```python
from falkner_skan import falkner_skan
import numpy as np
import sympy as sp
import matplotlib.pyplot as plt

### Define constants
m = 0.0733844181517584 # Exponent of the edge velocity ("a")
Vinf = 7 # Velocity at the trailing edge
nu = 1.45e-5 # kinematic viscosity
c = 0.08 # chord, in meters
x = sp.symbols("x") # x as a symbolic variable

### Get the nondimensional solution
eta, f0, f1, f2 = falkner_skan(m=m) # each returned value is a ndarray

### Dimensionalize the solution
ue = Vinf * (x / c) ** m
y = eta / (
    sp.sqrt(
        ((m + 1) / 2) * (ue / (nu * x))
    )
)
u = f1 * ue

### Get parameters of interest
Re_local = ue * x / nu
dFS = sp.sqrt(nu * x / ue)

theta_over_dFS = np.trapz(
    f1 * (1 - f1),
    dx=eta[1]
) * np.sqrt(2 / (m + 1))
dstar_over_dFS = np.trapz(
    (1 - f1),
    dx=eta[1]
) * np.sqrt(2 / (m + 1))
H = dstar_over_dFS / theta_over_dFS

dudy_wall = u[1] / y[1]  # approximately
Cf = nu * dudy_wall / (0.5 * ue ** 2)

### Plot parameters of interest
plt.ion()

```

```python
# Generate discrete values of parameters
x_over_c = np.linspace(1 / 100, 1, 100)
theta_d = sp.lambdify(x, theta_over_dFS * dFS, "numpy")(
x_over_c * c)
H_d = sp.lambdify(x, H, "numpy")(x_over_c * c)
Cf_d = sp.lambdify(x, Cf, "numpy")(x_over_c * c)

# Plot it
fig, ax = plt.subplots()

plt.subplot(311)
plt.plot(x_over_c, theta_d)
plt.title(r"$\theta$ Distribution")
plt.xlabel(r"$x/c$")
plt.ylabel(r"$\theta$ (unitless)")
plt.grid(True)

plt.subplot(312)
plt.plot(x_over_c, np.tile(H_d, len(x_over_c)))
plt.title(r"$H$ Distribution")
plt.xlabel(r"$x/c$")
plt.ylabel(r"$H$ (unitless)")
plt.grid(True)

plt.subplot(313)
plt.plot(x_over_c, Cf_d)
plt.title(r"$C_f$ Distribution")
plt.xlabel(r"$x/c$")
plt.ylabel(r"$C_f$ (unitless)")
plt.grid(True)

```