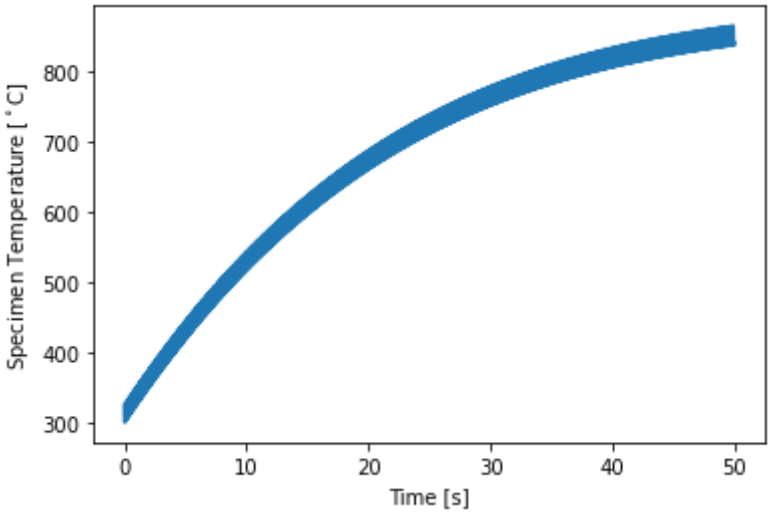
In [8]:

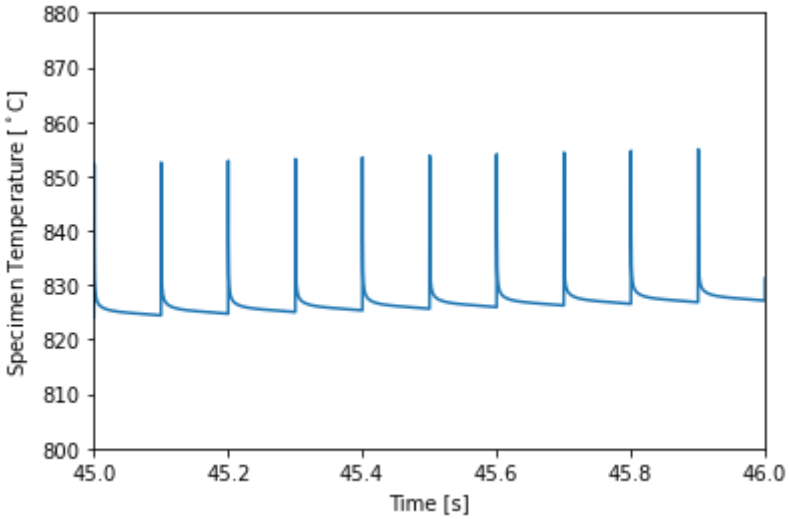
```
import numpy as np
from scipy.sparse import dia_matrix, identity
import matplotlib.pyplot as plt
import matplotlib.animation as animation
from IPython import display
import time as TIME

t1 = TIME.time()

''' 定数 '''
time = 50                        # time : 時間 (s)
time_point = 10000001           # time_point : 時間方向にとる点の数 = len(t)
x_L = 1.2e-3                        # x_L : x方向の長さ(試料+ダミー試料) (m)
x_L2 = 2.0e-3                   # x_L2 : x方向の長さ(銅板) (m)
n = 35                    # x_point = n : x方向にとる点の数

dt = time / (time_point-1) #(s)
dx = x_L / (n-1)             #(m)
dx2 = x_L2 / (n-1)             #(m)

k = 174                         #熱伝導率(W/m K)
rho = 19300                      #密度 (kg/m^3)
c = 138                     #比熱 (J/kg K)
alpha = k/(rho*c)         #熱拡散率(m^2/s)
r = alpha * dt / (dx)**2

k2 = 398                         #熱伝導率(W/m K)
rho2 = 8900                      #密度 (kg/m^3)
c2 = 385                     #比熱 (J/kg K)
alpha2 = k2/(rho2*c2)          #熱拡散率(m^2/s)
r2 = alpha2 * dt / (dx2)**2

s = 5.67e-8

T0 = 298.15   #初期温度 (K)
qL = 38e6       #レーザーの熱流束 (W/m^2)
#qL = 0
qP = 84e3    #プラズマの熱流束 (W/m^2)

Tinf = 298.15  #冷却水の温度 (K)
h = 500 #熱伝達率 (W/m^2 K)


print(alpha * dt / dx**2)
print(alpha2 * dt / dx2**2)
''' 行列の作成 '''

""" x """
x = np.zeros(n)
for i in range(n):
    x[i] = dx*i

x2 = np.zeros(n)
for i in range(n):
    x2[i] = dx2*i

""" t """
t = np.zeros(time_point)
for i in range(time_point):
    t[i] = dt*i
```

```
""" q """
q = np.zeros(time_point)

for i in range(time_point):
    if t[i]*1000 % 100 <= 0.2:
        q[i] = qL + qP
    else:
        q[i] = qP


""" matrix A """
data = np.array([np.ones(n), -2.0*np.ones(n), np.ones(n)])

data[2][1] = 0
data[0][n-2] = 0
data[1][0] = 0
data[1][n-1] = 0

offsets = np.array([-1, 0, 1])
B = dia_matrix((data, offsets), shape=(n, n))
E = identity(n)

A = E + r * B
A2 = E + r2 * B

""" T """
T = np.zeros(n)
T = T + T0

T2 = np.zeros(n)
T2 = T2 + T0

''' 準備 '''
print("dt = {}".format(dt))
print("dx = {}".format(dx))
print("dx2 = {}".format(dx2))
print('qP = {:.2e}'.format(qP))
print('qL = {:.2e}'.format(qL))
print('h = {:.2e}'.format(h))

print(alpha * dt / dx**2)
if alpha * dt / dx**2 > 0.5:
    print('安定ではありません')

print(alpha2 * dt / dx2**2)
if alpha2 * dt / dx2**2 > 0.5:
    print('安定ではありません')

''' main '''
Ts = []
TTT = []
ttt = []
for i in range(time_point):
    tmp0 = 2*dx*(q[i]-s*(T[0]**4))*r/k + (1-2*r)*T[0] + 2*r*T[1]
    #tmp0 = 2*dx*qP*r/k + (1-2*r)*T[0] + 2*r*T[1]
    tmp1 = k2*dx*(T2[1]-T2[0])/(k2*dx2) + T[n-2]
    tmpn = 2*r2*T2[n-2] + (1-2*r2)*T2[n-1] - (2*dx2*r2*h/k2)*(T2[n-1]-Tinf)

    T = A.dot(T)
    T2 = A2.dot(T2)

    T[0] = tmp0
    T[n-1] = tmp1
    T2[0] = tmp1
```

```
    T2[n-1] = tmpn

    Ts.append(T[0])

    if i == (time_point-1)/10:
        print('10%')
        t50 = TIME.time()
        print(t50-t1)
    #if i == time_point - 2:
        #aaa = T
        #bbb = T2

    if 99.799994 <= t[i] <= 99.800201:
        TTT.append(T)
        ttt.append(t[i])

plt.plot(t, Ts)
plt.xlabel("Time [s]")
plt.ylabel("Specimen Temperature [$¥mathrm{^¥circ C}$]")
plt.show()

#拡大図

plt.plot(t, Ts)
plt.xlim(45,46)
plt.ylim(800,880)
plt.xlabel("Time [s]")
plt.ylabel("Specimen Temperature [$¥mathrm{^¥circ C}$]")
plt.show()

t2 = TIME.time()

print('計算時間 : {} s'.format(t2-t1))
```

```
0.26222747866136026
0.1678418211002481
dt = 5e-06
dx = 3.529411764705882e-05
dx2 = 5.882352941176471e-05
qP = 8.40e+04
qL = 3.80e+07
h = 5.00e+02
0.26222747866136026
0.1678418211002481
10%
44.777206897735596
```





計算時間：274.1765944957733 s

In [ ]: