamos Python 2.7 (pero también se puede usar ejecutar en versiones más recientes Python 3.x). El proyecto fue desarrollado en conjunto con mi compañero Agustín Lezcano (Legajo 11956). Para la realizar las operaciones matriciales se creó una librería llamada *Operaciones.py* que se adjunta al final del informe.

Parte A: Análisis en la microescala de una lámina de un material compuesto

Hipótesis consideradas: Comportamiento elástico o lineal (de las fibras y la matriz), fibras infinitamente largas y espaciadas periódicamente (ordenamiento cuadrado o hexagonal).

Se planteó la *Regla de las mezclas* para determinar las propiedades del material compuesto.

$$E_1 = E_f * V_f + E_m * V_m$$

$$E_2 = \frac{E_f * E_m}{E_f * V_m + E_m * V_f}$$

$$G_m = \frac{E_m}{2*(1+v_m)}$$

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

$$v_{12} = vf_{12} * V_f + v_m * V_m$$

donde
E_1 y E_2 son los módulos de Young del compuesto,
E_m: Módulo de Elasticidad Longitudinal de la matriz,
E_f: Módulo de Elasticidad Longitudinal de la fibra,
vf_{12}: Coeficiente de Poisson de la fibra,
v_m: Coeficiente de Poisson de la matriz,
V_f: Fracción de volumen de las fibras,
V_m: Fracción de volumen de la matriz,
G_m: Módulo de elasticidad transversal.

Implementación en Python

```
"""
PARTE A: Una  lamina de compuesto de  polimero  reforzado  con  grafito
"""
import numpy as np
import Operaciones
import matplotlib.pyplot as plt

#Inicializacion datos
Em= 4.62*(10**9)
```

```
Ef1=233*(10**9)
Ef2=23.1*(10**9)
vm=0.36           #Coeficiente de Poisson
vf12=0.2
Gf12=8.96*(10**9)
Vf=np.arange(0,1.1,0.1)  #Fraccion de volumen

#Calculo
Vm = Operaciones.restar(1,Vf)
vc = Operaciones.division(vm,Vm)
E1 = Operaciones.suma(Operaciones.multiplicar(Ef1,Vf), Operaciones.multiplicar(Em,Vm))
E2 =
Operaciones.division(Ef2*Em,Operaciones.suma(Operaciones.multiplicar(Em,Vf),Operaciones.mul
tiplicar(Ef2,Vm)))
Gm = Em/(2*(1+vm))
G12 =
Operaciones.division(Gf12*Gm,Operaciones.suma(Operaciones.multiplicar(Gf12,Vm),Operaciones
.multiplicar(Gm,Vf)))
v12 = Operaciones.suma(Operaciones.multiplicar(vf12,Vf),Operaciones.multiplicar(vm,Vm))

#Grafico 1
plt.plot(Vf,E1)
plt.legend(["E1"])
plt.plot(0.6,E1[6], marker="o", color="blue")
plt.xlabel("Vf")
plt.title("Carlos Bustillo - Agustin Lezcano")
plt.grid()
plt.show()

#Grafico 2
plt.plot(Vf,E2)
plt.plot(Vf,G12)
plt.legend(["E2","G12"])
plt.plot(0.6,E2[6], marker="o", color="blue")
plt.plot(0.6,G12[6], marker="o", color="orange")
plt.xlabel("Vf")
plt.title("Carlos Bustillo - Agustin Lezcano")
plt.grid()
plt.show()

#Grafico 3
plt.plot(Vf,v12)
plt.legend(["v12"])
plt.plot(0.6,v12[6], marker="o", color="blue")
plt.xlabel("Vf")
plt.title("Carlos Bustillo - Agustin Lezcano")
plt.grid()
plt.show()

#Obtencion de valores
print("E1",format(E1[6],'.3e'))
print("E2",format(E2[6],'.3e'))
```

print("v12",v12[6])
print("G12",format(G12[6],'.3e'))

<u>Resultados y gráficas obtenidas</u>

a) Utilizando la regla de las mezclas se determino los valores de las constantes elásticas E_1, E_2, G_{12} y v_{12} para la lámina usando una fracción de volumen de la fibra $V^f = 0.6$.

```
('E1', '1.416e+11')
('E2', '8.885e+09')
('v12', 0.264)
('G12', '3.306e+09')
carlos@cabustillo13:~/Documentos/Proyectos/Materiales-Compuestos$
```

Ilustración 1: Valores obtenidos para $V^f = 0.6$.

b) Grafique E_1, E_2, G_{12} y v_{12} en función de V^f considerando $0 \leq V^f \leq 1$.

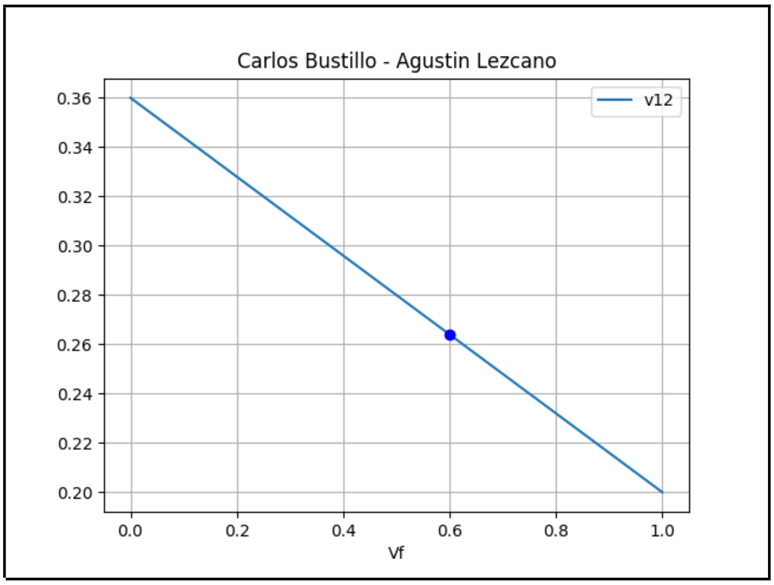


Ilustración 2: Gráfica v_{12} vrs V^f

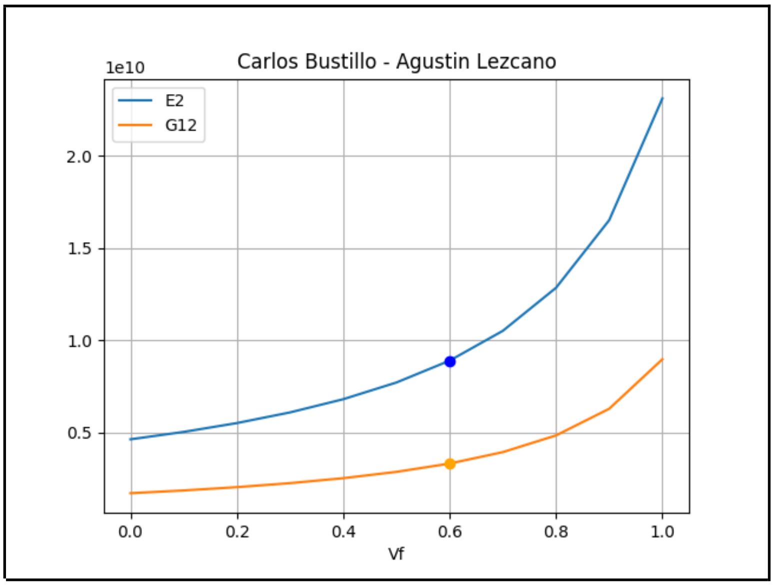


Ilustración 3: Gráfica E_2 y G12 vrs V^f

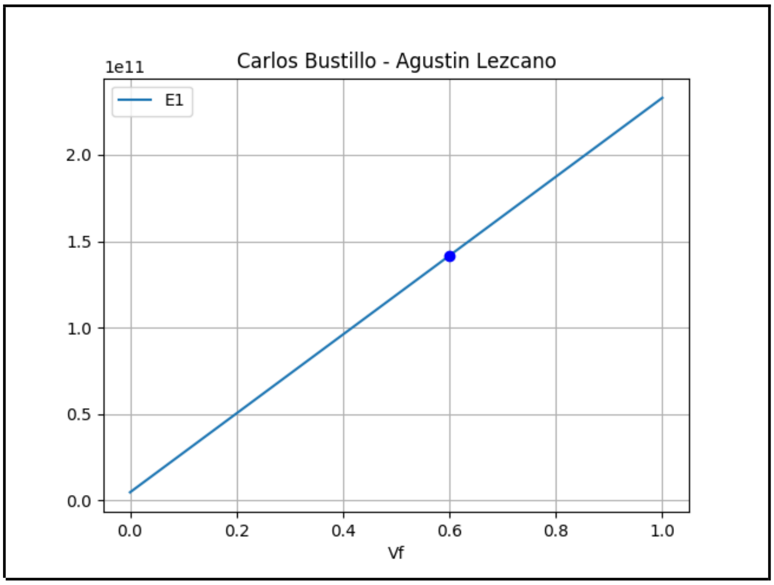


Ilustración 4: Gráfica E_1 vrs V^f

Parte B: Análisis en la macroescala de una lámina de un material compuesto

Se considera en este caso un material ortotrópico, es decir que existen 3 planos de simetría mutuamente perpendiculares en el mismo (en este caso se consideran los ejes coordenados x,y,z).

Para definir el comportamiento de la lámina se parte de la *Ley de Hooke* generalizada y se llega a la siguiente expresión:

$$\begin{bmatrix} \sigma_1 \\ \sigma_2 \\ \sigma_3 \\ \tau_{23} \\ \tau_{31} \\ \tau_{12} \end{bmatrix} = \begin{bmatrix} C_{11} & C_{12} & C_{13} & C_{14} & C_{15} & C_{16} \\ C_{21} & C_{22} & C_{23} & C_{24} & C_{25} & C_{26} \\ C_{31} & C_{32} & C_{33} & C_{34} & C_{35} & C_{36} \\ C_{41} & C_{42} & C_{43} & C_{44} & C_{45} & C_{46} \\ C_{51} & C_{52} & C_{53} & C_{54} & C_{55} & C_{56} \\ C_{61} & C_{62} & C_{63} & C_{64} & C_{65} & C_{66} \end{bmatrix} \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3 \\ \gamma_{23} \\ \gamma_{31} \\ \gamma_{12} \end{bmatrix}$$

Ilustración 5: Expresión General

Siendo C la matriz de rigidez, su inversa es la matriz de flexibilidad S, entonces $\epsilon = S*\sigma$. Estas matrices son simétricas. EL caso presentado en la Ilustración 4 es el caso más general de comportamiento elástico. Para nuestro análisis se considera que la placa es de material

ortrotrópico, entonces coincidiendo los planos de simetría con los planos del sistema de coordenadas de referencia, la relación tensión-deformación se reduce a: Se puede observar en la ecuación de la ilustración 5 que el comportamiento puede ahora definirse mediante nueve constantes independientes.

$$\begin{bmatrix} \sigma_1 \\ \sigma_2 \\ \sigma_3 \\ \tau_{23} \\ \tau_{31} \\ \tau_{12} \end{bmatrix} = \begin{bmatrix} C_{11} & C_{12} & C_{13} & 0 & 0 & 0 \\ C_{12} & C_{22} & C_{23} & 0 & 0 & 0 \\ C_{13} & C_{23} & C_{33} & 0 & 0 & 0 \\ 0 & 0 & 0 & C_{44} & 0 & 0 \\ 0 & 0 & 0 & 0 & C_{55} & 0 \\ 0 & 0 & 0 & 0 & 0 & C_{66} \end{bmatrix} \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3 \\ \gamma_{23} \\ \gamma_{31} \\ \gamma_{12} \end{bmatrix}$$

Ilustración 6: Material Ortotrópico

También se ve que las distorsiones angulares y las deformaciones longitudinales están desacopladas de las tensiones normales y las tensiones tangenciales, respectivamente. Además de lo considerado anteriormente se considera que sobre la placa actuá una fuerza en una sola dirección perpendicular al eje z, es decir, se considera movimiento plano. Entonces la ecuación vista anteriormente se puede definir como: Esta matriz es la llamada *Matriz de flexibilidad reducida.*

$$\begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \gamma_{12} \end{bmatrix} = \begin{bmatrix} S_{11} & S_{12} & 0 \\ S_{12} & S_{22} & 0 \\ 0 & 0 & S_{66} \end{bmatrix} \begin{bmatrix} \sigma_1 \\ \sigma_2 \\ \tau_{12} \end{bmatrix}$$

Ilustración 7: Matriz de Flexibilidad Reducida

<u>Implementación en Python</u>

```
import matplotlib.pyplot as plt
import numpy as np
if __name__ == "__main__":

    #Definicion de constantes
```

```
x=200                    #x:largo    [mm]
y=100                    #y:ancho    [mm]
z=0.2                   #z: espesor [mm]
E1=155                    #GPa = 1 kN/mm^2
E2=12.1                   #GPa = 1 kN/mm^2
E3=12.1                   #GPa = 1 kN/mm^2
v23=0.458
v12=0.248
v13=0.248
G23=3.2                   #GPa = 1 kN/mm^2
G12=4.4                   #GPa = 1 kN/mm^2
G13=4.4                   #GPa = 1 kN/mm^2

sigma = [4/(y*z), 0, 0]     #sigma 1, 2 y 3

# Inciso a)
epsylon3 = (-v13/E1)* sigma[0] + (-v23/E2)*sigma[1]
print("Calculo de epsylon3: ", epsylon3)

#Inciso b)
#Matriz de flexibilidad reducida

S11=1/E1
S12=-v12/E1
S22=1/E2
S66=1/G12

S = [[S11, S12, 0],
   [S12, S22, 0],
   [0, 0, S66]]

#Restricciones de coef. de Poisson
v21=(v12/E1)*E2
v31=(v13/E1)*E3

#Matriz de rigidez reducida
Q = [[E1/(1-v12*v21), (v12*E2)/(1-v21*v12), 0],
   [(v12*E2)/(1-v21*v12), E2/(1-v21*v12), 0],
   [0, 0, G12]]

INV=np.linalg.inv(S)

print("Matrices S: ", S)
print("Matrices Q: ", Q)

# Inciso c)Comparacion entre S y Q
if (np.all(Q)==np.all(INV)):
   print("Las matrices son iguales")

#Inciso d)
dominio = np.arange(-np.pi/2,np.pi/2,(np.pi/100))
```

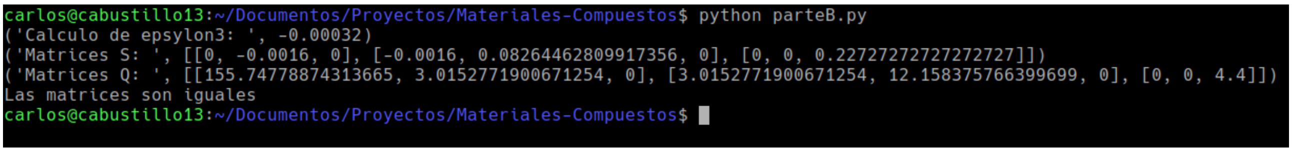
```
    S11n=[]
    S12n=[]
    S22n=[]
    S16n=[]
    S26n=[]
    S66n=[]

    for i in dominio:
        S11n.append( S11 * np.power(np.cos(i),4) + (2*S12+S66) * np.power((np.cos(i)*np.sin(i)),2) + S22 * np.power(np.sin(i),4))
        S12n.append( S12 * (np.power(np.sin(i),4) + np.power(np.cos(i),4))+(S11+S22-S66)*np.power((np.sin(i)*np.cos(i)),2))
        S22n.append(S11*np.power(np.sin(i),4)+(2*S12+S66)*np.power((np.sin(i)*np.cos(i)),2)+S22*np.power(np.cos(i),4))
        S16n.append((2*S11-2*S12-S66)* np.sin(i)*np.power(np.cos(i),3) - (2*S22-2*S12-S66)* np.cos(i)*np.power(np.sin(i),3))
        S26n.append((2*S11-2*S12-S66)* np.cos(i)*np.power(np.sin(i),3) - (2*S22-2*S12-S66)* np.sin(i)*np.power(np.cos(i),3))
        S66n.append(2* (2*S11+2*S22-4*S12-S66)* np.power((np.sin(i)*np.cos(i)),2)+S66*(np.power(np.sin(i),4)+np.power(np.cos(i),4)))

    #Graficos
    plt.figure()
    plt.subplot(2,3,1)
    plt.plot(dominio,S11n)
    plt.subplot(2,3,2)
    plt.plot(dominio,S12n)
    plt.subplot(2,3,3)
    plt.plot(dominio,S22n)
    plt.subplot(2,3,4)
    plt.plot(dominio,S16n)
    plt.subplot(2,3,5)
    plt.plot(dominio,S26n)
    plt.subplot(2,3,6)
    plt.plot(dominio,S66n)
    plt.show()
```

<u>Resultados y gráficas obtenidas</u>

a) Permita calcular la deformación ϵ_3, suponiendo que la lámina está en un estado de tensión plana.

b) Calcule las matrices de flexibilidad y rigidez reducidas.

c) Verificar que las dos matrices obtenidas en el inciso anterior sean inversas.



```
carlos@cabustillo13:~/Documentos/Proyectos/Materiales-Compuestos$ python parteB.py
('Calculo de epsylon3: ', -0.00032)
('Matrices S: ', [[0, -0.0016, 0], [-0.0016, 0.08264462809917356, 0], [0, 0, 0.22727272727272727]])
('Matrices Q: ', [[155.74778874313665, 3.0152771900671254, 0], [3.0152771900671254, 12.158375766399699, 0], [0, 0, 4.4]])
Las matrices son iguales
carlos@cabustillo13:~/Documentos/Proyectos/Materiales-Compuestos$
```

Ilustración 8: Resultados de los inciso a) , b) y c)

d) Considerando coordenadas globales permita graficar los valores de las seis elementos de la matriz de flexibilidad reducida transformada como una función de la orientación del ángulo θ entre $-\pi/2 \leq \theta \leq \pi/2$.

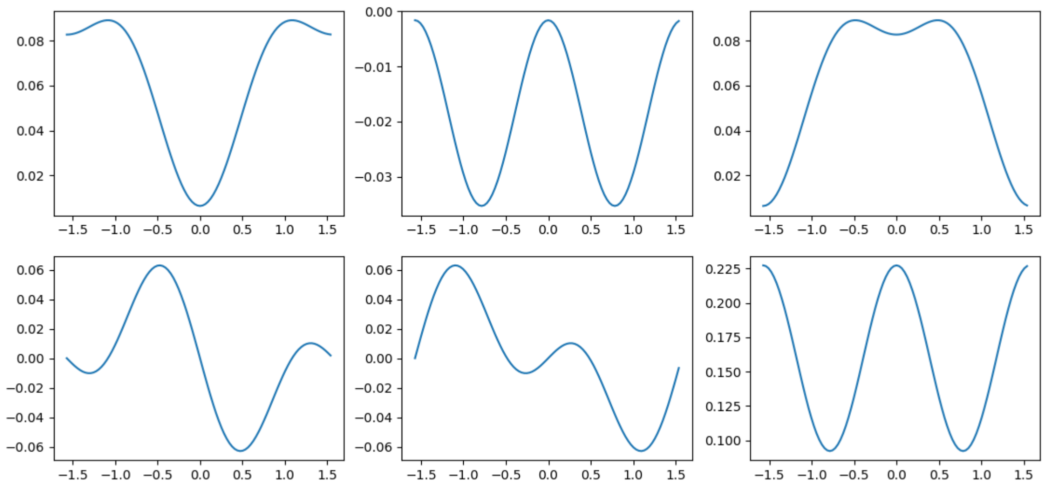


Ilustración 9: Grafica de los seis elementos de la matriz de flexibilidad reducida transformada.

<u>Librerías auxiliares utilizadas: Operaciones.py</u>

```
#Division miembro a miembro
def division(numerador,denominador,op = False):
    resultado=list()
    if (op==False):
        for i in range(len(denominador)):
            resultado.append(numerador/denominador[i])
    if (op==True):
        for i in range(len(numerador)):
            resultado.append(numerador[i]/denominador)
    return resultado

#Restar/Sumar punto a punto
def restar(minuendo,sustraendo):
    resultado=list()
```

```
    for i in range(len(sustraendo)):
        resultado.append(minuendo-sustraendo[i])
    return resultado

#Multiplicar miembro a miembro
def multiplicar(multiplicando,multiplicador):
    resultado=list()
    for i in range(len(multiplicador)):
        resultado.append(multiplicando*multiplicador[i])
    return resultado

#Suma miembro a miembro
def suma(aux1,aux2):
    resultado = list()
    for i in range(len(aux1)):
        resultado.append(aux1[i]+aux2[i])
    return resultado
```