```
import numpy as np
import matplotlib.pyplot as plt

np.set_printoptions(precision=3)

def Transform(theta):
    m = np.cos( np.deg2rad(theta) )
    n = np.sin( np.deg2rad(theta) )
    return np.array([
        [m**2, n**2, 2*m*n],
        [n**2, m**2, -2*m*n],
        [-m*n, m*n,  m**2 - n**2]], np.float64)


theta = np.array([+30,-30,0,0,-30,+30])

N = theta.size
h = .15*10**-3
H = N*h

Z = np.arange(N+1)*h - .5*H

E1 = 155 * 10**9
E2 = 12.1 * 10**9
v12 = .248
G12 = 4.4 * 10**9

S = np.array([
    [1/E1, -v12/E1, 0],
    [-v12/E1, 1/E2, 0],
    [0,  0       , 1/G12]], np.float64)

T = Transform(theta)
T_ = np.rollaxis(T, 2)

Sbar = np.einsum('...jk,kl,...lm->...jm', T.T, S, T_)
Qbar = np.linalg.inv(Sbar)

A = np.sum( np.diff(Z)[:, None, None] * Qbar, axis=0)
B = (1/2)*np.sum( np.diff(Z**2)[:, None, None] * Qbar, axis=0)
D = (1/3)*np.sum( np.diff(Z**3)[:, None, None] * Qbar, axis=0)

ABD = np.block([[A,B],[B,D]])

ABD[np.abs(ABD) < 10**-8] = 0

abd = np.linalg.inv(ABD)
```

```
Xt = 1500 * 10**6
Xc = 1250 * 10**6

Yt =   50 * 10**6
Yc =  200 * 10**6
S  =  100 * 10**6


alpha_1 = -0.018 * 10**-6
alpha_2 = 24.3   * 10**-6

beta_1 =  146 * 10**-6
beta_2 = 4770 * 10**-6


alpha = np.array([alpha_1, alpha_2, 0])
beta  = np.array([beta_1, beta_2, 0])

alpha_bar = np.matmul(T.T, alpha[:, None]).reshape(N, 1, 3)
beta_bar = np.matmul(T.T, beta[:, None]).reshape(N, 1, 3)

NM = np.array([[0,0,0,1,0,0]]).T

N_t = np.sum( np.diff(Z)[:, None] *  np.sum( Qbar * alpha_bar, axis=2 ), axis=0)
M_t = (1/2)*np.sum( np.diff(Z**2)[:, None] *  np.sum( Qbar * alpha_bar, axis=2 ),
axis=0)
NM_t = np.concatenate([N_t, M_t]).reshape(6, 1)

N_m = np.sum( np.diff(Z)[:, None] *  np.sum( Qbar * beta_bar, axis=2 ), axis=0)
M_m = (1/2)*np.sum( np.diff(Z**2)[:, None] *  np.sum( Qbar * beta_bar, axis=2 ),
axis=0)

NM_m = np.concatenate([N_m, M_m]).reshape(6, 1)

dT = -150

NM_r = 25*NM + dT*NM_t

Z_ = np.hstack((Z[:N//2], Z[-N//2:]))

ref_surf = np.dot(abd, NM_r)
surf_strain = ref_surf[:3]
surf_curve = ref_surf[3:]

Strains = surf_strain + np.dot(surf_curve, Z_[None,:])
Sigma_xyz = np.matmul(Qbar, Strains.T[:,:,None])
Sigma_123 = np.matmul(T_, Sigma_xyz)
```