数値計算の理論と実際

第7回 数値積分

1 台形公式

定積分

$$\int_a^b f(x)dx \tag{1}$$

を数値的に計算することを考える。分点 $a = x_0 < x_1 < \cdots < x_n = b$ に対して、

$$\int_a^b f(x)dx = \int_{x_0}^{x_1} f(x)dx + \int_{x_1}^{x_2} f(x)dx + \cdots \int_{x_{n-1}}^{x_n} f(x)dx \tag{2}$$

である。$f(x)$ を各区間ごとに簡単に積分できる関数で近似して積分し、それを足しあわせて定積分の近似値とするのが基本的なアイデアである。

各区間毎に直方体で近似するのが**区分求積法**である。それよりは1次関数、つまり台形で近似する方がよさそうである。k 番目の区間の面積は

$$\frac{1}{2}\left\{f(x_{k-1}) + f(x_k)\right\}(x_k - x_{k-1}) \tag{3}$$

である。したがって

$$\int_a^b f(x)dx = \frac{1}{2}\sum_{k=1}^{n}\left\{f(x_{k-1}) + f(x_k)\right\}(x_k - x_{k-1}) \tag{4}$$

である。これが**台形公式**である。特に区間幅が h で均等の場合、

$$\begin{aligned}\int_a^b f(x)dx &= \frac{h}{2}\sum\left\{f(x_{k-1}) + f(x_k)\right\} &(5)\\ &= \frac{h}{2}\left\{f(x_0) + 2\sum_{k=1}^{n-1} f(x_k) + f(x_n)\right\} &(6)\end{aligned}$$

となる。

1.1 練習問題

積分

$$\int_{-1}^{1}(1-x)e^{-x}dx \tag{7}$$

を台形公式で計算するプログラムを実装せよ。簡単なのでサンプルコードはない。真値は $3.086161\cdots$ であるが、区間数を $4, 8, 16\cdots$ と変更し、真値との誤差を計算せよ。

2　シンプソンの公式

台形公式は各区間を 1 次関数で近似した。同様に 2 つ分の区間を 2 次関数で近似することも考えられる。区間 $[a, b]$ を $2n$ 等分して、2 つの区間 $[x_{2i}, x_{2i+2}]$ ごとに、3 点

$$(x_{2i}, f_{2i}), \quad (x_{2i+1}, f_{2i+1}), \quad (x_{2i+2}, f_{2i+2})$$

通る 2 次関数を考える。点 x_{2i+1} を原点にとった、新しい座標系上で

$$Y = AX^2 + BX + C \tag{8}$$

とおくと、これは 3 点

$$(-h, f_{2i}), \quad (0, f_{2i+1}), \quad (h, f_{2i+2})$$

を通るので、

$$Ah^2 - Bh + C = f_{2i}, \quad C = f_{2i+1}, \quad Ah^2 + Bh + C = f_{2i+2}$$

となる。したがって

$$\begin{aligned}
\int_{x_{2i}}^{x_{2i+2}} f(x)dx &= \int_{-h}^{h} (AX^2 + BX + C)dX & (9)\\
&= \frac{h}{3}\left\{2(Ah^2 + C) + 4C\right\} & (10)\\
&= \frac{h}{3}\left\{f_{2i} + 4f_{2i+1} + f_{2i+2}\right\} & (11)
\end{aligned}$$

となり、すべての区間で合計すると定積分が計算される。これが**シンプソンの公式**である。

2.1　練習問題

式 (7) をシンプソン公式で積分するプログラムを実装し、台形公式と比較せよ。

3　モンテカルロ積分

乱数を用いて定積分を計算する方法である。円周率の計算を例にする。半径 1 の円の面積は π である。モンテカルロ積分では、$[0, 1)$ の乱数を 2 つ発生させ、それぞれ x, y 座標に見立てる。この空間には円の第 1 象限のみ含まれている。乱数が存在する領域の面積は 1 であるので、乱数を十分多く発生させれば、円の中に存在する乱数の数の割合は $\pi/4$ になってそうである。すなわち割合に 4 をかけると円周率になる。コードにすると次のようになる。c++ では drand48() 関数で $[0, 1)$ の一様乱数を生成することができる。

```
#include<iostream>
#include<cstdio>
#include<cstdlib>
#include<cmath>

using namespace std;

int main(){

  int n;
  cerr << "input n" << endl;
  cin >> n;
  cerr << n << endl;

  srand48( time(NULL));  #乱数の初期化

  int n2 = 0;
  for( int i=0; i<n; i++){
    double x = drand48();
    double y = drand48();
    double r2 = x*x + y*y;
    if( r2 < 1.0)  n2 ++;
  }

  cerr << n2 << "\t" << 4.0*n2/n << endl;

}
```

3.1 練習問題

式 (7) をモンテカルロ積分し、どれくらい乱数を増やせば、台形公式と同じくらいの精度を達成できるかについて調べよ。