数值分析与计算

北京邮电大学软件学院 漆 涛

September 20, 2019

目录

第一章：误差分析

误差来源

1. 模型误差

2. 观测误差

3. 截断误差

4. 舍入误差

定义 1 (绝对误差，绝对误差界)

$$\delta(a) = x - a, \quad |\delta(a)| \leqslant \Delta(a)$$

定义 2 (相对误差，相对误差界)

$$\varepsilon(a) = \frac{x-a}{x}, \quad |\varepsilon(a)| \leqslant E(a)$$

相对误差的近似值

$$\varepsilon'(a) = \frac{x-a}{a}$$

$$\varepsilon'(a) - \varepsilon(a) = \frac{\varepsilon'(a)^2}{1+\varepsilon'(a)} > 0$$

只要 $\varepsilon'(a) \ll 1$， $\varepsilon(a)$ 与 $\varepsilon'(a)$ 相差更小.

相对误差，相对误差界通常表示为百分数.

有效数字

不妨假设 x, a 均为正数.

十进制的四舍五入

$$x = (0.d_1 \cdots d_{p-1} d_p d_{p+1} \cdots)10^m, \qquad (d_1 \neq 0)$$

对 d_{p+1} 四舍五入:

$$a = \begin{cases} (0.d_1 \cdots d_p)10^m & \text{当 } d_{p+1} < 5 \\ (0.d_1 \cdots d_p)10^m + 10^{m-p} & \text{当 } d_{p+1} \geqslant 5 \end{cases}$$

习惯上称 a 具有 p 位有效数字.

定义 3 (有效数字) *β 进制近似数*

$$a = (0.d_1 d_2 \cdots d_p)\beta^m, \qquad (d_1 \neq 0)$$

如果其绝对误差不超过最后一位数的半个单位. 即

$$|\delta(a)| \leq \frac{1}{2}\beta^{m-p}$$

则称 a 有 p 位 β 进制有效数字.

定理 4 *近似数* $a=(0.d_1d_2\cdots d_p)\beta^m$ *是* x *具有* p *位数字的近似数, 则* a *是所有字长为* p *的实数中最接近* x *的数.*

有效数字是对相对误差的整数估计.

定理 5 *设* $a=(0.d_1d_2\cdots d_p)\beta^m$ *是具有* p *位有效数字近似数* $(p\geqslant 1)$*, 则其相对误差满足:*

$$|e(a)|\leqslant\frac{1}{2d_1\beta^{p-1}-1}$$

当 $d_1,d_2,\cdots,d_{p-1}$ *不全为零时*

$$|e_r(a)|\leqslant\frac{\beta^{-p+1}}{2d_0}$$

定理 6 *设近似数*

$$a = (d_0.d_1d_2\cdots d_{p-1}d_p\cdots)\beta^m$$

的相对误差界为

$$|e_r(a)| \leqslant \frac{\beta^{-p+1}}{2(d_0+1)}$$

并且 $d_1, d_2, \cdots, d_{p-1}$ *不全为零, 则* a *至少具有* p *位有效数字.*

证明:

$$\begin{aligned} |e(a)| &= |xe_r(a)| \leq (d_0+1)\beta^m |e_r(a)| \\ &\leqslant \frac{1}{2}\beta^{m-p+1} \end{aligned}$$

十进制与二进制有效数字

$$\log_2 10 \approx 3.01，\ \log_{10} 2 \approx 0.3322$$

十进制有效数字$\times 3 \approx$ 二进制有效数字

计算机浮点数系统与舍入误差

$$a = (d_0.d_1 d_2 \cdots d_{p-1})\beta^m, \quad (L \leqslant m \leqslant U)$$

名称	β	p	L	U
IEEE 单精度	2	24	-126	127
IEEE 双精度	2	53	-1022	1023
Cray 计算机	2	48	-16383	16384
HP 计算机	10	12	-499	499
一种 IBM 大型机	16	6	-64	63

IEEE 754 单精度 32 位浮点数

1	8	23
s	e	$m = b_1 b_2 \cdots b_{23}$

s: 符号位

e: 偏移指数

m: 尾数

代表规格化数:

$$(-1)^s(1+0.m)2^{e-127}$$

e	m	值
0	0	± 0
0	$\neq 0$	$\pm(0.m)2^{-126}$ 非规格化数
$1 \sim 254$	任意	$(-1)^s(1+0.m)2^{e-127}$
255	0	$\pm\infty$
255	$\neq 0$	NaN

单精度浮点数极值表(正值)

最小非规格化数	00000001	2^{-149}
最大非规格化数	007FFFFF	$(1-2^{-23})2^{-126}$
最小规格化数	00800000	2^{-126}
1	3F800000	1
最大规格化数	7F7FFFFF	$(1-2^{-24})2^{128}$
∞	7F800000	

$[-2^{24}, 2^{24}]$ 中的整数可以准确表示.

$fl(2^{24}+1) = fl(2^{24}+2), \quad fl(2^{25}) = fl(2^{25}+1)$

$\pi = (1.1001\,0010\,0001\,1111\,1011\,0101\,0100)_2 2^1$

$$s = 0, m = 1 + 127$$

$$m = 1001\,0010\,0001\,1111\,1011\,011$$

$$0\ \ 10000000\ \ 1001\,0010\,0001\,1111\,1011\,011$$

$$40400FDB$$

little endian

$$DB\,0F\,40\,40$$

IEEE 754 双精度 64 位浮点数

1	11	52
s	e	$m = b_1 b_2 \cdots b_{52}$

s: 符号位

e: 偏移指数

m: 尾数

代表规格化数:

$$(-1)^s (1 + 0.m) 2^{e-1023}$$

e	m	值
0	0	± 0
0	$\neq 0$	$\pm (0.m) 2^{-1022}$ 非规格化数
$1 \sim 2046$	任意	$(-1)^s (1 + 0.m) 2^{e-1023}$
2047	0	$\pm\infty$
2047	$\neq 0$	NaN

$[-2^{53}, 2^{53}]$ 整数可以准确表示.

第二章：方程求根

数列收敛速度度量

问题敏感性

二分法

压缩映像原理

牛顿法

割线法，抛物线法简介

数列收敛速度的度量: 数列 $\{x_n\}$ 收敛于 x.

1. p 次多项式收敛: 存在实数 $C, p > 1$ 使得当 n 充分大时,

$$|x_n - x| \leqslant C/n^p$$

2. 线性收敛: 存在实数 $C > 0, 0 < h < 1$ 使得当 n 充分大时,

$$|x_n - x| \leqslant Ch^n$$

3. 超线性收敛: 对于任意的 $0 < h < 1$, 存在实数 $C > 0$ 使得当 n 充分大时,

$$|x_n - x| \leqslant Ch^n$$

4. 高阶收敛(r阶收敛): 存在实数 $C > 0, 0 < h < 1, r > 1$ 使得当 n 充分大时,

$$|x_n - x| \leqslant Ch^{r^n}$$

收敛速度的判据：数列 $\{x_n\}$ 收敛于 x.

定理 7 *如果*

$$\lim_{n\to\infty}\frac{\left|x_{n+1}-x\right|}{|x_n-x|}=c<1$$

则序列线性收敛

如果

$$\lim_{n\to\infty}\frac{\left|x_{n+1}-x\right|}{|x_n-x|}=0$$

则序列超线性收敛.

如果

$$\lim_{n\to\infty}\frac{\left|x_{n+1}-x\right|}{|x_n-x|^r}=c$$

则 r 阶收敛

问题敏感性

考虑方程 $f(x) = \Delta y$ 的解在 $\Delta y = 0$ 附近的扰动.

$$
\begin{aligned}
f(x^*) &= 0 \\
f(x^* + \Delta x) &= \Delta y \\
\chi(f) = cond(f) &= \left|\frac{\Delta x}{\Delta y}\right| \approx \frac{1}{|f'(x^*)|}
\end{aligned}
$$

二分法

```
double root(double (*f)(double),
            double a, double b)
{
    // 前提条件: f(a) < 0;  f(b) > 0
    do {
        double mid = a + (b-a)/2;
        double v = f(mid);
        if( mid 可以接受} break;
        (v < 0) ?  (a = mid) : (b = mid);
    } while(1);
    return mid;
}
```

二分法的收敛速度: 线性收敛， $h = 1/2$.

压缩映像原理

压缩映像: $|f(x)-f(y)| \leqslant L|x-y| \quad (0<L<1)$

定理 8 X *为完备距离空间.* D *为* X *的闭子集.* $f: D \to D$ *为压缩映像. 则* f *在* D *中有唯一的不动点* x^*.

任取初始值 $x_0 \in D$. *做迭代*

$$x_{n+1} = f(x_n) \tag{1}$$

则序列 $\{x_n\}$ *满足*

1. $\lim_{n\to\infty} x_n = x^*$

2. $|x_n - x^*| \leqslant \frac{L^n}{1-L}|x_1 - x_0|$

3. $|x_n - x^*| \leqslant \frac{L}{1-L}|x_n - x_{n-1}|$

证明: 假设 $m > n$.

$$\begin{aligned}
|x_m - x_n| &\leqslant |x_m - x_{m-1}| + \cdots + \left|x_{n+1} - x_n\right| \\
&\leqslant \left(L^{m-n} + \cdots + L\right)|x_n - x_{n-1}| \\
&\leqslant \frac{L^n}{1-L}|x_1 - x_0|
\end{aligned}$$

函数 $\varphi(x)$ 满足

1. $\varphi[a,b] \subseteq [a,b]$

2. 当 $x \in (a,b)$ 时，$|\varphi'(x)| \leqslant L < 1$.

则 $\varphi(x)$ 在 $[a,b]$ 上为压缩映像.

定义 9 (局部收敛) *设* $\varphi(x)$ *有不动点* x^*. *如果存在* x^* *的一个邻域* $R = |x - x_*| \leqslant \delta$, *使得对于任意的* $x_0 \in R$, *由公式(1)确定的迭代产生的序列* $\{x_k\}$ *均属于* R, *并且收敛到* x^*. *则称不动点迭代(1)具有局部收敛性.*

定理 10 *设* x^* *为函数* $\varphi(x)$ *的不动点, 而* $\varphi'(x)$ *连续. 如果* $|\varphi'(x^*)| < 1$, *则迭代(1) 局部收敛.*

收敛速度

定理 11 *如果*

$\varphi'(x^*) = \varphi''(x^*) \cdots = \varphi^{(p-1)}(x) = 0, \varphi^{(p)}(x^*) \neq 0$

则不动点迭代(1) p 阶收敛.

证明: 将 $\varphi(x)$ 在 x^* 处泰勒展开:

$$\varphi(x) = \varphi(x^*) + \frac{\varphi^{(p)}(\xi)}{p!}(x - x^*)^p$$

$$\lim_{n\to\infty} \frac{\left|x_{n+1} - x^*\right|}{|x_n - x^*|^p} = \frac{\left|\varphi^{(p)}(x^*)\right|}{p!}$$

最常见的: $\varphi'(x^*) \neq 0$. 一阶收敛.

最理想的: $\varphi'(x^*) = 0, \varphi''(x^*) \neq 0$. 二阶收敛.

$f(x) = x^2 - 3$ 的根是 $x^* = \sqrt{3}$.

不同的不动点迭代:

1. $\varphi(x) = x^2 + x - 3, \quad \varphi'(x^*) = 2\sqrt{3} + 1 > 1.$

2. $\varphi(x) = 3/x, \quad \varphi'(x^*) = -1.$

3. $\varphi(x) = x - (x^2 - 3)/4, \quad \varphi'(x^*) \approx 0.134.$

4. $\varphi(x) = (x + 3/x)/2, \quad \varphi'(x^*) = 0$

埃特金迭代加速收敛技术(Aitken Δ^2 加速方法)

$$\begin{cases} x_1 = \varphi(x_0) \qquad x_1 - x^* = \varphi'(\eta)(x_1 - x_0) \\ x_2 = \varphi(x_1) \qquad x_2 - x^* = \varphi'(\zeta)(x_2 - x_1) \end{cases}$$

$$x^* = x_0 - \frac{(x_1 - x_0)^2}{x_2 - 2x_1 + x_0} = x_0 - (\Delta x_0)^2 / \Delta^2(x_0)$$

Steffensen 迭代格式:

$$\begin{cases} x_0 = \text{适当选取} \\ y = \varphi(x_n) \\ z = \varphi(y) \\ x_{n+1} = x_n - \dfrac{(y - x_n)^2}{z - 2y + x_n} \end{cases} \tag{2}$$

Setffensen 迭代相当于对于

$$\psi(x) = x - \frac{[\varphi(x) - x]^2}{\varphi(\varphi(x)) - 2\varphi(x) + x}$$

的普通不动点迭代.

定理 12 *若 x^* 是 ψ 的不动点, 则其也是 $\varphi(x)$ 的不动点.*

若 x^* 是 $\varphi(x)$ 的不动点，$\varphi''(x)$ 存在，$\varphi'(x^*) \neq 1$，则 x^*也是 $\psi(x)$ 的不动点，并且 Steffensen 迭代收敛，并且是二阶收敛的.

牛顿法

$$\begin{cases} x_0 = \text{适当选取} \\ x_{n+1} = x_n - \dfrac{f(x_n)}{f'(x_n)} \end{cases} \tag{3}$$

$$\phi(x) = x - \frac{f(x)}{f'(x)}, \quad \phi'(x) = \frac{f(x)f''(x)}{[f'(x)]^2}$$

计算机求平方根快速算法.

$$x^2 - c = 0$$

牛顿迭代格式

$$x_{n+1} = \frac{1}{2}\left(x_n + \frac{c}{x_n}\right)$$

$$x_{n+1} - \sqrt{c} = \frac{(x_n - \sqrt{c})^2}{2x_n}$$

$$x_{n+1} + \sqrt{c} = \frac{(x_n + \sqrt{c})^2}{2x_n}$$

$$\frac{x_{n+1} - \sqrt{c}}{x_{n+1} + \sqrt{c}} = \left(\frac{x_n - \sqrt{c}}{x_n + \sqrt{c}}\right)^2$$

$$\frac{x_n - \sqrt{c}}{x_n + \sqrt{c}} = \left(\frac{x_0 - \sqrt{c}}{x_0 + \sqrt{c}}\right)^{2^n}$$

$$x_n - x_{n+1} = \frac{x_n^2 - c}{2x_n} \geqslant 0$$

IEEE 单精度浮点数: $\quad c = 2^p 1.m$

$$\sqrt{c} = \begin{cases} 2^{p/2}\sqrt{1.m} & p \text{ 为偶数;} \\ \sqrt{2}\,2^{(p-1)/2}\sqrt{1.m} & p \text{ 为奇数} \end{cases}$$

问题: 求$\sqrt{c}$, $\quad c \in [1,2]$

取初始值 $x_0 = pc + q$.

1. p, q 使得 $\max_{c\in[1,2]} |x_0 - \sqrt{c}|$ 达到最小.

2. p, q 使得 $\max_{c\in[1,2]} (x_0 - \sqrt{c})^2$ 达到最小.

3. p, q 使得 $\max_{c\in[1,2]} \left(\frac{x_0-\sqrt{c}}{x_0-\sqrt{c}}\right)^2$ 达到最小.

4. p, q 使得

$$\max_{c\in[1,2]} \left|x_1 - \sqrt{c}\right| = \max_{c\in[1,2]} \frac{(x_0 - \sqrt{c})^2}{2x_0}$$

达到最小.

$$\begin{aligned} p &= \sqrt{2}-1, \\ q &= \frac{9-3\sqrt{2}}{8}, \\ \max_{c\in[1,2]} \left| x_0 - \sqrt{c} \right| &= \frac{5\sqrt{2}-7}{8} \\ &\approx 0.0088834 \approx 2^{-6.81} \end{aligned}$$

求 $\lfloor\sqrt{c}\rfloor$ (c 为正整数)的牛顿迭代法.

$$x_{n+1} = \left\lfloor \frac{1}{2}\left(x_n + \left\lfloor \frac{c}{x_n} \right\rfloor\right)\right\rfloor$$

定理 13 *如果* $x_0 \geqslant \lfloor\sqrt{c}\rfloor$, *则*

$$\lfloor\sqrt{c}\rfloor \leqslant x_1 < x_0$$

如果 $x_0 = \lfloor\sqrt{c}\rfloor$, *则*

$$x_0 \leqslant x_1 \leqslant \lfloor\sqrt{c}\rfloor + 1$$

证明略

x_0 的选取:

$$x_0 = 2^s, \quad 2^{s-1} < \lfloor\sqrt{c}\rfloor \leqslant 2^s$$

```
int isqrt(unsigned c)
{
     if(c <= 1) retrun c;
     unsigned c1 = c - 1;
     int s = 1;
     if(c1 > 65535) {  s += 8; c1 >>= 16;  }
     if(c1 > 255  ) {  s += 4; c1 >>= 8;   }
     if(c1 > 15   ) {  s += 2; c1 >>= 4;   }
     if(c1 > 3    )  ++s;

     int x0 = 1 << s;
     int x1 = (x0 + (c >> s)) >> 1;
     while( x1 < x0 )
     {
         x0 = x1;
         x1 = (x0 + c/x0) >> 1;
     }
    return x0;
}
```

$c \leqslant 16785407$ ，需要 4 次整数除法(c/x0).

s 并不严格符合 $2^{s-1} < \lfloor \sqrt{c} \rfloor \leqslant 2^s$

应用 `std::upper_bound` 函数:

c	$\lfloor\sqrt{c}\rfloor$	s	x_0
$[1,4)$	1	0	1
$[4,9)$	2	1	2
$[9,25)$	$(2,4]$	2	4
$[25,81)$	$(4,8]$	3	8
$\cdots$	$\cdots$	$\cdot$	$\cdots$
$\left[(2^{s-1}+1)^2,(2^s+1)^2\right)$	$\left(2^{s-1},2^s\right]$	s	2^s
$\cdots$	$\cdots$	$\cdot$	$\cdots$
$\left[(2^{14}+1)^2,(2^{15}+1)^2\right)$	$\left(2^{14},2^{15}\right]$	15	32768
$\left[(2^{15}+1)^2,2^{32}\right)$	$\left(2^{15},2^{16}\right]$	16	65536

```
unsigned  ss[16] = {
4,          9,          25,          81,
17*17,      33*33,      65*65,       129*129,
257*257,    513*513,    1025*1025,   2049*2049,
4097*4097,  8193*8193,  16385*16385, 32769*32769,
};
int isqrt(unsigned c)
{
     if(c <= 1) retrun c;
     unsigned* temp =
                std::upper_bound(c, ss, ss + 16);
     int s = temp - ss;
     int x0 = 1 << s;
     int x1 = (x0 + (c >> s)) >> 1;
     while( x1 < x0 )
     {
         x0 = x1;
         x1 = (x0 + c/x0) >> 1;
     }
    return x0;
}
```

应用 `std::lower_bound` 函数.

c	$\lfloor\sqrt{c}\rfloor$	s	x_0
$(0,3]$	1	0	1
$(3,8]$	2	1	2
$(8,24]$	$(2,4]$	2	4
$(24,80]$	$(4,8]$	3	8
$\cdots$	$\cdots$	$\cdot$	$\cdots$
$\left((2^{s-1}+1)^2-1,(2^s+1)^2-1\right]$	$\left(2^{s-1},2^s\right]$	s	2^s
$\cdots$	$\cdots$	$\cdot$	$\cdots$
$\left((2^{14}+1)^2-1,(2^{15}+1)^2-1\right]$	$\left(2^{14},2^{15}\right]$	15	32768
$\left((2^{15}+1)^2-1,2^{32}\right]$	$\left(2^{15},2^{16}\right]$	16	65536

```
unsigned  ss[16] = {
3,          8,          24,          80,
16*18,      32*34,      64*66,       128*130,
256*258,    512*514,    1024*1026,   2048*2050,
4096*4098,  8192*8194,  16384*16386, 32768*32770,
};

int isqrt(unsigned c)
{
     if(c <= 1) retrun c;
     unsigned* temp =
                std::lower_bound(c, ss, ss + 16);
     int s = temp - ss;
     int x0 = 1 << s;
     int x1 = (x0 + (c >> s)) >> 1;
     while( x1 < x0 )
     {
         x0 = x1;
         x1 = (x0 + c/x0) >> 1;
     }
    return x0;
}
```

如果 c 在区间 $[1, 2^{32})$ 上均匀分布.

```
int isqrt(unsigned c)
{
     if(c <= 1) retrun c;
     int s = 0;
     if       (c > 32768*32770)   s = 16;
     else if ( c >16384*16386)   s = 15;
     else if ( c > 8192*8194)    s = 14;
     else if ( c < 4096*4098)    s = 13;
     else if ( c > 2048*2050)    s = 12;
     else if ( c > 1024*1026)    s = 11;
     else if ( c >  512*514)     s = 10;
     else if ( c >  256*258)     s = 9;
     else if ( c >  128*130)     s = 8;
     else if ( c >   64*66)      s = 7;
     else if ( c >   32*34)      s = 6;
     else if ( c >   16*18)      s = 5;
     else if ( c >    8*10)      s = 4;
     else if ( c >    4*6)       s = 3;
     else if ( c >    2*4)       s = 2;
     else if ( c >    1*3)       s = 1;
     //else                        s = 0;
     int x0 = 1 << s;
     int x1 = (x0 + (c >> s)) >> 1;
     while( x1 < x0 ) {
         x0 = x1;
         x1 = (x0 + c/x0) >> 1;
     }
    return x0;
}
```

比较次数数学期望

$$(3/4) + (3/16)2 + (3/64)3 + \cdots = 4/3$$

计算机求倒数 $1/c$.

$$\frac{1}{x} = c$$

$$\begin{aligned} x_0 &= \text{适当选取} \\ x_{n+1} &= x_n(2 - cx_n) \\ x_{n+1} - c^{-1} &= -c(x_n - c^{-1})^2 \\ c(x_n - c^{-1}) &= \Big(c(x_n - c^{-1})\Big)^{2^n} \\ x_{n+1} - x_n &= x_n(1 - cx_n) \geqslant 0 \end{aligned}$$

问题:　求 c^{-1},　$c \in [1,2]$

取初始值 $x_0 = pc + q$.

迭代: $x_{n+1} = x_n(2 - cx_n)$

1. p, q 使得 $\max_{c\in[1,2]} \left| x_0 - c^{-1} \right|$ 达到最小.

2. p, q 使得 $\max_{c\in[1,2]} c(x_0 - c^{-1})^2$ 达到最小.

3. p, q 使得 $\max_{c\in[1,2]} \left(c(x_0 - c^{-1}) \right)^2$ 达到最小.

求 p, q 使得 $\max_{c\in[1,2]}\left|x_0 - c^{-1}\right|$ 达到最小.

$$\begin{aligned}
x_0 &= -\frac{1}{2}c + \sqrt{2}, \\
\max_{c\in[1,2]}\left|x_0 - c^{-1}\right| &= 1.5 - \sqrt{2} \\
&\approx 0.08579 \\
&\approx 2^{-3.54}
\end{aligned}$$

求 p, q 使得 $\max_{c\in[1,2]}\left(c(x_0 - c^{-1})\right)^2$ 达到最小.

$$\left(c(x_0 - c^{-1})\right)^2 = (c(pc + q) - 1))^2$$

$p = -\frac{8}{17}, q = -3p = \frac{24}{17}$

$cx_1 - 1 < \left(\frac{1}{17}\right)^2 \approx 2^{-8.1749}$,
一次迭代后有约 8 个有效数字.

求 $a^{-1} \mod 2^{32}$ 的牛顿迭代法.

$$\begin{cases} x_0 & = & \text{奇数} \\ x_{n+1} & = & x_n(2 - ax_n) \mod 2^{32} \end{cases}$$

定理 14 *如果* $x_n a \equiv 1 \mod m$, *则*

$$x_{n+1} a \equiv 1 \mod m^2$$

.

证明. 证毕

取 x_0 为奇数保证 $x_0 a \equiv 1 \mod 2$.

取 $x_0 = a + 2[(a+1)\&4]$ 有 $x_0 a \equiv 1 \mod 16$.

取 $x_0 = a^2 + a - 1 \mod 2^{32}$ 有 $x_0 a \equiv 1 \mod 16$.

重根情况

$$f(x)=(x-x^*)^m g(x), \quad g(x^*)\neq 0$$

$f(x^*)=f'(x^*)=\cdots=f^{(m-1)}(x^*)=0, f^{(m)}(x^*)\neq 0$

$\varphi(x)=x-f(x)/f'(x), \varphi'(x^*)=1-1/m$

如果 m 已知，则

$$\varphi(x)=x-m\frac{f(x)}{f'(x)}$$

二次收敛.

$$\mu(x)=\frac{f(x)}{f'(x)}$$

x^* 是其单重根.

$$\varphi(x)=x-\frac{\mu(x)}{\mu'(x)}=x-\frac{f(x)f'(x)}{[f'(x)]^2-f(x)f''(x)}$$

二阶收敛.

$$x^4-4x^2+4=0,\quad x^*=2 \text{ 二重根}$$

1. 牛顿法 $x_{n+1}=x_n-\frac{x_n^2-2}{4x_n}=\frac{1}{4}(3x_n+\frac{2}{x_n})$

2. $x_{n+1}=x_n-\frac{x_n^2-2}{2x_n}=\frac{1}{2}(x_n+\frac{2}{x_n})$

3. $x_{n+1}=\frac{4x_n}{x_n^2+2}$

简化牛顿法:

$$\varphi(x) = x - cf(x), \quad x_{n+1} = x_n - cf(x_n)$$

可取 $c = 1/f'(x_0)$.

牛顿下山法: 保证 $\left|f(x_{n+1})\right| < |f(x_n)|$

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

即

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

λ 从 1 开始, 依次减半.

弦截法:

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

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

收敛速度: $\frac{1+\sqrt{5}}{2} \approx 1.618$ 阶收敛.

抛物线法: 在 x_n, x_{n-1}, x_{n-2} 三点上做 $f(x)$ 的二次插值. 这个抛物线的零点作为 x_{n+1}.

$$\begin{aligned} p_2(x) &= f(x_n) + f[x_n, x_{n-1}](x - x_n) \\ &\quad + f[x_n, x_{n-1}, x_{n-2}](x - x_n)(x - x_{n-1}) \end{aligned}$$

两个零点:

$$x_{n+1} = x_n - \frac{2f(x_n)}{\omega \pm \sqrt{\omega^2 - f(x_n)f[x_n, x_{n-1}, x_{n-2}]}}$$

其中 $\omega = f[x_n, x_{n-1}] - f(x_n, x_{n-1}, x_{n-2}]$.

取较为接近 x_n 的那个点. 即 $\pm$ 号与 ω 同号.

收敛速度:

$$\frac{\left|x_{n+1} - x^*\right|}{|x_n - x^*|^{1.840}} \to \left|\frac{f'''(x^*)}{6f'(x^*)}\right|^{0.42}$$

1.840: $x^3 - x^2 - x - 1$ 的解.

多项式

$$x^n + a_1 x^{n-1} + a_2 x^{n-2} + \cdots + a_{n-1} x + a_n = 0$$

矩阵

$$\begin{pmatrix} -a_1 & -a_2 & \cdots & -a_{n-1} & -a_n \\ 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \end{pmatrix}$$

的特征值.