

プログラミング演習1

— 課題4 数値解析法 —

3. 数値積分

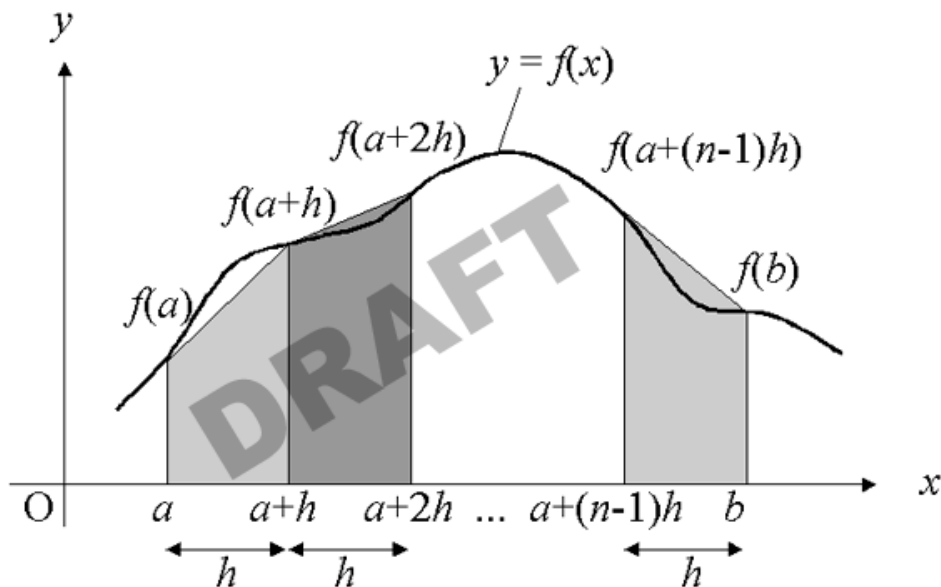
3.1 台形公式

台形公式の1区間分の近似式は

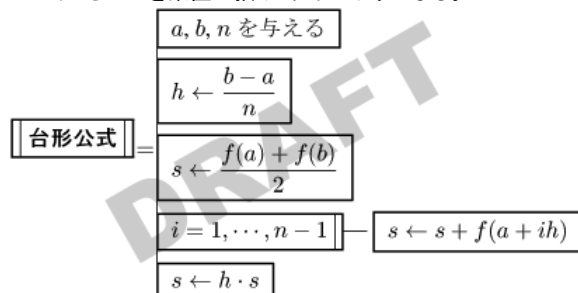
$$I = \int_a^b f(x) dx \approx (b-a)[f(a) + f(b)] / 2$$

積分範囲を n 分割したときの複合公式は

$$I_1 = h \left[\frac{f(a)+f(b)}{2} + \sum_{i=1}^{n-1} f(a+ih) \right], h = \frac{b-a}{n}$$



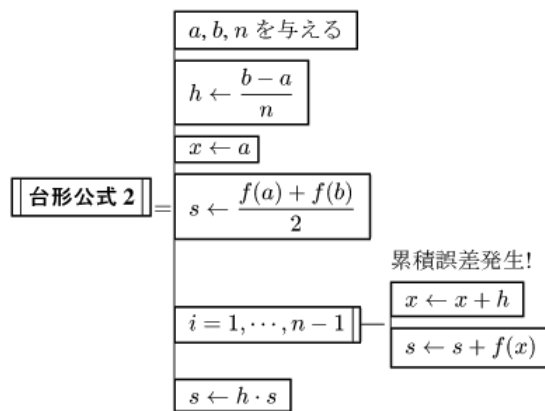
この式からPADを素直に描くと以下になる。



ここで、 $f(a+ih)$ のカッコ内の乗算をなくすために $x_i = a+ih$ とおき、

$$x_0 = a, \quad x_{i+1} = x_i + h$$

というように帰納的に計算すると、以下になる。



しかし、このアルゴリズムでは加算による丸め誤差の累積が発生しやすい。実際、台形公式による積分計算を単精度(float)で行うプログラム ([trapez_err0.c](#))を用いて、上記2つのアルゴリズムを比較してみる。ただし、アルゴリズムの主要部分は穴埋めする必要がある。

```

1: #include <stdio.h>
2: //define float double
3:
4: float f(float x)
5: {
6:     return 4.0 / (1 + x * x); // f(x) = 4/(1+x^2)
7: }
8:
9: float Trapez(float a, float b, float (*f)(float), int n)
10: /*
11:     累積誤差がない単精度の数値積分ルーチン（台形公式）
12: */
13: {
14:     int i;
15:     float h, s;
16:
17:     h = ____?____;
18:     s = ____?____;
19:     for (i = 1; i < n; i++) {
20:         s = ____?____; // 累積誤差はない
21:     }
22:     return h * s;
23: }
24:
25: float Trapez2(float a, float b, float (*f)(float), int n)
26: /*
27:     累積誤差がある単精度の数値積分ルーチン（台形公式）
28: */
29: {
30:     int i;
31:     float h, x, s;
32:
33:     h = ____?____;
34:     x = ____;
35:     s = ____?____;
36:     for (i = 1; i < n; i++) {
37:         x = ____?____; // 累積誤差発生要因
38:         s = ____?____;
39:     }
40:     return h * s;
41: }
42:
43: int main(void)
44: {
45:     int n;
46:     float a, b;
47:
48:     a = 0; b = 1;
49:     printf("# 分割数, 累積誤差なし, 累積誤差あり\n");
50:     for (n = 40; n <= 10000; n += 20)
51:         printf("%5d, % .15e, % .15e\n", n, Trapez(a, b, f, n), Trapez2(a, b, f, n));
52:     return 0;
53: }
  
```

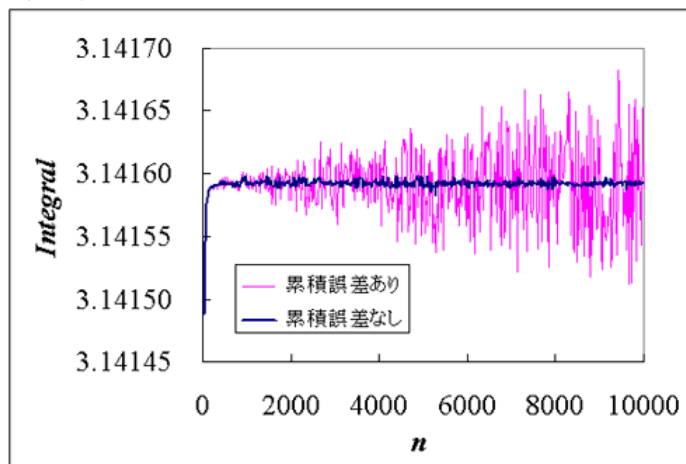
実行結果が長いので、VC++の開発環境で実行したときに スクロールバーを使って画面バッファの上の方にさかのぼっても 実行結果の最初の部分が消えてしまうことがある。そのときには、実行ウィンドウのタイトルバーを 右クリックして「プロパティ(P)」を選択し、次に「レイアウト」タブを選択し、「画面バッファのサイズ」欄の「高さ(H)」を 501 以上に設定する。「OK」をクリックし、「同じタイトルのウィンドウに適用する(S)」をチェックしたあと、再び「OK」をクリックし、[Enter]キーを 押していったん実行ウィンドウを閉じる。再び実行すれば、今度は実行結果がすべて残っているはずである。

「コマンドプロンプト」を使う場合は、trapez_err.exe ファイルがあるところ (例えば Z:\proen1\kadai4\trapez_err\Debug) まで cd コマンドなどで移動し、

Z:\proen1\kadai4\trapez_err\Debug>trapez_err > trapez_err.csv

とコマンドを実行すると、リダイレクトにより [CSVファイル](#) に出力される。

計算結果を示すと以下のようになる。



練習問題

- (1) 上記の結果を確認せよ。
- (2) [倍精度\(double\)](#)で計算し、単精度の結果と比較してみよ。

単精度から倍精度にするには

```
#define float double
```

を用いて置き換えると楽である。

[【目次】](#) | [【1.】](#) | [【2.】](#) | [【3.のつづき】](#) | [【付録1】](#) | [【付録2】](#) | [【付録3】](#)