

# 2021 年度 システムプログラミング実験

## 第 1 回 テクニカルライティング 実験レポート（模範解答）

実験日：2021 年 10 月 14 日（木）

提出日：2021 年 10 月 14 日（木）

学修番号：10194040

氏名：柴田祐樹

班名：Z

共同実験者：下川原英理

## 1 はじめに

本書ではシステムプログラミング実験第一回の課題を実施した結果を報告する。課題は関数の収束値を計算機を用いて解析するものである。解析結果をグラフと数値により示し、その結果について考察を行う。

## 2 実験の説明

本実験では以下の関係式の収束性を計算機により確かめる。

$$\lim_{x \rightarrow 0} \frac{\sin x}{x} = 1 \quad (1)$$

この式が正しいことの証明を付録に記載してある。これを計算機により確かめる方法をここで述べる。

数値実験により、徐々に 0 へ近づいていく  $x$  に対する  $x^{-1} \sin x$  の値を計算し、記録する。ここで、計算機でこれを実現するために、どのように  $x$  を小さくしていくか、という法則を明示する必要がある。これに関して次に説明する。

繰り返し回数を  $n = 1, 2, \dots, N$  とする。 $x$  の点列  $x_1, x_2, \dots, x_N$  として以下式 (2) の等比数列を用いて、 $x \rightarrow 0$  の極限を模擬することとする。

$$x_{n+1} = 0.9x_n, n = 1, 2, \dots, N - 1 \quad (2)$$

ただし、 $x_1 = 3$  とする。 $3 < \pi$  であり、 $\sin x$  の周期性を加味すればこの値から実験を始めれば十分である。式 (1) 左辺の分数は  $x = 0$  で不定形となるが、上記等比数列では  $x = 0$  とならずかつ漸近的に  $x$  が 0 へ近づくため、今回の目的に適していると考ええる。

式 (2) の数列に対し、以下式 (3) で定義される  $f_n$  を計算する。

$$f_n = \frac{\sin x_n}{x_n} \quad (3)$$

このようにして生成したデータ点の 2 次元グラフによる可視化と、実際の数値の値から結果を考察する。

## 3 実験結果

前節で説明した方法を C++ 言語<sup>\*1</sup>により実装した。実験環境の仕様を次に示す。

- Central Processing Unit: AMD Ryzen Threadripper 3970X<sup>\*2</sup>, 4.5 GHz
- 主記憶: Double Data Rate 4 Synchronous Dynamic Random-Access Memory<sup>\*3</sup>, 1200 MHz, 256 bit
- コンパイラ: g++ version 11.1.0<sup>\*4</sup>
- コンパイルコマンド: g++ -std=c++14 x2.cpp (x2.cpp はソースプログラムのファイル名である。)

---

<sup>\*1</sup> <https://isocpp.org/std/the-standard>

<sup>\*2</sup> <https://www.amd.com/en/products/cpu/amd-ryzen-threadripper-3970x>

<sup>\*3</sup> <https://www.jedec.org/standards-documents/docs/jesd79-4a>

<sup>\*4</sup> <https://gcc.gnu.org/>

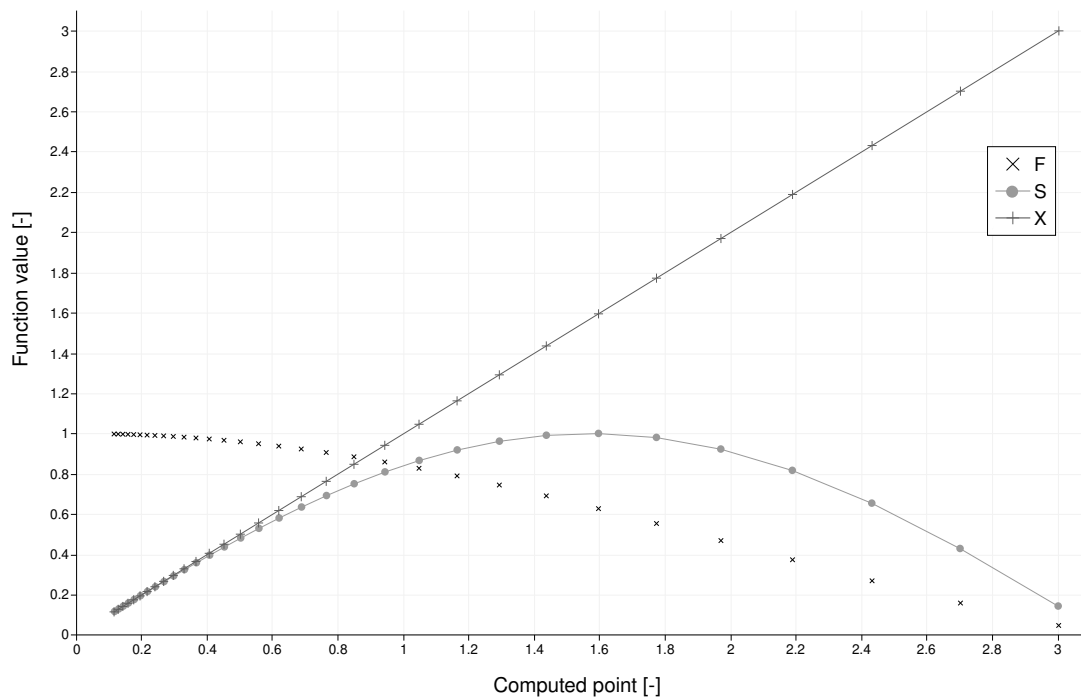


図1 実験結果のグラフ。Fは式(3)で定義される $f_n$ , Sは $\sin x_n$ , Xは $x_n$ の値を縦軸 $y$ ,  $x_n$ の値を横軸 $x$ として,  $n = 1, 2, \dots, N$ に対しそれぞれ表示している。

- Operating System: Arch Linux<sup>\*5</sup> (カーネルは 5.14.8-arch1-1 を使用)
- 数値型: 倍精度浮動小数点数<sup>\*6</sup>

式(2)と式(3)を $x_1 = 3, N = 32$ として計算した結果を図1に示す。グラフ中の座標を $(x, y)$ としたとき、図中のFは $(x_n, f_n)$ , Sは $(x_n, \sin x_n)$ , Xは $(x_n, x_n)$ を描画したものである。図1より、 $x$ が0に近づくに従い、 $x^{-1} \sin x$ の値が1へ近づくことが定性的に確認できる。この図からどれほど1へ近づいているを定量的に読み取ることは困難であるため、実際の計算値を見る必要がある。この計算値を表1に示す。有効数字は5桁としている。この表から、計算の範囲内で1を超えることなく単調に $f_n$ は1へ近づいていることを確認できる。

課題では $N = 32$ と $x_1 = 3$ と設定した実験を行うように指定されていたが、より細かく見るために本実験では $N = 64$ ,  $x_1 = 3, -3$ とした2つの実験をそれぞれ追加で行った。負の値を調べるのは $x \rightarrow -0$ の場合を検証するためである。また、より0に近い点を評価するために式(2)の公比を0.85とした。図2に、この結果を示す。図の凡例と座標系の見方は図1と同様である。

図2より、 $x$ が正負どちらから0へ近づいたとしても、 $x^{-1} \sin x$ の値が1へ近づくことが定性的に確認できる。つぎに、定量的にこの傾向を見るために計算値を確認する。この計算値を表2へ示す。有効数字は6桁としている。表では $n = 29, 30, \dots, 46$ の結果についてのみ記載し、他の範囲については冗長であるため記載していない。 $n \geq 47$ の範囲では $|1 - f_n| < 10^{-6}$ となり常に単調に $f_n$ が1へ近づいていることを確認している。また、 $x_n \approx 10^{-3}$ のとき $f_n \approx 10^{-6}$ となっていることが表より確認できる。

<sup>\*5</sup> <https://archlinux.org/>

<sup>\*6</sup> <https://www.gnu.org/software/gsl/doc/html/ieee754.html>

表 1 計算結果. 式 (2), 式 (3) で定義される  $x_n [-], f_n [-]$  の値を  $n = 1, 2, \dots, N$  について記載している.

$x_n$	$f_n$	$x_n$	$f_n$	$x_n$	$f_n$	$x_n$	$f_n$
3.0000	0.047040	1.2914	0.74433	0.55591	0.94929	0.23930	0.99048
2.7000	0.15829	1.1623	0.78959	0.50032	0.95880	0.21537	0.99229
2.4300	0.26874	1.0460	0.82736	0.45028	0.96655	0.19383	0.99375
2.1870	0.37315	0.94143	0.85869	0.40526	0.97285	0.17445	0.99494
1.9683	0.46844	0.84729	0.88457	0.36473	0.97798	0.15700	0.99590
1.7715	0.55318	0.76256	0.90586	0.32826	0.98214	0.14130	0.99668
1.5943	0.62705	0.68630	0.92333	0.29543	0.98552	0.12717	0.99731
1.4349	0.69049	0.61767	0.93762	0.26589	0.98826	0.11446	0.99782

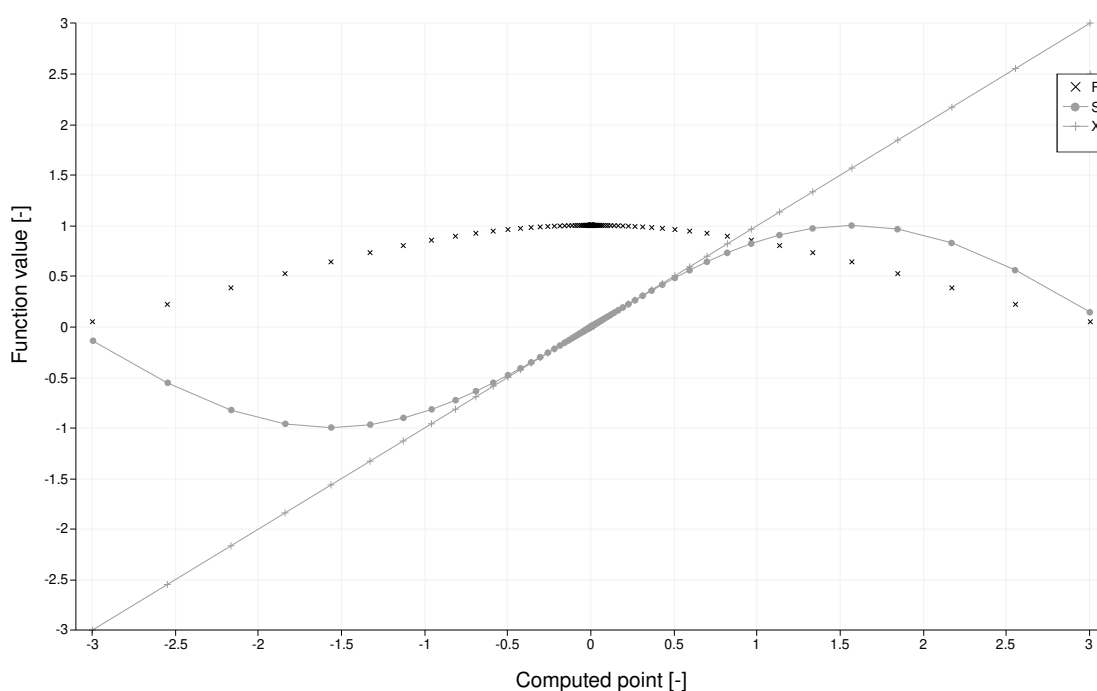


図 2 追加実験の結果に対するグラフ. F は式 (3) で定義される  $f_n$ , S は  $\sin x_n$ , X は  $x_n$  の値を縦軸  $y$ ,  $x_n$  の値を横軸  $x$  として,  $n = 1, 2, \dots, N$  に対しそれぞれ表示している.

## 4 考察

実験結果より部分的にはあるが, 式 (1) が満たされる傾向を確認した. 追加実験により,  $x \approx 10^{-3}$  において  $x^{-1} \sin x \approx 10^{-6}$  となっていることから, 収束速度は  $O(x^2)$  であると考えられる. 実際に, これは式 (1) の分数項の分母分子に 3 次の項までのテイラー級数展開を用いて, この項を変形すると,  $x = 0$  付近において,

$$\frac{\sin x}{x} = \frac{x - \frac{1}{6}x^3}{x} = 1 - \frac{1}{6}x^2 = 1 + O(x^2) \quad (4)$$

となることから, 妥当であると考え.

式 (1) の厳密な証明は, すべての  $x = 0$  付近の値に対する検証を必要とするため [1], 本実験の結果

表2 追加実験の結果. 式 (2), 式 (3) で定義される  $x_n [-]$ ,  $f_n [-]$  の値を左 2 列は  $x_1 = -3$  として計算した結果を  $n = 29, 30, \dots, 46$ , 右 2 列は  $x_1 = -3$  として  $n = 29, 30, \dots, 46$  について計算した結果を記載している.

$10^{-2}x_n$	$f_n$	$10^{-2}x_n$	$f_n$	$10^{-2}x_n$	$f_n$	$10^{-2}x_n$	$f_n$
-3.16848	0.999833	-0.733874	0.999991	3.16848	0.999833	0.733874	0.999991
-2.69321	0.999879	-0.623793	0.999994	2.69321	0.999879	0.623793	0.999994
-2.28923	0.999913	-0.530224	0.999995	2.28923	0.999913	0.530224	0.999995
-1.94584	0.999937	-0.450690	0.999997	1.94584	0.999937	0.450690	0.999997
-1.65497	0.999954	-0.383087	0.999998	1.65497	0.999954	0.383087	0.999998
-1.40587	0.999967	-0.325624	0.999998	1.40587	0.999967	0.325624	0.999998
-1.19499	0.999976	-0.276780	0.999999	1.19499	0.999976	0.276780	0.999999
-1.01574	0.999983	-0.235263	0.999999	1.01574	0.999983	0.235263	0.999999
-0.863381	0.999988	-0.199974	0.999999	0.863381	0.999988	0.199974	0.999999

はこの式が成立することを十分に示すものとは言えない. しかしながら, この実験はどのような関数の収束性についても, それが計算可能であるならば評価することができるものであり, 厳密な証明への動機を与えることができることから, 有用なものであると考える.

## 5 おわりに

本書ではシステムプログラミング実験第一回の課題を実験により検証した結果を報告した. 実験結果から, 定性的に, また部分的ではあるが定量的に関数の収束性を確かめることが可能であると示した. 今後, より自明でない, 例えば微分方程式の解の収束性などを検証し, この実験のさらなる有用性を示せると考える. 本書の執筆に, ソースコードの作成を含めて 3 時間を要した. ただし付録は除く.

## 参考文献

- [1] 松坂和夫: 解析入門 上, 第 1 版, 岩波書店, 2018

## 6 付録

図 3 に課題の数値実験に用いたソースコードを示す. また, 図 4 に追加実験で用いたソースコードを示す.

### 6.1 与式の収束性の証明

式 (1) の収束性は,  $\sin x$  の  $x$  についての微分が  $\cos x$  であることを示すために必要である. この証明には三角形 2 つと円弧を用いた証明がよく知られているが, ここでは, テイラー展開を用いた三角関数の定義による証明を記載する.  $\exp x$  は以下の級数により表すことが可能である.

$$\exp x = 1 + x + \frac{1}{2}x^2 + \frac{1}{3!}x^3 + \dots, x \in \mathbb{R}$$

$x \mapsto ix$  と置き換えれば,

$$\exp(ix) = 1 + ix + \frac{1}{2}x^2 - \frac{1}{3!}ix^3 + \cdots, x \in \mathbb{R}$$

となる. ここで,  $i^2 = -1$  は虚数単位である. 上記の実部を  $\cos x$ , 虚部を  $\sin x$  として定義する. これにより, 式 (1) は,

$$\lim_{x \rightarrow 0} \frac{\sin x}{x} = \lim_{x \rightarrow 0} \frac{x - \frac{1}{3!}x^3 + \frac{1}{5!}x^5 + \cdots}{x} = \lim_{x \rightarrow 0} \left[ 1 - \sum_{i=1} \frac{(-1)^{2i} x^{2i}}{(2i+1)!} \right] = 1$$

のように変形される. よって, 式 (1) が示される.

テイラー展開による三角関数の定義が幾何学的定義と同等であることを示す.  $\cos^2 x + \sin^2 x = 1$  は  $(a+ib)(a-ib) = a^2 + b^2$  の性質から, 次のように示される.

$$\cos^2 x + \sin^2 x = (\cos x + i \sin x)(\cos x - i \sin x) = \exp(ix) \exp(-ix) = 1$$

これより,  $\forall x \in \mathbb{R} (\sin x, \cos x \in [-1, 1])$  とわかる. 加法定理は以下のように示される.

$$\cos(a+b) + i \sin(a+b) = \exp i(a+b) = \exp(ia) \exp(ib) = (\cos a + i \sin a)(\cos b + i \sin b),$$

すなわち,

$$\cos(a+b) + i \sin(a+b) = \cos a \cos b - \sin a \sin b + i(\sin a \cos b + \cos a \sin b)$$

となるため恒等式から,  $\sin, \cos$  についてのそれぞれの加法定理が得られる. 周期性については,

$$\exp(ix) = -1, 0 > x \tag{5}$$

を満たす最小の解を<sup>\*7</sup>  $x = \pi$  とすれば,

$$-1 = \exp(i\pi) = \cos \pi, 0 = \sin \pi$$

となり, 整数  $n$  に対し,

$$\exp i(x + 2\pi n) = \exp(2i\pi n) \exp(ix) = [\exp(i\pi)]^{2n} \exp(ix) = (-1)^{2n} \exp(ix) = \exp(ix)$$

であるから, すなわち,

$$\cos(x + 2\pi n) + i \sin(x + 2\pi n) = \cos x + i \sin x$$

となるため, 恒等式を解けば,  $\sin, \cos$  それぞれの周期性が得られる.  $\pi$  が式 (5) を満たす最小の解であることから,  $\sin(\pi/2) = 1$  など得られる.

---

<sup>\*7</sup> 厳密解は得られない. ニュートン法などを使う.

---

```
1 #include <iostream>
2 #include <fstream>
3 #include <cmath>
4 #include <iomanip>
5 using namespace std;
6 int main(){
7
8     double x = 3;
9     double r = 0.9;
10    int N=32;
11    ofstream f_out("x2.csv");
12    f_out << setprecision(5);
13    f_out << "x,F,X,sin\n";
14    for(int i=0;i<N;++i){
15        double s = sin(x);
16        f_out << x << ", " << s/x << ", " << x << ", " << s << "\n";
17        x = r*x;
18    }
19    return 0;
20 }
```

---

図3 課題の数値実験に用いたソースコード.

---

```

1 #include <iostream>
2 #include <fstream>
3 #include <cmath>
4 #include <iomanip>
5 using namespace std;
6 int main(){
7
8     double x = 3;
9     double r = 0.85;
10    int N=64;
11    ofstream f_out("x2-2.csv");
12    f_out << setprecision(6);
13    f_out << "x,F,X,sin\n";
14    for(int i=0;i<N;++i){
15        double s = sin(x);
16        f_out << x << ", " << s/x << ", " << x << ", " << s << "\n";
17        x = r*x;
18    }
19    f_out = ofstream("x2-3.csv");
20    f_out << setprecision(6);
21    f_out << "x,F,X,sin\n";
22    x = -3;
23    for(int i=0;i<N;++i){
24        double s = sin(x);
25        f_out << x << ", " << s/x << ", " << x << ", " << s << "\n";
26        x = r*x;
27    }
28    return 0;
29 }

```

---

図4 追加実験に用いたソースコード.