

科学計算研究室 Python ゼミ

～ 11. 偏微分方程式 その1～

2021-04-12 福田 浩

1 原理

偏微分方程式は，下記のようにして解くことが出来る．まず，微小区間 h に対して $u_{i,j}$ の前後の格子点上の関数値 $u_{i-1,j}$, $u_{i+1,j}$ を用いて差分近似を考える．

$$\left(\frac{\partial u}{\partial x}\right)_{i,j} = \frac{u_{i,j} - u_{i-1,j}}{h} \quad (1)$$

$$\left(\frac{\partial u}{\partial x}\right)_{i,j} = \frac{u_{i+1,j} - u_{i,j}}{h} \quad (2)$$

$$\left(\frac{\partial u}{\partial x}\right)_{i,j} = \frac{u_{i+1,j} - u_{i-1,j}}{2h} \quad (3)$$

式 1 を後退差分近似，式 2 を前進差分近似，式 3 を中央差分近似と呼ぶ．
2 次の偏導関数は，さらにこの考え方を進めて，

$$\left(\frac{\partial^2 u}{\partial x^2}\right)_{i,j} = \frac{(\partial u / \partial x)_{i,j} - (\partial u / \partial x)_{i-1,j}}{h} \quad (4)$$

$$= \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{h^2} \quad (5)$$

で示すことが出来る．

2 放物型偏微分方程式

$$\frac{\partial u}{\partial t} = a \frac{\partial^2 u}{\partial x^2} \quad (6)$$

のタイプの偏微分方程式を，放物型偏微分方程式と呼ぶ．代表的な例として，1 次元の熱伝導方程式がある．1 次元の熱伝導方程式は，下記の通りである．

$$\rho C \frac{\partial T}{\partial t} = \lambda \frac{\partial^2 T}{\partial x^2} \quad (7)$$

ここで， T は温度 (K), x は位置 (m), ρ は密度 (kg/m³), C は比熱 (J/(kg·K)), λ は熱伝導率 (W/(m·K)) である．これを差分化すると，

$$\frac{T_i^{N+1} - T_i^N}{\Delta t} = \frac{\lambda}{\rho C} \left\{ \frac{T_{i+1}^N - 2T_i^N + T_{i-1}^N}{\Delta x^2} \right\} \quad (8)$$

$$T_i^{N+1} = T_i^N + \frac{\lambda \Delta t}{\rho C} \left\{ \frac{T_{i+1}^N - 2T_i^N + T_{i-1}^N}{\Delta x^2} \right\} \quad (9)$$

ここで上の添え字は時間 t を表し，下の添え字は位置 x を表す．

3 課題

放物型偏微分方程式の数値解法プログラムを Python で実装せよ

- 300K に熱した長さ 1m の鉄の棒の両端を 100K の熱浴につけた時の、鉄棒全体の温度の時間変化を出力せよ
- 以下の物性値を使え
 - 密度 ρ : 7874 kg/m³
 - 比熱 C : 461 J/(kg·K)
 - 熱伝導率 λ : 80.4 W/(m·K)
- 長さ方向には 10 分割し、時間変化は 0~100 秒まで計算せよ。

4 参考1: C++ソースコード

```
1  #include <iostream>
2  #define N 10
3  using namespace std;
4
5  int main(void){
6
7      double u_old[N+1], u_new[N+1], lamb, rho, c, dx;
8
9      lamb = 80.4;
10     rho = 7874;
11     c = 461;
12     dx = 0.1;
13
14     for(int i=1; i<N; i++) u_old[i] = 1000;
15     u_old[0]=300;
16     u_old[N]=300;
17
18     for(int t=0;t<3600;t++){
19         for(int i=1;i<N;i++){
20             u_new[i] = u_old[i] + lamb/(rho*c*dx*dx)*(u_old[i+1] - 2*u_old[i] + u_old[i-1]);
21         }
22         for(int i=1; i<N; i++){
23             u_old[i] = u_new[i];
24         }
25         if((t%10==0)&&(t%60==0)) {
26             cout << t << "\t";
27             for(int i=0; i<N+1; i++) cout << u_old[i] << "\t";
28             cout << endl;
29         }
30     }
31     return 0;
32 }
```