2 分法で方程式を解こう

樋口さぶろお

龍谷大学理工学部数理情報学科

数値計算法 L07(2010-06-04)

今日の目標

1. 公式のない方程式 $f(x) = 0$ を解く 2 分法のプログラムが書けるようになろう.
2. 方程式を解くアルゴリズムの速さを評価できるようになろう.

hig3.net

こんな問題?

方程式を解くこんな問題?

$f(a_0)$ と $f(b_0)$ は異符号．区間 (a_0, b_0) にある，$f(x) = 0$ を満たす x を求めて．ただし，$f(x)$ は連続だけど微分できない．$\rightsquigarrow$ $\boxed{\text{Newton 法はだめ}}$

区間 $(a_0, b_0) = \{x | a_0 < x < b_0\}$.
このとき解ってあるの?

中間値の定理

$f(x)$ が区間 $[a_0, b_0]$ で連続で，$f(a_0) < f(b_0)$ ならば，$f(a_0) < k < f(b_0)$ を満たす任意の値 k に対して，

$$f(c) = k, a_0 < c < b_0$$

を満たす c が少なくとも 1 つ存在する．

数学 III

対応: 　　　　　　　　 不等号は逆向きでも同じ．

2 分法 (bisection) ののり

栗原 §2.5 (a_0, b_0) から始めて，反復で (a_n, b_n) を決めていく．区間 (a_n, b_n) が短くなったら終了．
簡単のため $f(x)$ が増加関数，つまり $f(a_0) < 0 < f(b_0)$ のときに限定．
$x_n = \frac{a_n+b_n}{2}$ ぐらいかな？ と思ってみる．
場合分け

- $f(x_n) = 0$ のとき x_n が解．(ぴったりになることはまれ)
- $f(x_n) > 0$ のとき，$f(a_n)$ と $f(x_n)$ は異符号．解は (a_n, x_n) に存在．
  - $\leadsto$ 区間 $(a_{n+1}, b_{n+1}) = (a_n, x_n)$.
- $f(x_n) < 0$ のとき，$f(a_n)$ と $f(x_n)$ は同符号．解は (x_n, b_n) に存在．
  - $\leadsto$ 区間 $(a_{n+1}, b_{n+1}) = (x_n, b_n)$.

こうすればいつでも $f(a_n) < 0 < f(b_n)$ のまま．

$f(x)$ が単調とは限らないとき

以下のようにして, $f(a_n)$ と $f(b_n)$ がいつでも異符号 (か 0) であるようにできる.

場合分け

- $f(x_n)$ と $f(a_n)$ が異符号のとき. 解は (a_n, x_n) に存在.
  - $\rightsquigarrow$ 区間 $(a_{n+1}, b_{n+1}) = (a_n, x_n)$.
- $f(x_n)$ と $f(a_n)$ が同符号のとき. $f(x_n)$ と $f(b_n)$ は異符号. 解は (x_n, b_n) に存在.
  - $\rightsquigarrow$ 区間 $(a_{n+1}, b_{n+1}) = (x_n, b_n)$.

がんばれば, 数列 $\{a_n, b_n\}$ の, 場合分けによる漸化式として書けるね.

p, q が異符号の書き方

同符号も似たのりで.

2 分法の誤差と終了条件

真の解 x_*.
n 回目の反復での解の推定 $\frac{a_n+b_n}{2}$.
誤差は $\frac{a_n+b_n}{2} - x_*$.
誤差 (の最大値)=

反復の終了条件

誤差が, 指定された大きさ ϵ より小さくなったら終了
毎回, 誤差は $\frac{1}{2}$ ずつになっていく!
$b_n - a_n = 2^{-n}(b_0 - a_0)$.

2 分法のアルゴリズムの概要

今日は日本語で書いてます.

```c
double f(double x); /* 方程式は f(x)=0 */

int main(void){
  double a,b;  /* (a,b) が現在の区間 */
  double x;    /* 中点 */
  double eps=1e-10;  /*実現したい誤差の上限. 例です.*/

  a=...;
  b=...;    /* f(a)と f(b)が異符号であるような初期条件 */
  while( /*誤差がepsより大きい*/ ){
    x=(a+b)/2.0;

    if( /*f(a)と f(x) が異符号か0 */){
      /* aとxの間に解があるはずなので区間を更新 */
    } else {
      /*この場合, f(a)と f(x) が同符号,
        したがって f(x)と f(b)は異符号.
        xとbの間に解があるはずなので区間を更新 */
    }
  }
  printf("%f\n",(a+b)/2);
  return 0;
}
```

何で 3 個もやるの?

どれも反復

2 分法は

	反復法	Newton 法	2 分法
一言紹介	素朴	高速で標準的	頑健
収束性	発散することも	発散することも	
f の条件	連続	微分可能	
選択の余地	A, 初期値 x_0	初期値 x_0	
速度	遅い (1 次収束)	速い (2 次収束)	

- Newton 法でうまくいったらそれでいい.
- $f(x)$ の様子がよくわからないときや, Newton 法が収束しないときは 2 分法で様子をつかむ.

いいアルゴリズムって?

- 正確 (答えの誤差が小さい)
- 高速
- (パラメタの調節がいらない. 例外的なケースでも間違えた結果を出さない)
- エコ
  - (=時間) をたくさん使わない ($\simeq$ 高速)
  - メモリ (変数) をたくさん使わない
  - ディスクをたくさん使わない
- 人に優しい (書くのが簡単, 単純)

反復法, Newton 法, 2 分法などの, 反復により方程式の解を求める方法は, 反復の回数を増やせば正確になっていくのが普通なので, 高速 $\simeq$ 正確.

アルゴリズムの評価方法

- 数学のり
- 人海戦術のり (いろんなケースで実際やってみる)

アルゴリズムの速度の評価の準備

方程式 $f(x)=0$ を考える. $x=x_*$ が (未知の) 解だとする.
$x=x_*$ のまわりでテイラー展開. $f^{(0)}(x_*)=0$ に注意して,

$$\begin{aligned} f(x) =& \sum_{n=1}^{\infty} \frac{f^{(n)}(x_*)}{n!}(x-x_*)^n \\ =&(x-x_*)\sum_{n=1}^{\infty} \frac{f^{(n)}(x_*)}{n!}(x-x_*)^{n-1} \\ =&(x-x_*)F(x). \quad F(x) \text{ の定義} \end{aligned}$$

$F(x_*)=c$ (そこで発散してない). $F(x_*)=0$ なら f は x 軸に接してる.

$$\begin{aligned} f'(x) =& F(x)+(x-x_*)F'(x) \\ f'(x_*) =& F(x_*)+0\cdot F'(x_*) \end{aligned}$$

反復法の速度

数列 x_n が x_* に収束する '速さ' を考えたい.

漸化式 $x_{n+1} = g(x_n)$ の不動点として x_* が求まる. $g(x) = x - Af(x)$.

$$x_{n+1} - x_* = x_n - x_* - Af(x_n)$$

$$x_{n+1} - x_* = x_n - x_* - A(x_n - x_*)F(x_n)$$

誤差 $\epsilon_n = x_n - x_*$. ϵ_n が十分小さくなった後を考えると,

$$x_{n+1} - x_* \simeq (x_n - x_*)(1 - AF(x_*))$$

$$\epsilon_{n+1} = (1 - AF(x_*)) \cdot \epsilon_n$$

$$\epsilon_n = (1 - AF(x_*))^n \epsilon_0 = \epsilon_0 \mathrm{e}^{(\log(1-AF(x_*))n}$$

$|1 - AF(x_*)| = |1 - Af'(x_*)| < 1$ でないと, 誤差 $\epsilon_n \to 0$ とならない. 前回の授業

2 分法の速度

$$\epsilon_n = |b_n - a_n|.$$

$$\epsilon_{n+1} = \frac{1}{2}\epsilon_n.$$

.

p 次収束

$\epsilon_{n+1} = r \times \epsilon_n^p$

反復法と 2 分法は 1 次収束．

$p = 1 \rightsquigarrow$ 誤差が等比数列.

$$\epsilon_n = r^n \epsilon_0 = c_1 \mathrm{e}^{-c_2 n}$$

Newton 法の速度

実は $\boxed{\qquad}$ 次収束. 栗原 §2.4

$$\epsilon_{n+1} = r\epsilon_n^p.$$

$p > 1$ のとき超高速!

$$\epsilon_n = C_1 \mathrm{e}^{c_2 \mathrm{e}^{-c_3 n}}$$

アイデア
反復法って

$$\epsilon_n = (1 - AF(x_*))^n \epsilon_0$$

$1 - AF(x_*) = 0$ だったら超よくない?

$$A = \frac{1}{F(x_*)} = \frac{1}{f'(x_*)} \quad \text{つまり} \quad g(x) = x - \frac{1}{f'(x_*)} f(x)$$

あっ x_* が事前にわかってるわけじゃなかった.

$$g(x) = x - \frac{1}{f'(x)} f(x)$$

1 次収束と 2 次収束の勝負

$\epsilon_0 = 1.$

	1 次収束 $\epsilon_{n+1} = 0.1\epsilon_n$	2 次収束 $\epsilon_{n+1} = 0.1\epsilon_n^2$
ϵ_0	1	1
ϵ_1	10^{-1}	
ϵ_2	10^{-2}	
ϵ_3	10^{-3}	
ϵ_4	10^{-4}	
ϵ_5	10^{-5}	
指数部が		

Newton 法が 2 次収束であることの導出

$f(x) = (x - x_*)F(x)$. $f'(x) =$

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \qquad \text{Newton 法の漸化式}$$

$$\begin{aligned}
x_{n+1} - x_* &= x_n - x_* - \frac{(x_n - x_*)F(x_n)}{F(x_n) + (x_n - x_*)F'(x_n)} \\
&= \frac{(x_n - x_*)(F(x_n) + (x - x_*)F'(x_n)) - (x_n - x_*)F(x_n)}{F(x_n) + (x - x_*)F'(x_n)} \\
&= \frac{(x_n - x_*)^2 F'(x_n)}{F(x_n) + (x_n - x_*)F'(x_n)} \\
&\simeq \frac{F'(x_*)}{F(x_*) + 0 \cdot F'(x_*)}(x_n - x_*)^2 \quad x_n \text{ が } x_* \text{ の近く}
\end{aligned}$$

$$\epsilon_{n+1} = \frac{F'(x_*)}{F(x_*)}\epsilon_n^2$$

$\left|\frac{F'(x_*)}{F(x_*)}\right| > 1$ でも, x_0 が x_* の十分近くならダイジョーブ!

例題

$f(x) = +x^3 - 3x$ とする.

1. (必要なら増減表を書いて)$y = f(x)$ のグラフを描こう.
2. 方程式 $f(x) = 0$ を解くのに 2分法を用いる. $(a_0, b_0) = (-2, +3)$ から始めるとき, 繰り返し回数 $n \to +\infty$ でどの解が見つかるか.
3. 方程式 $f(x) = 0$ を解くのに 2分法を用いる. $(a_0, b_0) = (-6, +4)$ から始めるとき, 繰り返し回数 $n \to +\infty$ でどの解が見つかるか.
4. 3のとき, 誤差 $b_n - a_n$ が 10^{-2} より小さくなる n はいくつか.

2 分法の仲間たち

- **2 分探索** 何かの順序に並べられた, 有限個のデータの中から, 望みのデータを高速に見つけ出す方法. アルゴリズム・演習 (2 年前期)
- **2 分木** 2 分探索を高速で行うために設計されたデータ構造. 応用アルゴリズム (3 年前期)

お知らせ

1. 自宅の個人 PC に Visual C++ (Studio) 2008/2010 をインストールしたけどうまく使えない, という質問を何回かいただいています. 困った状態のスクリーンショットを添付してメールで問い合わせてくれたら, 解決できるかもしれません.
2. これまでの演習問題のうち何個かの解答プログラム例を公開しています. 演習問題のページからどうぞ.
3. 来週から quiz やります.
4. 演習:質問予約の順位決定ポリシー変更してみます (実験)