```python
# -*- coding: utf-8 -*-
import math
import matplotlib.pyplot as plt
import numpy as np

# Given values that won't change in the main loop
R = 287
P_amb = 26.500
T_amb = 223.252
P_exit = P_amb
h_c = 43000
Fst = 0.06

def Velocity(M,Gamma,T,R):
    u = M*math.sqrt(Gamma*R*T)
    return u
def m_flow(density,u,A):
    m_flow = density*u*A
    return m_flow
def TfromTo(To, Gamma, M):
    T = To/(1+((Gamma-1)/2)*M**2)
    return T
def TofromT(T, Gamma, M):
    To = T*(1+((Gamma-1)/2)*M**2)
    return To
def rho(P, R, T):
    density = P/(R*.001*T)
    return density
def FfromCp(To_amb, To_max, h_c, Cp, Fst):
    F = ((To_max/To_amb)-1)/(((h_c)/(Cp*0.001*To_amb))-(To_max/To_amb))
    if F>Fst:
        F = Fst
    return F
def calcCp(Gamma,R):
    Cp = (Gamma/(Gamma-1))*R
    return Cp
def m_spec(m_e, F):
    m_a = m_e/(1+F)
    return m_a
def calcThrust(m_amb, F, u_in, u_out):
    thrust = m_amb*((1+F)*u_out-u_in)
    return thrust
def calcTo_max (To_amb,h_c,F,Cp):
    To_max = (F*h_c/(Cp*0.001)+To_amb)/(1+F)
    return To_max
def propeffic(u_in, u_out, F, h_c):
    np = (2*(u_in/u_out))/(1+(u_in/u_out))
#    np = (I*u_in)/((1+F)*((u_out**2)/2)-((u_in**2)/2))
    return np
def thermeffic(u_in, u_out, To_max, T_amb):
    nth = ((u_out**2)-(u_in**2))/(2*Cp*(To_max-To_amb))
#    nth = ((1+F)*(u_out**2)-(u_in**2))/(2*F*h_c)
    return nth
```

```
def arearatio(M): #I used Gamma = 1.33 just to simplify trying to type this equation
    A_ratio = (1/M)*((2/2.33)*(1+(0.33/2)*M**2))**(2.33/0.66)
    return A_ratio

# Initialize the empty lists that will hold our graph data
Mlist = []
Ilist = []
TSFClist = []
To_maxlist = []
nthlist = []
nplist = []
nolist = []
Flist = []
Cplist = []
Tamblist = []
Toamblist = []
Texitlist = []
A_ratiolist = []

# Get the range of Mach numbers to iterate the program through
start = float(raw_input('Please enter starting Mach Number: '))
end = float(raw_input('Please enter ending Mach Number: '))
# Calculate needed values based on given values, with 0.01 step in M to smooth graphs
for M in np.arange(start,end,0.01):
    # Given values that need to be reset each iteration
    To_max = 2600
    Gamma = 1.40
    To_amb = TofromT(T_amb, Gamma, M)
    u_inlet = Velocity(M, Gamma, T_amb, R)
    # Getting to the combustor Gamma changes to 1.33
    Gamma = 1.33
    Cp = calcCp(Gamma, R)
    F = FfromCp(To_amb, To_max, h_c, Cp, Fst)
    if F == Fst:
        To_max = calcTo_max(To_amb, h_c, F, Cp)
    T_exit = TfromTo(To_max, Gamma,M)
    density_exit = rho(P_exit,R,T_exit)
    u_exit = Velocity(M, Gamma, T_exit, R)
    A_ratio = arearatio(M)
#    m_exit = m_flow(density_exit,u_exit,A_ratio)
#    m_amb = m_spec(m_exit,F)
#    thrust = calcThrust(m_amb, F, u_inlet, u_exit)
#    I = thrust/m_amb
    I =(1+F)*u_exit-u_inlet
    TSFC = F/((1+F)*u_exit-u_inlet)
    np = propeffic(u_inlet, u_exit, F, h_c)
    nth = thermeffic(u_inlet, u_exit, To_max, To_amb)
    no = np*nth

    # Write our calculations to the end of the list so we can graph them after the loop
    A_ratiolist.append(A_ratio)
    Mlist.append(M)
    Ilist.append(I)
```

```
    TSFClist.append(TSFC)
    To_maxlist.append(To_max)
    nthlist.append(nth)
    nplist.append(np)
    nolist.append(no)
    Cplist.append(Cp)
    Flist.append(F)
    Tamblist.append(T_amb)
    Toamblist.append(To_amb)
    Texitlist.append(T_exit)

#Now to plot these values!
plt.figure(1)
plt.figure(1).subplots_adjust(hspace=0.35)
plt.subplot(211)
plt.plot(Mlist, Ilist)
plt.xlabel('Mach Number, M')
plt.ylabel('Specific Thrust, I')
plt.title('Specific Thrust vs Mach Number')

plt.figure(1)
plt.subplot(212)
plt.plot(Mlist, TSFClist)
plt.xlabel('Mach Number, M')
plt.ylabel('TSFC')
plt.title('TSFC vs Mach Number')

plt.figure(2)
plt.figure(2).subplots_adjust(hspace=0.35)
plt.subplot(111)
plt.plot(Mlist, To_maxlist)
plt.xlabel('Mach Number, M')
plt.ylabel('To4')
plt.title('To4 vs Mach Number')

plt.figure(3)
plt.figure(3).subplots_adjust(hspace=0.6)
plt.subplot(311)
plt.plot(Mlist, nthlist)
plt.xlabel('Mach Number, M')
plt.ylabel('nth')
plt.title('nth vs Mach Number')

plt.figure(3)
plt.subplot(312)
plt.plot(Mlist, nplist)
plt.xlabel('Mach Number, M')
plt.ylabel('np')
plt.title('np vs Mach Number')

plt.figure(3)
plt.subplot(313)
plt.plot(Mlist, nolist)
```

```
plt.xlabel('Mach Number, M')
plt.ylabel('no')
plt.title('no vs Mach Number')

plt.figure(4)
plt.subplot(111)
plt.plot(Mlist, A_ratiolist)
plt.xlabel('Mach Number, M')
plt.ylabel('A/A*')
plt.title('A/A* vs Mach Number')

plt.figure(5)
plt.subplot(111)
plt.plot(Mlist, Flist)
plt.xlabel('Mach Number, M')
plt.ylabel('F')
plt.title('F vs Mach Number')

plt.show()
```