/* ---------- *

Simpson の公式

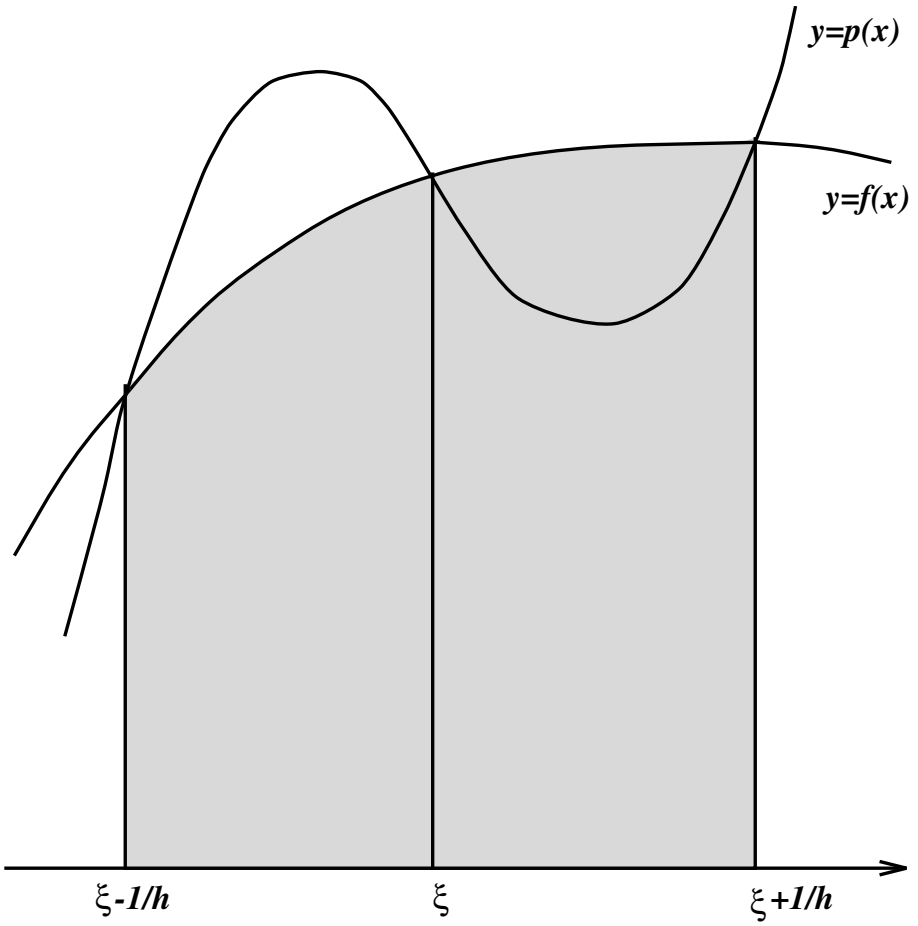
定積分 $\int_a^b f(x)\,dx$ を計算する一つの方法として、次の近似式が良く使われる。

$$\int_a^b f(x)\,dx \sim \frac{h}{3}\left(y_0 + 4y_1 + 2y_2 + 4y_3 + 2y_4 + 4y_5 + \cdots + 2y_{n-2} + 4y_{n-1} + y_n\right),$$

ここで、自然数 n は偶数とし、 $x_i = a + \frac{i}{n}(b-a)$, $y_i = f(x_i)$ $(i = 0, 1, 2, \cdots, n)$ とする。この公式の証明は、比較的やさしい。じっさい、 Taylor の定理から導かれる等式

$$\int_{\xi-\frac{1}{h}}^{\xi+\frac{1}{h}} f(x)\,dx \sim \int_{\xi-\frac{1}{h}}^{\xi+\frac{1}{h}} p(x)\,dx = \frac{h}{3}\left(f(\xi - \frac{1}{h}) + 4f(\xi) + f(\xi + \frac{1}{h})\right) + O(h^5)$$

を $\frac{n}{2}$ 個足し合わせることにより、この公式は証明される。



ここで、 $p(x)$ は次のような条件を満足する 2 次の多項式を表す:

$$p\left(\xi - \frac{1}{h}\right) = f\left(\xi - \frac{1}{h}\right),\ p(\xi) = f(\xi),\ p\left(\xi + \frac{1}{h}\right) = f\left(\xi + \frac{1}{h}\right).$$

A, B, F(X), M の定義をいろいろと変えて、各人で数値実験を行ってみよう。

```
* ____________________________________________________________________________ */

/* ____________________________________ simpson.c ____________________________ */

#include <stdio.h>
#define A 0.                            /* 積分をする区間 [a, b]                */
#define B 1.
#define F(X) ((X)*(X))                  /* 被積分関数 f(x)                      */
#define M 20                            /* 区間 [a, b] の分割数 n               */

double simpson_rule()                   /* Simpson の公式を用いて ∫_a^b f(x)dx を計算する関数 */
{
    long i;                             /* long 整数を使う                      */
    double h, s, y[M + 1];              /* メッシュ h, 積分値 s, 分点 i/n での関数値 y_i */

    h = 1. / (double) M;                /* h = 1/n                              */
    for (i = 0; i <= M; ++i)            /* y_i = f(i/n)   (i = 0, 1, 2, ..., n) */
        y[i] = F((double) i / (double) M);
                                        /* 以下は                               */
                                        /* s = h/3 (y_0 + 4y_1 + 2y_2 + 4y_3 + ... + 4y_{n-1} + y_n) */
                                        /* の計算                               */
    s = y[0];
    for (i = 1; i < M; ++i)
    {
        if (i % 2 == 1)                 /* ただし i % 2= i を 2 で割った余り    */
            s += 4. * y[i];
        else
            s += 2. * y[i];
    }
    s += y[M];
    s *= (h / 3.);
    return s;                           /* s の値を返す                         */
}

main()
{
    double s;

    s = simpson_rule();                 /* 上で定義した関数 simpson_rule() を使う */
    printf("%.16f\n", s);               /* s を小数点以下 16 桁表示する         */
}
```