```
import numpy as np
import sympy as sp
from sympy import Symbol

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


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

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

S  =  100 * 10**6

F1 = (1/Xt) - (1/Xc)
F2 = (1/Yt) - (1/Yc)

F11 = 1/(Xt*Xc)
F22 = 1/(Yt*Yc)

F66 = 1/S**2

F12 = -np.sqrt(F11*F22)

NM = np.array([
    [-5.79, .85, 1.415],
    [-9.02, .876, -.772],
    [-7.52, .618, .0858],
    [7.52, -.618, -.0858],
    [9.02, -.876, .772],
    [5.79, -.85, -1.415]]) * 10**6

NM_t = np.array([
    [.368, -.141, .0722],
    [.368, -.141, -.0722],
    [-.356, -.0977, 0],

    [-.356, -.0977, 0],
    [.368, -.141, -.0722],
    [.368, -.141, .0722]]) * 10**6
```

```
dT = -150

compression, tension = (np.inf, np.inf)

c_ply = 0
t_ply = 0

mx_var = Symbol('mx')

for i in range(N):
    s1 = NM[i, 0] * mx_var + NM_t[i, 0] * dT
    s2 = NM[i, 1] * mx_var + NM_t[i, 1] * dT
    t12 = NM[i, 2] * mx_var + NM_t[i, 2] * dT

    equ = F11*s1**2 + F22*s2**2 + F66*t12**2 + F1*s1 + F2*s2 + F12*s1*s2

    c, t = sp.solve(equ-1, mx_var)

    print('Mx= %.2f / %.2f (Nm/m) in (%d) ply at Z=%.3fmm' %(c, t, theta[i],
Z_[i]*10**3))

    compression = min(np.abs(c), compression)
    tension = min(np.abs(t), tension)

    if np.abs(c) == compression: c_ply = i
    if np.abs(t) == tension: t_ply = i

print('--------------------')
print('FPF from -Mx=%.2f (Nm/m) in (%d) ply at Z=%.3fmm' %(-compression,
theta[c_ply], Z_[c_ply]*10**3))
print('FPF from Mx=%.2f (Nm/m) in (%d) ply at Z=%.3fmm' %(tension, theta[t_ply],
Z_[t_ply]*10**3))
```