現象数理1（2020年度 第12回授業）

黒岩大史 kuroiwa@riko.shimane-u.ac.jp

今回の授業日と課題〆切

対面：12月24日　　オンデマンド推奨期間：12月17日〜1月13日　　課題〆切：1月13日

非線形方程式

今回と次回の授業では、方程式 $f(x)=0$ の解法を取り扱います。

2次方程式 $ax^2+bx+c=0$ の解は解の公式により

$$x=\frac{-b\pm\sqrt{b^2-4ac}}{2a}$$

で求められます。3次方程式はカルダーノ (G.Cardano) の公式、4次方程式はフェラリ (L.Ferrari) の公式によって解を求めることが出来ます。一般に、n 次方程式

$$a_nx^n+a_{n-1}x^{n-1}+\cdots+a_1x+a_0=0\quad(a_n\neq0)$$

は、複素数の範囲で n 個 (重複を考慮に入れて) の解を持つことが知られていますが、しかしながら、5次以上の代数方程式に対する解の公式（四則演算と冪根 (べきこん) の有限回の組合せによるもの）は存在しません（この事実は、アーベル (H.L.Abel) とガロア (E.Galois) によって独立に示されています)。つまり、5次以上の方程式を解く場合には、

解は存在するのに、解の公式が無い !!

のです。しかし「近似解」であれば、数値解析的に（誤差を理論的に抑えながらコンピュータを用いて）求めることができる場合があります。今回と次回では、非線形方程式 $f(x)=0$ の近似解法のうち、二分法とニュートン法について学習します。

1　二分法

二分法は、中間値の定理に基づいたアルゴリズムです。中間値の定理の証明には、実数の性質と関数の連続性が本質的に効いています（ここでは、定理の証明は省きます)。$\emptyset\neq I\subset\mathbb{R}$ のとき、関数 $f:I\to\mathbb{R}$ が連続であるとは、

$$\forall x\in I,\ \forall\varepsilon>0,\quad \exists\delta>0\ \text{ s.t. }\ \forall x'\in I(|x'-x|<\delta),\ |f(x')-f(x)|<\varepsilon$$

が成り立つときをいいます。

中間値の定理

$f:[a,b]\to\mathbb{R}$ が連続であるとする。もし $f(a)f(b)<0$ ならば、区間 $[a,b]$ 内に $f(x)=0$ の解が存在する。

※ $f(a)f(b)<0$ というのは、$f(a)<0<f(b)$ または $f(b)<0<f(a)$ のどちらかの場合です。

この定理に基づいて $f(x)=0$ に収束する数列を作り、$f(x)=0$ の近似解を求めるアルゴリズムを述べます。以下では $f:[a,b]\to\mathbb{R}$ は連続、かつ $f(a)<0<f(b)$ の場合のみ述べます（$f(b)<0<f(a)$ の場合も同様にできますが省略します）。また、停止条件に関わる $\varepsilon>0$ は事前に与えておきます。

アルゴリズム (二分法)

(I) $k=1$, $a_1=a$, $b_1=b$, $c_1=\dfrac{a+b}{2}$ とおき (II) へ。

(II) $f(c_k)\neq 0$ かつ $2\varepsilon<b_k-a_k$ ならば (III) へ。そうでなければ (IV) へ

(III)
- $f(c_k)>0$ ならば $a_{k+1}=a_k$, $b_{k+1}=c_k$ とおき、
- $f(c_k)<0$ ならば $a_{k+1}=c_k$, $b_{k+1}=b_k$ とおく。

そして k の値を 1 増やし、$c_k=\dfrac{a_k+b_k}{2}$ とおく。(II) へ。

(IV) c_k を $f(x)=0$ の近似解とする。

重要！（以下は授業中に説明する）

1. まずはこのアルゴリズムを理解するため、$f(a)<0<f(b)$ となる連続関数 $f(x)$ と a,b を与え、グラフを描いて下さい。(I) から順に実施し、a_k,b_k,c_k を x 軸上に取ってみましょう。その際、必ず $f(a_k)<0<f(b_k)$ が成立することを確認して下さい。
2. このアルゴリズムは必ず停止します。つまり、ある自然数 k において $b_k-a_k\leq 2\varepsilon$ となります。その理由を説明するため、一般に b_k-a_k が a,b を用いてどのように表されるか答えて下さい。
3. このアルゴリズムが停止した際の近似解 c_k は、区間 $[a,b]$ 内のある真の解 $\bar{x}$ との差の絶対値が ε 以下であることが判ります。すなわち、

$$\exists\bar{x}\in[a,b]\ \text{s.t.}\ f(\bar{x})=0\ \text{かつ}\ |c_k-\bar{x}|\leq\varepsilon$$

が判ります。それはなぜかを答えて下さい。

※上のアルゴリズムでは数列 $\{a_k\}$, $\{b_k\}$, $\{c_k\}$ を作るような説明にしています。Python でもリストを

用いてプログラミングすることも出来ますが、リストを使ってもメリットがない（後で途中の点を持ち出す必要がない）ため、以下の例題のようにリストを使わずにプログラミングした方が良いです。

例題 1. 二分法を用いて、区間 $[0,2]$ にある $x^2=3$ の近似解を求めて下さい。ただし、真の解と近似解の差の絶対値は $0.000001(=10^{-6})$ 以下となるようにして下さい。

[解き方] `f(x)=x*x-3` となるように関数を定め、二分法のアルゴリズムに沿ってプログラミングします。`k` についての部分は無くても動作しますが、終了時の繰り返し回数を知るために付けています。

2020-12-1.py

```
def f(x):
   return x*x-3
k = 1
a = 0
b = 2
c = (a + b) / 2
e = 0.000001
while f(c)!=0 and 2*e<b-a:
   if f(c) > 0:
      b = c
   else:
      a = c
   k = k + 1
   c = (a + b) / 2
print(c,k)
```

練習 1. 二分法のアルゴリズムを用いて、以下のプログラムを作成して下さい。

1. 区間 $[2,3]$ には $x^2=5$ の解が一つだけ存在します。この解に対して差の絶対値が 0.000001 以下であるような近似解を求め表示するプログラムを作成して下さい。(`2020-12-2.py`)
2. $\sqrt[3]{5}$ の近似解 x のうち、$|\sqrt[3]{5}-x|\leq 0.000001$ となるものを一つ見つけ、表示するプログラムを作成して下さい。(ヒント：$\sqrt[3]{5}$ がどのような多項式の解であるかを求め、解が含まれる区間を決定せよ。`2020-12-3.py`)
3. 誕生月日から次のようにして作られる多項式の近似解 x （ただし誤差の絶対値は 0.000001 以下）を一つ見つけ、表示するプログラムを作成して下さい。(`2020-12-4.py`)

$$x^5+x^4+2x^3-4x^2+(\text{誕生月})x-(\text{誕生日})=0$$

今日のまとめ

- 二分法のアルゴリズムとアイディアを理解し、練習 1 が出来ること。
- 二分法のアルゴリズムに関する「重要！」（p. 2）の部分を理解し、説明できること。

2 確認問題・発展問題

(1) は今回の復習で、第 14 回授業時に問われるような形式で書かれた問題ですので、理解度・定着度を確認しながら問いてみて下さい。提出の際、筆記部分の (1)(b)〜(d) は写真に撮って 1 ファイルの PDF にして提出して下さい。(2)、(3) は発展問題です。余裕があればトライしてみて下さい。(3) は二分法で用いたようなアイディアを用いることで、最大 7 回程度で終わるようプログラミングすることが出来ます。

(1) 次のアルゴリズムについて、以下の問いに答えよ。

STEP1: $i=1$、$a=1$、$b=2$、$c=(a+b)/2$ とする。STEP2 へ進む。

STEP2: $b-a \geq 2\cdot 10^{-6}$ ならば STEP3 へ進み、そうでなければ STEP4 へ進む。

STEP3: まず i、a、b の値を画面に表示する。そして $c^2-3 \leq 0$ ならば a に c の値を代入し、$c^2-3>0$ ならば b に c の値を代入する。$c=(a+b)/2$ とし、i の値を 1 増加させ、STEP2 へ進む。

STEP4: i と c の値を表示し、プログラムを終了する。

(a) このアルゴリズムに従って動作するプログラムを作成して下さい。(`2020-12-5.py`)

(b) このアルゴリズムが必ず停止する理由を述べて下さい。

(c) STEP4 で表示される i の値を、$\log_{10} 2 = 0.3010$ として求めて下さい。

(d) このアルゴリズムが停止した際、$|c-\sqrt{3}| \leq 0.000001$ であることが判ります。それはなぜかを答えて下さい。

(2) 人とコンピュータでジャンケンをするプログラムを作成して下さい。0 をグー、1 をチョキ、2 をパーとし、人は `a=int(input())` でじゃんけんの手をキーボードから入力して下さい。プログラムの先頭に「`import random`」を書き、`b=random.randint(0,2)` を実行すると、0 か 1 か 2 がランダムに得られるので、これをコンピュータ側のじゃんけんの手と考え、プログラミングして下さい。(`2020-12-6.py`)

```
ジャンケンポン (グー 0, チョキ 1, パー 2):2
コンピュータは 2 です。
あいこです。あいこでしょ(グー 0, チョキ 1, パー 2):1
コンピュータは 1 です。
あいこです。あいこでしょ(グー 0, チョキ 1, パー 2):0
コンピュータは 2 です。
コンピュータの勝ちです。
```

(3) まずは以下のプログラムを実行して下さい。これは、配列 `x` に入る数字を（前の数に 1 か 2 を加え、単調増加になるように）決めた後、キーボードから入力した数 `n` に対して `x[i]=n` をみたす `i` が

あるかどうかを調べるプログラムです。このプログラムでは、このことをチェックするために合計で100回のif文を実行していることになりますが、この配列x[i]に入力されている数が単調増加であることを考慮すると、あまり良いアルゴリズムではありません。if文の実行回数の期待値を少しでも小さくする方法を考え、10行目以降を書き直して下さい。(2020-12-7.py)

※if文の実行回数が最大で7回程度となる方法があります。

```
import random
x = [0]*100
x[0]=1
for i in range(1,100):
   x[i]=x[i-1]+random.randint(1,2)
for i in range(0,100):
   print x[i],
print
n=int(input("自然数を入力して下さい："))
#これより上は変えない
m=0
for i in range(0,100):
   if x[i]==n:
       m=i
if m>0:
   print(n,"はx(i)の中に存在します。")
else:
   print(n,"はx(i)の中に存在しません。")
```