数値解析 レポート No. 15　　学籍番号　　氏名

[〆切] 2021/2/3 19:00

[**XIV**. 偏微分方程式の数値解法]

XIV-B. XIV-A で求めた 温度分布 $T(x,y)$ を初期条件 $T(x,y,t=0)$ として鉄板が断熱された環境に移された場合を考える。時間の経過とともに熱が拡散して一様な温度分布に近づいていく様子を数値的に求める。解くべき方程式は 2 次元の熱拡散方程式 $\frac{\partial T(x,y,t)}{\partial t} = a\left\{\frac{\partial^2 T(x,y,t)}{\partial x^2} + \frac{\partial^2 T(x,y,t)}{\partial y^2}\right\}$ $(0 \le x \le 1, 0 \le y \le 1)$ である。ここで a は温度伝導率であり $a=1$ とする。両辺を XIV-A と同じ格子点を用いて離散化し、オイラー法を用い時間幅を $\Delta t = 0.001$ として $t = 0.001, 0.01, 0.1, 1$ での温度分布を求めるプログラムを作成せよ。温度分布は XIV-A と同様に数値で出力するか等高線を用いた図を作成して示すこと。作成したプログラムも提出すること。プログラミング言語は問わない。

熱拡散方程式の両辺を差分化すると $t \ge 0$, $h \le x \le Nh$, $h \le y \le Nh$ において

$$(左辺) = \frac{T(x,y,t+\Delta t) - T(x,y,t)}{\Delta t}$$

$$(右辺) = \frac{\frac{T(x+h,y,t)-T(x,y,t)}{h} - \frac{T(x,y,t)-T(x-h,y,t)}{h}}{h} + \frac{\frac{T(x,y+h,t)-T(x,y,t)}{h} - \frac{T(x,y,t)-T(x,y-h,t)}{h}}{h}$$

$$= \frac{1}{h^2}\left\{T(x+h,y,t) + T(x-h,y,t) + T(x,y+h,t) + T(x,y-h,t) - 4T(x,y,t)\right\}$$

$$\therefore\ T(x,y,t+\Delta t) = T(x,y,t) + \frac{\Delta t}{h^2}\left\{T(x+h,y,t) + T(x-h,y,t) + T(x,y+h,t) + T(x,y-h,t) - 4T(x,y,t)\right\}$$

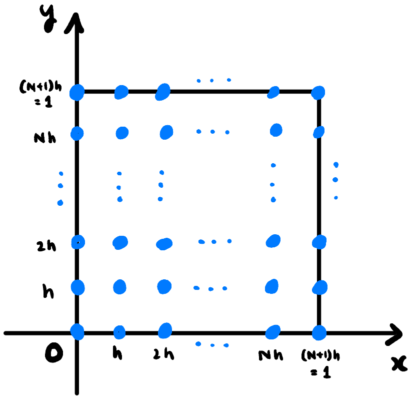
$$= \left\{1 - \frac{4\Delta t}{h^2}\right\} T(x,y,t) + \frac{\Delta t}{h^2}\left\{T(x+h,y,t) + T(x-h,y,t) + T(x,y+h,t) + T(x,y-h,t)\right\} \quad \cdots ①$$

となる。また 断熱されている条件は $x=0,1$ において $\frac{\partial T(x,y,t)}{\partial x} = 0$,

$y=0,1$ において $\frac{\partial T(x,y,t)}{\partial y} = 0$ と表せる。

$x=0$ における条件を差分化すると

$$\frac{T(h,y,t) - T(0,y,t)}{h} = 0 \iff T(0,y,t) = T(h,y,t) \quad \cdots ②$$



となる。$x=1$, $y=0,1$ における条件についても同様に差分化すると

$$T(x,0,t) = T(x,h,t) \quad \cdots ③$$

$$T((N+1)h,y,t) = T(Nh,y,t) \quad \cdots ④$$

$$T(x,(N+1)h,t) = T(x,Nh,t) \quad \cdots ⑤$$

となる。

②～⑤の条件のうち鉄板の四隅の点 $(x, y) = (0, 0), (0, (N+1)h), ((N+1)h, 0), ((N+1)h, (N+1)h)$ ではそれぞれ2つの条件が存在し、

$$\left.\begin{aligned}
T(0,0,t) &= \begin{cases} T(0,h,t) \\ T(h,0,t) \end{cases} \\
T(0,(N+1)h,t) &= \begin{cases} T(0,Nh,t) \\ T(h,(N+1)h,t) \end{cases} \\
T((N+1)h,0,t) &= \begin{cases} T(Nh,0,t) \\ T((N+1)h,h,t) \end{cases} \\
T((N+1)h,(N+1)h,t) &= \begin{cases} T((N+1)h,Nh,t) \\ T(Nh,(N+1)h,t) \end{cases}
\end{aligned}\right\} \quad ⑥$$

である。①～⑤において鉄板の四隅の点 $(x, y) = (0, 0), (0, (N+1)h), ((N+1)h, 0), ((N+1)h, (N+1)h)$ での $T(x,y,t)$ を含む条件は存在しないため⑥で示した条件はどれを選んでも影響は鉄板の四隅の点のみで閉じているため⑥の条件の選び方によって系全体のふるまいが変化することはない。ここでは⑥で得られる2つずつの条件を半分ずつ加味した以下の条件を考える。

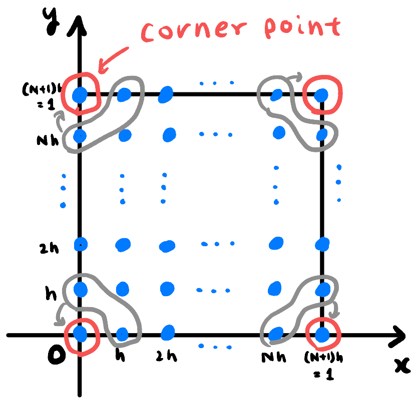
(T_corner_points_update (N, T))

$$T(0,0,t) = \frac{1}{2}\{T(0,h,t) + T(h,0,t)\}$$

$$T(0,(N+1)h,t) = \frac{1}{2}\{T(0,Nh,t) + T(h,(N+1)h,t)\}$$

$$T((N+1)h,0,t) = \frac{1}{2}\{T(Nh,0,t) + T((N+1)h,h,t)\}$$

$$T((N+1)h,(N+1)h,t) = \frac{1}{2}\{T((N+1)h,Nh,t) + T(Nh,(N+1)h,t)\}$$



四隅の点以外の鉄板の辺上の点では②～⑤の条件が成り立つ。

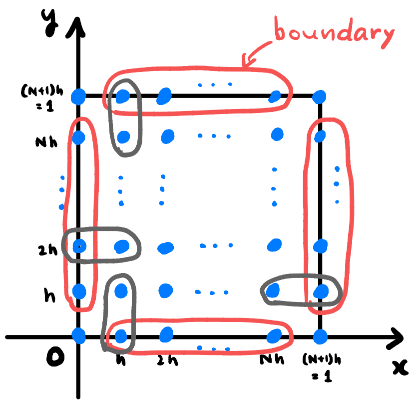
(T_boundary_update (N, T))

$$T(0,y,t) = T(h,y,t)$$

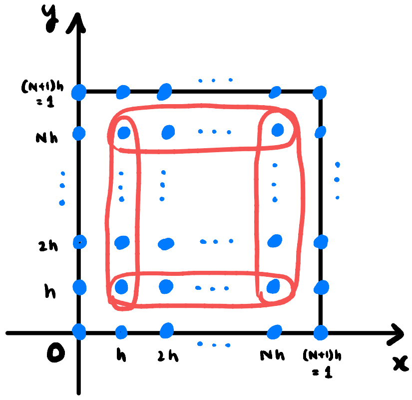
$$T(x,0,t) = T(x,h,t)$$

$$T((N+1)h,y,t) = T(Nh,y,t)$$

$$T(x,(N+1)h,t) = T(x,Nh,t)$$



鉄板の辺の1つ内側の行と列を考える。



②〜⑤の両辺を入れかえた式より

$$T(h,y,t) = T(0,y,t) \quad \cdots ②'$$

$$T(x,h,t) = T(x,0,t) \quad \cdots ③'$$

$$T(Nh,y,t) = T((N+1)h,y,t) \quad \cdots ④'$$

$$T(x,Nh,t) = T(x,(N+1)h,t) \quad \cdots ⑤'$$

である。まず $x=h,\ h \le y \le Nh$ の列について ① は

$$T(h,y,t+\Delta t) = \left\{1 - \frac{4\Delta t}{h^2}\right\} T(h,y,t) + \frac{\Delta t}{h^2}\left\{T(2h,y,t) + \underbrace{\overset{②'\text{より}}{T(0,y,t)}}_{=T(h,y,t)} + T(h,y+h,t) + T(h,y-h,t)\right\}$$

$$= \left\{1 - \frac{3\Delta t}{h^2}\right\} T(h,y,t) + \frac{\Delta t}{h^2}\left\{T(2h,y,t) + T(h,y+h,t) + T(h,y-h,t)\right\} \quad \cdots ⑦$$

となる。次に $h \le x \le Nh,\ y=h$ では

$$T(x,h,t+\Delta t) = \left\{1 - \frac{4\Delta t}{h^2}\right\} T(x,h,t) + \frac{\Delta t}{h^2}\left\{T(x+h,h,t) + T(x-h,h,t) + T(x,2h,t) + \underbrace{\overset{③'\text{より}}{T(x,0,t)}}_{=T(x,h,t)}\right\}$$

$$= \left\{1 - \frac{3\Delta t}{h^2}\right\} T(x,h,t) + \frac{\Delta t}{h^2}\left\{T(x+h,h,t) + T(x-h,h,t) + T(x,2h,t)\right\} \quad \cdots ⑧$$

同様に $x=Nh,\ h < y \le Nh$ において

$$T(Nh,y,t+\Delta t) = \left\{1 - \frac{4\Delta t}{h^2}\right\} T(Nh,y,t) + \frac{\Delta t}{h^2}\left\{\underbrace{\overset{④'\text{より}}{T((N+1)h,y,t)}}_{=T(Nh,y,t)} + T((N-1)h,y,t) + T(Nh,y+h,t) + T(Nh,y-h,t)\right\}$$

$$= \left\{1 - \frac{3\Delta t}{h^2}\right\} T(Nh,y,t) + \frac{\Delta t}{h^2}\left\{T((N-1)h,y,t) + T(Nh,y+h,t) + T(Nh,y-h,t)\right\} \quad \cdots ⑨$$

最後に $h \le x < Nh,\ y=Nh$ において

$$T(x,Nh,t+\Delta t) = \left\{1 - \frac{4\Delta t}{h^2}\right\} T(x,Nh,t) + \frac{\Delta t}{h^2}\left\{T(x+h,Nh,t) + T(x-h,Nh,t) + \underbrace{\overset{⑤'\text{より}}{T(x,(N+1)h,t)}}_{=T(x,Nh,t)} + T(x,(N-1)h,t)\right\}$$

$$= \left\{1 - \frac{3\Delta t}{h^2}\right\} T(x,Nh,t) + \frac{\Delta t}{h^2}\left\{T(x+h,Nh,t) + T(x-h,Nh,t) + T(x,(N-1)h,t)\right\} \quad \cdots ⑩$$

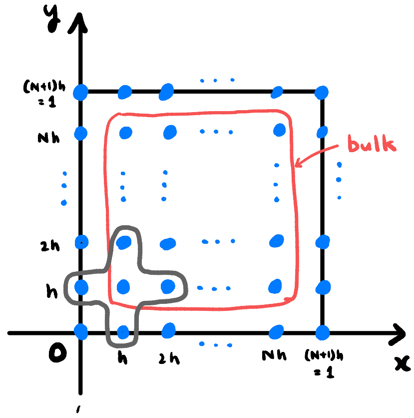
と表される。これより内側の点では ① の条件のみである。従って

①, ⑦〜⑩ でバルク部分の条件がそろった。

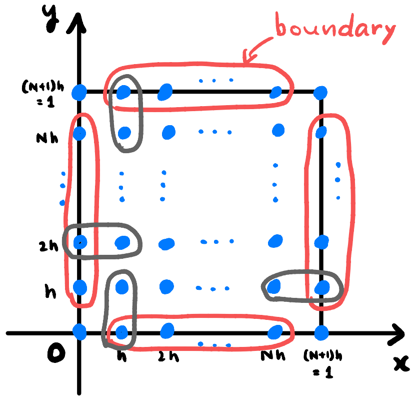
↳ T_bulk_update (N, T, T_next)

コードは以下のようになる.

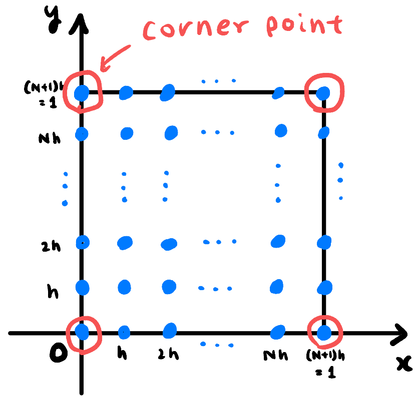
```
import numpy as np
import matplotlib.pyplot as plt

def T_bulk_update(N, T, T_next):
    for i in range(1, N+1):
        for j in range(1, N+1):
            T_tmp = 0; count = 0
            if i >= 2:
                T_tmp += T[i-1, j]; count += 1
            if i <= N - 1:
                T_tmp += T[i+1, j]; count += 1
            if j >= 2:
                T_tmp += T[i, j-1]; count += 1
            if j <= N - 1:
                T_tmp += T[i, j+1]; count += 1
            T_next[i, j] = (1 - count * (dt/h**2)) * T[i,j] + (dt/h**2) * T_tmp
    for i in range(1, N+1):
        for j in range(1, N+1):
            T[i, j] = T_next[i, j]
```




```
def T_boundary_update(N, T, T_next):
    for i in range(1, N+1):
        T[i, 0] = T_next[i, 1]
        T[i, N+1] = T_next[i, N]
    for j in range(1, N+1):
        T[0, j] = T_next[1, j]
        T[N+1, j] = T_next[N, j]

def T_corner_points_update(N, T):
    T[0, 0] = (T[0, 1] + T[1, 0])/2.
    T[0, N+1] = (T[0, N] + T[1, N+1])/2.
    T[N+1, 0] = (T[N+1, 1] + T[N, 0])/2.
    T[N+1, N+1] = (T[N+1, N] + T[N, N+1])/2.
```



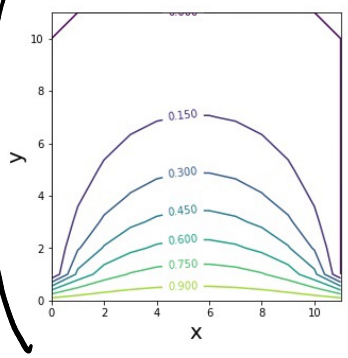

```
def T_plot(T):
    fig = plt.figure(figsize=(5, 5))
    ax = fig.add_subplot(111)
    cntr = ax.contour(T.transpose())
    ax.clabel(cntr)
    ax.set_aspect('equal')
    ax.set_xlabel('x', fontsize=20)
    ax.set_ylabel('y', fontsize=20)
    plt.show()
```



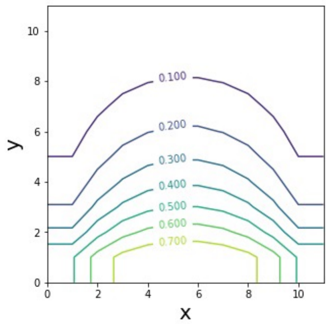

```
N = 10
T = np.zeros((N+2)**2)
T = T.reshape(N+2, N+2) # T[X, Y]
T[:, 0] = 1

for k in range(200):
    for i in range(1, N+1):
        for j in range(1, N+1):
            T[i, j] = 0.25 * (T[i-1, j] + T[i+1, j] + T[i, j-1] + T[i, j+1])

print("t = 0")
T_plot(T)

h = 1/(N+1)
dt = 0.001
```

```
T_next = np.zeros((N+2)**2)
T_next = T_next.reshape(N+2, N+2) # T_next[X, Y]
for k in range(1001):
    T_bulk_update(N, T, T_next)
    T_boundary_update(N, T, T_next)
    T_corner_points_update(N, T)
    if k == 0:
        print("t = ", dt * (k+1))
        T_plot(T)
    elif k == 9:
        print("t = ", dt * (k+1))
        T_plot(T)
    elif k == 99:
        print("t = ", dt * (k+1))
        T_plot(T)
    elif k == 999:
        print("t = ", dt * (k+1))
        T_plot(T)
```

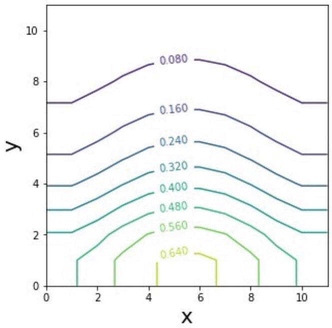
t = 0



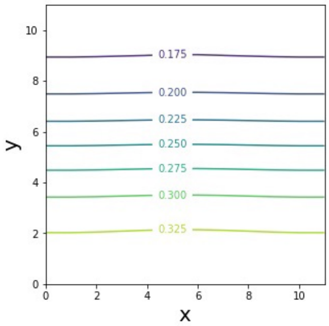
t = 0.001



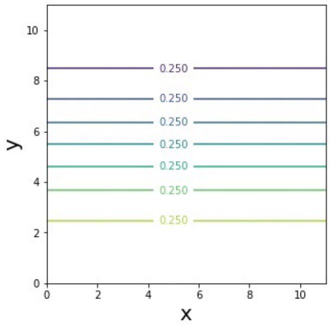
t = 0.01



t = 0.1

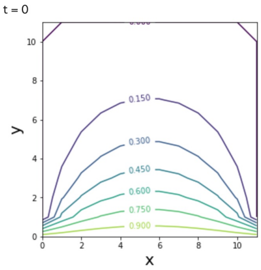


t = 1

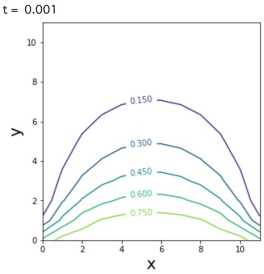


おまけ　境界の温度をすべて0にした場合

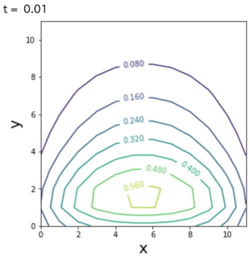
t = 0



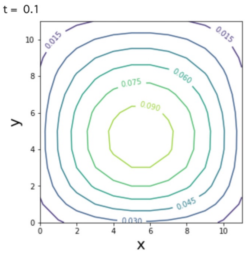
t = 0.001



t = 0.01



t = 0.1



t = 1.0