回顾

第二章 非线性方程$f(x)=0$的解法

2.1 引言

2.2 二分法与试值法

2.3 不动点迭代法

2.4 牛顿-拉夫森法（简称：牛顿迭代法）

2.5 割线法

2.6 迭代收敛的加速办法(选讲)

回顾

2.3.3 不动点迭代法的收敛性分析

不动点迭代法的收敛条件

由于对方程f(x)=0可以构造不同的迭代公式,但迭代公式:

$$x_{k+1}=g(x_k), \qquad k=0, 1, 2\ldots$$

并非总是收敛的。那么,当迭代函数φ(x)满足什么条件时，相应的迭代公式才收敛呢?

即使迭代收敛时，也不可能迭代很多次，而是迭代有限次后就停止，这就需要估计迭代值的误差，以便适时终止迭代。

不动点原理 （压缩映像定理） ----回顾

定理2.1 设$g(x)$在$[a,\ b]$上连续，且一阶导数连续，若

(1) $a \le g(x) \le b$对一切$x\in[a,b]$都成立，

(2) 存在$0 \le L < 1$，使得$|g'(x)| \le L$对$\forall x\in[a,b]$成立，

则函数$f(x) = x - g(x)$在$[a,b]$中有唯一的零点x^*。

x^* 称为$g(x)$的不动点，即 $\boxed{g(x^*) = x^*}$

简证： 取$f(x) = x - g(x)$, 我们有

$$f(a) = a - g(a) \le 0,\ f(b) = b - g(b) \ge 0.$$

$\Rightarrow$ $f(x)$在$[a,b]$上有零点。

唯一性：反证法，假设存在 x^*，$y^*\in[a,\ b]$ 使得

$$\begin{cases} x^* = g(x^*) \\ y^* = g(y^*) \end{cases}$$

$\Rightarrow$ $|x^*-y^*| = |g(x^*)-g(y^*)| = |g'(\xi)|\cdot|x^*-y^*| \le L|x^*-y^*|$ 矛盾！

例2.7 用不同方法求 $x^2-3=0$的根$\sqrt{3}$，取 $x_0=2$.
讨论合理性和收敛性

- 迭代公式1： $x_{k+1}=x_k^2+x_k-3$
- 迭代公式2： $x_{k+1}=3/x_k$
- 迭代公式3： $x_{k+1}=x_k-(x_k^2-3)/4$
- 迭代公式4： $x_{k+1}=(x_k+3/x_k)/2$
- 计算结果：

---回顾

k	x_k	公式1	公式2	公式3	公式4
0	x_0	2	2	2	2
1	x_1	3	1.5	1.75	1.75
2	x_2	9	2	1.734375	1.732143
3	x_3	87	1.5	1.732361	1.732051
⋮	⋮	⋮	⋮	⋮	⋮

精确值：
$\sqrt{3}=1.7320508\ldots$

怎么判断收敛的迭代公式的速度快慢呢？

收敛速度—收敛阶 (非常重要)

定义2.1 设迭代 $x_{k+1}=g(x_k)$ 收敛到 $g(x)$ 的不动点 x^*。记绝对误差 $e_k=x_k-x^*$，若

$$\lim_{k\to\infty}\left|\frac{e_{k+1}}{e_k{}^p}\right|=C\neq 0 \quad (C\text{为常数})$$

则称该迭代为以收敛阶p收敛到x^*。数C称为渐近误差常数。

(1) 当 $p=1$ 时称为线性收敛，此时 $|C|<1$;

(2) 当 $p=2$ 时称为二次收敛，或平方收敛;

(3) 当 $p>1$ 时称为超线性收敛。

□ 不动点迭代中, 若迭代数列$\{x_k\}$收敛, 且$g'(x^*)\neq 0$，则

$$e_{k+1}=x_{k+1}-x^*=g(x_k)-g(x^*)=g'(\xi)e_k$$

取极限得 $\lim_{k\to\infty}\left|\frac{e_{k+1}}{e_k}\right|=g'(x^*)\neq 0 \Longrightarrow$ 线性收敛.

2.3.4 不动点迭代法的算法实现

迭代法就是通过有限次的计算，来求出给定方程的满足精度要求的近似根。它的突出优点是算法的逻辑结构简单，且在计算时，中间结果若有扰动，仍不会影响计算结果。

计算步骤

1)确定有根区间的范围$[a,b]$，使$f(a)f(b)<0$；

2)确定方程$f(x)=0$的等价形式$x=\varphi(x)$及初始值x_0；为确保迭代收敛，要求$\varphi(x)$满足定理2.1的条件；

3)建立迭代格式$x_{k+1}=\varphi(x_k)$，计算出$x_{k+1}(K=0,1,2\cdots)$；

4)若$|x_{K+1}-x_K|<\varepsilon$($\varepsilon$为事先给定的精度)，则终止迭代，输出$x_{k+1}$为$x$的近似值；否则，继续迭代。

算法流程图

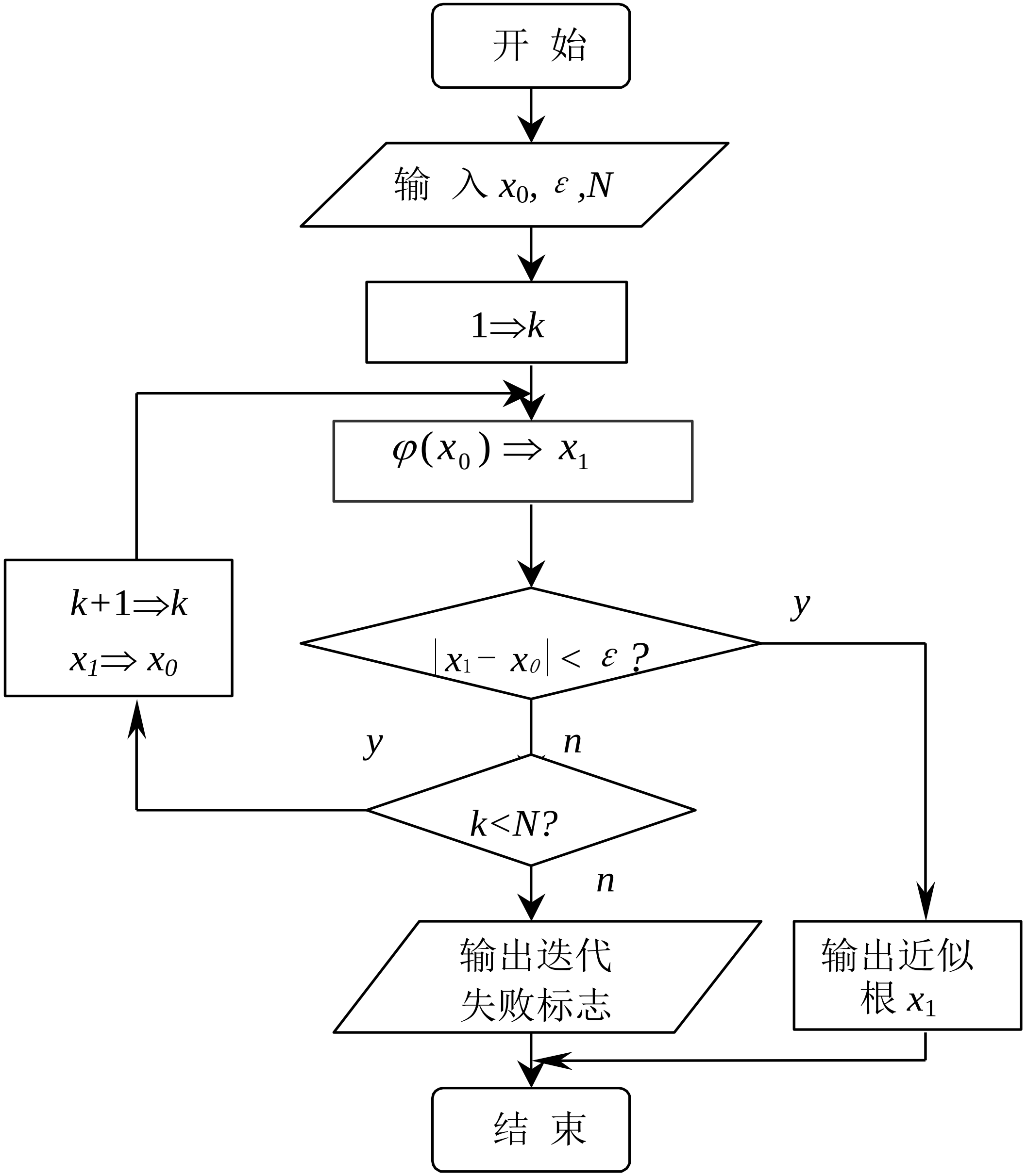


图 迭代法的算法流程图

程序:

fixed_Iteration.m

程序:

fixed_Iteration.m

```
function [k,p,err,P] = fixed_Iteration(g,p0,tol,max1)
%Input   -  g is the iteration function
%          -  p0 is the initial guess for the fixed-point
%          -  tol is the tolerance
%          -  max1 is the maximum number of iterations
%Output-  k is the number of iterations
%          -  p is the approximation to the fixed-point
%          - err is the error in the approximation
%          -  P' contains the sequence {pn}
% f=@(x)  2*sqrt(x-1);
% p0=1; tol=1e-4; max1=500;
% fixed_Iteration(f, p0, tol, max1)
P(1)= p0;

for k=2:max1
    P(k)=g(P(k-1));
    err=abs(P(k)-P(k-1));
    relerr=err/(abs(P(k))+eps);
    p=P(k);
    if (err<tol) | (relerr<tol),   break;     end
End

P=P'
```

作业 2.2

2. 当

$$g(x) = -4 + 4x - \frac{1}{2}x^2$$

时，研究不动点迭代的性质。

(a) 求解 $g(x) = x$，且证明 $P = 2$ 和 $P = 4$ 是不动点。

(b) 用起始值 $p_0 = 1.9$ 计算 p_1, p_2 和 p_3。

(c) 用起始值 $p_0 = 3.8$ 计算 p_1, p_2 和 p_3。

(d) 对于(b)和(c)中的 p_k，寻找误差 E_k 和相对误差 R_k。

(e) 从定理 2.3 中可得出什么结论？

2.4 牛顿-拉夫森法（简称:牛顿迭代法）

用迭代法可逐步精确方程$f(x)=0$根的近似值，但必须要找到$f(x)=0$的等价方程$x=\varphi(x)$，如果$\varphi(x)$选得不合适，不仅影响收敛速度，而且有可能造成迭代格式发散。

能否找到一种迭代方法，既结构简单，收敛速度快，又不存在发散的问题。这就是本节要介绍的牛顿迭代法。

这是为什么要研究牛顿迭代法的原因

2.4.1 牛顿迭代法的基本思想

2.4.2 牛顿迭代法的几何解释

2.4.3 牛顿迭代法的收敛性分析

2.4.3 牛顿迭代法的算法实现

2.4.1 牛顿迭代法的基本思想

牛顿迭代法是一种重要和常用的迭代法，它的基本思想是将非线性函数 $f(x)$ 逐步线性化，从而将非线性方程 $f(x)=0$ 近似地转化为线性方程求解。

对于方程 $f(x)=0$，设其近似根为 x_k，函数 $f(x)$ 可在 x_k 附近作泰勒展开。

$$f(x)=f(x_k)+f'(x_k)(x-x_k)+\boxed{\frac{f''(\xi)}{2!}(x-x_k)^2},$$ ξ **在 x_k 和 x 之间.**

忽略高次项，用其线性部分作为函数f(x)的近似，有

$$f(x)\approx f(x_k)+f'(x_k)(x-x_k)$$

设f(x)=0的根 x^*，则有 $f(x^*)=0$，即

$$f(x_k)+f'(x_k)(x^*-x_k)\approx 0$$

2.4.1 牛顿迭代法的基本思想

$$x^* \approx x_k - \frac{f(x_k)}{f'(x_k)}$$

$$f(x_k) + f'(x_k)(x^* - x_k) \approx 0$$

将左端取为x_{k+1}，即x_{k+1}是比x_k更接近于x^*的近似值，即

$$\boxed{x_{k+1} \approx x_k - \frac{f(x_k)}{f'(x_k)}, k = 0, 1, 2, \cdots}$$

这就是著名的牛顿迭代公式，相应的迭代函数为

$$\varphi(x) \approx x - \frac{f(x)}{f'(x)}.$$

2.4.4 牛顿迭代法的基本思想---（非常重要）

例2.8 试建立计算$\sqrt[3]{a}$ 的牛顿迭代格式，并求 $\sqrt[3]{411.791}$ 的近似值，要求迭代误差不超过0.005。

解：令 $x = \sqrt[3]{a}, f(x) = x^3 - a = 0$

则牛顿迭代格式为

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)} = x_k - \frac{x_k^{\ 3} - a}{3x_k^2} = \frac{2}{3}x_k + \frac{a}{3x_k^2}$$

7^3=343，8^3=512，f(7)f″(7)<0而f(8)f″(8)>0。

取x_0=8，有

$$x_{k+1} = x_1 = \frac{2}{3}x_0 + \frac{a}{3x_0^2} = \frac{2}{3}*8 + \frac{411.791}{3*8*8} \approx 7.478078$$

$$x_1=\frac{2}{3}x_0+\frac{a}{3x_0^2}=\frac{2}{3}*8+\frac{411.791}{3*8^2}\approx 7.439\ 078$$

$$x_2=\frac{2}{3}x_1+\frac{a}{3x_1^2}=\frac{2}{3}*7.478\ 078+\frac{411.791}{3*7.478\ 078^2}\approx 7.439\ 956$$

$$\left|x_2-x_1\right|=0.038\ 122$$

$$x_3=\frac{2}{3}x_2+\frac{a}{3x_2^2}=\frac{2}{3}*7.439\ 956+\frac{411.791}{3*7.439\ 956^2}\approx 7.439\ 760$$

$$\left|x_3-x_2\right|=0.000\ 196$$

于是取$x^*\approx 7.439\ 760$

2.4.4 牛顿迭代法的基本思想

例2.9 用牛顿迭代法求$x = e^{-x}$ 的根，ε=10^{-4}

解：

因 $f(x) = xe^x - 1,\ f'(x) = e^x(x+1)$ 建立迭代公式

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

取x_0=0. 5, 逐次计算得

x_1=0. 57102

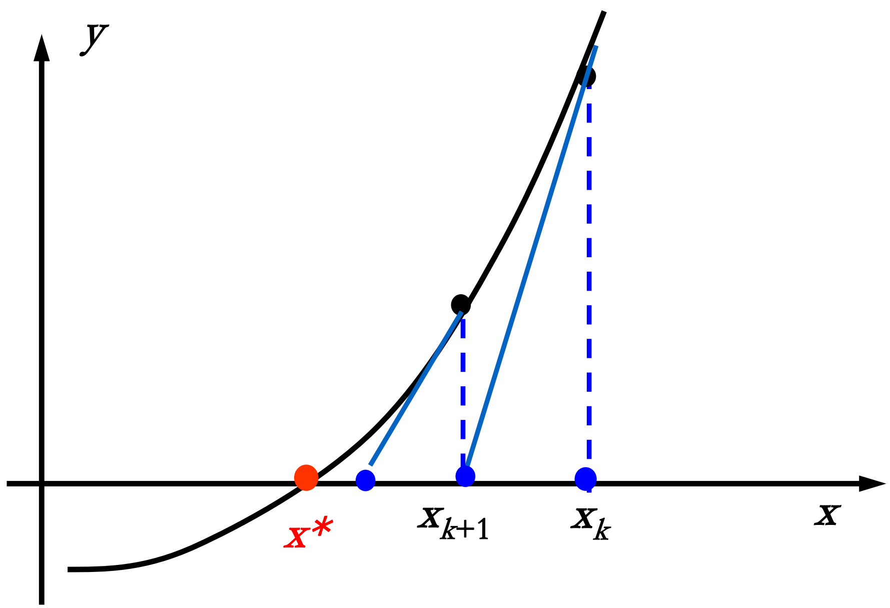
x_2=0. 56716

x_3=0. 56714

2.4.1 牛顿迭代法的几何解释

方程$f(x)=0$的根x^*是曲线$y=f(x)$与x轴交点的横坐标，设x_k是根x^*的某个近似值，过曲线$y=f(x)$的横坐标为x_k的点$P_k=(x_k,f(x_k))$引切线交x轴于x_{k+1}，并将其作为x^*新的近似值。

$$x_{k+1} \approx x_k - \frac{f(x_k)}{f'(x_k)}, \; k=0,1,2,\cdots$$



重复上述过程,一次次用切线方程可求解方程f(x)=0的根。

2.4.3 牛顿迭代法的收敛性分析

例2.10：只用加减乘除运算，设计一个二阶收敛算法计算$\sqrt{a}$ $(a>0)$

解：转化为求 $x^2-a=0$ 的正根

Newton 迭代：$x_{k+1}=x_k-\dfrac{f(x_k)}{f'(x_k)}=x_k-\dfrac{x_k^2-a}{2x_k}=\dfrac{1}{2}\left(x_k+\dfrac{a}{x_k}\right)$

$$\Rightarrow\quad x_{k+1}-\sqrt{a}=\frac{x_k^2+a}{2x_k}-\sqrt{a}=\frac{x_k^2+a-2x_k\sqrt{a}}{2x_k}=\frac{\left(x_k-\sqrt{a}\right)^2}{2x_k}$$

$$\Rightarrow\quad \frac{x_{k+1}-\sqrt{a}}{\left(x_k-\sqrt{a}\right)^2}=\frac{1}{2x_k}\Rightarrow\frac{1}{2\sqrt{a}}\Rightarrow$$ 二阶收敛

回顾收敛阶的定义

设迭代 $x_{k+1}=g(x_k)$ 收敛到 $g(x)$ 的不动点 x^*。记绝对误差 $e_k=x_k-x^*$，若

$$\lim_{k\to\infty}\left|\frac{e_{k+1}}{e_k^{\,p}}\right|=C\neq 0\quad (C\text{为常数})$$

则称该迭代为以收敛阶 p 收敛到 x^*。数 C 称为渐近误差常数。

2.4.3 牛顿迭代法的收敛性分析

定理2.3 设 $f(x)$ 在其零点 x^* 的某个邻域内二阶连续可导且 x^*是单根，则存在 x^* 的某个δ邻域 $N(x^*)=[x^*-\delta, x^*+\delta]$, 使得对 $\forall x_0\in N(x^*)$, 牛顿迭代法产生的如下迭代定义的序列以不低于二阶的收敛速度收敛到 x^*:

$$x_{k+1}=x_k-\frac{f(x_k)}{f'(x_k)}, k=0,1,2,\cdots$$

(局部收敛定理)

证明：若x^*是方程f(x)=0的单根，则有

$$f(x^*)=0, \quad f'(x^*)\neq 0.$$

Newton 法可以看作下面的不动点迭代：

$$x_{k+1}=\varphi(x_k)=x_k-\frac{f(x_k)}{f'(x_k)}$$

考虑Taylor展开式

$$0=f(x^*)=f(x_k)+f'(x_k)(x^*-x_k)+\frac{1}{2}f''(\xi)(x^*-x_k)^2 \qquad \xi\in[x^*,x_k]$$

可得

$$x_k-x^*=\frac{f(x_k)}{f'(x_k)}+\frac{f''(\xi)}{2f'(x_k)}(x^*-x_k)^2$$

$$x_k-\frac{f(x_k)}{f'(x_k)}-x^*=\frac{f''(\xi)}{2f'(x_k)}(x^*-x_k)^2$$

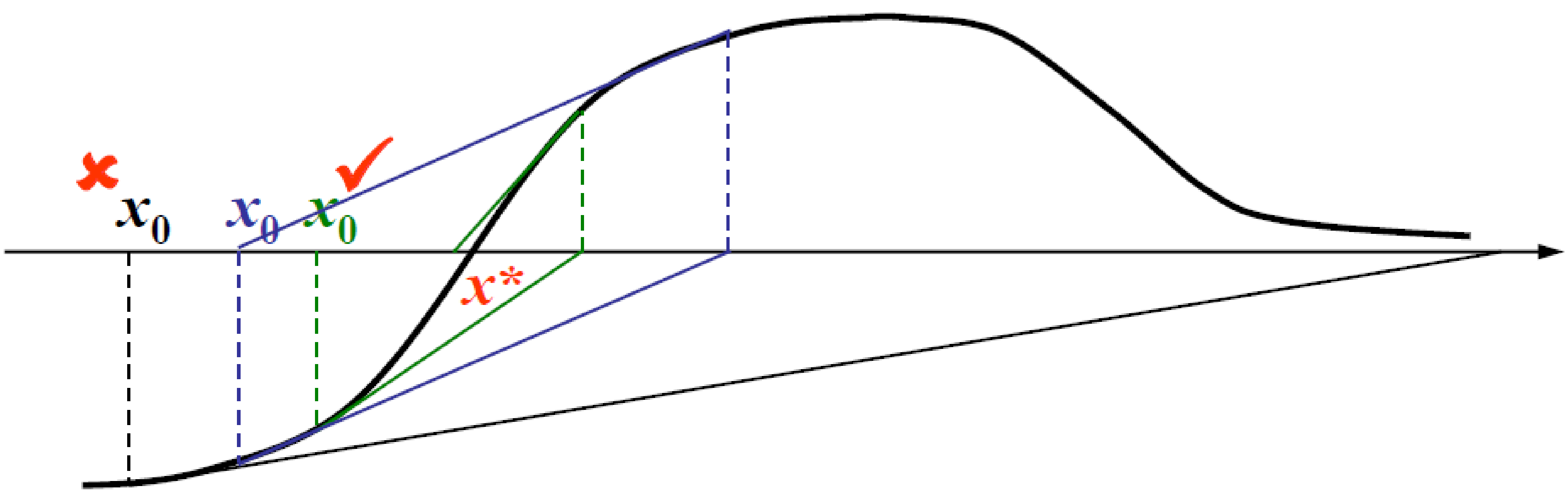
$$x_{k+1}-x^*=\frac{f''(\xi)}{2f'(x_k)}(x^*-x_k)^2$$

于是 $\lim\limits_{k\to\infty}\dfrac{\left|x^{*}-x_{k+1}\right|}{\left|x^{*}-x_{k}\right|^{2}}=\dfrac{\left|f''(x^{*})\right|}{2\left|f'(x^{*})\right|}$

所以 $\lim\limits_{k\to\infty}\dfrac{\left|e_{k+1}\right|}{\left|e_{k}\right|^{2}}=\lim\limits_{k\to\infty}\dfrac{\left|x^{*}-x_{k+1}\right|}{\left|x^{*}-x_{k}\right|^{2}}=\dfrac{\left|f''(x^{*})\right|}{2\left|f'(x^{*})\right|}$

Newton 法至少

二阶 局部收敛



非常重要

牛顿迭代法对初值x_0的要求比较高，x_0必须充分靠近x^*，才能保证局部收敛。

2.4.3 牛顿迭代法的收敛性分析　（非常重要）

重根情形

- 设 x^* 是 $f(x)$ 的 $m(m\geq 2)$ 重根，Newton法是否收敛？

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

Taylor 展式 ⟹ $$f(x) = \frac{1}{m!} f^{(m)}(\xi_1)(x - x^*)^m$$

Taylor 展式 ⟹ $$f'(x) = \frac{1}{(m-1)!} f^{(m)}(\xi_2)(x - x^*)^{m-1}$$

Taylor 展式 ⟹ $$f''(x) = \frac{1}{(m-2)!} f^{(m)}(\xi_3)(x - x^*)^{m-2}$$

Newton 迭代：$\varphi(x) = x - \dfrac{f(x)}{f'(x)}$

⟹ $$\varphi'(x^*) = \lim_{x\to x^*} \varphi'(x) = \lim_{x\to x^*} \frac{f(x) f''(x)}{\left[f'(x)\right]^2} = 1 - \frac{1}{m}$$

⟹ 线性收敛。　且重数 m 越高，收敛越慢。

2.4.3 牛顿迭代法的收敛性分析

提高收敛速度—提高收敛阶

法一：取 $\varphi(x) = x - m\dfrac{f(x)}{f'(x)}$ $\Rightarrow$ $\varphi'(x^*) = 0$ $\Rightarrow$ 二阶收敛

但 *m* 通常无法预先知道！

法二：将求 $f(x)$ 的重根转化为求另一个函数的单根。

令 $\mu(x) = \dfrac{f(x)}{f'(x)}$，则 x^* 是 $\mu(x)$ 的单重根。

构造针对 $\mu(x)$ 的具有二阶收敛的 Newton 迭代：

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

2.4.4 牛顿迭代法的算法实现

计算步骤

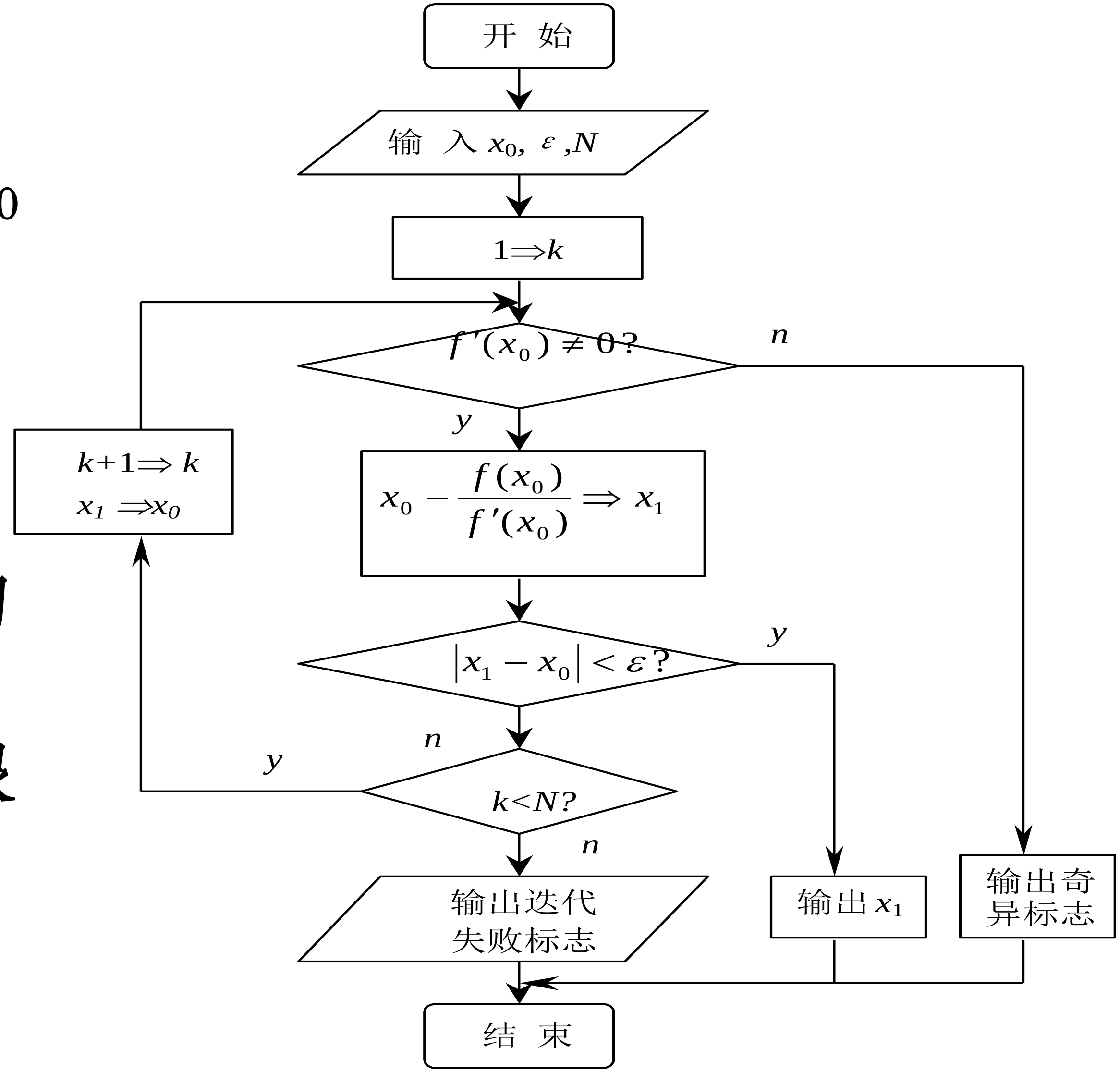
① 给出初始近似值x_0及精度ε

② 计算

$$\left[x_0-\frac{f(x_0)}{f'(x_0)}\right]\to x_1$$

① 若$|x_1-x_0|<\varepsilon$则转向4)；否则转向2)

② 输出满足精度的根x_1，结束。



Matlab源程序：

Newton_Iteration.m

```
function [p0,err,k,y]=Newton_Iteration(f,df,p0,delta,epsilon,max1)
%Input    - f is the object function
%         - df is the derivative of f
%         - p0 is the initial approximation to a zero of f
%         - delta is the tolerance for p0
%         - epsilon is the tolerance for the function values y
%         - max1 is the maximum number of iterations
%Output - p0 is the Newton-Raphson approximation to the zero
%         - err is the error estimate for p0
%         - k is the number of iterations
%         - y is the function value f(p0)
% f=@(x)  x*exp(x)-1; df=@(x)  exp(x)*(x+1);
% p0=1; delta=1e-4; epsilon=1e-3;  max1=500;
% Newton_Iteration(f,df, p0, delta, epsilon, max1)
for k=1:max1
    p1=p0-f(p0)/df(p0);
    err=abs(p1-p0);
    relerr=2*err/(abs(p1)+delta);
    p0=p1;
    y=f(p0);
    if (err<delta)|(relerr<delta)|(abs(y)<epsilon),
   break,
   end
end
```

作业2.3

4\. 设 $f(x)=x^3-3x-2$。

(a) 求出牛顿－拉夫森公式 $p_k=g(p_{k-1})$。

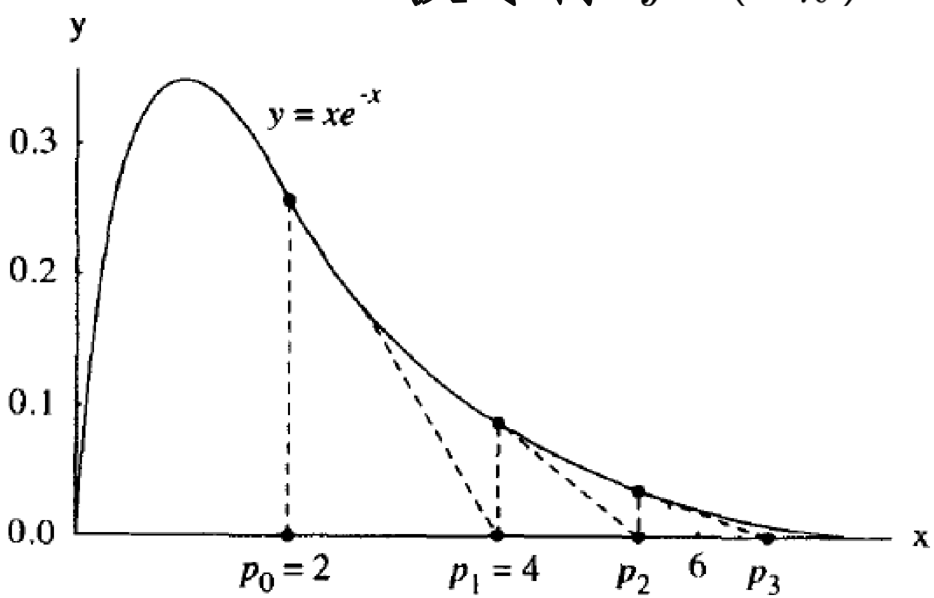
(b) 从 $p_0=2.1$ 开始，求 p_1, p_2, p_3 和 p_4。

(c) 序列是二次收敛的还是线性收敛的?

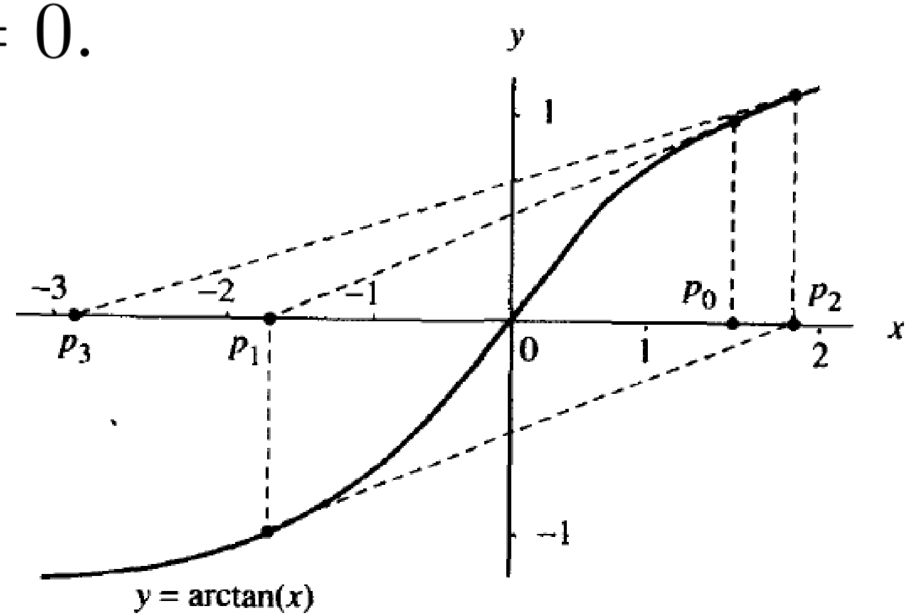
进一步分析

牛顿迭代法虽然具有收敛速度快的优点，但每迭代一次都要计算导数$f'(x_k)$，当f(x)比较复杂时，不仅每次计算$f'(x_k)$带来很多不便，而且还可能十分麻烦。如果用不计算导数的迭代方法，往往只有线性收敛的速度。

缺陷：　1. 被零除 $f'(x_k)=0.$



2. 迭代产生一个发散序列



3. 迭代产生一个发散循环序列

2.5 割线法

本节介绍的割线法便是一种不必进行导数运算的求根方法。割线法在迭代过程中不仅用到前一步x_k处的函数值，而且还使用x_{k-1}处的函数值来构造迭代函数，这样做能提高迭代的收敛速度。

2.5.1 割线法的基本思想

为避免计算函数的导数$f'(x_k)$，使用差商

$$\frac{f(x_k)-f(x_{k-1})}{(x_k-x_{k-1})}$$

替代牛顿公式中的导数$f'(x_k)$，便得到迭代公式

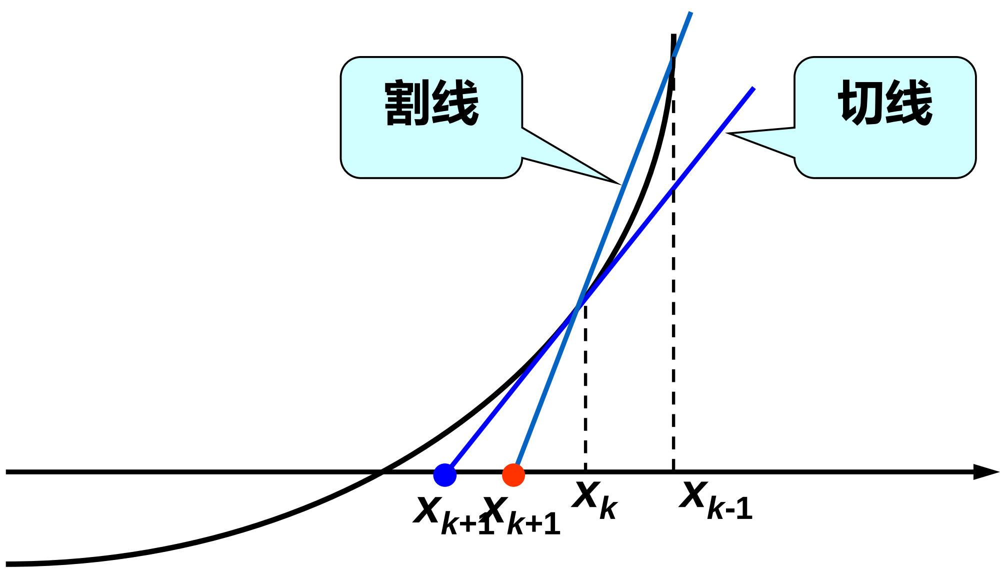
$$x_{k+1}=x_k-\frac{f(x_k)}{f(x_k)-f(x_{k-1})}(x_k-x_{k-1}),\ k=1,2,3,\cdots$$

称为割线迭代公式，相应的迭代法称为割线法。

这种方法称为多点迭代法。

2.5.2 割线法的几何意义 (非常重要)

用过曲线上两点P0(x0,f(x0))、P1(x1,f(x1))的割线来代替曲线，用割线与x轴交点的横座标作为方程的近似根x2 。 再过P1点和点 P2(x2,f(x2))作割线求出x3。以此类推，当收敛时可求出满足精度要求的xk。



切线斜率 ≈ 割线斜率

$$f'(x_k) \approx \frac{f(x_k) - f(x_{k-1})}{x_k - x_{k-1}}$$

- ✓ 只需计算函数值，避免计算导数；
- ✓ 需要两个初始点；
- ✓ 收敛比牛顿迭代法稍慢，但对初始点要求同样高。

2.5.2 割线法的几何意义

例2.11 用割线法求方程$x=e^{-x}$在$x_0=0.5$初始值邻近的一个根。要求$|x_{k+1}-x_k|<0.0001$。

解：取$x_0=0.5$、$x_1=0.6$，令$f(x)= x-e^{-x}$ 利用割线迭代公式，有

$$x_{k+1}=x_k-\frac{(x_k-e^{-x_k})}{(x_k-x_{k-1})-(e^{-x_k}-e^{-x_{k-1}})}(x_k-x_{k-1})$$

计算结果如下表

k	x_k	$\lvert x_{k+1}-x_k\rvert$
0	0. 500 00	
1	0. 600 00	
2	0. 567 54	0. 032 46
3	0. 567 15	0. 000 39
4	0. 567 14	0. 000 01

取近似根$x_4=0.56714$,即可满足精度要求。

2.5.3 割线法的收敛性分析 （重要）

定理2.4 设 x^* 是 $f(x)$ 的单重零点，$f''(x)$ 在 x^* 的某个邻域内连续，则存在 x^* 的一个邻域 $N(x^*)=[x^*-\delta, x^*+\delta]$，使得当 $x_0, x_1 \in N(x^*)$ 时，由两步弦截法产生的序列收敛到 x^*，且收敛阶为

$$\left(\sqrt{5}+1\right)/2 \approx 1.618$$

2.5.4 割线法的算法实现

割线法的收敛速度比牛顿迭代法要慢，但优点是不需要计算导数。

1.计算步骤

(1)选定初值x_0、x_1，并计算相应的函数值$f(x_0)$、$f(x_1)$；

(2)计算

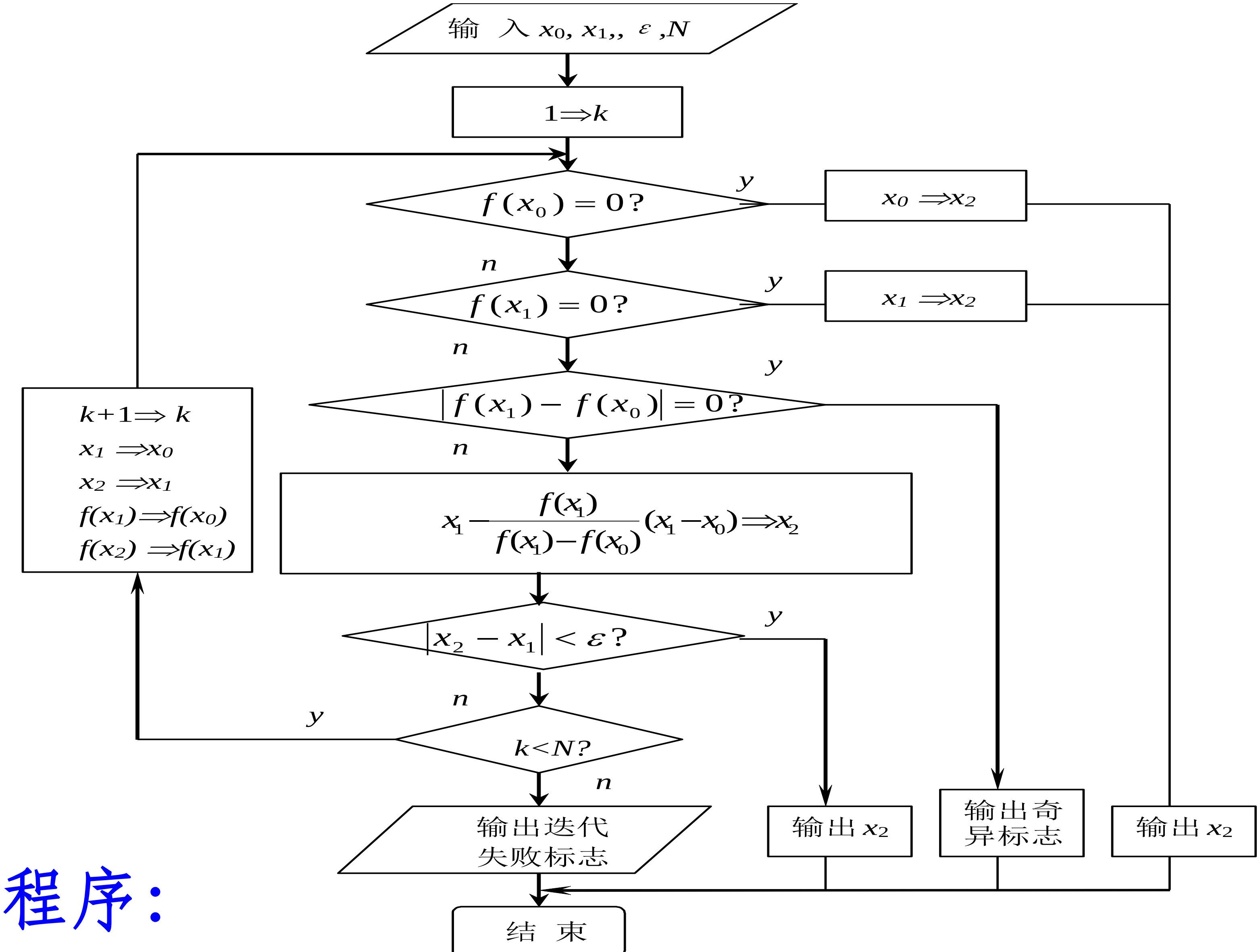
$$x_2 = x_1 - \frac{f(x_1)}{f(x_1) - f(x_0)}(x_1 - x_0) \quad x_1 \to x_0, x_2 \to x_1$$

(3)若$|x_1-x_0|<\varepsilon$(ε为给定的精度要求)，

则转(4)，否则转向(2)；

(4)输出满足精度要求的根x_1，结束。

2.5.4 割线法的算法实现



Matlab源程序：

Gexianfa_Iteration.m

Matlab源程序：

Gexianfa_Iteration.m

```
function [p1,err,k,y]=Gexianfa_Iteration(f,p0,p1,delta,epsilon,max1)

%Input    - f is the object function
%         - p0 and p1 are the initial approximations to a zero of f
%         - delta is the tolerance for p1
%         - epsilon is the tolerance for the function values y
%         - max1 is the maximum number of iterations
%Output - p1 is the secant method approximation to the zero
%         - err is the error estimate for p1
%         - k is the number of iterations
%         - y is the function value f(p1)
% f=@(x)  x*exp(x)-1;  ;
% p0=1; p1=1.2; delta=1e-4; epsilon=1e-3;  max1=500;
% Gexianfa_Iteration(f,p0, p1, delta, epsilon, max1)

for k=1:max1
    p2=p1-f(p1)*(p1-p0)/(f(p1)-f(p0));
    err=abs(p2-p1);
    relerr=2*err/(abs(p2)+delta);
    p0=p1;
    p1=p2;
    y=f(p1);
    if (err<delta)|(relerr<delta)|(abs(y)<epsilon),
     break,
    end
end
```

2.6 迭代收敛的加速办法(选讲)

对于收敛的迭代过程足够多次，就可以使结果达到任意的精度。但有时迭代过程收敛缓慢，从而使计算量变得很大。因此，可考虑对迭代作加速处理。

2.6.1 埃特金过程

埃特金加速技术可加速任意线性收敛的序列。

2.6.1 埃特金(Aitken)过程

定义2.2 设有序列 $\{p_n\}_{n=0}^{\infty}$,用如下表达式定义前向微分Δp_n:

$$\Delta p_n = p_{n+1} - p_n,\ n \geq 0.$$

高阶$\Delta^k p_n$ 可定义为

$$\Delta^k p_n = \Delta^{k-1}(\Delta p_n),\ k \geq 2.$$

定义2.3（埃特金加速）设序列$\{p_n\}_{n=0}^{\infty}$ 线性收敛到极限 ***p***, 而且对所有$n \geq 0$, 有 $p - p_n \neq 0$. 如果存在实数**A**, 且|**A**|<**1**, 满足

$$\lim_{n\to\infty} \frac{p-p_{n+1}}{p-p_n} = A.$$

则定义为

$$q_n = p_n - \frac{(\Delta p_n)^2}{\Delta^2 p_n} = p_n - \frac{(p_{n+1}-p_n)^2}{p_{n+2}-2p_{n+1}+p_n}$$

的序列$\{q_n\}_{n=0}^{\infty}$收敛到 ***p***, 且比 $\{p_n\}_{n=0}^{\infty}$快，即$\lim\limits_{n\to\infty} \frac{q-q_n}{p-p_n} = 0.$

2.6.1 埃特金过程

例2.12 用埃特金方法对不动点迭代序列

$$p_0 = 0.5,\ p_{k+1} = e^{-p_k},\ k = 0, 1, 2, \cdots$$

进行加速.

$$q_n = p_n - \frac{(p_{n+1} - p_n)^2}{p_{n+2} - 2p_{n+1} + p_n}$$

表 2.10 线性收敛序列 $\{p_n\}$

n	p_n	$E_n = p_n - p$	$A_n = \dfrac{E_n}{E_{n-1}}$
1	0.606530660	0.039387369	−0.586616609
2	0.545239212	−0.021904079	−0.556119357
3	0.579703095	0.012559805	−0.573400269
4	0.560064628	−0.007078663	−0.563596551
5	0.571172149	0.004028859	−0.569155345
6	0.564862947	−0.002280343	−0.566002341

表 2.11 用埃特金 Δ^2 过程得到的序列 $\{q_n\}$

n	q_n	$q_n - p$
1	0.567298989	0.000155699
2	0.567193142	0.000049852
3	0.567159364	0.000016074
4	0.567148453	0.000005163
5	0.567144952	0.000001662
6	0.567143825	0.000000534

显然，序列$\{q_n\}_{n=0}^{\infty}$的收敛要比$\{p_n\}_{n=0}^{\infty}$快。

作业2.5

6. 序列 $p_n = 1/(4^n + 4^{-n})$ 线性收敛到 0。利用埃特金公式(4)求 q_1，q_2 和 q_3，从而加速收敛。

n	p_n	q_n
0	0.5	−0.26437542
1	0.23529412	
2	0.06225681	
3	0.01562119	
4	0.00390619	
5	0.00097656	

另外两种加速方法自学：

米勒法和斯蒂芬森加速法

本章教学要求及重点难点

- 理解方程根的基本概念
- 掌握二分法、迭代法、牛顿迭代法、割线法求方程近似根的基本方法
- 掌握迭代过程的敛散性证明
- 掌握迭代法、Newton迭代法的算法设计思想
- 重点：迭代法求根原理、迭代算法的基本思想
- 难点：迭代法的收敛性分析与证明
- 了解：埃特金加速技术

课堂作业

(1) 分别运用牛顿迭代法和割线法设计算法求解函数

$$f(x) = x^2 - 2x$$

在区间[-1, 0.5]内满足误差小于0.01的根。

注：无需私信或QQ邮箱发我程序或运行结果，只需截图发到群里即可。