第五章　　方程求根的数值解

/* Solutions of Nonlinear Equations */

HUST

引入

在科学与工程计算中经常需求解方程$f(x)=0$的根

① 当$f(x)=a_n x^n+a_{n-1}x^{n-1}+\ldots\ldots a_1 x+a_0$ 时

$n=1,2,3,4$时,可用求根公式求解

$n \geqslant 5$时,不能用公式表示方程的根

② 对于一般的非线性方程$f(x)=0,\ x\in R$，只能求出其近似值，我们探讨其数值解法——逐步逼近法.

1.初始近似根x_0

2. $x_k \rightarrow$ 递推关系 $\rightarrow x_{k+1}$　　(x_k收敛于真解x^*)

HUST

5.1 根的隔离和二分法

为了确定初始近似根 x_0， 必须知道f(x)=0 的根的大致范围。

若f(x)=0 在(a,b) 内有一个根，称(a,b)为f(x)=0 的有根区间;

若f(x)=0 在 (a,b)只有一个根，称(a,b)为f(x)=0 的隔根区间。

当(a,b) 为隔根区间时，可取$x_0\in(a,b)$.

Def: 根的隔离——求f(x)=0的隔根区间的过程.

根的隔离的依据

Th5.1 设函数f(x)在[a,b]上连续，且有f(a)f(b)<0, 则方程 f(x)=0 在[a,b]内至少有一个根.

注: ① [a,b]为有根区间

② 当 f(x) 满足Th5.1 的条件且在[a,b] 上单调时，

f(x)在[a,b]内只有一根;

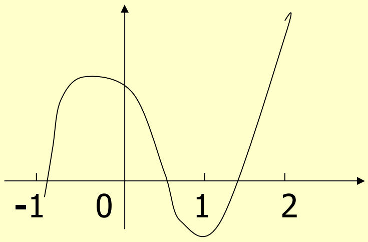
即 (a,b)为隔根区间.

根的隔离的方法

图象法:作出 y=f(x)的草图，由曲线y=f(x) 与x轴的交点的大致位置来确定隔根区间.

例 隔根区间: (-1,0),(0,1),(1,2)

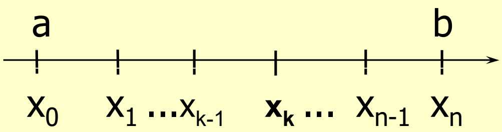
区间端点上函数值f(x) 异号



逐步搜索法

已知: [a,b]为 f(x)=0的有根区间，且[a,b] 较大,

求一个缩小的有根区间



- 取步长h=(b-a)/n
- 从x_0=a出发以h 为步长向右搜索直至找到第一个点x_k=a+kh 满足 $f(a)f(x_k)\le 0$, 则得缩小得有根区间$[x_{k-1},x_k]$.
- 取初始近似根为x_{k-1}或x_k，其误差限为h.

Algorithm

step 1: $x_0 = a,\ h=(b-a)/n$

step 2: If $(f(x_0)f(x_0+h) \le 0)$ 输出(x_0,x_0+h)

Else $x_0 = x_0+h$, goto step 2

例 求方程$f(x)=x^3-x-1=0$ 的有根区间。

解: $\because f(0)=-1<0,\ f(+\infty)>0$ $\therefore (0,+\infty)$为有根区间

从$x=0$ 出发, 步长$h=0.5$向右计算, 则

x	0	0.5	1.0	1.5	2.0
f(x)	-	-	-	+	+

得缩小的有根区间为（1.0，1.5）

可取初始近似根$x_0=1.0$或$x_0=1.5$

小结：当 h很小时，得到很小的有根区间，

取$x^* \in (x_0,x_0+h)$，从而可算得任意精度的近似根，

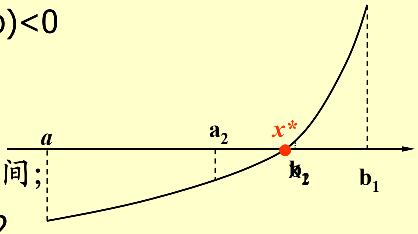
但h越小计算量越大，利用此法求近似根仍不十分理想。

5.1.2 二分法 /* Bisection Method */

Chapter 5 Solutions of equations in one varible

思想: 将有根区间 [a,b]逐次减半（二分），使有根区间缩小直到误差容许范围内，然后取区间中点为真根x^* 的近似值。

设$f(x)=0$的有根区间为$[a,b]$且$f(a)f(b)<0$



(1) 取$x_0=(a+b)/2$

If $f(x_0)=0$，则 x_0为 $f(x)=0$的根;

else if $f(a)f(x_0)<0$, 则$[a,x_0]$为有根区间;

记 $a_1=a,\ b_1=x_0=(a+b)/2$

else $f(x_0)f(b)<0$,则$[x_0,b]$ 为有根区间,记 $a_1=x_0=(a+b)/2,\ b_1=b$

$\therefore$ 得缩小的有根区间$[a_1,b_1]$且$b_1-a_1=(b-a)/2$，$[a,b]$包含$[a_1,b_1]$

(2) 将$[a_1,b_1]$ 二等分，其中点$x_1=(a_1+b_1)/2$，计算$f(x_1)$，重复(1),

或$f(x_1)=0$ 则$x^*=x_1$，或得有根区间 $[a_2,b_2]$ 且$b_2-a_2=(b_1-a_1)/2$.

(3) ……反复进行，则得到有根区间套

$$[a,b]\supset[a_1,b_1]\supset[a_2,b_2]\supset\ldots\supset[a_k,b_k]\ldots\ni x^*$$

$$b_k-a_k=\tfrac{1}{2}(b_{k-1}-a_{k-1})=\ldots=\tfrac{1}{2^k}(b-a)$$

记 $[a_k,b_k]$ 的中点为 $x_k=(a_k+b_k)/2$ 并作为根的近似值，

从而有近似根序列：$x_0, x_1, x_2, \ldots\ldots, x_k, \ldots..$

$$\because \lim_{k\to\infty}(b_k-a_k)=\lim_{k\to\infty}\frac{b-a}{2^k}=0 \text{ 且 } \forall k,\ x^*\in[a_k,b_k] \quad \therefore \lim_{k\to\infty}x_k=x^*$$

二分法是收敛的

将有限次二分的结果x_k 作为根的近似值，其误差为多少呢？

$$\because x^*, x_k\in[a_k,b_k],\ x_k=\frac{a_k+b_k}{2} \qquad \therefore |x^*-x_k|\le\frac{1}{2}(b_k-a_k)=b_{k+1}-a_{k+1}$$

从而误差估计式 $|x^*-x_k|\le\dfrac{b-a}{2^{k+1}}$

$$|x^*-x_k|\le|b_{k+1}-a_{k+1}|=\frac{b-a}{2^{k+1}}$$

于是用二分法解f(x)=0，使误差不超过ε的终止准则：

（1） 先验估计 $|x^*-x_k|\le\dfrac{b-a}{2^{k+1}}<\varepsilon \Rightarrow k>\dfrac{\ln(b-a)-\ln\varepsilon}{\ln 2}-1$

（2） 后验估计 $b_{k+1}-a_{k+1}<\varepsilon$

Algorithm

Step1. 输入a, b, ε, δ

Step2. X=(a+b)/2

Step3. if (|f(x)|<δ 或b-x<ε) 输出x stop

else if (f(a)f(x)<0) b=x

else a=x

Step4. Goto Step2

例题

例 5.2 求方程$f(x)=x^3-x-1$在区间(1,1.5)内的根，要求精确到小数点后的第二位，($\varepsilon=10^{-2}/2$)，用四位小数计算。

解：① $a=1,b=1.5$ 且$f(a)<0$, $f(b)>0$

精度要求为 $\varepsilon=10^{-2}/2=0.005$ 由误差估计式$|x^*-x_k|\le(b-a)/2^{k+1}$

得$0.5/2^{k+1}<0.005$，从而 $2^{k+1}>100$，取$k=6$ 即可。

② $x_0=\frac{1}{2}(a+b)=1.25$ $f(x_0)<0$

$\because f(x_0)f(b)<0$ $\therefore$ 令$a_1=x_0=1.25$, $b_1=b=1.5$

新的有根区间(a_1,b_1) 取 $x_1=\frac{1}{2}(a_1+b_1)=1.375$，$f(x_1)>0$ $f(a_1)f(x_1)<0$

$\therefore a_2=a_1=1.25$, $b_2=x_1=1.375$ 从而得有根区间(a_2,b_2)， … …

二分法分析

① 简单;收敛性有保证;
② 对 $f(x)$ 要求不高(只要连续即可)。

HW:
作业五 #1

① 无法求复根及偶重根;
② 收敛慢：线性收敛速度。

注：用二分法求根，最好先给出 $f(x)$ 草图以确定根的大概位置。或用搜索程序，将$[a, b]$分为若干小区间，对每一个满足条件 $f(a_k)\cdot f(b_k)<0$ 的区间调用二分法程序，可求$[a, b]$内的多个根。

5.2 迭代法

Chapter 5 Solutions of equations in one varible

思想：先给出f(x)=0的一个初始近似根x_0，再反复使用某一公式校正这个初始根，使之逐步精确化，直到满足精度要求为止。

$$\begin{cases} x_0 & \text{迭代初值} \\ x_{k+1}=g(x_k) \quad k=0,1,2,\ldots & \text{迭代格式} \end{cases}$$

g(x)称为迭代函数。

如何构造迭代格式？——不动点迭代法/* Fixed-Point Iteration */

$$f(x)=0 \xleftrightarrow{\text{等价变换}} x=g(x)$$

$f(x)$ 的根 ⟷ $g(x)$ 的不动点

思路

从 x_0 出发，计算 $x_1=g(x_0)$, $x_2=g(x_1)$, …, $x_{k+1}=g(x_k)$, … 若 $\{x_k\}_{k=0}^{\infty}$ 收敛，即存在 x^* 使得 $\lim_{k\to\infty} x_k = x^*$，且 g 连续，则由 $\lim_{k\to\infty} x_{k+1} = \lim_{k\to\infty} g(x_k)$，得 $x^*=g(x^*)$，即x^* 是 g 的不动点，也就是$f(x)=0$的根。

HUST

迭代法的几何解释

Chapter 5 Solutions of equations in one varible

x=g(x) 的解即为曲线y=g(x) 与直线y=x 的交点p*

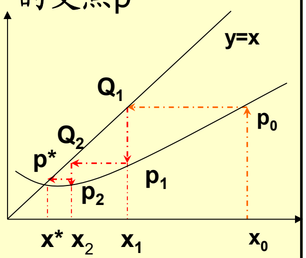
初始值x_0, 得y=g(x) 上点$p_0(x_0,g(x_0))$

…$p_1(x_1,g(x_1))$　$x_1=g(x_0)$

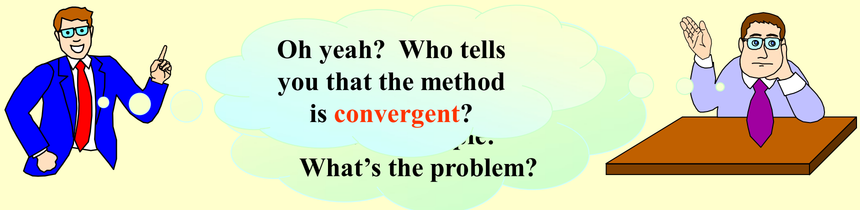
…$p_2(x_2,g(x_2))$　$x_2=g(x_1)$

……

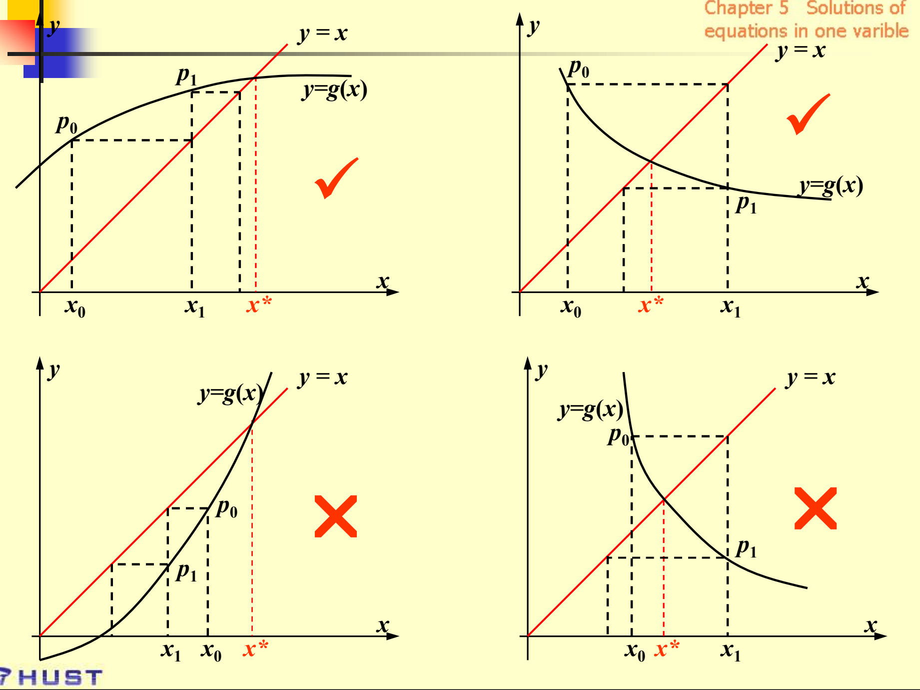
…$p_{k+1}(x_{k+1},g(x_{k+1}))$　$x_{k+1}=g(x_k)$



当由$x_{k+1}=g(x_k)$ 所决定的点列$x_1,x_2,\ldots,x_k,\ldots$收敛到$x^*$.
则点$p_1,p_2,\ldots\ldots p_k,\ldots\ldots$逐步逼近交点p*.



HUST



例：求 $x^3-x-1=0$ 在 $x=1.5$ 附近的一个根（用六位有效数字近似）

解 I： $x^3-x-1=0$ 等价于 $x=\sqrt[3]{x+1}$ 取 $g(x)=\sqrt[3]{x+1}$ 取 $x_0=1.5$ 则

$x_1=\sqrt[3]{1.5+1}=1.35721$　　　$x_7=\sqrt[3]{x_6+1}=1.32472$

$x_2=\sqrt[3]{1.35721+1}=1.33086$　　　$x_8=\sqrt[3]{x_7+1}=1.32472=x_7$

…　　　$\Rightarrow x^*\approx 1.32472$

II： $x^3-x-1=0$ 等价于 $x=x^3-1$； $g(x)=x^3-1$

迭代格式为 $x_{k+1}=x_k^3-1$ $(k=0,1,2,\ldots)$

取 $x_0=1.5$ 则 $x_1=1.5^3-1=2.375$， $x_2=x_1^3-1=12.3976$

…………

序列 $\{x_k\}$ 发散

注：(1)必须适当选取 x_0 及 $g(x)$ 才能使迭代公式所求的序列 $\{x_k\}$ 收敛；

(2) $\{x_k\}$ 收敛时，k 次迭代的结果的误差 $\varepsilon_k=x_k-x^*=$?

迭代法收敛性分析

$f(x)=0 \Leftrightarrow x=g(x)$, 从而有迭代格$x_{k+1}=g(x_k)$ $k=0,1,2,\ldots\ldots$

Th5.3 设迭代函数 $g(x)$ 在$[a,b]$上具有连续的一阶导数，且当

(1) $x\in[a,b]$ 时，$a\leqslant g(x)\leqslant b$;

(2) 存在正数$L<1$，对 $x\in[a,b]$ 有$|g'(x)|\leqslant L<1$成立，

则 $x=g(x)$ 在 $[a,b]$ 上有唯一解 x^*，且对任意的初始近似值 $x_0\in[a,b]$迭代过程$x_{k+1}=g(x_k)(k=0,1,2,\ldots\ldots)$ 收敛且 $\lim\limits_{k\to\infty}x_k=x^*$

证明： ① x^*的存在性

作 $h(x)=x-g(x)$，则$h(x)$ 在$[a,b]$上连续；

而 $h(a)=a-g(a)\leqslant 0$，$h(b)=b-g(b)\geq 0$；

由连续函数性质，必有$x^*\in[a,b]$ 使 $h(x^*)=0$ 即$x^*=g(x^*)$ ✓

② x^*的唯一性

若$x^{\triangle}\in[a,b]$,且$x^{\triangle}=g(x^{\triangle})$，则$x^*-x^{\triangle}=g(x^*)-g(x^{\triangle})$,

那么$x^*-x^{\triangle}=g'(\xi)(x^*-x^{\triangle})$，$\xi$在$x^*$与$x^{\triangle}$之间且$\xi\in[a,b]$，

则$(x^*-x^{\triangle})(1-g'(\xi))=0$，而$|g'(\xi)|\leq L<1$， $\therefore x^*=x^{\triangle}$ ✓

③ 迭代过程的收敛性

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

由条件(1)知$x_k\in[a,b]$，则$\xi\in(a,b)$，

$$\begin{aligned}\therefore |x^*-x_{k+1}| &= |g'(\xi)||x^*-x_k| \\ &\leq L|x^*-x_k|\leq L^2|x^*-x_{k-1}|\leq\ldots \\ &\leq L^{k+1}|x^*-x_0|\to 0,\quad (\because 0\leq L<1)\end{aligned}$$

$$\therefore \lim_{k\to\infty}x_k=x^*$$ ✓

说明

- Th中条件(1) 迭代序列$\{x_k\}$均在$[a,b]$内；
- 条件(2)保证x_k与x^*间的距离随k增加而减少并最终趋于0。
- 对$x^3-x-1=0$ 的两种格式分析

$$g(x)=\sqrt[3]{x+1}\qquad g(x)=x^3-1\qquad [a,b]=[1,2]$$

误差估计式

Th5.4 在 Th5.3的条件下，有如下误差估计式

$$|x^*-x_k| \le \frac{|x_{k+1}-x_k|}{1-L} \quad \text{后验估计}$$

$$|x^*-x_k| \le \frac{L^k}{1-L}|x_1-x_0| \quad \text{先验估计}$$

证:

$\because |x^*-x_{k+1}| \le L|x^*-x_K|$

$$\begin{aligned}\therefore |x_{k+1}-x_k| &= |x^*-x_k-(x^*-x_{k+1})| \\ &\ge |x^*-x_k|-|x^*-x_{k+1}| \\ &\ge |X^*-X_K|-L|X^*-X_K| \\ &= (1-L)|X^*-X_K|\end{aligned}$$

$$\therefore \quad |X^*-X_K| \le \frac{|X_{k+1}-X_K|}{1-L}$$

又 $|x_{k+1}-x_K|$

$$\begin{aligned}&= |g(x_k)-g(x_{k-1})| \\ &= |g'(\xi)(x_k-x_{k-1})|,\ \xi \in (a,b), \\ &\le L|x_k-x_{k-1}| \\ &\le L^2|x_{k-1}-x_{k-2}| \\ &\cdots\cdots \\ &\le L^k|x_1-x_0|\end{aligned}$$

$$\therefore \quad |x^*-x_k| \le \frac{L^k}{1-L}|x_1-x_0|$$

停机准则: ①先验估计

$\because |x^*-x_k| \le \frac{L^k}{1-L}|x_1-x_0| \le \varepsilon$

$\therefore k > [\ln \frac{\varepsilon(1-L)}{|x_1-x_0|} \div \ln L]$

②后验估计

$|x^*-x_k| \le \frac{1}{1-L}|x_{k+1}-x_k| \le \varepsilon$ 若L不接近于1

实用方法: $|x_{k+1}-x_k| < \varepsilon$

Algorithm: Fixed-Point Iteration

Find a solution to $x = g(x)$ given an initial approximation x_0.

Input: initial approximation x_0; tolerance TOL; max. num. of iterations N_{max}.

Output: approximate solution x or message of failure.

Step 1 Set $i = 1$;

Step 2 While ($i \le N_{max}$) do steps 3-6

Step 3 Set $x = g(x_0)$; /* compute x_i */

Step 4 If $|x - x_0| < TOL$ then Output (x); /* successful */
STOP;

Step 5 Set i++;

Step 6 Set $x_0 = x$;
/* update x_0 */

当 x 很大时，此处可改为 $\left|\frac{x-x_0}{x}\right| < TOL$

Step 7 Output (The method failed after N_{max} iterations); /* unsuccessful */
STOP.

例5.3求方程 $x=e^{-x}$ 在 $x_0=0.5$ 附近的近似根,要求精确到小数后三位.

解: 此时 $f(x)= x-e^{-x} =0$

$f(0.5)<0 \quad f(0.6)>0$

∴ [0.5,0.6] 为f(x)=0 的有根区间。

取 $g(x)=e^{-x}$; 从而迭代格式 $x_{k+1}=e^{-x_k} \quad k=0,1,2,\cdots\cdots$

判定收敛性:

当 $x\in[0.5,0.6]$, $|g'(x)|=|-e^{-x}|=e^{-x}\le e^{-0.5}\approx 0.607<1$

∴ 迭代格式收敛

取 $x_0=0.5$，精度要求 $\varepsilon=10^{-3}/2=0.0005$ 迭代,

结果见p129表

Chapter 5 Solutions of equations in one varible

k	x_k	$x_k - x_{k-1}$
0	0.5	
1	0.60653	0.10653
2	0.54524	−0.06129
3	0.57970	0.03446
4	0.56007	−0.01963
5	0.57117	0.0110
6	0.56486	−0.00631
7	0.56844	0.00358
8	0.56641	−0.00203
9	0.56756	0.00115
10	0.56691	−0.00065

注: 定理条件非必要条件，可将 $[a, b]$ 缩小，定义局部收敛性。

HUST

局部收敛性

Th5.3 的迭代收敛条件之一**:** $x \in [a,b]$，$|g'(x)| \le L<1$

在$[a,b]$较大时，其该条件不易满足，考虑局部收敛性 ——

Def: 若在x^*的某邻域$\triangle$:$|x-x^*|\le\delta$, 迭代过程对任意的初始值 $x_0\in\triangle$均收敛，则称其具有局部收敛性。

Th5.5 设$g(x)$在$x=g(x)$ 的根x^*邻近有连续的一阶导数，且 $|g'(x^*)|<1$，则迭代过程$x_{k+1}=g(x_k)$具有局部收敛性。

局部收敛性

Chapter 5 Solutions of equations in one varible

Th5.5 设$g(x)$在$x=g(x)$ 的根x^*邻近有连续的一阶导数，且 $|g'(x^*)|<1$，则迭代过程$x_{k+1}=g(x_k)$具有局部收敛性**.**

分析**:** 在$\triangle$:$|x-x^*|\le\delta$即$[x^*-\delta, x^*+\delta]$应用**Th5.3**来证明.

证 $\because |g'(x^*)|<1$且$g'(x)$ 在x^*的邻近连续

$\therefore$ 存在充分小的邻域$\triangle$：$|x-x^*|\le\delta$使

$x \in\triangle$时，$|g'(x)| \le L<1$ （L为常数）

而 $g(x)-g(x^*)=g'(\xi)(x-x^*)$

又 $x \in\triangle$时$\xi\in\triangle$，有$|g'(\xi)| \le L<1$.

$\therefore |g(x)-x^*|=| g(x)-g(x^*) | \le L|x-x^*|<|x-x^*|\le\delta$，即$g(x) \in\triangle$。

$\therefore$ $g(x)$在x^*的 δ 邻域$\triangle$内满足**Th5.3**收敛条件(1) (2);

$\therefore x_{k+1}=g(x_k)$ 对任意 $\mathbf{x_0}\in \triangle$ 收敛，即具有局部收敛性。

例5.4 求$x^3-2x-5=0$ 在$x_0=2$ 附近的实根。

解：由$x^3-2x-5=0$ 得 $x=\sqrt[3]{2x+5}$ $\Rightarrow$ $g(x)=\sqrt[3]{2x+5}$

$$\begin{cases} x_{k+1}=g(x_k)=\sqrt[3]{2x_k+5} \\ x_0=2 \end{cases} \qquad g'(x)=\tfrac{2}{3}(2x+5)^{-\frac{2}{3}}$$

∵ $g'(x_0)<1/6$ 且$g'(x)$在$x_0=2$邻近连续

∴ 迭代格式$x_{k+1}=g(x_k)$ 在$x_0=2$的邻域内具有局部收敛性

$x_1=\sqrt[3]{2x_0+5}=2.0800838$

$x_2=2.0923507,\ x_3=2.0942170,$

$x_4=2.0945006,\ x_5=2.0945438,$

$x_6=2.0945503,$

$x^*=2.0945514815$

误差逐步减小，减小速度为6^{-k}

注： 构造 $x=\frac{1}{2}(x^3-5)$ $\quad g(x)=\frac{1}{2}(x^3-5)$ $\quad g'(x)=\frac{3}{2}x^2$ $\quad g'(2)=6>1$

如取 $x_0=2$，则 $x_1=1.5,\ x_2=-0.125,\ x_3=-2.500,$

$x_4=-10.312,\ x_5=-551.2,\cdots,$ 是发散序列。

迭代法在实时系统设计中的应用

假定实时系统由N个实时任务构成集合 $\Gamma=\{t_1, t_2, \ldots, t_N\}$

$\Gamma=\{t_i=\langle C_i, T_i, D_i, p_i\rangle \mid i=1, \ldots, N\}$

$R_i\leq D_i,\ i=1, 2, \ldots, N$

$$R_i=C_i+\sum_{\forall t_j\in\Gamma,\, p_j>p_i}\left\lceil\frac{R_i}{T_j}\right\rceil C_j$$

$$R_i^{(n+1)}=C_i+\sum_{\forall t_j\in\Gamma,\, p_j>p_i}\left\lceil\frac{R_i^{(n)}}{T_j}\right\rceil C_j$$

取迭代初值 $R_i^{(0)}=0$

如果 $R_i^{(n+1)}=R_i^{(n)}$ 表明迭代收敛，$R_i=R_i^{(n+1)}$

如果 $R_i^{(n+1)}>D_i$ 表明任务t_i是不可调度的.

5.3.1 迭代过程的收敛速度

迭代格式 $x_{k+1}=g(x_k)$的收敛速度依赖于什么？ $|x^*-x_k|\le\frac{L^k}{1-L}|x_1-x_0|$

若$|g'(x)|\le L<1$:

当 $L\approx0$ 时　　收敛快，

当 $L\approx1$ 时　　收敛慢，

而 $L>1$ 时，　不收敛(发散)。

收敛速度用收敛阶来衡量：

Def5.2 迭代序列$\{x_k\}$收敛于$f(x)=0$ 的根x^* ($x_k\to x^*$),

记第k 步迭代的误差为 $e_k=x_k-x^*(k=0,1,2,\ldots\ldots)$,

若有某个实数 $p\ge1$和非零常数C 使 $\lim_{k\to\infty}\frac{|e_{k+1}|}{|e_k|^p}=C\ (k\ne0)$

则称$\{x_k\}$是 p阶收敛的。

注： p的大小反映收敛速度的快慢，p越大，收敛越快

p=1 —— 线性收敛； p=2 —— 平方收敛

p>1 —— 超线性收敛

Th5.6 对于迭代过程$x_{k+1}=g(x_k)$，如果迭代函数g(x)在根x*的邻近有连续二阶导数，且 $|g'(x^*)|<1$

(1) 当$g'(x^*)\ne0$ 时，迭代过程线性收敛;

(2) 当$g'(x^*)=0$ 而$g''(x^*)\ne0$ 时，迭代过程平方收敛.

分析： 用泰勒公式证

（1）$\frac{e_{k+1}}{e_k}\to g'(x^*)$　　（2）$\frac{e_{k+1}}{e_k^2}\to\frac{g''(x^*)}{2}$

注： 推广的结论Th5.7

HW:
作业五 **#2**

注：构造迭代函数的一般方法

$x=x+\lambda(x)f(x)$， $g(x)=x+\lambda(x)f(x)$

由$|g'(x)|\le L<1$ → 选$\lambda(x)$

Review: $x_{k+1}=g(x_k)$

Th5.3 设迭代函数 g(x) 在[a,b] 上具有连续的一阶导数，且当

(1) $x \in [a,b]$ 时, $a \leqslant g(x) \leqslant b$;

(2) 存在正数L<1，对 $x \in [a,b]$ 有 $|g'(x)| \leqslant L<1$ 成立,

则 x=g(x) 在 [a,b] 上有唯一解 x*，且对任意的初始近似值 $x_0 \in [a,b]$ 迭代过程 $x_{k+1}=g(x_k)(k=0,1,2,.....)$ 收敛且 $\lim\limits_{k\to\infty} x_k = x^*$

Th5.4 在 Th5.3的条件下，有误差估计式：

$$|x^*-x_k| \le \frac{|x_{k+1}-x_k|}{1-L}$$

$$|x^*-x_k| \le \frac{L^k}{1-L}|x_1-x_0|$$

局部收敛性: 若对某邻域△:|x-x*|≤δ, 迭代过程对∀ x_0∈△均收敛。

Th5.5 设g(x)在x=g(x)的根x^*邻近有连续的一阶导数，且$|g'(x^*)|<1$，则迭代过程$x_{k+1}=g(x_k)$具有局部收敛性。

p阶收敛: 若有某个实数 p≥1和非零常数C 使 $\lim\limits_{k\to\infty}\frac{|e_{k+1}|}{|e_k|^p}=C\ (k \ne 0)$

5.3.2 迭代过程的加速

对于收敛的迭代过程，若收敛速度太慢，需改进以加速收敛。

/* accelerating convergence */

迭代公式的加工：对于迭代公式$x_{k+1}=g(x_k)$ k=0,1,2,……

若g′(x) 在求根范围内改变不大，取$g'(x) \approx a$

当$|g'(x)| \approx |a| \le L<1$

（1）x_k为 x*的近似值，迭代一次得 $\tilde{x}_{k+1}=g(x_k)$

$$\therefore x^*-\tilde{x}_{k+1}=g(x^*)-g(x_k)=g'(\xi)(x^*-x_k)\approx a(x^*-x_k)$$

$$\therefore (1-a)x^*-\tilde{x}_{k+1}\approx -ax_k \Rightarrow (1-a)x^*-\tilde{x}_{k+1}+a\tilde{x}_{k+1}\approx a\tilde{x}_{k+1}-ax_k$$

$$\therefore (1-a)(x^*-\tilde{x}_{k+1})\approx a(\tilde{x}_{k+1}-x_k) \Rightarrow x^*-\tilde{x}_{k+1}\approx\frac{a}{1-a}(\tilde{x}_{k+1}-x_k)$$

（2）将以上误差补偿给 $\tilde{x}_{k+1}$ 得更精确的近似根 $x_{k+1}=\tilde{x}_{k+1}+\frac{a}{1-a}(\tilde{x}_{k+1}-x_k)$

从而得迭代加速公式

$$\left.\begin{aligned}&\text{迭代}\quad \tilde{x}_{k+1}=g(x_k)\\&\text{改进}\quad x_{k+1}=\tilde{x}_{k+1}+\tfrac{a}{1-a}(\tilde{x}_{k+1}-x_k)\end{aligned}\right\}$$

例 用加速收敛的方法求$x=e^{-x}$ 在$x=0.5$ 附近的一个根，要求精度为 $\varepsilon=10^{-5}$。

分析：例5.3中，用简单迭代法迭代10次才达到精度10^{-3}，那么用改进的公式迭代多少次满足精度要求10^{-5}？

解：$g(x)=e^{-x}$且$g'(x)=-e^{-x}$，在$x=0.5$ 附近有：$g'(x)\approx-0.6$，

从而加速公式为
$$\begin{cases}\tilde{x}_{k+1}=e^{-x_k}\\ x_{k+1}=\tilde{x}_{k+1}-\dfrac{0.6}{1.6}(\tilde{x}_{k+1}-x_k)\end{cases}$$

取$x_0=0.5$，迭代结果为

k	0	1	2	3	4
x_k	0.5	0.56658	0.56713	0.56714	0.56714
$\tilde{x}_k$		0.60653	0.56746	0.56715	0.56715

Aitken 加速法

$$\left.\begin{array}{ll}\text{迭代} & \tilde{x}_{k+1}=g(x_k)\\ \text{改进} & x_{k+1}=\tilde{x}_{k+1}+\frac{a}{1-a}(\tilde{x}_{k+1}-x_k)\end{array}\right\}$$

缺点：要用导数$g'(x)$的近似值a，下面对它进一步改进。

如 $\tilde{x}_{k+1}=g(x_k)$　　$x^*-\tilde{x}_{k+1}\approx a(x^*-x_k)$

$\bar{x}_{k+1}=g(\tilde{x}_{k+1})$　　$x^*-\bar{x}_{k+1}\approx a(x^*-\tilde{x}_{k+1})$

$$\therefore \frac{x^*-\tilde{x}_{k+1}}{x^*-\bar{x}_{k+1}}\approx\frac{x^*-x_k}{x^*-\tilde{x}_{k+1}}\Rightarrow x^*\approx\bar{x}_{k+1}-\frac{(\bar{x}_{k+1}-\tilde{x}_{k+1})^2}{\bar{x}_{k+1}-2\tilde{x}_{k+1}+x_k}$$

注：新的改进值（不含**a** 的信息，对两次迭代值加工）

$$\begin{cases}\tilde{x}_{k+1}=g(x_k)\\ \bar{x}_{k+1}=g(\tilde{x}_{k+1})\\ x_{k+1}=\bar{x}_{k+1}-\dfrac{(\bar{x}_{k+1}-\tilde{x}_{k+1})^2}{\bar{x}_{k+1}-2\tilde{x}_{k+1}+x_k}\end{cases}$$ ——Aitken 加速法

例5.8 用Aitken 法解方程$x^3-x-1=0$.

说明：迭代格式 $x_{k+1}=x_k^3-1$ $(x_0=1.5)$ 是发散的

解：对 $x_{k+1}=x_k^3-1$ 利用Aitken 加速迭代公式

$$\tilde{x}_{k+1} = x_k^3 - 1$$

$$\bar{x}_{k+1} = \tilde{x}_{k+1}^3 - 1$$

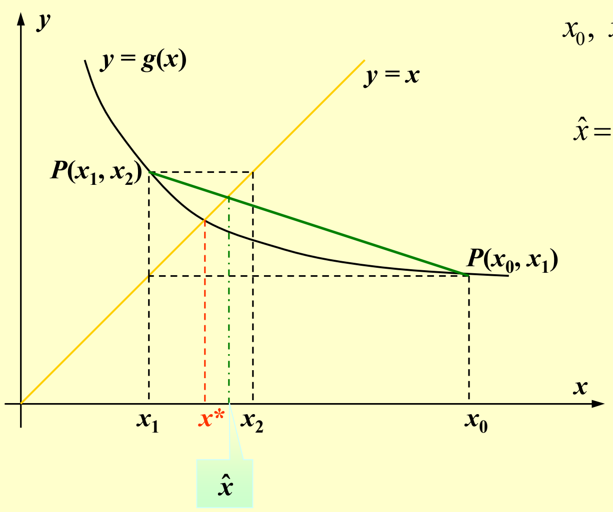
$$x^* \approx \bar{x}_{k+1} - \frac{(\bar{x}_{k+1} - \tilde{x}_{k+1})^2}{\bar{x}_{k+1} - 2\tilde{x}_{k+1} + x_k} \quad k = 0,1,2,\ldots$$

取$x_0=1.5$ 进行迭代，结果见P135表

k	$\tilde{x}_k$	$\bar{x}_k$	x_k
0			1.5
1	2.37500	123965	1.41629
2	1.84092	5.23888	1.35565
3	1.49140	2.31728	1.32895
4	1.34710	1.44435	1.32480
5	132518	1.32714	1.32472

发散的迭代公式被加速后有较好的收敛性。

Aitken 加速法的几何解释



$$x_0,\ x_1 = g(x_0),\ x_2 = g(x_1),$$

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

5.4 牛顿法

假设f(x)=0 有根x^*，下面从几何上来讨论

①则x^*为曲线y=f(x)与x轴的交点

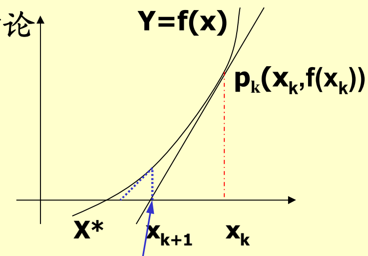
②若f(x)=0 有近似根x_k，在 y=f(x) 上找到点$p_k(x_k, f(x_k))$

③过P_k 作y=f(x) 的切线

$y=f(x_k)+f'(x_k)(x-x_k)$

其与x 轴交于点x_{k+1}，则 $0=f(x_k)+f'(x_k)(x_{k+1}-x_k)$

x_{k+1}比x_k更接近于x^*，反复作切线得点x_{k+2}，x_{k+3}，……→x^*



④由③得 $x_{k+1}=x_k-\dfrac{f(x_k)}{f'(x_k)}$ k=0,1,2,... —— Newton迭代法(切线法)

注：将非线性方程线性化 —— Taylor 展开

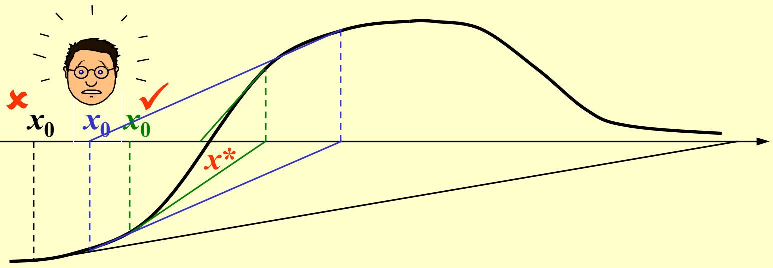
$f(x)=f(x_k)+f'(x_k)(x-x_k)+\dfrac{f''(\xi)}{2!}(x-x_k)^2$，$\xi$在 x_k 和 x 之间。

将 $(x^*-x_k)^2$ 看成高阶小量，则有：

$$0=f(x^*)\approx f(x_k)+f'(x_k)(x^*-x_k) \Rightarrow x^* \approx x_k-\frac{f(x_k)}{f'(x_k)} \Rightarrow x_{k+1}=x_k-\frac{f(x_k)}{f'(x_k)}$$

$$\begin{cases} x_{k+1}=x_k-\dfrac{f(x_k)}{f'(x_k)} & k=0,1,2,\ldots \\ x_0 \end{cases}$$

注：Newton's Method 收敛性依赖于x_0的选取。



下面给出Newton法的算法和步骤

（1）准备 取初始值x_0 及精度ε和最大迭代次数N， 置k=0

（2）迭代 if (f′(x_0)=0) stop (失败)

else

$$x_1=x_0-\frac{f(x_0)}{f'(x_0)}$$

（3）控制 if (|f(x_1)|<ε or |x_1- x_0 |<ε) stop ($x^*\approx x_1$)

else if （k=N） stop (发散)

else k=k+1;

x_0 =x_1 ;

Goto (2)

例5.8 用 Newton法求方程$f(x)=x^4-2x-4$的根，精确到0.01。

解 $\because f'(x)=4x^3-2$， 则Newton 迭代公式为 $x_{k+1}=x_k-\dfrac{x_k^4-2x_k-4}{4x_k^3-2}$

$f(1)=-5$，$f(2)=8$，选取$x_0=1.5$， $f'(1.5)=11.5$

(1) $f(1.5)=-1.9375, f'(1.5)=11.5$

$$x_1=1.5-\frac{(-1.9375)}{11.5}=1.668478$$

(2) $f(x_1)=0.412696, f'(x_1)=16.57896$

$$x_2=1.668478-\frac{0.412696}{16.57896}=1.643585$$

(3) $f(x_2)=0.010243, f'(x_2)=15.75974$

$$x_3=1.643585-\frac{0.010243}{15.75974}=1.642935$$

$|x_3-x_2|=0.00065<\varepsilon$

$x^*\approx 1.642935.$

(4) $f(x_3)=1.62\times10^{-6}$, $f'(x_3)=15.7387$

$$x_4=1.642935-\frac{1.62\times10^{-6}}{15.7387}=1.64293519$$

HW:
作业五 #3

Newton法的收敛性

$$x_{k+1}=x_k-\frac{f(x_k)}{f'(x_k)}=g(x_k) \quad \Rightarrow g(x)=x-\frac{f(x)}{f'(x)}$$

条件：若 f(x)有单根x*，且 f(x)在x*邻近具有连续二阶导数，
隐含： $f(x^*)=0$， $f'(x^*)\neq0$

应用 Th5.5讨论其收敛性

$\because \quad g'(x^*)=\left.\dfrac{f(x)\cdot f''(x)}{[f'(x)]^2}\right|_{x=x^*}=0$ 由Th5.4， Newton 法是局部收敛的

$$g''(x)=\frac{[f'(x)f''(x)+f(x)f'''(x)]\cdot[f'(x)]^2-f(x)f''(x)2f'(x)f''(x)]}{[f'(x)]^4}$$

$$\therefore \quad g''(x^*)=\frac{f''(x^*)}{f'(x^*)}$$

$\therefore$ 当 $f''(x^*)\neq0$时， $g''(x^*)\neq0$,

由Th5.6，Newton 法是平方收敛的（二阶收敛速度）。

例5.10 用Newton法求方程$xe^x-1=0$的根，取五位小数计算.

解: $f(x)=xe^x-1$, $f'(x)=e^x+xe^x$, Newton迭代公式 $x_{k+1}=x_k-\frac{x_ke^{x_k}-1}{e^{x_k}+x_ke^{x_k}}$

即 $x_{k+1}=x_k-\frac{x_k-e^{-x_k}}{1+x_k}$，取$x_0=0.5$ 迭代结果为

k	0	1	2	3
x_k	0.5	0.57102	0.56716	0.56714

Newton 法收敛速度快

例5.11 Newton法应用: $\sqrt{c}=?$ 解$x^2-c=0$，$f(x)=x^2-c$，$f'(x)=2x$

$$x_{k+1}=x_k-\frac{x_k^2-C}{2x_k} \quad 或 \quad x_{k+1}=\frac{1}{2}(x_k+\frac{C}{x_k}) \qquad (*)$$

(*)式 含义: $\sqrt{c}$ 的两个近似值x_k, $\frac{c}{x_k}$ 的算术平均是更好的近似值

注: $x_{k+1}=\frac{1}{2}(x_k+\frac{c}{x_k})$对任意$x_0>0$ 都为平方收敛

HUST

牛顿法初值的选取(了解)

Chapter 5 Solutions of equations in one varible

- 牛顿法是一种局部收敛法，如果初值 x_0 选择不当，可能得不到收敛的迭代序列。
- 为使牛顿法收敛，必须满足：用迭代公式算出的x_1比 x_0 更靠近准确根 x^*。
- 如果 $f'(x_0)=0$，则不能运用牛顿迭代公式，如果 $f'(x_0)$ 非常小，也不能得到很快的收敛序列。

$$x_1=x_0-\frac{f(x_0)}{f'(x_0)} \longrightarrow \underbrace{x_1-x^*}_{\varepsilon_1}=(\underbrace{x_0-x^*}_{\varepsilon_0})-\frac{f(x_0)}{f'(x_0)}$$

HUST

牛顿法初值的选取（续）

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)} \longrightarrow \underbrace{x_1 - x^*}_{\varepsilon_1} = (\underbrace{x_0 - x^*}_{\varepsilon_0}) - \frac{f(x_0)}{f'(x_0)}$$

$$\frac{\varepsilon_1}{\varepsilon_0} = 1 - \frac{f(x_0)}{f'(x_0)(x_0 - x^*)} = 1 + \frac{f(x_0)}{f'(x_0)(x^* - x_0)}$$

$$= \frac{f(x_0) + f'(x_0)(x^* - x_0)}{f'(x_0)(x^* - x_0)}$$

$$f(x^*) = f(x_0) + f'(x_0)(x^* - x_0) + \frac{f''(\xi)}{2!}(x^* - x_0)^2 = 0$$

$$f(x_0) + f'(x_0)(x^* - x_0) = \boxed{-\frac{f''(\xi)}{2!}(x^* - x_0)^2}$$

$$\frac{\varepsilon_1}{\varepsilon_0} = -\frac{f''(\xi)(x^* - x_0)^2}{2f'(x_0)(x^* - x_0)} = -\frac{f''(\xi)(x^* - x_0)}{2f'(x_0)}$$

$$\frac{\varepsilon_1}{\varepsilon_0} = -\frac{f''(\xi)(x^* - x_0)}{2f'(x_0)}$$

$$f(x^*) = f(x_0) + f'(\eta)(x^* - x_0) = 0 \longrightarrow x^* - x_0 = -\frac{f(x_0)}{f'(\eta)}$$

如果 $f'(x)$ 和 $f''(x)$ 在 x_0 附近变化不大，并且 $f''(x_0) \neq 0$，则可近似的认为：

$$f''(\xi) \approx f''(x_0) \qquad f'(\eta) \approx f'(x_0)$$

$$\frac{\varepsilon_1}{\varepsilon_0} = -\frac{f''(\xi)\left(-\frac{f(x_0)}{f'(\eta)}\right)}{2f'(x_0)} = \frac{f''(\xi)\cdot f(x_0)}{2f'(x_0)\cdot f'(\eta)} \approx \frac{f''(x_0)\cdot f(x_0)}{2[f'(x_0)]^2}$$

$$\frac{\varepsilon_1}{\varepsilon_0} \approx \frac{f''(x_0)\cdot f(x_0)}{2[f'(x_0)]^2}$$

为了满足 x_1 比 x_0 更靠近准确根 x^*，必须有：

$$|\varepsilon_1| < |\varepsilon_0|$$

即：

$$\left|\frac{\varepsilon_1}{\varepsilon_0}\right| < 1 \longrightarrow \left|\frac{f''(x_0)\cdot f(x_0)}{2[f'(x_0)]^2}\right| < 1$$

从而：

$$\boxed{[f'(x_0)]^2 > \frac{1}{2}|f''(x_0)|\cdot|f(x_0)|}$$ 条件 1

$$\boxed{f''(x_0) \neq 0}$$ 条件 2

Newton下山法/* Descent Method */

x_0 的选取很重要,如 x_0偏离x^* 较远，则可能发散。

例 用Newton 法解方程x^3 -x-1=0 的根

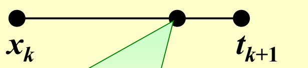
解 迭代公式为 $x_{k+1}=x_k-\dfrac{x_k^3-x_k-1}{3x_k^2-1}$， 取$x_0$=1.5 收敛很快。

k	0	1	2	3
x_k	1.5	1.34783	1.32520	1.32472

取x_0=0.6时， x_1= 17.9， ......， 发散。

为控制迭代发散， 介绍 Newton下山法：

原理：迭代的每一步满足$|f(x_{k+1})|<|f(x_k)|$, 令 $t_{k+1}=x_k-\dfrac{f(x_k)}{f'(x_k)}$



$\lambda t_{k+1}+(1-\lambda)x_k$ ， $\lambda\in[0,1]$

$$x_{k+1} = \lambda\left[x_k - \frac{f(x_k)}{f'(x_k)}\right] + (1-\lambda)x_k$$
$$= x_k - \lambda\frac{f(x_k)}{f'(x_k)}$$

$$\begin{cases} x_{k+1} = x_k - \lambda \dfrac{f(x_k)}{f'(x_k)} & 0 < \lambda \le 1 \\ |f(x_{k+1})| < |f(x_k)| \end{cases}$$

确定：$\lambda = 1 \to \lambda = \frac{1}{2} \to \lambda = \frac{1}{2^2} \to \cdots$

注：$\lambda = 1$ 时就是Newton's Method 公式。

当 $\lambda = 1$ 代入效果不好时，将 λ 减半计算，逐步试探。

例 $x^3 - x - 1 = 0$. $x_{k+1} = x_k - \lambda \dfrac{x_k^3 - x_k - 1}{3x_k^2 - 1}$

取 $x_0 = 0.6, t_1 = 17.9, \lambda = \dfrac{1}{32}$.

$$x_1 = \frac{1}{32}\bar{x}_1 + \frac{31}{32}x_0 = 1.140625$$

Algorithm: Newton's Descent Method

Find a solution to $f(x) = 0$ given an initial approximation x_0.

Input: initial approximation x_0; $f(x)$ and $f'(x)$; minimum step size of x_{min}; tolerance $TOL1$ for x; tolerance $TOL2$ for λ; maximum number of iterations N_{max}.

Output: approximate solution x or message of failure.

Step 1 Set $k = 1$;

Step 2 While ($k \le N_{max}$) do steps 3-10

Step 3 Set $\lambda = 1$;

Step 4 Set $x = x_0 - \lambda \dfrac{f(x_0)}{f'(x_0)}$; /* compute x_k */

Step 5 If $|x - x_0| < TOL1$ then Output (x); STOP; /* successful */

Step 6 If $|f(x)| < |f(x_0)|$ then $x_0 = x$; GOTO *Step 10* ; /* update x_0 */

Step 7 Set $\lambda = \lambda / 2$; /* update λ to descend */

Step 8 If $\lambda > TOL2$ then GOTO *Step 4* ; /* compute a better x_i */

Step 9 Set $x_0 = x_0 + x_{min}$; /* move forward anyway to avoid deadlock */

Step 10 Set k ++;

Step 11 Output (Method failed after N_{max} iterations); STOP. /* unsuccessful */

5.5 近似牛顿法

$x_{k+1}=x_k-\frac{f(x_k)}{f'(x_k)}$ 每一步迭代须计算$f'(x_k)$

简化 Newton法

$f'(x_k)$换为常数C, $x_{k+1}=x_k-\frac{f(x_k)}{C}$ ——推广的简化的Newton法

取$C=f'(x_0)$, $x_{k+1}=x_k-\frac{f(x_k)}{f'(x_0)}$ ——简化的Newton法

上述两种方法简化了迭代计算过程，但收敛速度受到影响。

由迭代函数g(x)=x-f(x)/c 得推广的简化Newton 法收敛

$$\Leftrightarrow |g'(x^*)|=|1-\frac{f'(x^*)}{C}|<1 \Leftrightarrow 0<\frac{f'(x^*)}{C}<2$$

此时，推广的简化 Newton 法局部收敛且一般为线性收敛。

弦截法

对 Newton法作另一种改进:

$$\because f'(x_k)\approx\frac{f(x_k)-f(x_0)}{x_k-x_0}=f[x_0,x_k] \Rightarrow x_{k+1}=x_k-\frac{f(x_k)}{f(x_k)-f(x_0)}(x_k-x_0)\ k=1,2,\ldots$$

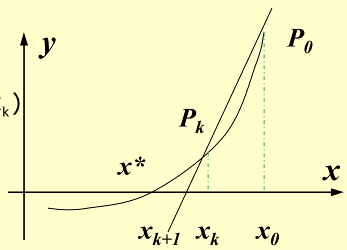
迭代函数: $g(x)=x-\frac{f(x)}{f(x)-f(x_0)}(x-x_0)$

用弦 $\overline{P_0P_k}$ 的斜率代替P_k 点的切线斜率$f'(x_k)$，弦为

$$y-f(x_k)=\frac{f(x_k)-f(x_0)}{x_k-x_0}(x-x_k)$$

其与x轴交点x_{k+1}: $0-f(x_k)=\frac{f(x_k)-f(x_0)}{x_k-x_0}(x_{k+1}-x_k)$

$$\therefore x_{k+1}=x_k-\frac{f(x_k)}{f(x_k)-f(x_0)}(x_k-x_0)$$



上述迭代法称为弦截法

弦截法的收敛性

迭代函数 $g(x)=x-\dfrac{f(x)}{f(x)-f(x_0)}(x-x_0)$

$$\because g'(x)=1-\left\{\left[\frac{f'(x)(f(x)-f(x_0))-f(x)f'(x)}{(f(x)-f(x_0))^2}\right](x-x_0)+\frac{f(x)}{f(x)-f(x_0)}\right\}$$

$$=1-\left\{\left[\frac{-f'(x)f(x_0)}{(f(x)-f(x_0))^2}\right](x-x_0)+\frac{f(x)}{f(x)-f(x_0)}\right\}$$

$$\therefore g'(x^*)=1+\frac{f'(x^*)}{f(x_0)}(x^*-x_0)=1-\frac{f'(x^*)}{\dfrac{f(x^*)-f(x_0)}{x^*-x_0}}$$

当 x_0 靠近 x^* 时，$0<|g'(x^*)|<1$，弦截法线性收敛。

为提高弦截法的收敛速度，介绍另一类型的弦截法。

快速弦截法

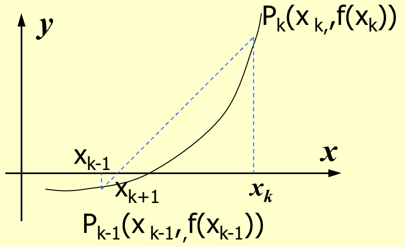
$$x_{k+1}=x_k-\frac{f(x_k)}{f'(x_k)}$$

$$\because f'(x_k)\approx f[x_{k-1},x_k]=\frac{f(x_k)-f(x_{k-1})}{x_k-x_{k-1}}\Rightarrow x_{k+1}=x_k-\frac{f(x_k)}{f(x_k)-f(x_{k-1})}(x_k-x_{k-1})$$

—— 快速弦截法

其几何意义:

x_{k+1} 为弦 $\overline{P_{k-1}P_k}$ 与 **x** 轴的交点



收敛性 如果:**f(x)** 在根 **x^*** 的邻域△:$|x-x^*|\le\delta$ 内具有二阶连续导数，且对 $x\in$△,有 $f'(x)\neq 0$，当 $x_0, x_1\in$△且△充分小时，快速弦截法按阶 $p=\frac{1+\sqrt{5}}{2}\approx 1.618$ 收敛到根 **x^***.

$$x_{k+1}=x_k-\frac{f(x_k)}{f(x_k)-f(x_{k-1})}(x_k-x_{k-1})$$ 计算x_{k+1}时须用x_k, x_{k-1}

Algorithm:

step1: 选取x_0, x_1 , 计算函数值$f_0=f(x_0)$, $f_1=f(x_1)$;

step2: 迭代 $x_2=x_1-\frac{f_1}{f_1-f_0}(x_1-x_0)$

step3: if **(**$|x_2-x_1|<\varepsilon_1$ or $|f(x_2)|<\varepsilon_2$**),** then $x^*\approx x_2$, stop.

else if （迭代次数≤N） then

{ $x_0=x_1, x_1=x_2, f_0=f(x_0), f_1=f(x_1)$;

goto step2

}

else 输出“迭代过程不收敛”, **stop.**

Chapter 5 Solutions of equations in one varible

例 求方程$f(x)=\sin x-(x/2)^2=0$ 的正根, 要求用快速弦截法, $\varepsilon_1=\varepsilon_2=10^{-4}/2$

解

n	$x_0=1$, $x_1=2$		$x_{n+1}-x_n$	$f(x_n)$
2	$x_2=x_1-\frac{f(x_1)}{f(x_1)-f(x_0)}(x_1-x_0)=$	1.86704	0.064316	+0.84981
3	$x_3=$	1.93135	0.002490	+0.003167
4	$x_4=$	1.93384	-0.000091	–0.000120
5	$x_5=$	1.93375	-0.000001	+0.000001
6	$x_6=$	1.93375		

得 $x^*\approx x_6=1.93375$

HUST

小结

本章的问题是：求解非线性方程 $f(x)=0$

二分法: $x_k=\frac{1}{2}(a_k+b_k)$ $|x^*-x_k|\le\frac{1}{2^{k+1}}(b-a)$

特点--简便、易掌握、对f(x)的要求不高,但收敛较慢。

简单迭代法: $f(x)=0 \Leftrightarrow x=g(x)$

收敛要求: $|g'(x)|\le L<1$

迭代格式: $x_{k+1}=g(x_k)$

停机准则: $|x_{k+1}-x_k|<\varepsilon$

大范围收敛与局部收敛性的理论　　收敛的阶

加速迭代: Aitken加速

牛顿迭代法: $x_{k+1}=x_k-\dfrac{f(x_k)}{f'(x_k)}$　二阶收敛　　Newton法的改进