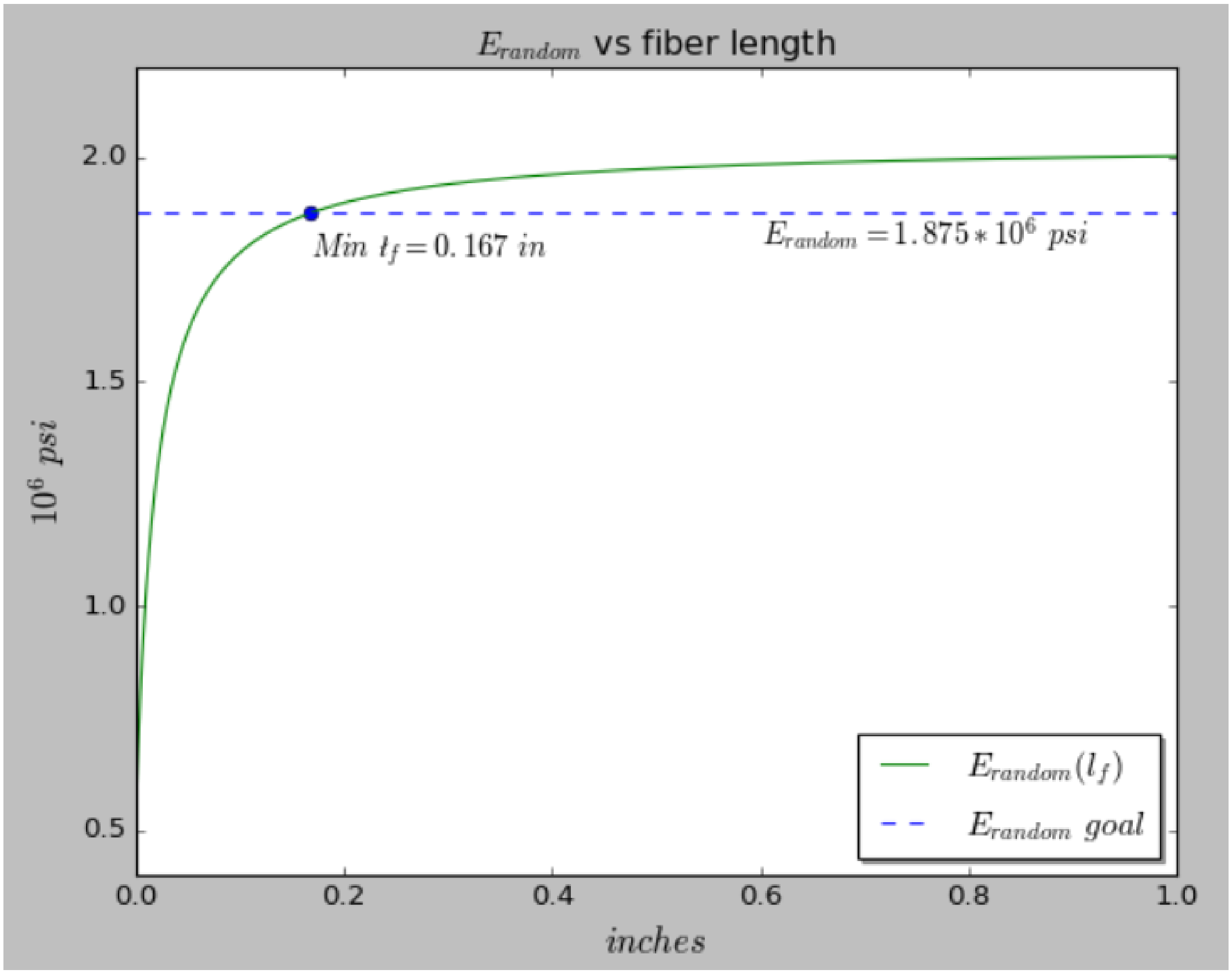
#1

- The new tensile modulus goal is 150% of the previous composite, resulting in Erandom of 1.875 *10^6 psi.
- First need to find the volume fractions from the mass fraction of 20%
- Use Halpin-Tsai equations modified for discontinuous fiber to find E1 and E2
- Substitute in Tsai and Pagano equation for Randomly oriented discontinuous fiber
- Now have an equation with **fiber lengt**h as single variable
- Set 3/8*E1 + 5/8*E2 – Erandom = 0 and solve for **fiber length**

  -1511576.3149403 + 150000.0(33717.0263788969*l_f/(3333.33333333333*l_f + 75.0) + 1)/(1 - 10.1151079136691/(3333.33333333333*l_f + 75.0)) = 0*

- The graph shows the minimum fiber length that will achieve the desired tensile modulus



#2

$$\bar{S}_{0^\circ} = \begin{bmatrix} 2.027e-11 & -5.270e-12 & 0.000e+00 \\ -5.270e-12 & 7.963e-11 & 0.000e+00 \\ 0.000e+00 & 0.000e+00 & 2.295e-10 \end{bmatrix} GPa^{-1}$$

$$\bar{S}_{45^\circ} = \begin{bmatrix} 7.971e-11 & -3.503e-11 & -2.968e-11 \\ -3.503e-11 & 7.971e-11 & -2.968e-11 \\ -2.968e-11 & -2.968e-11 & 1.104e-10 \end{bmatrix} GPa^{-1}$$

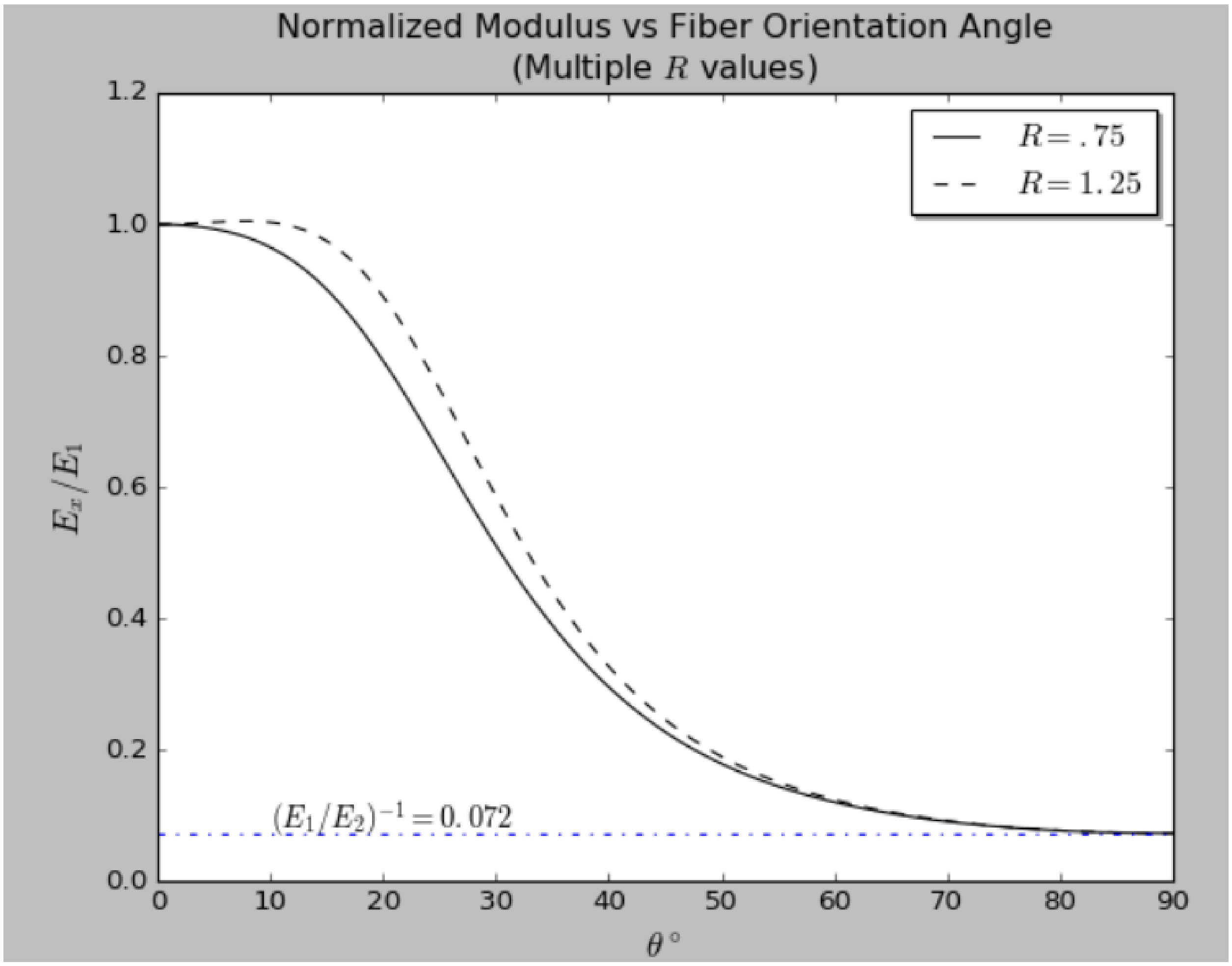
Assuming plane stress for the thin lamina. $\bar{S}_{0^\circ}$ is the same as S since no rotation has occurred.

$\bar{S}_{45^\circ}$ has introduced coupling between tensile and shear components since $\bar{S}_{16,\,45^\circ}$ and $\bar{S}_{26,\,45^\circ}$ are non-zero. We also see that the two tensile modulus values have become equal at 45º.

#3

Ex/E1 = 1/(E1*n**4/E2 + m**4 + m**2*n**2*(-2*v12 + (2*v12 + 2)/R))

Solved with algebraic system, see code #3 (includes assumptions about properties)



We can see that higher values of R result in equal or higher tensile modulus values across all orientation angles. As expected, both normalized modulus are 1 at 0°, since no transformation has occurred yet, so it is equivalent to E1/E1. Additionally, we see that Ex/E1 approaches E2/E1 as theta goes to 90°, which occurs since E1 and E2 are 90° apart.

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

Erandom = (1+.5)*1.25*10**6 #psi

Wf = .2
rho_f = 1.8 #g/cm**3
Ef = 30*10**6 #psi
lf = sp.var('l_f')
df = .0006 #in

Wm = 1 - Wf
rho_m = 1.14 #g/cm**3
Em = .4*10**6 #psi

rho_c = 1/( (Wf/rho_f) + (Wm/rho_m) )
Vf = Wf * (rho_c/rho_f)

eta_L = ((Ef/Em)-1) / ( (Ef/Em) + 2*(lf/df))
eta_T = ((Ef/Em)-1) / ( (Ef/Em) + 2)

E1 = ( (1 + 2*(lf/df)*eta_L*Vf) / (1 - eta_L*Vf) )*Em
E2 = ( (1+2*eta_T*Vf) / (1-eta_T*Vf) )*Em

f = .375*E1 + .625*E2 - Erandom
length, = sp.solve(f, lf)
E = sp.lambdify(lf, f+Erandom)
lf = np.linspace(0, 1, 1000)
fig, ax = plt.subplots()

pair = (length, E(length)/10**6)
ax.plot(*pair, 'o')
ax.annotate(r'$Min$ $\l_f={%.3f}$ $in$' %pair[0], xy=(length, pair[1]-.1),
fontsize=13)
plt.plot(lf, E(lf)/10**6, label='$E_{random}(l_f)$')

plt.axhline(Erandom/10**6, linestyle='--')
ax.annotate(r'$E_{random}=%.3f*10^6$ $psi$' %(Erandom/10**6), xy=(.6, Erandom/10**6
- .07), fontsize=13)

plt.title('$E_{random}$ vs fiber length')
plt.xlabel(r'$inches$', fontsize=15)
plt.ylabel(r'$10^6$ $psi$', fontsize=15)

legend = ax.legend(loc='lower right', shadow=True)

plt.show()
```

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

np.set_printoptions(precision=3)

sqrt = lambda e: np.sqrt(e)

Ef1 = 80*10**9
Ef2 = 80*10**9
Gf12 = 33.33*10**9
vf12 = .2
Vf = .6
sq_Vf = sqrt(Vf)

Em = 3.35*10**9
Gm = 1.24*10**9
vm = .35
Vm = 1 - Vf


E1 = (Ef1*Vf + Em*Vm)
E2 = ( (1-sq_Vf)/Em + sq_Vf/(Ef2*sq_Vf + Em*(1-sq_Vf)) )**-1
G12 = Gm * ( (Gm*(1-Vf) + Gf12*(1+Vf)) / (Gm*(1+Vf) + Gf12*(1-Vf)) )
v12 = vf12*Vf + vm*Vm


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

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)

T = Transform(0)
Sbar_0 = np.dot(T.T, np.dot(S, T))

T = Transform(45)
Sbar_45 = np.dot(T.T, np.dot(S, T))
```

```
import matplotlib.pyplot as plt
import numpy as np
import sympy as sp
#variables relevant to equation
E1, E2, m, n, v12, G12, R = sp.var('E1 E2 m n v12 G12 R')

#shear modulus equation
G12 = (R*E1)/(2*(1+v12))

#Transformed tensile modulus
Ex = E1/(m**4 + (n**2)*(m**2)*(E1/G12 - 2*v12) + (E1/E2)*(n**4))

#normalize modulus, equation to be plotted as function of theta
f = Ex/E1

#assume volume fractions
Vf = .625
Vm = 1 - Vf

#assume fiber poisson's ratios
vf12 = .215
vm = .325
v12 = vf12*Vf + vm*Vm
E1E2 = 13.8

theta = np.linspace(0,90,90)
m = np.cos(np.deg2rad(theta))
n = np.sin(np.deg2rad(theta))

fig, ax = plt.subplots()

R = .75
ExE1_1 = 1/(E1E2*n**4 + m**4 + m**2*n**2*(-2*v12 + (2*v12 + 2)/R))
plt.plot(theta, ExE1_1, 'k-', label=r'$R=.75$')

R = 1.25
ExE1_2 = 1/(E1E2*n**4 + m**4 + m**2*n**2*(-2*v12 + (2*v12 + 2)/R))
plt.plot(theta, ExE1_2, 'k--', label=r'$R=1.25$')

plt.axhline(1/E1E2, linestyle='-.')
ax.annotate(r'${(E_1/E_2)}^{-1}=%.3f$' %(1/13.8), xy=(10, 1/12.5), fontsize=13)

plt.title('Normalized Modulus vs Fiber Orientation Angle\n(Multiple $R$ values)')
plt.xlabel(r'$\theta^\circ$', fontsize=15)
plt.ylabel(r'$E_x/E_1$', fontsize=15)
legend = ax.legend(loc='upper right', shadow=True)
plt.xticks(np.linspace(0, 90, 10))
plt.show()
```