スクリプト言語プログラミング
Pythonによる数値解析

第9回講義資料

箕原辰夫

実数による数値計算

数値計算

- 直接解法
  - 有限の回数で解を求められるもの
  - 例：ガウスの消去法、QR分解など
- 反復解法
  - 繰返しで、解に近づいていくもの
  - 例：ニュートン法、二分法など

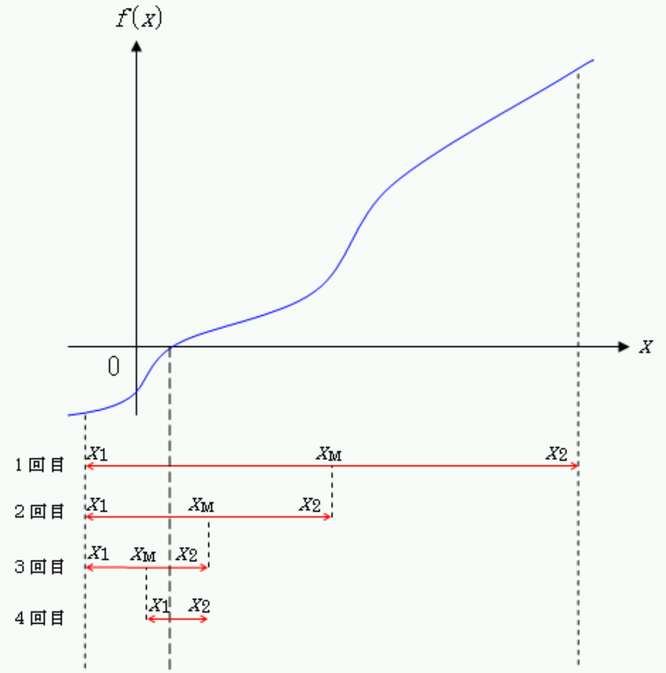
扱う数値計算

- 非線形方程式の解を求める（求根）
  - ２分法、はさみうち法、割線法
  - ニュートン法、ハーレー法
- マクローリン・テイラー展開
- 数値積分
  - ガウスの積分公式
  - 台形公式・シンプソンの公式
- 精度指定の値
  - πの値を求める

２分法

- 多項式の関数をf(x)とする。f(x) = 0のときのxの値を求める方法
- ２つのx座標を取り、f(x)の値が0より大きいか小さいかで、xの値を値の範囲を分ける⇒２分（bisection）法
- 区間a, bの間で求める
- 次のようなアルゴリズムで求める

$$a_0 = a, b_0 = b$$
$$c_{n+1} = \left(a_n + b_n\right)/2$$
$$f(c_{n+1}) > 0 \rightarrow a_{n+1} = a_n, b_{n+1} = c_{n+1}$$
$$f(c_{n+1}) < 0 \rightarrow a_{n+1} = c_{n+1}, b_{n+1} = b_n$$



はさみうち法

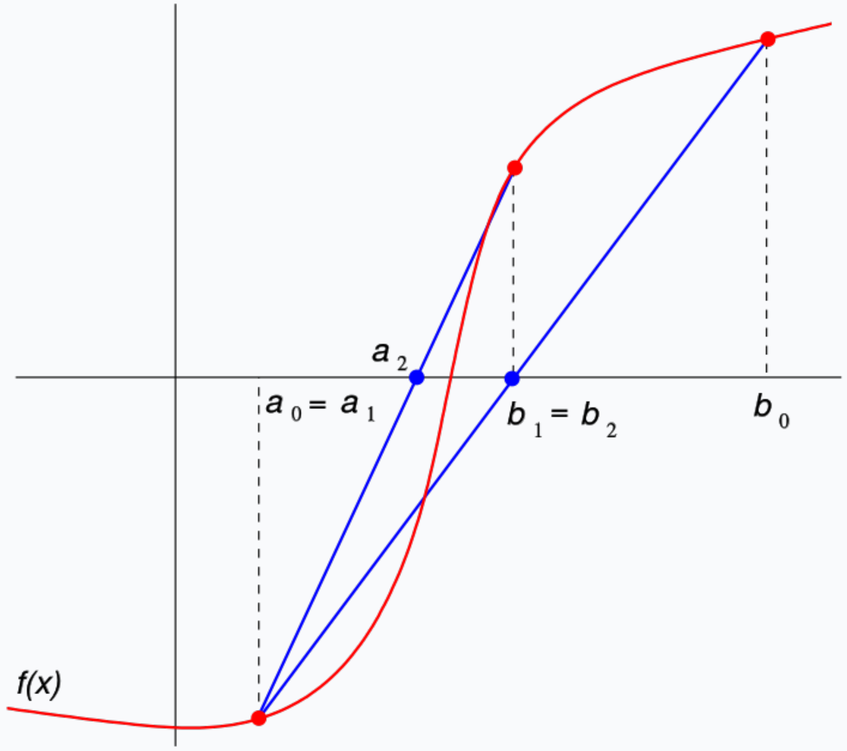
- はさみうち（squeeze）法も、区間a, bの間で求めるが、中間地点のcを求める際の式が異なる。

$$a_0 = a, b_0 = b$$

$$c_{n+1} = \frac{a_n f(b_n) - b_n f(a_n)}{f(b_n) - f(a_n)}$$

$$f(c_{n+1}) > 0 \rightarrow a_{n+1} = a_n, b_{n+1} = c_{n+1}$$

$$f(c_{n+1}) < 0 \rightarrow a_{n+1} = c_{n+1}, b_{n+1} = b_n$$

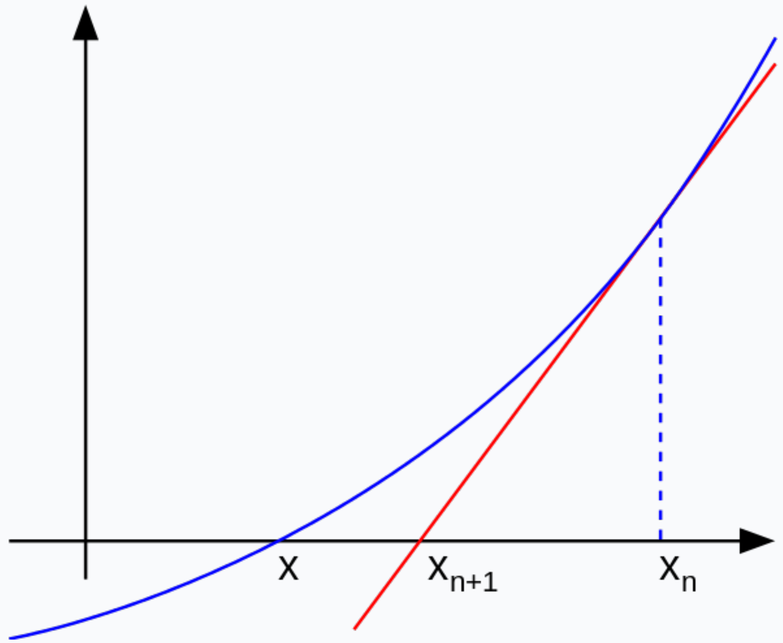


ニュートン法（ニュートン・ラフソン法）

$$f(x)=0$$

となるxの値を、以下の
数式によって漸次的に
求める方法

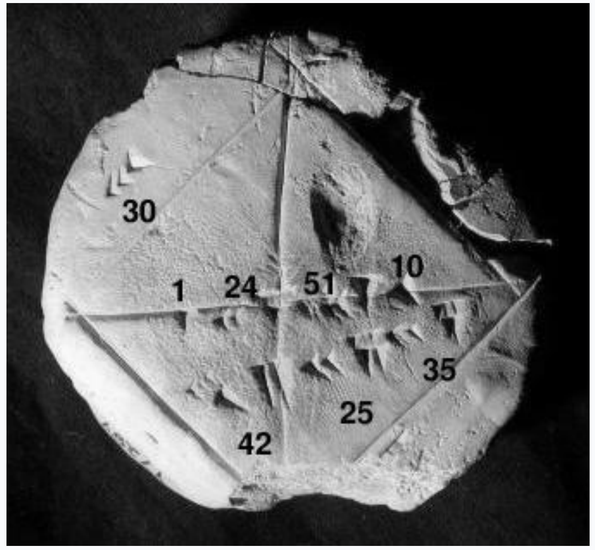
$$x_{n+1}=x_n-\frac{f(x_n)}{f'(x_n)}$$



- x_0は、どのような値でも良い。ただし解が複数あるものは、最初の値によって、どの解が求められるか決定する。ただし、関数によっては求められない場合もある。

平方根の求め方

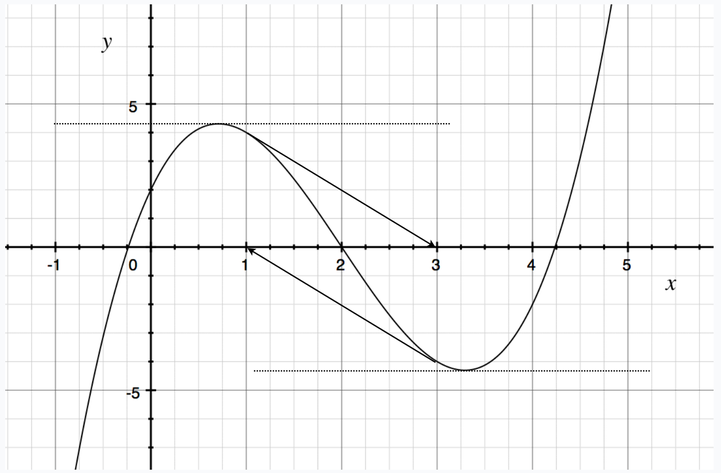
- ニュートン法でも求める
  - $x_{n+1} = x_n - f(x_n) / f'(x_n)$
- 平方根の場合
  - $f(x) = x^2 - n$
  - $f'(x) = 2x$
  - $x_{n+1} = x_n - (x_n^2 - n) / 2x_n$
  - $= x_n / 2 + n / 2x_n$
- Pythonの関数で記述する
  - **def** squareroot(n):
  - x = 1.0
  - **for** i **in** range(10):
  - x = x / 2.0 + n / (2.0 *x)
  - **return** x



バビロニアの粘土板 YBC 7289 （紀元前1800-1600年頃）
2の平方根の近似値は60進法で4桁、10進法では約6桁に相当する。1 + 24/60 + 51/602 + 10/603 = 1.41421296...。(Image by Bill Casselman)

ニュートン法で解を求められない初期値の地点

- $x^3-6x^2+7x+2=0$ の解を求めることを考える
- この3次方程式の解は、$2-\sqrt{5},2,2+\sqrt{5}$ の3つである。
- 初期値をx_0=1, あるいはx_0=3としたときには、互いの位置に振動してしまい、解は求まらない
- また、初期値を $x_0=2\pm\sqrt{\frac{5}{3}}$ としたときには、接線の傾きが水平になってしまい、解が求まらない。

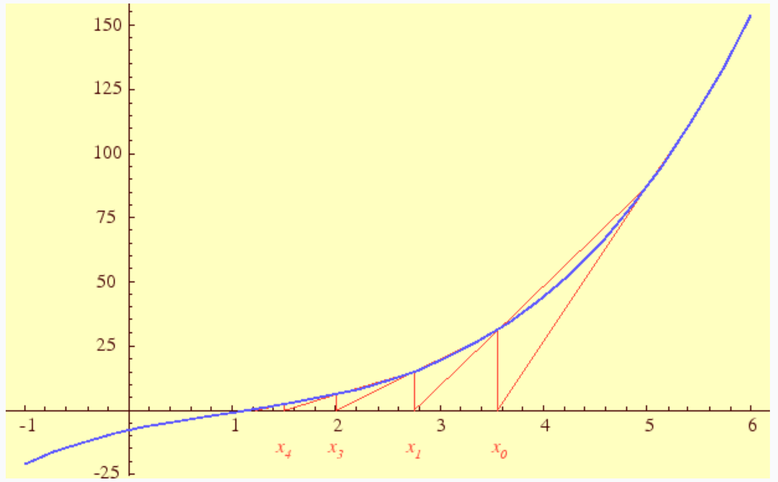


- 『よくわかる数値計算』戸川隼人他監修，日刊工業新聞社，2001年より

割線法

- ニュートン法の接線の代わりに、2点$(x_{i-1}, f(x_{i-1}))$と$(x_i, f(x_i))$を結ぶ直線（割線：secant）を使う。

$$x_{i+1} = x_i - f(x_i)\frac{x_i - x_{i-1}}{f(x_i) - f(x_{i-1})}$$



- 図と式は、「http://www.yamamo10.jp/yamamoto/lecture/2005/5E/nonlinear_equation/text/html/node6.html」より

ハレー法

- ハレー（Halley）法は、ニュートン・ラフソン法は、１階の微分に対して、２階の微分形も使って求める。

$$x_{n+1} = x_n - \frac{2f(x_n)f'(x_n)}{2[f'(x_n)]^2 - f(x_n)f''(x_n)}$$

- また、これを簡単化した次の式も用いられる。

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}\left[1 - \frac{f(x_n)}{f'(x_n)} \cdot \frac{f''(x_n)}{2f'(x_n)}\right]^{-1}$$

収束速度

- ２分法・はさみうち法…１次収束
- ニュートン法・割線法の収束速度…２次収束
- ハレー法の収束速度…３次収束

- 2分法・はさみうち法はa, bの区間内にある解が1つだけ求められる
- ニュートン法・割線法・ハレー法は、最初のx_0に近い解が1つだけ求められる

マクローリン・テイラー展開

- 微分形を用いて、関数の値を求める方法。テイラー級数を用いて，$f(x)$の値は、次のように求められる。

$$\sum_{n=0}^{\infty} \frac{f^{(n)}(a)}{n!}(x-a)^n$$

- 特に、a=0のときをマクローリン展開と呼ぶ。

$$\sum_{n=0}^{\infty} \frac{f^{(n)}(0)}{n!}x^n$$

階乗の表記など

- n ! = 1×2×3×…×n
- $\sum_{x=1}^{n} x = 1 + 2 + 3 + \ldots + n$
- $\prod_{x=1}^{n} x = 1 \times 2 \times 3 \times \cdots \times n$
- テイラー展開

$$f(x) = \sum_{n=0}^{\infty} \frac{f^{(n)}(a)}{n!}(x-a)^n = f(a) + f'(a)(x-a) + \frac{1}{2}f''(a)(x-a)^2 + \frac{1}{3}f'''(a)(x-a)^3 \cdots$$

- マクローリン展開

$$f(x) = \sum_{n=0}^{\infty} \frac{f^{(n)}(0)}{n!} = f(0) + f'(0)x + \frac{1}{2}f''(0)x^2 + \frac{1}{3}f'''(0)x^3 \cdots$$

マクローリン展開での関数値の求め方

- 関数値はマクローリン展開すると以下のように記述できる

$$e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!}$$

$$\sin x = \sum_{n=0}^{\infty} \frac{(-1)^n}{(2n+1)!} x^{2n+1}$$

$$\cos x = \sum_{n=0}^{\infty} \frac{(-1)^n}{(2n)!} x^{2n}$$

- 上記式を展開すると、次のようになる

$$\begin{aligned} e^x &= 1 + x + \frac{x^2}{2} + \frac{x^3}{3!} + \cdots \\ \sin x &= x - \frac{x^3}{3!} + \frac{x^5}{5!} + \cdots \\ \cos x &= 1 - \frac{x^2}{2!} + \frac{x^4}{4!} + \cdots \end{aligned}$$

https://eman-physics.net/math/taylor.html

マクローリン展開による関数値の求値

- 三角関数のマクローリン展開の公式

$$\sin x = \sum_{n=0}^{\infty} \frac{(-1)^n}{(2n+1)!} x^{2n+1}$$

$$\cos x = \sum_{n=0}^{\infty} \frac{(-1)^n}{(2n)!} x^{2n}$$

- 有限回の計算で求める（sin xの場合）

```
sin_x =0.0
for n in range( 100 ) :
    sin_x += ( 1 if n%2 == 0 else -1 ) / math.factorial( 2*n+1 ) * math.pow( x, 2*n+1 )
```