Apéndice C. Código Python

```
import numpy as np

# 1. PROPIEDADES DE LOS MATERIALES

# Propiedades de las capas de fibra de refuerzo
Ef1 = 24873  # MPa
Ef2 = 2344.1 # MPa
Ef3 = Ef2  # MPa
Gf12 = 682.72  # MPa
Gf13 = Gf12  # MPa
Gf23 = 559.03  # MPa
nuf12 = 0.37
nuf13 = nuf12
nuf23 = 0.503
nuf21 = nuf12 * (Ef2 / Ef1)
nuf31 = nuf13 * (Ef3 / Ef1)
nuf32 = nuf23 * (Ef3 / Ef2)

# Propiedades del honeycomb triangular Onyx
E1 = 85 # MPa
E2 = 85  # MPa
E3 = 256.02  # MPa
G12 = 32  # MPa
G13 = 94.82  # MPa
G23 = 94.82  # MPa
nu12 = 0.3333
nu13 = 0.1161
nu23 = 0.1161
nu21 = (nu12 * E2) / E1
nu31 = (nu13 * E3) / E1
nu32 = (nu23 * E3) / E2

# Propiedades de las regiones sólidas de Onyx
E = 1400  # MPa
```

```
nu = 0.35
G = E / (2 * (1 + nu))  # MPa

rho = 0.1  # Porosidad igual al 10%
E1solid = (1 - rho) * E  # MPa
E2solid = (1 - rho ** 0.5) * E  # MPa
E3solid = E2solid  # MPa
G12solid = G * (1 - rho) * (1 - rho ** 0.5) / ((1 - rho) + (1 - rho ** 0.5))  # MPa
G13solid = G12solid  # MPa
G23solid = (1 - rho ** 0.5) * G  # MPa
nu12solid = (1 - rho) * nu
nu13solid = nu12solid
nu23solid = (1 - rho ** 0.5) * nu
nu21solid = (1 - rho ** 0.5) * nu
nu31solid = nu21solid
nu32solid = nu21solid

# 2. MATRIZ DE CUMPLIMIENTO

# Matriz de cumplimiento para las fibras de Kevlar
Skevlar = np.array([
    [1 / Ef1, -nuf21 / Ef2, -nuf31 / Ef3, 0, 0, 0],
    [-nuf12 / Ef1, 1 / Ef2, -nuf32 / Ef3, 0, 0, 0],
    [-nuf13 / Ef1, -nuf23 / Ef2, 1 / Ef3, 0, 0, 0],
    [0, 0, 0, 1 / Gf23, 0, 0],
    [0, 0, 0, 0, 1 / Gf13, 0],
    [0, 0, 0, 0, 0, 1 / Gf12]
])

# Matriz de cumplimiento para las capas de honeycomb
Shoneycomb = np.array([
    [1 / E1, -nu21 / E2, -nu31 / E3, 0, 0, 0],
    [-nu12 / E1, 1 / E2, -nu32 / E3, 0, 0, 0],
    [-nu13 / E1, -nu23 / E2, 1 / E3, 0, 0, 0],
    [0, 0, 0, 1 / G23, 0, 0],
    [0, 0, 0, 0, 1 / G13, 0],
    [0, 0, 0, 0, 0, 1 / G12]
])

# Matriz de cumplimiento para las capas sólidas de Onyx
Ssolid = np.array([
    [1 / E1solid, -nu21solid / E2solid, -nu31solid / E3solid, 0, 0, 0],
    [-nu12solid / E1solid, 1 / E2solid, -nu32solid / E3solid, 0, 0, 0],
    [-nu13solid / E1solid, -nu23solid / E2solid, 1 / E3solid, 0, 0, 0],
    [0, 0, 0, 1 / G23solid, 0, 0],
    [0, 0, 0, 0, 1 / G13solid, 0],
    [0, 0, 0, 0, 0, 1 / G12solid]
])
```

```
# Matriz de cumplimiento para la pared de Onyx
Spared = Ssolid

# 3. TRANSFORMACIÓN DEL SISTEMA DE COORDENADAS

# 3.1 Transformación positiva
alphapos = 45 * (np.pi / 180)
c1 = np.cos(alphapos)
s1 = np.sin(alphapos)

Tpos = np.array([
    [c1**2, s1**2, 0, 0, 0, 2 * c1 * s1],
    [s1**2, c1**2, 0, 0, 0, -2 * c1 * s1],
    [0, 0, 1, 0, 0, 0],
    [0, 0, 0, c1, s1, 0],
    [0, 0, 0, -s1, c1, 0],
    [-c1 * s1, c1 * s1, 0, 0, 0, c1**2 - s1**2]
])

# 3.2 Transformación negativa
alphaneg = -45 * (np.pi / 180)
c2 = np.cos(alphaneg)
s2 = np.sin(alphaneg)

Tneg = np.array([
    [c2**2, s2**2, 0, 0, 0, 2 * c2 * s2],
    [s2**2, c2**2, 0, 0, 0, -2 * c2 * s2],
    [0, 0, 1, 0, 0, 0],
    [0, 0, 0, c2, s2, 0],
    [0, 0, 0, -s2, c2, 0],
    [-c2 * s2, c2 * s2, 0, 0, 0, c2**2 - s2**2]
])

ShoneycombPos = Tpos.T @ Shoneycomb @ Tpos
ShoneycombNeg = Tneg.T @ Shoneycomb @ Tneg
SsolidPos = Tpos.T @ Ssolid @ Tpos
SsolidNeg = Tneg.T @ Ssolid @ Tneg

# 4. MATRIZ INVERSA DE CUMPLIMIENTO

Cpared = np.linalg.inv(Spared)  # Matriz rigidez pared
Ckevlar = np.linalg.inv(Skevlar)  # Matriz rigidez capas Kevlar
ChoneycombCero = np.linalg.inv(Shoneycomb)  # Matriz rigidez capas honeycomb a 0°
ChoneycombPos = np.linalg.inv(ShoneycombPos)  # Matriz rigidez capas honeycomb a 45°
ChoneycombNeg = np.linalg.inv(ShoneycombNeg)  # Matriz rigidez capas honeycomb a -45°
CsolidPos = np.linalg.inv(SsolidPos)  # Matriz rigidez capas Onyx a 45°
CsolidNeg = np.linalg.inv(SsolidNeg)  # Matriz rigidez capas Onyx a -45°

# 5. FRACTIONES VOLUMÉTRICAS
```

```
xkevlar = 0.3042  # fracción volumétrica capas Kevlar
xpared = 0.41921  # fracción volumétrica pared de Onyx
xhoneycombPos = 0.069  # fracción volumétrica capas honeycomb a 45°
xhoneycombNeg = 0.069  # fracción volumétrica capas honeycomb a -45°
xhoneycombCero = 0.069  # fracción volumétrica capas honeycomb a 0°
xsolidPos = 0.0347  # fracción volumétrica capas Onyx a 45°
xsolidNeg = 0.0347  # fracción volumétrica capas Onyx a -45°

# 6. RIGIDEZ TOTAL

Ctotal = (xkevlar * Ckevlar) + (xpared * Cpared) + \
        (xhoneycombPos * ChoneycombPos) + (xhoneycombNeg * ChoneycombNeg) + \
        (xhoneycombCero * ChoneycombCero) + (xsolidPos * CsolidPos) + \
        (xsolidNeg * CsolidNeg)

Stotal = np.linalg.inv(Ctotal)

Ex = 1 / Stotal[0, 0]
Ey = 1 / Stotal[1, 1]
Ez = 1 / Stotal[2, 2]
Gxy = 1 / Stotal[5, 5]
Gyz = 1 / Stotal[4, 4]
Gzx = 1 / Stotal[3, 3]
vxy = -Stotal[0, 1] / Stotal[0, 0]
vzx = -Stotal[0, 2] / Stotal[2, 2]
vyz = -Stotal[1, 2] / Stotal[1, 1]
vxz = -Stotal[2, 0] / Stotal[0, 0]

print(Ex)
print(Ey)
print(Ez)
print(Gxy)
print(Gyz)
print(Gzx)
print(vxy)
print(vyz)
print(vxz)
```