

連分数と数値解析

情報数理工学・コンピュータサイエンス実験第二

Hidenori Ogata

Dept. Computer and Network Engineering, Univ. Electro-Communications

2019 年 11 月 6 日 (水)

1. 連分数

連分数 (continued fraction)

$$C = \frac{a_0}{b_0 + \frac{a_1}{b_1 + \frac{a_2}{b_2 + \ddots}}}$$

略記法 (のひとつ)

$$C = \left[\frac{a_0}{b_0} \right] + \left[\frac{a_1}{b_1} \right] + \left[\frac{a_2}{b_2} \right] + \dots$$

有限連分数 $C = \left[\frac{a_0}{b_0} \right] + \left[\frac{a_1}{b_1} \right] + \dots + \left[\frac{a_n}{b_n} \right],$

無限連分数 $C = \left[\frac{a_0}{b_0} \right] + \left[\frac{a_1}{b_1} \right] + \dots$

2. 有理数の連分数表示

(例) $163/31$ を連分数で表す. → **Euclid** の互除法

$$163 = 5 \times 31 + 8,$$

$$31 = 3 \times 8 + 7,$$

$$8 = 1 \times 7 + 1,$$

$$7 = 7 \times 1 \quad \text{終わり.}$$

(これより, 163 と 31 の最大公約数は 1 .)

上の計算から,

$$\begin{aligned}\frac{163}{31} &= 5 + \frac{8}{31} = 5 + \frac{1}{31/8} = 5 + \frac{1}{3 + \frac{7}{8}} \\ &= 5 + \frac{1}{3 + \frac{1}{8/7}} = 5 + \frac{1}{3 + \frac{1}{1 + \frac{1}{7}}}\end{aligned}$$

2. 有理数の連分数表示

単純連分数：次の形の連分数

$$k_0 + \cfrac{1}{\left\lfloor k_1 \right\rfloor} + \cfrac{1}{\left\lfloor k_2 \right\rfloor} + \cdots, \quad k_i \in \mathbb{Z}, \quad k_1, k_2, \dots > 0.$$

課題 1

- ① $5 + \cfrac{1}{\left\lfloor 3 \right\rfloor} + \cfrac{1}{\left\lfloor 1 \right\rfloor} + \cfrac{1}{\left\lfloor 7 \right\rfloor}$ が実際に $\frac{163}{31}$ に一致することを確認せよ.
- ② 任意に（正の）有理数を与えた時，その単純連分数表示を求めるプログラムを作成せよ
（ $k_0, k_1, \dots$ を求めるプログラムを作ればよい）.

3. 無理数の連分数表示

無限連分数

$$\begin{aligned}C &= \cfrac{a_0}{b_0} + \cfrac{a_1}{b_1} + \cfrac{a_2}{b_2} + \cdots \\&= \cfrac{a_0}{b_0 + \cfrac{a_1}{b_1 + \cfrac{a_2}{b_2 + \cdots}}}.\end{aligned}$$

無限連分数 C の第 n 近似分数

$$w_n = \cfrac{a_0}{b_0} + \cfrac{a_1}{b_1} + \cfrac{a_2}{b_2} + \cdots + \cfrac{a_n}{b_n}.$$

無限連分数 C の値：近似分数列の極限で定義する.

$$C \equiv \lim_{n \rightarrow \infty} w_n = \lim_{n \rightarrow \infty} \cfrac{a_0}{b_0} + \cfrac{a_1}{b_1} + \cfrac{a_2}{b_2} + \cdots + \cfrac{a_n}{b_n}.$$

3. 無理数の連分数表示

(例) $\sqrt{2}$ の連分数表示を求める.

$$\sqrt{2} = 1 + (\sqrt{2} - 1) \quad (\sqrt{2} \text{ の整数部分} + \text{小数部分})$$

$$= 1 + \frac{1}{1 + \sqrt{2}} = 1 + \frac{1}{2 + (\sqrt{2} - 1)}$$

($1 + \sqrt{2}$ の整数部分 + 小数部分)

$$= 1 + \frac{1}{2 + \frac{1}{1 + \sqrt{2}}} = 1 + \frac{1}{2 + \frac{1}{2 + \frac{1}{2 + (\sqrt{2} - 1)}}} = \dots,$$

$$\therefore \sqrt{2} = 1 + \frac{1}{2 + \frac{1}{2 + \ddots}}$$

3. 無理数の連分数表示

課題 2

- ① $(1 + \sqrt{5})/2$ の単純連分数表示を求めよ.
- ② 任意の (正の) 無理数に対しその単純連分数表示を求めるプログラムを作成せよ.
- ③ 前項のプログラムを用いて π の単純連分数表示を求めよ.
(注) π は C 言語では `M_PI` で与えられる.

(ヒント) C/C++では, 正の実数 (浮動小数点数) a の整数部分は `(int)a` で求められる.

```
double pi;  
int a;  
pi = M_PI;  
a = (int)pi;           → a = 3;  
a = (int)(-pi);        → a = - 3;
```

4. 連分数の計算（近似分数）

連分数を次のように計算するのは面倒である．

$$\begin{aligned} \frac{1}{1 + \frac{1}{2 + \frac{1}{3 + \frac{1}{4}}}} &= \frac{1}{1 + \frac{1}{2 + \frac{1}{\frac{13}{4}}}} = \frac{1}{1 + \frac{1}{2 + \frac{4}{13}}} \\ &= \frac{1}{1 + \frac{1}{\frac{30}{13}}} = \frac{1}{1 + \frac{13}{30}} = \frac{1}{\frac{43}{30}} = \frac{30}{43}. \end{aligned}$$

数列の漸化式を用いた効率的な計算法がある．

4. 連分数の計算 (近似分数)

連分数の近似分数の生成法

数列 $\{p_n\}, \{q_n\}$ を漸化式

$$\left. \begin{aligned} p_n &= a_n p_{n-2} + b_n p_{n-1} \\ q_n &= a_n q_{n-2} + b_n q_{n-1} \end{aligned} \right\} \quad (n = 1, 2, \dots),$$
$$p_{-1} = 0, \quad q_{-1} = 1, \quad p_0 = a_0, \quad q_0 = b_0.$$

で求めると,

$$(\text{第 } n \text{ 近似分数}) \quad w_n = \frac{p_n}{q_n} = \cfrac{a_0}{b_0} + \cfrac{a_1}{b_1} + \cdots + \cfrac{a_n}{b_n}.$$

4. 連分数の計算（漸化式）

（証明）

$$\begin{aligned}C &= \left\lfloor \frac{a_0}{b_0} \right\rfloor + \left\lfloor \frac{a_1}{b_1} \right\rfloor + \left\lfloor \frac{a_2}{b_2} \right\rfloor + \dots \\&= \frac{p_0}{q_0 + C_1} \left(C_1 = \left\lfloor \frac{a_1}{b_1} \right\rfloor + \left\lfloor \frac{a_2}{b_2} \right\rfloor + \dots \right) \\&= \frac{p_0}{q_0 + \frac{a_1}{b_1 + C_2}} \left(C_2 = \left\lfloor \frac{a_2}{b_2} \right\rfloor + \left\lfloor \frac{a_3}{b_3} \right\rfloor + \dots \right) \\&= \frac{p_0(b_1 + C_2)}{q_0(b_1 + C_2) + a_1} = \frac{\overbrace{b_1 p_0 + p_0 C_2}^{p_1}}{\underbrace{a_1 q_{-1} + b_1 q_0 + q_0 C_2}_{q_1}} = \frac{p_1 + p_0 C_2}{q_1 + q_0 C_2}.\end{aligned}$$

ここで $C_2 = 0$ とおいたものは第 1 近似分数 w_1 に一致するので、 $w_1 = p_1/q_1$ を得る。

4. 連分数の計算 (漸化式)

(証明 (続))

$$\begin{aligned}C &= \frac{p_1 + \frac{p_0 a_2}{b_2 + C_3}}{q_1 + \frac{q_0 a_2}{b_2 + C_3}} \left(C_3 = \cfrac{a_3}{b_3} + \cfrac{a_4}{b_4} + \dots \right) \\&= \frac{p_1(b_2 + C_3) + a_2 p_0}{q_1(b_2 + C_3) + a_2 q_0} = \frac{\overbrace{a_2 p_0 + b_2 p_1 + p_1 C_3}^{p_2}}{\underbrace{a_2 q_0 + b_2 q_1 + q_1 C_3}_{q_2}} = \frac{p_2 + p_1 C_3}{q_2 + q_1 C_3}.\end{aligned}$$

ここで $C_3 = 0$ とおいたものは第 2 近似分数 w_2 に一致するので、
 $w_2 = p_2/q_2$ を得る…



4. 連分数の計算（近似分数）

課題 3

- ① $\sqrt{2}$, $(1 + \sqrt{5})/2$ に対し, その近似分数 $w_0, w_1, w_2, \dots$ を求めよ.
- ② 前項の第 n 近似分数 w_n ともとの無理数との間の誤差の n に対する変化を調べ, そのグラフを描け.

(注意) 誤差のグラフを描くときは, 片対数グラフを使う.
gnuplot では

```
>set logscale y
```

とすれば縦軸が対数スケールになる.

5. 関数項連分数

関数項連分数

$$C(z) = \frac{a_0}{b_0 + \frac{a_1 z}{b_1 + \frac{a_2 z}{b_2 + \ddots}}}.$$

関数項連分数は他の形のものも考えられるが、本実験では上の形のもののみ考える。

5. 関数項連分数：冪級数を連分数で表す

$$f(z) = c_0 + c_1 z + c_2 z^2 + \cdots$$

商差法 (quotient-difference algorithms)

数列 $e_k^{(n)}$ ($k = 0, 1, \dots; n = 0, 1, \dots$),
 $q_k^{(n)}$ ($k = 1, 2, \dots; n = 0, 1, \dots$) を次の漸化式で生成する.

$$e_0^{(n)} = 0, \quad q_1^{(n)} = \frac{c_{n+1}}{c_n} \quad (n = 0, 1, 2, \dots),$$

$$e_k^{(n)} = q_k^{(n+1)} - q_k^{(n)} + e_{k-1}^{(n+1)}, \quad q_{k+1}^{(n)} = \frac{e_k^{(n+1)}}{e_k^{(n)}} q_k^{(n+1)} \\ (k = 1, 2, \dots; n = 0, 1, \dots).$$

$$\Rightarrow f(z) = \cfrac{c_0}{1} - \cfrac{q_1^{(0)} z}{1} - \cfrac{e_1^{(0)} z}{1} - \cfrac{q_2^{(0)} z}{1} - \cfrac{e_2^{(0)} z}{1} - \cdots$$

(証明) P.Henrici “Applied and Computational Complex Analysis, Vol.2”,
John Wiley & Sons, 1977 を参照.

5. 関数項連分数

$$f(z) = c_0 + c_1 z + c_2 z^2 + \cdots,$$

$$e_0^{(0)} (= 0)$$

$$q_1^{(0)} (= c_1/c_0)$$

$$e_0^{(1)} (= 0)$$

$$e_1^{(0)}$$

$$q_1^{(1)} (= c_2/c_1)$$

$$q_2^{(0)}$$

$$e_0^{(2)} (= 0)$$

$$e_1^{(1)}$$

$$e_2^{(0)}$$

$$q_1^{(2)} (= c_3/c_2)$$

$$q_2^{(1)}$$

$$q_3^{(0)}$$

$$e_0^{(3)} (= 0)$$

$$e_1^{(2)}$$

$$e_2^{(1)}$$

$$e_3^{(0)}$$

$$q_1^{(3)} (= c_4/c_3)$$

$$q_2^{(2)}$$

$$q_3^{(1)}$$

$$q_4^{(0)}$$

$$e_0^{(4)} (= 0)$$

$$e_1^{(3)}$$

$$e_2^{(2)}$$

$$e_3^{(1)}$$

$$\cdots$$

$$f(z) = \cfrac{c_0}{1} - \cfrac{q_1^{(0)} z}{1} - \cfrac{e_1^{(0)} z}{1} - \cfrac{q_2^{(0)} z}{1} - \cfrac{e_2^{(0)} z}{1} - \cdots$$

5. 関数項連分数

課題 4

① 超幾何関数

$$F(\alpha, 1; \gamma; z) = \sum_{n=0}^{\infty} \frac{(\alpha)_n}{(\gamma)_n} z^n = 1 + \frac{\alpha}{\gamma} z + \frac{\alpha(\alpha+1)}{\gamma(\gamma+1)} z^2 + \cdots,$$
$$(a)_0 = 1, \quad (a)_n = a(a+1)(a+2) \cdots (a+n-1) \\ (n = 1, 2, \dots)$$

に対して,

$$q_k^{(n)} = \frac{(\gamma + n + k - 2)(\alpha + n + k - 1)}{(\gamma + n + 2k - 3)(\gamma + n + 2k - 2)},$$
$$e_k^{(n)} = \frac{k(\gamma - \alpha + k - 1)}{(\gamma + n + 2k - 2)(\gamma + n + 2k - 1)}$$

となることを示せ.

5. 関数項連分数

課題 4 (続)

- ② 関数 $\arctan z = \int_0^z \frac{d\zeta}{1+\zeta^2}$ は次のように超幾何関数で表されることを示せ.

$$\arctan z = zF\left(\frac{1}{2}, 1; \frac{3}{2}; -z^2\right).$$

- ③ $\arctan z$ の連分数表示を次の形で求めよ
($\arctan z$ は奇関数であることに注意).

$$\arctan z = \cfrac{a_0 z}{1} + \cfrac{a_1 z^2}{1} + \cfrac{a_2 z^2}{1} + \cdots.$$

- ④ $\arctan z$ に対し次の連分数表示が成り立つことを示せ.

$$\arctan z = \cfrac{z}{1} + \cfrac{1^2 z^2}{3} + \cfrac{2^2 z^2}{5} + \cfrac{3^2 z^2}{7} + \cfrac{4^2 z^2}{9} + \cdots.$$

5. 関数項連分数

課題 4 (続 2)

- ❶ 関数 $\arctan z$ の連分数表示に $z = 1, 1/\sqrt{3}$ を代入することにより, 次の π の連分数表示が得られることを示せ.

$$\begin{aligned}\pi &= \cfrac{4}{1 + \cfrac{1^2}{3 + \cfrac{2^2}{5 + \cfrac{3^2}{7 + \cdots}}}} \\ &= \cfrac{2\sqrt{3}}{1 + \cfrac{1^2/3}{3 + \cfrac{2^2/3}{5 + \cfrac{3^2/3}{7 + \cdots}}}}.\end{aligned}$$

- ❷ 前項のふたつの連分数の近似分数 $w_n = p_n/q_n$ を計算し, π への収束の様子を確かめよ (近似分数の誤差のグラフを描け).

5. 関数項連分数

商差法は丸め誤差に弱い.

課題 5

$\arctan z$ に対して, 連分数の係数 $-q_k^{(0)}, -e_k^{(0)}$ を商差法によって倍精度演算により計算し, k を大きくするに連れて係数の精度が悪くなることを確かめよ.



多倍長演算

本実験ではライブラリ **exflib** を用いる.

5. 関数項連分数

exflib は C++, Fortran, MATLAB で動作する.

⇒ 最小限の C++ の知識

C++ のプログラム例 (青字はおまじない)

```
// C++ ではコメントにこの記号を使う (/...*/ も使える)
#include <iostream> // C における <stdio.h> の代わり
#include <iomanip>
#include <math.h>
using namespace std;
int main(void){
    cout << setiosflags(ios::scientific);
    double pi = M_PI; // 円周率は M_PI で与えられる.
    cout << setprecision(15) << pi << endl;
    // pi を 15 桁で出力する. endl は改行.
    double e; // C++ ではプログラム途中でも変数の型宣言ができる.
    e = M_E; // Napier の数 (自然対数の底)
    cout << setprecision(15) << e << endl;
    return 0;
}
```

5. 関数項連分数

C++プログラムのコンパイルと実行

前頁の C++ プログラムを（仮に）foo.cpp と名前をつけて保存する。

```
> g++ -o foo foo.cpp  
> ./foo
```

すると、pi（円周率）と e（Napier の数）の値が出力される。

P.S. 繰り返し文の書き方.

```
for (int i=1; i<=10; ++i) // プログラムの途中で型宣言できる.  
    cout << setw(3) << i << endl;  
// setw(3)... 整数を 3 桁で出力.
```

課題 6

- ① 前頁の C++ プログラムを作成しコンパイルして実行せよ。
- ② $n = 1, 2, \dots, 10$ に対し $n^2, n^3, 1/n$ を計算・表示するプログラムを作成し、実行せよ。

5. 関数項連分数

exflib のプログラム例

```
#define PRECISION 100 // 10 進 100 桁精度の計算を行う.  
#include <iostream>  
#include <iomanip>  
#include <math.h>  
#include <exfloat.h>  
using namespace std;  
int main(void){  
    cout << setiosflags(ios::scientific);  
    exfloat pi = "#pi"; // 多倍長浮動小数点数の型宣言.  
    exfloat e = "#e";  
    exfloat x = "1.23"; // 数値は文字列で指定する.  
    cout << "pi = " << setprecision(100) << pi << endl;  
    cout << "e = " << setprecision(100) << e << endl;  
    cout << "x = " << setprecision(100) << x << endl;  
    return 0;  
}
```

5. 関数項連分数

exflib プログラムのコンパイルと実行

前頁のプログラムを（仮に）foo.cpp と名前をつけて保存する。

```
> g++ -o foo foo.cpp -lexfloat  
> ./foo
```

π , e , 1.23 の値が 100 桁表示される。

課題 7

- ① 前頁の exflib プログラムを作成・コンパイル・実行せよ。
- ② $\arctan z$ に対する連分数の係数 $-q_k^{(0)}, -e_k^{(0)}$ を、今度は商差法を用いて 10 進 100 桁の多倍長演算で計算し、理論値と符合することを確かめよ。

6. 連分数による解析接続

解析接続とは？

解析関数の定義域を（解析関数として）拡張すること．

（例）

$$f(z) = \sum_{n=0}^{\infty} z^n.$$

収束半径 1 $\rightarrow$ 定義域は $|z| < 1$. ところが, 無限和を計算すると,

$$f(z) = \frac{1}{1-z}.$$

右辺の定義域は $\mathbb{C} \setminus \{1\}$.

$\therefore f(z)$ は $\mathbb{C} \setminus \{1\}$ に解析接続される.

6. 連分数による解析接続

冪級数で与えられた解析関数 $f(z) = \sum_{n=0}^{\infty} c_n z^n$.

... 収束円 $|z| < R$ においてのみ（絶対かつ一様）収束する.

$$\left(R^{-1} = \limsup_{n \rightarrow \infty} |c_n|^{1/n} \quad \text{収束半径} \right)$$

連分数による解析接続

冪級数で与えられた解析関数 $f(z)$ を連分数で表すと、収束円 $|z| < R$ の外で $f(z)$ が収束して計算できるようになる.

6. 連分数による解析接続 (例)

$$f(z) = 1 - \frac{z}{2} + \frac{z^2}{3} - \cdots \left(= \frac{\log(1+z)}{z} \right) \quad (|z| < 1),$$

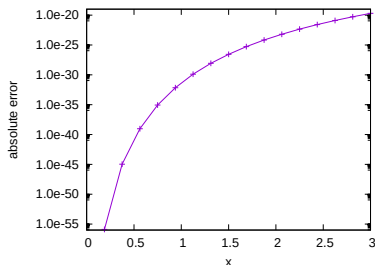
$$f(z) = \cfrac{a_0}{1} + \cfrac{a_1 z}{1} + \cfrac{a_2 z}{1} + \cdots$$

商差法により連分数に書き直す.

n	a_n	n	a_n
0	1.0000...	7	0.28571 42857 14285...
1	0.5000...	8	0.22222...
2	0.16666...	9	0.27777...
3	0.33333...	10	0.22727 27272 72727...
4	0.20000...	11	0.27272 72727 27272...
5	0.30000...	12	0.23076 92307 69230...
6	0.21428 57142 85714...	14	0.26923 07692 30769...

6. 連分数による解析接続

$0 < z = x < 3$ における解析接続の誤差



● 縦軸：絶対誤差

● 横軸： $x(=z)$

収束円 $|z| < 1$ 外で解析接続が
高精度で計算されている。

課題 8

$f(z) = 1 - z/2 + z^2/3 - \cdots (= \log(1+z)/z)$ の連分数表示を、係数を商差法で数値計算することにより求め、その連分数関数が収束円 $|z| < 1$ の外で収束して $\log(1+z)/z$ に一致することを確認せよ。

6. 連分数による解析接続：ゼータ関数への応用

ゼータ (zeta) 関数

$$\zeta(s) = \sum_{n=1}^{\infty} \frac{1}{n^s} = 1 + \frac{1}{2^s} + \frac{1}{3^s} + \cdots \quad (\operatorname{Re} s > 1).$$

$\zeta(s)$ の計算を，解析関数の連分数による解析接続により行う．
次の補助的な関数を考える（収束半径 1 の解析関数）．

$$f_s(z) \equiv \sum_{n=0}^{\infty} \frac{(-1)^n}{(n+1)^s} z^n = 1 - \frac{z}{2^s} + \frac{z^2}{3^s} - \cdots.$$

* Abel の連続性定理により，

$$\sum_{n=1}^{\infty} \frac{(-1)^{n-1}}{n^s} = \lim_{x \rightarrow 1-0} f_s(x).$$

6. 連分数による解析接続：ゼータ関数への応用

$$\zeta(s) = \frac{f_s(1)}{1 - 2^{1-s}} \quad \left(s \neq 1, f_s(1) = \lim_{x \rightarrow 1-0} f_s(x) \right).$$

$f_s(1)$ が計算できれば $\zeta(s)$ が計算できる.

$$\begin{aligned} \therefore \sum_{n=1}^{2N} \frac{(-1)^{n-1}}{n^s} &= 1 - \frac{1}{2^s} + \frac{1}{3^s} - \frac{1}{4^s} + \cdots - \frac{1}{(2N)^s} \\ &= 1 - \frac{1}{2^2 \cdot 1^s} + \frac{1}{3^s} - \frac{1}{2^s 2^s} + \cdots + \frac{1}{(2N-1)^s} - \frac{1}{2^s N^2} \\ &= 1 + \frac{1}{2^s} + \frac{1}{3^s} + \cdots + \frac{1}{(2N)^2} - 2^{1-s} \left(1 + \frac{1}{2^s} + \cdots + \frac{1}{N^s} \right), \end{aligned}$$

$N \rightarrow \infty$ として $f_s(1) = (1 - 2^{1-s})\zeta(s)$ を得る. ■

6. 連分数による解析接続：ゼータ関数への応用

課題 9

- ① $s = 2, 3, 4$ について $f_s(z)$ の連分数表示を求め、収束円 $|z| < 1$ の外で連分数表示が収束して関数値を計算できることを確かめよ.
- ② 次式の成立を数値的に確かめよ.

$$\zeta(2) = \sum_{n=1}^{\infty} \frac{1}{n^2} = \frac{\pi^2}{6}, \quad \zeta(4) = \sum_{n=1}^{\infty} \frac{1}{n^4} = \frac{\pi^4}{90}.$$

- ③ 次式の成立を数値的に確かめよ ($\zeta(s)$ の解析接続).

$$\zeta(0) = -\frac{1}{2}, \quad \zeta(-1) = -\frac{1}{12}, \quad \zeta(\text{負の奇数}) = 0.$$

5. 関数項連分数：ゼータ関数への応用

参考 課題9の第3問は次を意味している.

①

$$1 + 1 + 1 + \cdots = -\frac{1}{2}, \quad 1 + 2 + 3 + \cdots = -\frac{1}{12} \quad !?$$

② $s = -1, -3, \dots$ は $\zeta(s)$ の零点である ($\zeta(s)$ の自明な零点).

Riemann 予想

$\zeta(s)$ の非自明な零点はすべて、
複素平面の直線 $\operatorname{Re} s = \frac{1}{2}$ の上にある.

6. (発展課題) Fourier 変換

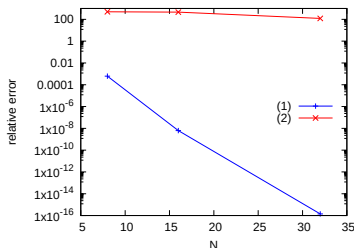
Fourier 変換

$$\mathcal{F}[f](\xi) = \int_{-\infty}^{\infty} f(x) \exp(-2\pi i \xi x) dx.$$

$f(x)$ が減衰の遅い関数の場合、従来の数値積分公式で Fourier 変換を計算するのは難しい。

次の積分を DE 公式で数値計算した。

$$(1) \int_{-\infty}^{\infty} \frac{dx}{1+x^2},$$
$$\times (2) \int_{-\infty}^{\infty} \frac{e^{-2\pi i x}}{1+x^2} dx.$$



6. (発展課題) Fourier 変換

Fourier 変換 $\mathcal{F}[f](\xi)$ を解析接続の数値計算を用いて計算する.

$$\mathcal{F}[f](\xi) = \lim_{\epsilon \rightarrow 0+} \left\{ \int_{-\infty}^0 f(x) e^{-2\pi i(\xi + i\epsilon)x} dx + \int_0^{\infty} f(x) e^{-2\pi i(\xi - i\epsilon)x} dx \right\},$$

$$\mathcal{F}[f](\xi) = \lim_{\epsilon \rightarrow 0+} \{ \mathfrak{F}_+[f](\xi + i\epsilon) - \mathfrak{F}_-[f](\xi - i\epsilon) \},$$

$$\begin{aligned} \mathfrak{F}_+[f](\zeta) &= \int_{-\infty}^0 f(x) \exp(-2\pi i \zeta x) dx \\ &= \int_0^{\infty} f(-x) \exp(2\pi i \zeta x) dx, \end{aligned}$$

$$\mathfrak{F}_-[f](\zeta) = - \int_0^{\infty} f(x) \exp(-2\pi i \zeta x) dx.$$

$\mathfrak{F}_{\pm}[f](\zeta)$ は上／下半平面 $\pm \operatorname{Im} \zeta > 0$ で解析的である.

6. (発展課題) Fourier 変換

$\mathfrak{F}_{\pm}[f](\zeta)$ を上／下半平面 $\pm \operatorname{Im} \zeta > 0$ で計算し、それらを実軸 $\mathbb{R}$ 上に解析接続することにより $\mathcal{F}[f](\xi)$ を計算する。

- ① $\mathfrak{F}_{\pm}[f](\zeta)$ の Taylor 級数を上／下半平面 $\pm \operatorname{Im} \zeta > 0$ で求める。
複素定数 $\zeta_0^{(\pm)}$ を上／下半平面 $\pm \operatorname{Im} \zeta > 0$ 上に適当に取る。

$$\begin{aligned}\mathcal{F}_{\pm}[f](\zeta) &= \sum_{n=0}^{\infty} c_n^{(\pm)} (\zeta - \zeta_0^{(\pm)})^n, \\ c_n^{(\pm)} &= \frac{1}{n!} \frac{d^n}{d\zeta^n} \mathfrak{F}_{\pm}[f](\zeta) \Big|_{\zeta=\zeta_0^{(\pm)}} \\ &= \pm \frac{1}{n!} \int_0^{\infty} (\pm 2\pi i x)^n f(\mp x) e^{\pm 2\pi i \zeta_0^{(\pm)} x} dx.\end{aligned}$$

係数 $c_n^{(\pm)}$ を与える積分は、被積分関数が指数関数的に減衰するので、従来の数値積分公式（DE 公式など）で簡単に計算できる。

6. (発展課題) Fourier 変換

- ② 前項で Taylor 級数の形で求めた $\mathfrak{F}_{\pm}[f](\zeta)$ を, 連分数に変換することにより, 実軸 $\mathbb{R}$ 上に解析接続する.

$$\mathfrak{F}_{\pm}[f](\zeta) = \frac{a_0^{(\pm)}}{1 + \frac{a_1^{(\pm)}(\zeta - \zeta_0^{(\pm)})}{1 + \frac{a_2^{(\pm)}(\zeta - \zeta_0^{(\pm)})}{1 + \ddots}}},$$

連分数の係数 $a_n^{(\pm)}$ は, すでに説明したように, Taylor 級数の係数 $c_n^{(\pm)}$ から商差法により求める.

③

$$\mathcal{F}[f](\xi) = \mathfrak{F}_+[f](\xi) - \mathfrak{F}_-[f](\xi).$$

6. (発展課題) Fourier 変換

課題 10*

Fourier 変換

$$\mathcal{F}[(1+x^2)^{-1}](\xi) = \pi \exp(-2\pi|\xi|),$$

$$\mathcal{F}[\tanh(\pi x)](\xi) = \frac{-i}{\sinh(\pi\xi)}$$

を上述の方法により計算し、その精度（誤差）を求めよ。

定数 $\zeta_0^{(\pm)}$ は $\zeta_0^{(\pm)} = \pm i$ とせよ。

数値積分に用いる DE 公式のプログラムは、用意したものを
書き直して用いること（最後の頁参照）。

* Taylor 係数 $c_n^{(\pm)}$ は次式で計算すること ($\zeta_0^{(\pm)} = \xi_0^{(\pm)} + i\eta_0^{(\pm)}$).

$$c_n^{(\pm)} = \frac{1}{2\pi\eta_0^{(\pm)} n!} \int_0^\infty \left(\frac{ix}{\eta_0^{(\pm)}} \right)^n f\left(\frac{-x}{2\pi\eta_0^{(\pm)}} \right) \exp\left(i \frac{\xi_0^{(\pm)}}{\eta_0^{(\pm)}} x \right) \exp(-x) dx.$$

6. (発展課題) Fourier 変換：数値積分に対する DE 公式

台形公式：解析関数の全無限区間積分に対して精度がよい.

$$\int_{-\infty}^{\infty} g(u) du \simeq h \sum_{k=-\infty}^{\infty} g(kh) \simeq h \sum_{k=-N_-}^{N_+} g(kh).$$

半無限区間積分 $\int_0^{\infty} f(x)dx$ に対しては？

→ 変数変換 $x = \psi(u)$ により全無限区間積分にして、
台形公式を適用する.

$$\begin{aligned} \int_0^\infty f(x)dx &= \int_{-\infty}^\infty f(\psi(u))\psi'(u)du \\ &\simeq h \sum_{k=-N_-}^{N_+} f(\psi(kh))\psi'(kh), \end{aligned}$$

$x = \psi(u) : (-\infty, +\infty)$ を $(0, +\infty)$ に写す変数変換.

6. (発展課題) Fourier 変換 : 数値積分に対する DE 公式

変数変換 $x = \psi(u)$ をどう選ぶか？

→ 被積分関数 $f(\psi(u))\psi'(u)$ が二重指数関数的に速く減衰して、少ない標本点数 $N_+ + N_- + 1$ で台形公式の無限和を打ち切れるようにする.

指数関数的減衰する関数の積分 $\int_0^\infty f_0(x)e^{-x}dx$ の場合は,

$$\psi(u) = \exp(u - \exp(-u)),$$

$$\psi'(u) = (1 + \exp(-u)) \exp(u - \exp(-u)),$$

$$\int_0^\infty f(x)dx = \int_0^\infty f_0(x)e^{-x}dx \simeq h \sum_{k=-N_-}^{N_+} f(\psi(kh))\psi'(kh).$$

$N_\pm$ の決め方 : | 被積分関数 | が十分小さい項で無限和を打ち切る.

6. (発展課題) Fourier 変換 : DE 公式による自動積分

刻み幅 $h = 2^{-1}, 2^{-2}, \dots$ の台形公式による積分値を計算する.

I_h : 刻み幅 h の DE 公式.

$$I_h = h \sum_k f(\psi((2k+1)h)) + \frac{1}{2} I_{2h}.$$

刻み幅 h の DE 公式 (台形公式) の誤差 : $C \exp\left(-\frac{c}{h}\right)$ (C, c : 正定数).

刻み幅 $2h$ の DE 公式の誤差 (I : 厳密な積分値)

$$|I_{2h} - I_h| \simeq |I_{2h} - I| = C \exp\left(-\frac{c}{2h}\right) = |I_h - I|^{1/2}.$$

許容誤差を ϵ として, $|I_{2h} - I_h| \leq c_{\text{safe}} \epsilon^{1/2}$ となったら, I_h は許容誤差を満たすと判定して自動積分を終了し, I_h を積分近似値として返す
(c_{safe} : 安全因子. 0.2 位の値に取ればよい).

補足：T_EX でレポートを書くためのために

- 連分数を T_EX で書くには？

- `\cfrac` コマンドを用いる. T_EX ファイルのプリアンブル (`\begin{document}` の前) に `\usepackage{amsmath}` を入れること.
- (T_EX ソース) `\cfrac{a}{b+\cfrac{c}{d}}`
- (出力)
$$\frac{a}{b + \frac{c}{d}}$$

- 花文字 $\mathscr{F}$ を T_EX で書くには？

- `\mathscr{F}`
- T_EX ファイルのプリアンブルに `\usepackage{mathrsfs}` を入れること.

- ドイツ文字 (フラクトゥール体) $\mathfrak{F}$ を T_EX で書くには？

- `\mathfrak{F}`
- T_EX ファイルのプリアンブルに `\usepackage{amsmath,amssymb}` を入れること.

- 手計算で解く課題は自宅学習,
プログラミング課題のみ計算機室で行うこと.
- DE 型数値積分公式のプログラムは,
<http://www.uec-ogata-lab.jp> の「教育ページ」から
ダウンロードして用いること.
- 締切: 2019 年 12 月 13 日 (金).
- レポート提出場所: 西 4 号館 4 階事務室前のメールボックス.