数値解析 レポート No. 3　　学籍番号　　氏名

[〆切] 2020/10/15 19:00

[**II.** 有限回の呪縛]

II-B. $\sqrt{x+1}$ を $x=0$ でテイラー展開して得られた 2 次の多項式を $p_2(x)$ と置く。$x \in [0,1]$ の範囲のすべての x に対して

$$|\sqrt{x+1} - p_2(x)| \leq \frac{1}{16}$$

となることを示せ。

$f(x) = \sqrt{x+1}$ とおくと 1階導関数, 2階導関数, 3階導関数 はそれぞれ

$$f'(x) = \frac{1}{2}(x+1)^{-\frac{1}{2}}, \quad f''(x) = -\frac{1}{4}(x+1)^{-\frac{3}{2}}, \quad f'''(x) = \frac{3}{8}(x+1)^{-\frac{5}{2}}$$

となる. 従って $x \in [0,1]$ の範囲において

$$|f'''(x)| \leq \qquad \cdots ①$$

となる. 剰余項付きテイラーの定理より $f(x)$ は 2次のテイラー多項式 $P_2(x)$ と それに対するラグランジュの剰除項 $R_2(x)$ を用いて

$$f(x) = P_2(x) + R_2(x) \qquad \cdots ②$$

と表せる. $x \in [0,1]$ において $R_2(x)$ は $c_x \in [0,1]$ を用いて

$$R_2(x) = \qquad \cdots ③$$

となる. ② より問いにある不等式左辺は

$$\left|\sqrt{x+1} - P_2(x)\right| = |f(x) - P_2(x)| = |R_2(x)|$$

と書き換えられ、③, ① および $|x| \leq 1$ より

$$|R_2(x)| =$$

となり問いにある不等式が示された. 」

数値解析 レポート No. 3　　学籍番号　　　　氏名

[〆切] 2020/10/15 19:00

[**III.** 求根アルゴリズム (1)]

III-A. (1) x の多項式 $f(x) = x^6 - 7x^4 + 11x^3 - 10$ が与えられた時、$x \in [1, 2]$ の範囲にある $f(x)$ の根を 2 分法を用いて求める場合を考える。解の存在する範囲を 10^{-8} 以下に限定するには 2 分法の繰り返し操作を最低何回行えば良いか答えよ。

(2) 2 分法により $f(x)$ の根を求めるプログラムを作成して問 1 で求めた回数繰り返し、繰り返し操作後の解の存在する x の区間の両端の値を 10 進 11 桁以降を切り捨てて求めよ。またその区間の両端の値の平均値を求め根の近似値として 10 進 11 桁以降を切り捨てて求めよ。作成したプログラムも提出すること。プログラミング言語は問わない。

(1) 2分法では1回の操作で解が存在する区間の幅は $\frac{1}{2}$ になっていく。従ってn回操作を繰り返すと

$$\inf\left\{n \in \mathbb{N} \,\middle|\, 2 \cdot \frac{1}{2^n} \leq 10^{-8}\right\} \quad \cdots ①$$

対数関数は単調増加関数であるため、不等号の両辺の対数をとっても不等号の向きは変化しない。

$\log_2$(左辺) =

$\log_2$(右辺) =

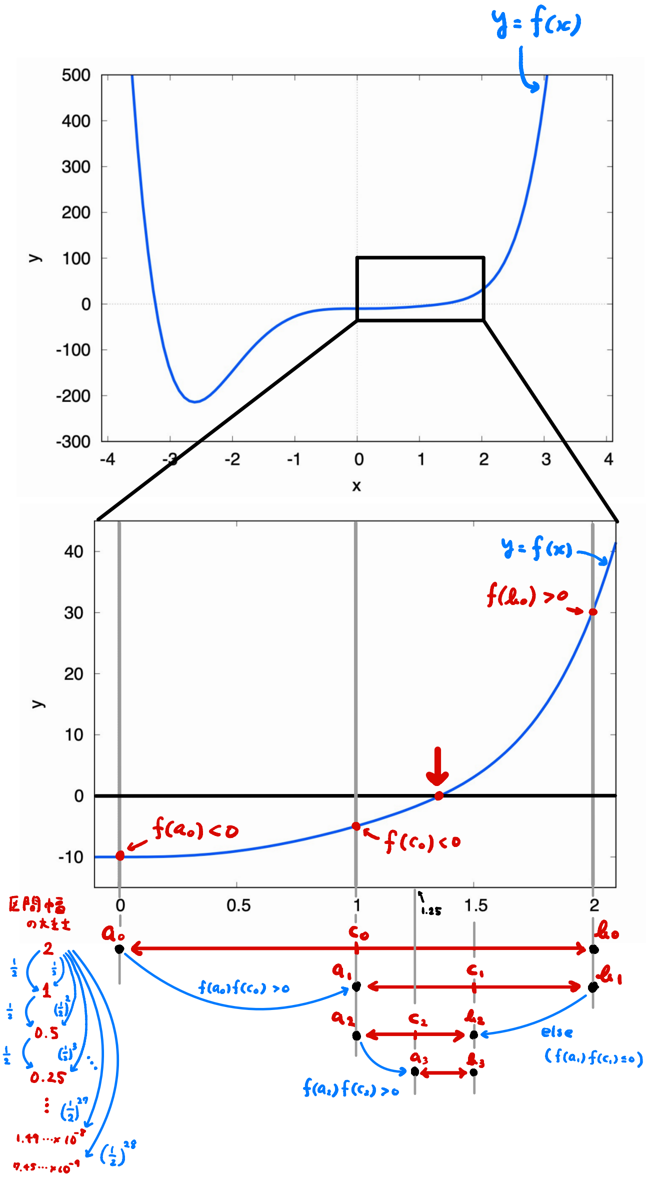
より

また $8\log_2 10 = 26.57\cdots$ より

① =

=

従って　　回」



(2) 作成したプログラムの出力より

解が存在する区間幅

[1.357271470, 1.357271477]」

根の近似値

1.357271473」

Python 3.6 の例

```
#def f(x):
    #return x**6 - 7*x**4 + 11*x**3 - 10

def f(x):
    x2 = x*x; x3 = x2*x
    return ((x2-7)*x+11)*x3-10 # ホーナー法 (Horner's rule)
```

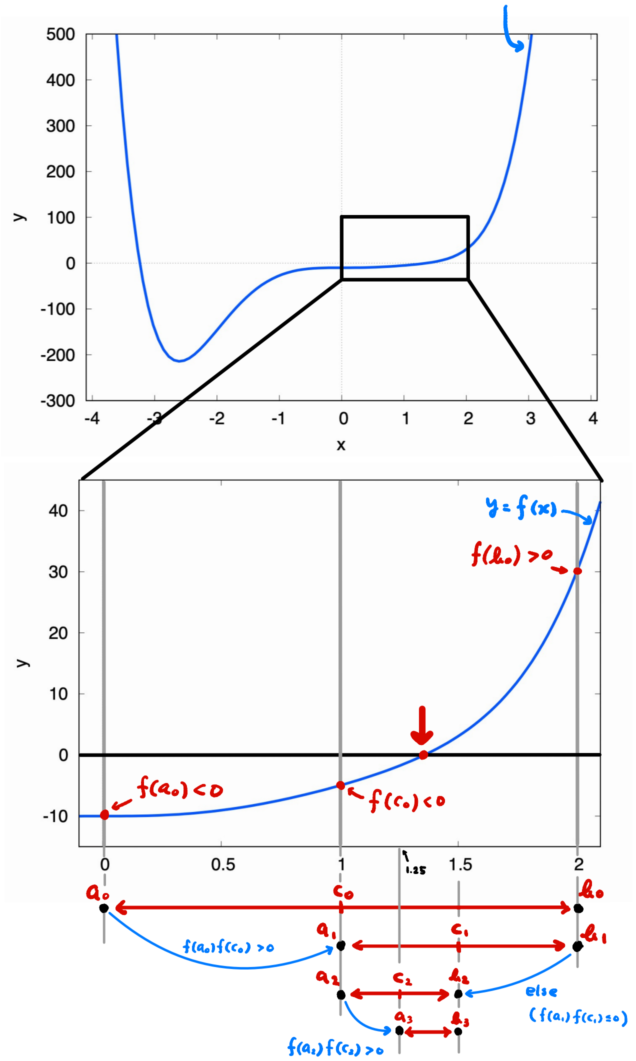
キーコード

```
a = 0; b = 2
for i in range(28):
  #c = (a+b)/2
  c = a + (b-a)/2# 平均値の計算を旨くする
  if f(a) * f(c) > 0:
    a = c
  else:
    b = c
```

⊙ 絶対誤差と残差をプロット

2020 10/15

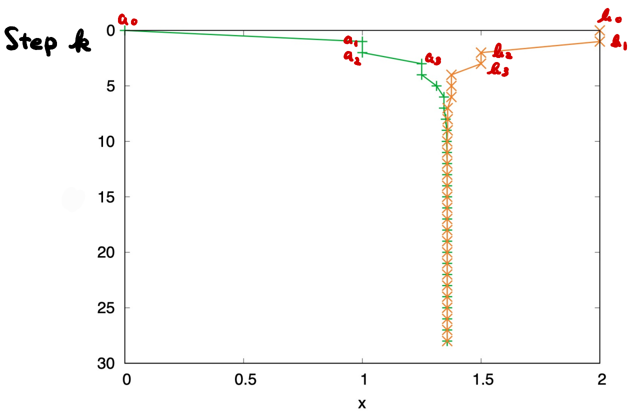
$y = f(x) = x^6 - 7x^4 + 11x^3 - 10$

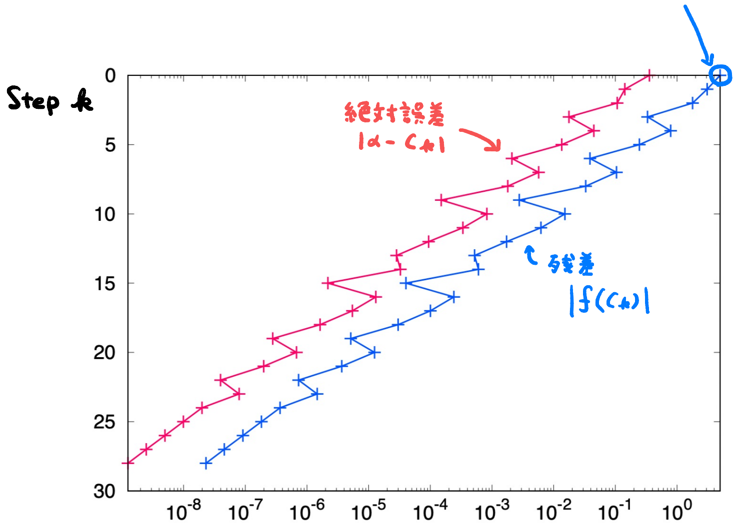


(i = 0) [0.000000000000000, 2.000000000000000] 残差 5.000000000000000
(i = 1) [1.000000000000000, 2.000000000000000] 残差 3.078125000000000
(i = 2) [1.000000000000000, 1.500000000000000] 残差 1.790771484375000
(i = 3) [1.250000000000000, 1.500000000000000] 残差 0.332431793212891
(i = 4) [1.250000000000000, 1.375000000000000] 残差 0.789903104305267
(i = 5) [1.312500000000000, 1.375000000000000] 残差 0.245758912526071
(i = 6) [1.343750000000000, 1.375000000000000] 残差 0.038826270741993
(i = 7) [1.343750000000000, 1.359375000000000] 残差 0.104560935900508
(i = 8) [1.351562500000000, 1.359375000000000] 残差 0.033145030229402
(i = 9) [1.355468750000000, 1.359375000000000] 残差 0.002770684097975
(i = 10) [1.355468750000000, 1.357421875000000] 残差 0.015204592856266
(i = 11) [1.356445312500000, 1.357421875000000] 残差 0.006221317340325
(i = 12) [1.356933593750000, 1.357421875000000] 残差 0.001726408364700
(i = 13) [1.357177734375000, 1.357421875000000] 残差 0.000521864805256
(i = 14) [1.357177734375000, 1.357299804687500] 残差 0.000602340029376
(i = 15) [1.357238769531250, 1.357299804687500] 残差 0.000040254676435
(i = 16) [1.357269287109375, 1.357299804687500] 残差 0.000240800798071
(i = 17) [1.357269287109375, 1.357284545898438] 残差 0.000100271994263
(i = 18) [1.357269287109375, 1.357276916503906] 残差 0.000030008392281
(i = 19) [1.357269287109375, 1.357273101806641] 残差 0.000005123208735
(i = 20) [1.357271194458008, 1.357273101806641] 残差 0.000012442575109
(i = 21) [1.357271194458008, 1.357272148132324] 残差 0.000003659679020
(i = 22) [1.357271194458008, 1.357271671295166] 残差 0.000000731765899
(i = 23) [1.357271432876587, 1.357271671295166] 残差 0.000001463956300
(i = 24) [1.357271432876587, 1.357271552085876] 残差 0.000000366095135
(i = 25) [1.357271432876587, 1.357271492481232] 残差 0.000000182835397
(i = 26) [1.357271462678909, 1.357271492481232] 残差 0.000000091629863
(i = 27) [1.357271462678909, 1.357271477580070] 残差 0.000000045602768
(i = 28) [1.357271470129490, 1.357271477580070] 残差 0.000000023013550

解の存在する区間幅 [1.357271470129490, 1.357271477580070]
根の近似値 1.357271473854780
残差 0.000000023013550

$|f(c_0)| =$





- 毎ステップ必ず絶対誤差・残差が減少するわけではない。
- このケースでは絶対誤差と残差が似た動きをしている

残差が小さければ絶対誤差も小さい

数値解析 レポート No. 3　　学籍番号　　氏名

[〆切] 2020/10/15 19:00

[**II.** 有限回の呪縛]

II-B. $\sqrt{x+1}$ を $x=0$ でテイラー展開して得られた 2 次の多項式を $p_2(x)$ と置く。$x \in [0,1]$ の範囲のすべての x に対して

$$|\sqrt{x+1} - p_2(x)| \leq \frac{1}{16}$$

となることを示せ。

$f(x) = \sqrt{x+1}$ とおくと 1階導関数, 2階導関数, 3階導関数 はそれぞれ

$$f'(x) = \frac{1}{2}(x+1)^{-\frac{1}{2}}, \quad f''(x) = -\frac{1}{4}(x+1)^{-\frac{3}{2}}, \quad f'''(x) = \frac{3}{8}(x+1)^{-\frac{5}{2}}$$

となる. 従って $x \in [0,1]$ の範囲において

$$|f'''(x)| \leq \frac{3}{8} \quad \cdots ①$$

となる. 剰余項付きテイラーの定理より $f(x)$ は 2次のテイラー多項式 $P_2(x)$ と それに対するラグランジュの剰余項 $R_2(x)$ を用いて

$$f(x) = P_2(x) + R_2(x) \quad \cdots ②$$

と表せる. $x \in [0,1]$ において $R_2(x)$ は $c_x \in [0,1]$ を用いて

$$R_2(x) = \frac{x^3}{3!} f'''(c_x) \quad \cdots ③$$

となる. ② より問いにある不等式左辺は

$$|\sqrt{x+1} - P_2(x)| = |f(x) - P_2(x)| = |R_2(x)|$$

と書き換えられ、③, ① および $|x| \leq 1$ より

$$|R_2(x)| = \frac{|x|^3}{3!}|f'''(c_x)| \leq \frac{1}{3!} \cdot \frac{3}{8} = \frac{1}{16}$$

となり問いにある不等式が示された. ┘

数値解析 レポート No. 3

学籍番号 ______ 氏名 ______

[〆切] 2020/10/15 19:00

[**III.** 求根アルゴリズム (1)]

III-A. (1) x の多項式 $f(x) = x^6 - 7x^4 + 11x^3 - 10$ が与えられた時、$x \in [1, 2]$ の範囲にある $f(x)$ の根を 2 分法を用いて求める場合を考える。解の存在する範囲を 10^{-8} 以下に限定するには 2 分法の繰り返し操作を最低何回行えば良いか答えよ。

(2) 2 分法により $f(x)$ の根を求めるプログラムを作成して問 1 で求めた回数繰り返し、繰り返し操作後の解の存在する x の区間の両端の値を 10 進 11 桁以降を切り捨てて求めよ。またその区間の両端の値の平均値を求め根の近似値として 10 進 11 桁以降を切り捨てて求めよ。作成したプログラムも提出すること。プログラミング言語は問わない。

(1) 2分法では1回の操作で解が存在する区間の幅は $\frac{1}{2}$ になっていく。従ってn回操作を繰り返すと

$$\inf\left\{n \in \mathbb{N} \;\middle|\; 2 \cdot \frac{1}{2^n} \leq 10^{-8}\right\} \quad \cdots ①$$

infimum (下限) ⇔ supremum (上限)

対数関数は単調増加関数であるため、不等号の両辺の対数をとっても不等号の向きは変化しない。

$\log_2(\text{左辺}) = \log_2\left(2 \cdot \frac{1}{2^n}\right) = -n+1$

$\log_2(\text{右辺}) = \log_2 10^{-8} = -8\log_2 10$

より

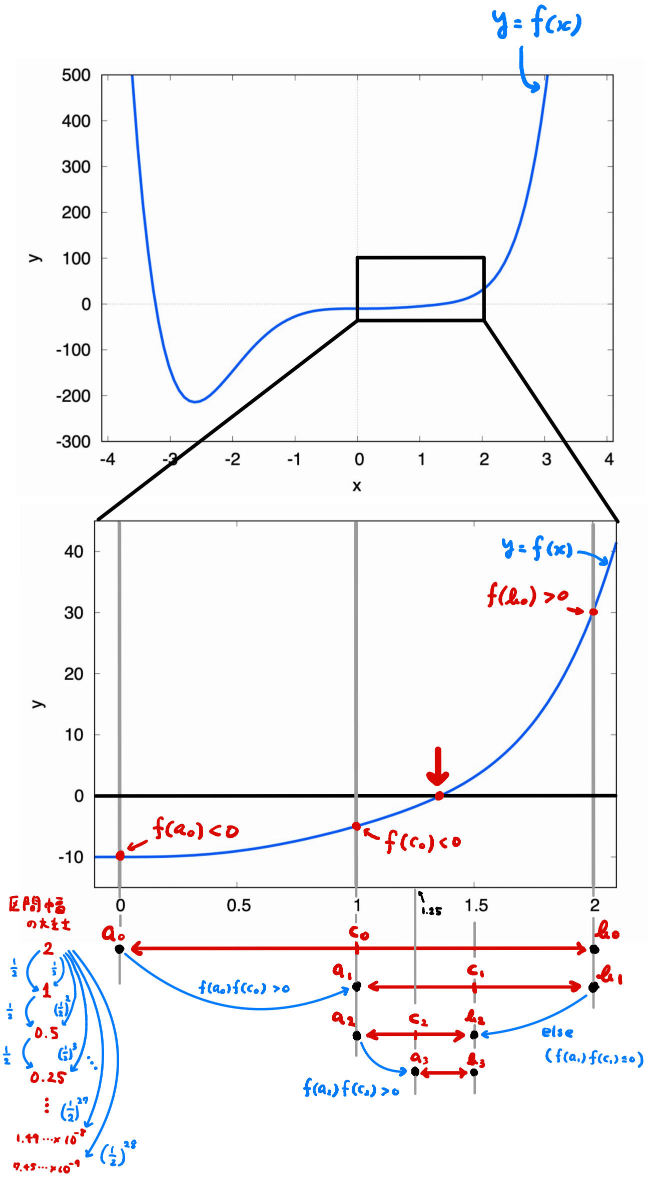
$$-n+1 \leq -8\log_2 10 \iff n \geq 8\log_2 10 + 1$$

また $8\log_2 10 = 26.57\cdots$ より

$$① = \inf\{n \in \mathbb{N} \mid n \geq 8\log_2 10 + 1\}$$

$$= \lceil 8\log_2 10 + 1 \rceil = 28$$

従って 28回」



(2) 作成したプログラムの出力より

解が存在する区間幅

[1.357271470, 1.357271477]」

根の近似値

1.357271473」

Python 3.6 の例

```
#def f(x):
    #return x**6 - 7*x**4 + 11*x**3 - 10

def f(x):
    x2 = x*x; x3 = x2*x
    return ((x2-7)*x+11)*x3-10 # ホーナー法 (Horner's rule)
```

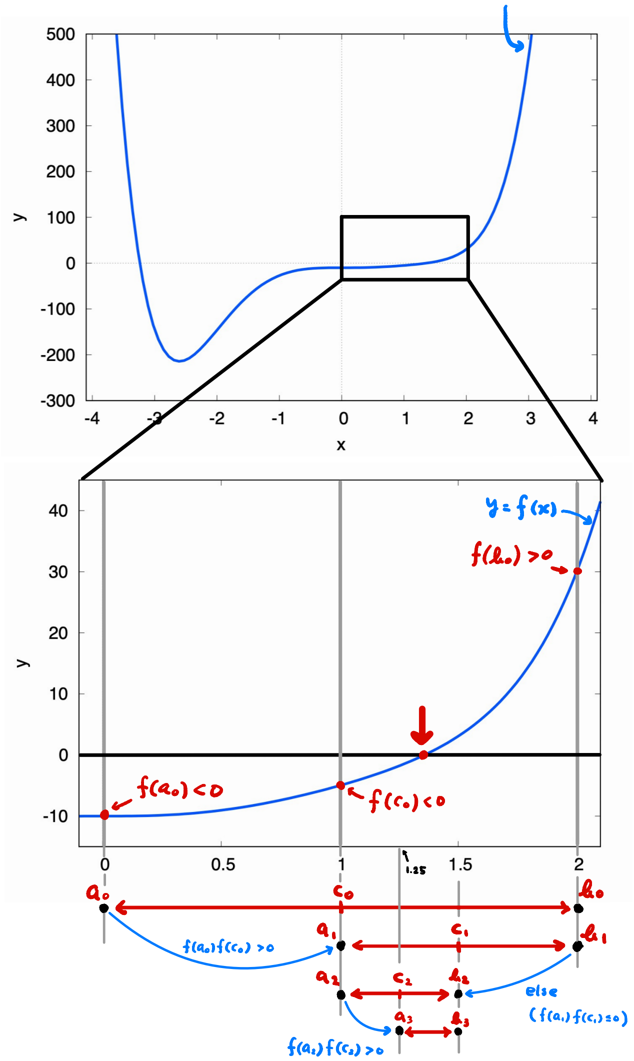
キーコード

```
a = 0; b = 2
for i in range(28):
  #c = (a+b)/2
  c = a + (b-a)/2# 平均値の計算を旨くする
  if f(a) * f(c) > 0:
    a = c
  else:
    b = c
```

⊙ 絶対誤差と残差をプロット

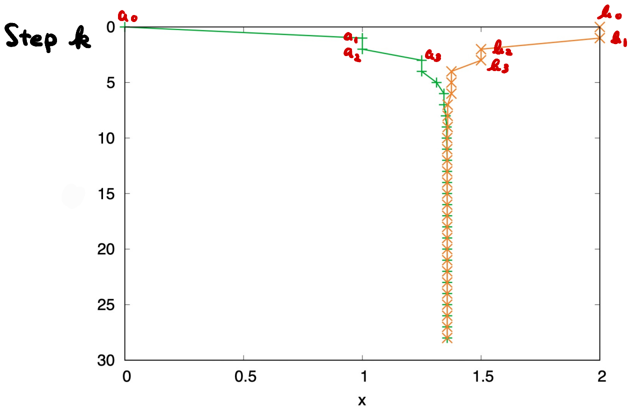
2020
10/15

$$y = f(x) = x^6 - 7x^4 + 11x^3 - 10$$

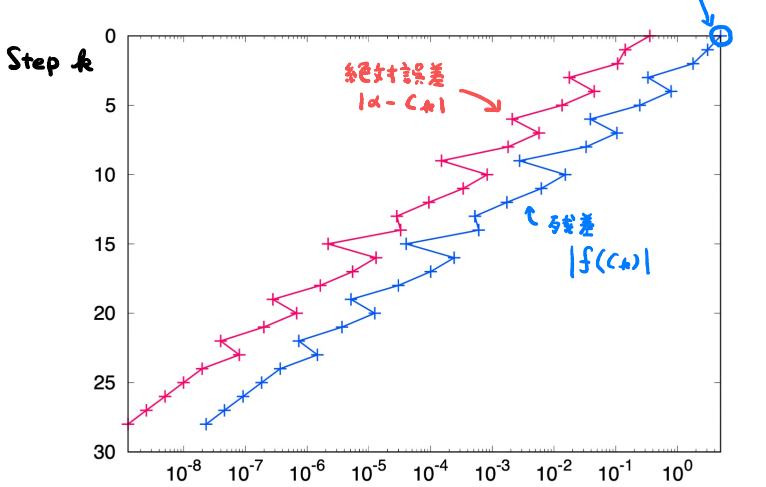


(i = 0) [0.000000000000000, 2.000000000000000] 残差 5.000000000000000
(i = 1) [1.000000000000000, 2.000000000000000] 残差 3.078125000000000
(i = 2) [1.000000000000000, 1.500000000000000] 残差 1.790771484375000
(i = 3) [1.250000000000000, 1.500000000000000] 残差 0.332431793212891
(i = 4) [1.250000000000000, 1.375000000000000] 残差 0.789903104305267
(i = 5) [1.312500000000000, 1.375000000000000] 残差 0.245758912526071
(i = 6) [1.343750000000000, 1.375000000000000] 残差 0.038826270741993
(i = 7) [1.343750000000000, 1.359375000000000] 残差 0.104560935900508
(i = 8) [1.351562500000000, 1.359375000000000] 残差 0.033145030229402
(i = 9) [1.355468750000000, 1.359375000000000] 残差 0.002770684097975
(i = 10) [1.355468750000000, 1.357421875000000] 残差 0.015204592856266
(i = 11) [1.356445312500000, 1.357421875000000] 残差 0.006221317340325
(i = 12) [1.356933593750000, 1.357421875000000] 残差 0.001726408364700
(i = 13) [1.357177734375000, 1.357421875000000] 残差 0.000521864805256
(i = 14) [1.357177734375000, 1.357299804687500] 残差 0.000602340029376
(i = 15) [1.357238769531250, 1.357299804687500] 残差 0.000040254676435
(i = 16) [1.357269287109375, 1.357299804687500] 残差 0.000240800798071
(i = 17) [1.357269287109375, 1.357284545898438] 残差 0.000100271994263
(i = 18) [1.357269287109375, 1.357276916503906] 残差 0.000030008392281
(i = 19) [1.357269287109375, 1.357273101806641] 残差 0.000005123208735
(i = 20) [1.357271194458008, 1.357273101806641] 残差 0.000012442575109
(i = 21) [1.357271194458008, 1.357272148132324] 残差 0.000003659679020
(i = 22) [1.357271194458008, 1.357271671295166] 残差 0.000000731765899
(i = 23) [1.357271432876587, 1.357271671295166] 残差 0.000001463956300
(i = 24) [1.357271432876587, 1.357271552085876] 残差 0.000000366095135
(i = 25) [1.357271432876587, 1.357271492481232] 残差 0.000000182835397
(i = 26) [1.357271462678909, 1.357271492481232] 残差 0.000000091629863
(i = 27) [1.357271462678909, 1.357271477580070] 残差 0.000000045602768
(i = 28) [1.357271470129490, 1.357271477580070] 残差 0.000000023013550

解の存在する区間幅 [1.357271470129490, 1.357271477580070]
根の近似値 1.357271473854780
残差 0.000000023013550



$$|f(c_0)| = |f(1)|$$
$$= |1^6 - 7 \cdot 1^4 + 11 \cdot 1^3 - 10|$$
$$= |-5| = 5$$



- 毎ステップ必ず絶対誤差・残差が減少するわけではない。

- このケースでは絶対誤差と残差が似た動きをしている

  残差が小さければ絶対誤差も小さい