

# 現象と数学 レポート課題

桂田 祐史

2020年9月2日公開, 2020年9月3日 11:50 更新

<http://nalab.mind.meiji.ac.jp/~mk/report-kadai-katsurada/>

(2020/9/3 11:50 加筆) [http://nalab.mind.meiji.ac.jp/~mk/gensyou20200903\\_teisei/](http://nalab.mind.meiji.ac.jp/~mk/gensyou20200903_teisei/)にも目を通して下さい。

レポートは Oh-o! Meiji に単一の A4 サイズ PDF で提出して下さい。

PDF は自分のやりやすい方法で作って構いません。手書きしたものをスキャンして PDF 化しても良いです。

参考: 「授業の提出物を PDF 形式で用意する方法」<sup>1</sup>

(1) 次の (a), (b) のどちらか (または両方).

(a) 人間の計算の速さと、現在使われているコンピューター (自分が持っているものでも、報道等で紹介されるものでも良い) の計算の速さを比べなさい。これが正解というものはないです (人間やコンピューターの選び方にもよるし、計算の種類にもよるし)。比べ方によって、簡単に 2,3 桁違った結果になると思われるので、その値の正確さで評価するつもりはありません。自分で何をどうやって調べて考察して得た結論かが大事です。答えて下さい、というよりは、個人的な調査結果を聞かせて下さい、という感じです。

(b) この授業を聴いて、数値計算についての考えに変化が生じたら、それを説明して下さい。

(2) 二分法の C++ プログラム `bisection.cpp`<sup>2</sup> を読んで実行し、アルゴリズムと実行結果の説明をしなさい。また自分が選んだ方程式を解くようにプログラムを書き換えて<sup>3</sup>、その結果の説明をしなさい。

`interval.hpp` のあるディレクトリで、以下の 3 コマンドを実行する

```
curl -O http://nalab.mind.meiji.ac.jp/~mk/misc/20200903/bisection.cpp;  
c++ bisection-skv.cpp;  
./a.out
```

実行すると、`探す区間の左端  $\alpha$  , 右端  $\beta$  , 許容精度  $\epsilon =$`  と表示されるので、例えば `0 1 1e-14` と入力して enter キーを押して下さい (1e-14 は  $10^{-14}$  という意味です)。

(セミコロン ; は本来不要ですが、コピペするとき、行がくっついてしまう事故が起こることを考えて入れています。)

<sup>1</sup>[http://nalab.mind.meiji.ac.jp/~mk/how\\_to\\_pdf/](http://nalab.mind.meiji.ac.jp/~mk/how_to_pdf/)

<sup>2</sup><http://nalab.mind.meiji.ac.jp/~mk/misc/20200903/bisection.cpp>

<sup>3</sup>C++ 言語で書かれているけれど、方程式を変えるだけならば、C++ の知識はほぼ不要です。

**2020/9/3 8:40 加筆** プログラム中に `if (in(0,fb))` のように、関数 `in()` を使っているところがあります。これは、「区間 `fb` に `0` が属していれば」ということです。 $f(b)$  の値が `0` に近く (`0` かもしれない)、正であると保証することも、負であると保証することも出来ない、ということです。

`interval.hpp` を入手するには (授業資料にも書いたけれど)

ターミナルで以下の2つのコマンドを実行

```
curl -O http://verifiedby.me/kv/simple/interval-simple-0.4.48.tar.gz;
tar xzf interval-simple-0.4.48.tar.gz
```

を実行します。`interval-simple-0.4.48.tar.gz` は柏木先生のサイトにあるわけですが、何か障害が発生することもありますね。そのときは次のようにして下さい。

```
curl -O http://nalab.mind.meiji.ac.jp/~mk/misc/20200903/interval-simple-0.4.48.tar.gz;
tar xzf interval-simple-0.4.48.tar.gz
```

**2020/9/3 8:20 加筆**  $\cos x - x = 0$  という方程式についての二分法について、C 言語を使う授業でネタにしていたことがあります。講義動画で時間を気にして少し舌足らずだったので、補足しておきます。

- この方程式はただ1つの実数解を持つこと。  
<http://nalab.mind.meiji.ac.jp/~mk/syori2-2005/jouhousyori2-2005-11/node6.html>  
(自分で方程式  $f(x) = 0$  を選ぶときに、 $f$  のグラフくらいは描いておきましょう。)
- 例えば C 言語で二分法のプログラムを書くとうどうなるか。今回の C++ プログラムとは (区間演算以外にも) ちょっと違っていますが参考まで。

<http://nalab.mind.meiji.ac.jp/~mk/syori2-2005/bisection.c>

(この C のプログラムは区間演算していないので、符号の判定を間違えることがあります。そういう意味では厳密には正しく動かない。)

**2020/9/3 8:35, 9:10 加筆** C++ について補足説明しておこうか、という意味のことを口走ったことを思い出しました。

- ソースプログラムの名前の末尾が `.c` ではなく、`.cc` や `.cpp` であることが多い。コンパイルには、`cc` や `gcc` でなく、`c++` や `g++` を使う。
- 入出力として `printf()`, `scanf()` のような関数を使う代わりに、`cout <<` や `cin >>` を使う。  
それもあって、`#include` も違ってきます。
- 今回の `interval` のような型をユーザーが定義できるような仕組みがある。(そのおかげで、例えば区間の足し算、掛け算、大小比較などが `+`, `*`, `>` などのできる。)

ただし、今回の課題を行うのに、そのあたりを書き換える必要はほとんどないはずですが。

書き換えるとしたら関数 `f()` の中身の式くらいでしょう。そういうところは C 言語と同じように書ける場合が多いと思います。

**2020/9/3 10:00 加筆** `interval-simple-0.4.48.tar.gz` には、`interval.hpp` 以外に、`test-interval.cc` というサンプル・プログラムがついています。これを見ると、`interval.hpp` の使い方が分かります。この解説メモをつけておきます。

<http://nalab.mind.meiji.ac.jp/~mk/misc/20200903/test-interval-kaisetsu/node9.html>