連続系アルゴリズム
レポート課題6

連絡先：mail@myuuuuun.com

平成 27 年 11 月 30 日

レポート内で使用したプログラムは GitHub にアップロードしています。 https://github.com/myuuuuun/various/tree/master/ContinuousAlgorithm/HW6/

問題 1

3 点、5 次の Gauss 公式を求めよ。

3 点のガウス-ルジャンドル積分公式を求める。3 次までのルジャンドル多項式は、

- $P_0 = 1$
- $P_1 = x$
- $P_2 = \frac{3}{2}x^2 - \frac{1}{2}$
- $P_3 = \frac{5}{2}x^3 - \frac{3}{2}x$

であった（正規化を行った場合）。P_3 の零点は $x_1 = -\sqrt{\frac{5}{3}}, x_2 = 0, x_3 = \sqrt{\frac{5}{3}}$ の 3 点であるので、この 3 点を標本点とする。
次に各標本点での重みを計算する。選点直交性から、

$$\begin{aligned}\frac{1}{w_i} &= \sum_{r=0}^{2} \frac{P_r(x_i)^2}{(P_r \cdot P_r)} \\ &= \frac{1}{2} + \frac{3}{2}P_1(x_i)^2 + \frac{5}{2}P_2(x_i)^2\end{aligned}$$

であるので、

$$\begin{aligned}\frac{1}{w_1} = \frac{1}{w_3} &= \frac{1}{2} + \frac{3}{2}\left(\pm\sqrt{\frac{3}{5}}\right)^2 + \frac{5}{2}\left(\frac{2}{5}\right)^2 = \frac{9}{5} \\ \frac{1}{w_2} &= \frac{1}{2} + \frac{5}{2} - \left(\frac{1}{2}\right)^2 = \frac{9}{8}\end{aligned}$$

となる。
したがって、

$$\begin{aligned}x_1 &= -\sqrt{\frac{5}{3}}, x_2 = 0, x_3 = \sqrt{\frac{5}{3}} \\ w_1 &= \frac{5}{9}, w_2 = \frac{8}{9}, w_3 = \frac{5}{9}\end{aligned}$$

問題 2

適当な関数の積分値を DE 積分で求めよ。

$\int_{-1}^{1} x^2\,dx = \frac{3}{2}$ を DE 積分で計算する。積分範囲は-5〜5 で固定、ステップ幅は 2, 1, 0.5, ..., 2^{-8} とする。

ソースコード: hw6-1.cpp

```
#define _USE_MATH_DEFINES
#include <iostream>
#include <cmath>
#include <limits>
const double EPS = 1.0e-16;
using namespace std;

// [-1, 1] の範囲でDE 積分
double DE_integration1(double (*integrand)(double),
        double partition, double lower_b, double upper_b){
  double step = (upper_b - lower_b) * 1.0 / partition;
  double sum = 0;
  double f_inner;
  double point;
  for(int i=0; i<=partition; ++i){
    point = lower_b + step * i;
    f_inner = tanh(M_PI / 2 * sinh(point));
    sum += integrand(f_inner) * cosh(point)
                / pow(cosh(M_PI*sinh(point)/2), 2);
  }

  return sum * M_PI * step / 2;
}

double func1(double x){
  return x * x;
}

int main(){
  double f;
  cout.precision(std::numeric_limits<double>::max_digits10);

  for(int i=0; i<10; ++i){
    f = DE_integration1(func1, 5*pow(2, i), -5, 5);
    cout << f << endl;
  }

  return 0;
}
```

結果は [0.832774000015436, 0.41691937705316179, 0.6658855800450576, 0.66666666653570195, 0.66666666666666663, 0.66666666666666696, 0.66666666666666663, 0.66666666666666696, 0.66666666666666663, 0.66666666666666707] となって、h=0.25 付近で誤差が最小になった。

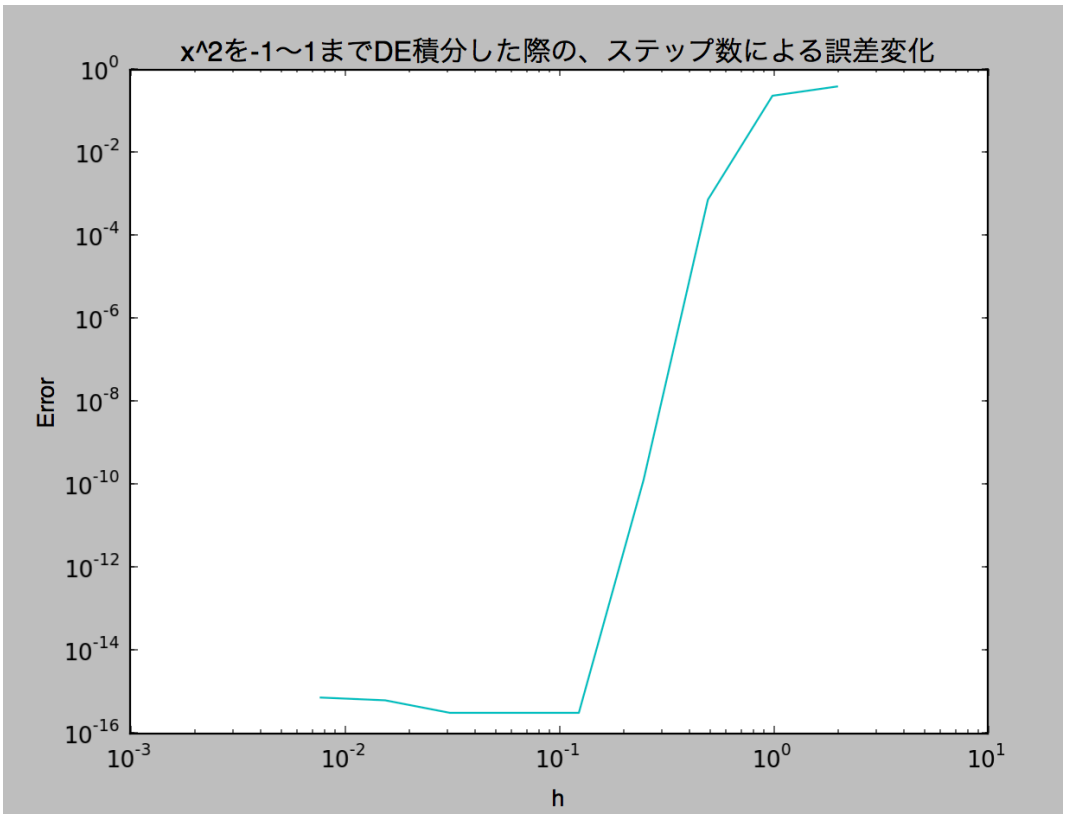
x^2を-1〜1までDE積分した際の、ステップ数による誤差変化
Error
h
10^0
10^-2
10^-4
10^-6
10^-8
10^-10
10^-12
10^-14
10^-16
10^-3
10^-2
10^-1
10^0
10^1

図 1: DE 積分の誤差