第6章 方程求根

张利军
zlj@nju.edu.cn
http://cs.nju.edu.cn/zlj

目录

- 根的搜索
- 迭代法
- Newton法
- 弦截法与抛物线法
- 代数方程求根

引言

- 许多数学物理问题可归结为解方程$f(x)=0$
  - $f(x)$可以是代数多项式，或超越函数
  - 超越函数，指变量之间的关系不能用有限次加、减、乘、除、乘方、开方运算表示的函数
- 方程$f(x)=0$的解x^*称为它的**根**，或称为$f(x)$的**零点**
- 设函数$f(x)$在$[a,b]$上连续，且$f(a)f(b)<0$，根据连续函数的性质，可知$f(x)=0$在区间(a,b)内一定有实根
  - $[a,b]$为$f(x)=0$的**有根区间**

逐步搜索法

- 假定$f(a) < 0, f(b) > 0$
- 从有根区间$[a, b]$的左端点$x_0 = a$出发，按预定步长h（例如取$h = \frac{b-a}{N}, N$为正整数）一步步向右跨
  - 每跨一步，进行一次根的搜索，即检查节点$x_k = a + kh$上的函数值$f(x_k)$的符号
  - 一旦发现节点x_k与端点a的函数值异号，即$f(x_k) > 0$，则可确定一个缩小了的有根区间$[x_{k-1}, x_k]$，其宽度为h
  - 若$f(x_k) = 0$，则x_k即为所求的根

例6.1

- 考察方程$f(x) = x^3 - x - 1 = 0$。注意到$f(0) < 0, f(2) > 0$，知$f(x)$在区间$(0,2)$内至少有一个实根
  - 设从$x = 0$出发，取$h = 0.5$为步长向右进行根的搜，列表格记录各个节点上函数值的符号，发现在区间$(1.0, 1.5)$内必有一根

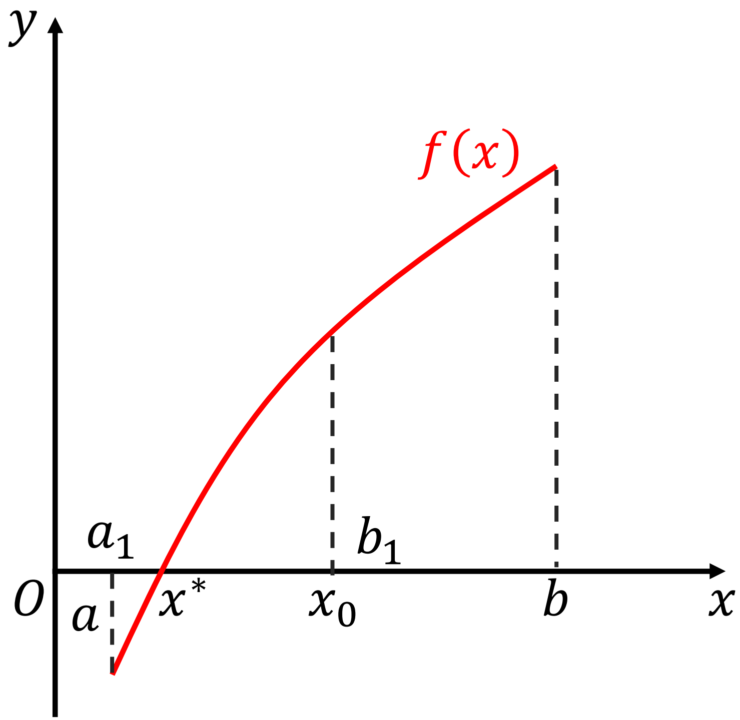
x	0	0.5	1.0	1.5
$f(x)$的符号	$-$	$-$	$-$	$+$

讨论

- 步长h的选择是个关键
  - 只要步长h取得足够小，利用这种方法可以得到具有任意精度的近似根
  - 不过当h缩小时，所要搜索的步数相应增多，从而使计算量增大
  - 因此，如果精度要求比较高，单用这种逐步搜索方法是不合算的
- 下面的二分法可以看作逐步搜索方法的一种改进

二分法

- 考察有根区间$[a,b]$，取中点$x_0=\frac{(a+b)}{2}$将它分为两半，然后检查$f(x_0)$与$f(a)$是否同号
  - 若同号，则说明所求根x^*在x_0右侧，这时令$a_1=x_0, b_1=b$；
  - 否则，x^*在x_0的左侧，这时令$a_1=a, b_1=x_0$
  - 新有根区间$[a_1,b_1]$的长度仅为$[a,b]$的一半



二分法（续）

- 对压缩了的有根区间$[a_1,b_1]$施行同样的手续
  - 即用中点$x_1=(a_1+b_1)/2$将区间$[a_1,b_1]$再分半
  - 然后通过根的搜索判定所求的根在x_1的哪一侧
  - 从而确定一个新的有根区间$[a_2,b_2]$，其长度是$[a_1,b_1]$的一半
- 如此反复二分，即可得出一系列有根区间

$$[a,b]\supseteq[a_1,b_1]\supseteq[a_2,b_2]\supseteq\cdots\supseteq[a_k,b_k]\supseteq\cdots$$

  - 其中每个区间都是前一个区间的一半，$[a_k,b_k]$的长度$b_k-a_k=(b-a)/2^k$，在$k\to\infty$时趋于0
  - 若无限地做二分操作，这些区间将收缩于一点x^*，也即所求根

二分法（续）

- 每次二分后，设取有根区间$[a_k, b_k]$的中点 $x_k = \frac{a_k+b_k}{2}$作为根的近似值
  - 在二分过程中可以获得一个近似根的序列$x_0, x_1, \cdots, x_k, \cdots$，该序列必以根$x^*$为极限
  - 在实际计算时，不可能完成这个无限过程，其实也没有这种必要，因为数值分析的结果允许带有一定的误差
- 由于

$$|x^* - x_k| \le (b_k - a_k)/2 = (b-a)/2^{k+1} \qquad (6.1.1)$$

  - 只要二分次数足够多，k充分大，便有$|x^* - x_k| < \varepsilon$，这里$\varepsilon$为预定的精度

$$|x^* - x_k| \le \frac{b-a}{2^{k+1}} \qquad (6.1.1)$$

例6.2

- 求方程$f(x) = x^3 - x - 1 = 0$在区间$(1.0, 1.5)$内的一个实根，要求精确到小数点后第二位
  - $a = 1.0, b = 1.5$，而$f(a) < 0, f(b) > 0$
  - 取(a, b)的中点$x_0 = 1.25$，将区间二等分
  - 由于$f(x_0) < 0$，也即$f(x_0)$与$f(a)$同号，故所求根x^*在x_0右侧，这时令$a_1 = x_0 = 1.25, b_1 = b = 1.5$，得到新的有根区间$[a_1, b_1]$
  - 如此反复二分下去，按误差估计式(6.1.1)，只需二分$k = 6$次，即可达到预定精度

$$|x^* - x_6| \le 0.005$$

例6.2（续）

- 二分法的计算结果

k	a_k	b_k	x_k	$f(x_k)$的符号
0	1.00	1.5	1.25	−
1	1.25		1.375	+
2		1.375	1.3125	−
3	1.3125		1.3438	+
4		1.3438	1.3281	+
5		1.3281	1.3203	−
6	1.3203		1.3242	−

二分法的计算步骤

1. 准备 计算$f(x)$在有根区间$[a,b]$端点处的值 $f(a),f(b)$
2. 二分 计算$f(x)$在区间中点$\frac{a+b}{2}$处的值 $f\left(\frac{a+b}{2}\right)$
3. 判断
   - 若$f\left(\frac{a+b}{2}\right)=0$则$\frac{(a+b)}{2}$是根，计算结束
   - 若$f\left(\frac{a+b}{2}\right)$与$f(a)$异号，则根位于区间$\left(a,\frac{a+b}{2}\right)$内，这时以$(a+b)/2$代替$b$

二分法的计算步骤（续）

- 若$f\left(\frac{a+b}{2}\right)$与$f(a)$同号，则根位于区间$\left(\frac{a+b}{2},b\right)$内，这时以$(a+b)/2$代替$a$

4. 反复执行二分和判断，直到区间$[a,b]$的长度缩小到误差允许范围内，此时区间中点$(a+b)/2$即可作为所求根

- 二分法的优点是算法简单，而且收敛性总能得到保证

目录

- 根的搜索
- 迭代法
- Newton法
- 弦截法与抛物线法
- 代数方程求根

迭代法

- 考察下列形式的方程：

$$x = \varphi(x) \tag{6.2.1}$$

  - 这种方程是隐式的，因而不能直接得出它的根

- 如果给出根的某个猜测值x_0，将它代入式(6.2.1)的右端，即可求得$x_1 = \varphi(x_0)$

- 然后，又可取x_1作为猜测值，进一步得到$x_2 = \varphi(x_1)$

迭代法（续）

- 如此反复迭代，如果按公式

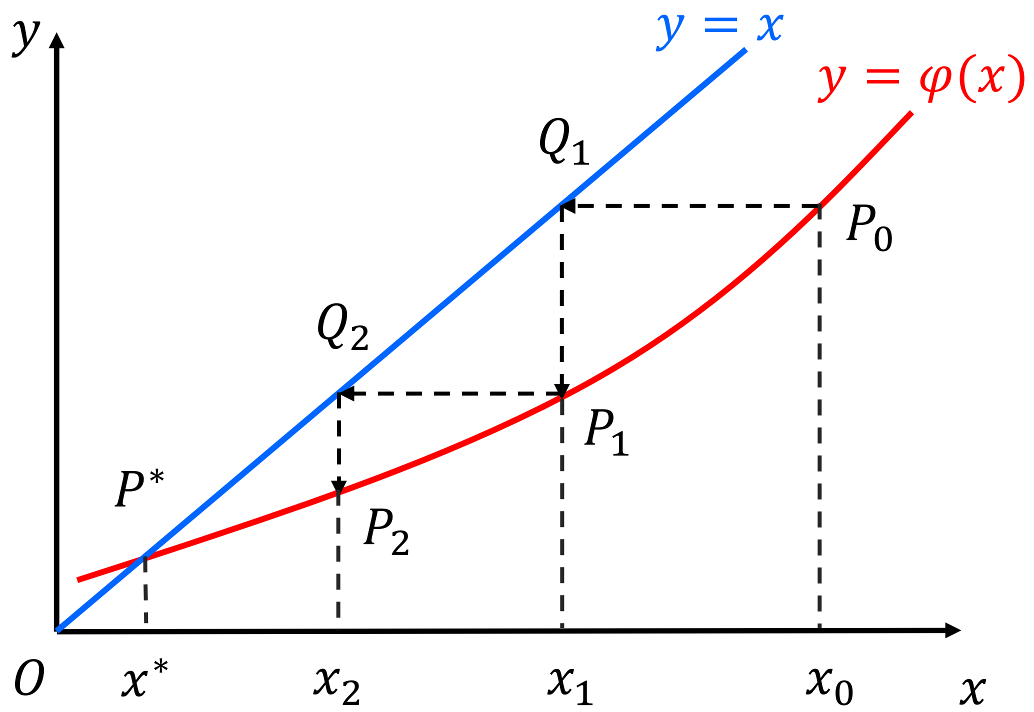
$$x_{k+1} = \varphi(x_k), \qquad k = 0,1,\cdots \qquad (6.2.2)$$

确定的数列$\{x_k\}$有极限$x^* = \lim_{k\to\infty} x_k$，则称迭代过程式(6.2.2)收敛， 这时极限值$x^*$显然就是方程(6.2.1)的根

- 上述迭代法是一种逐次逼近法，基本思想是将隐式方程$x = \varphi(x)$归结为一组显式的计算公式(6.2.2)
  - 迭代过程是逐步显式化的过程

几何解释

- 方程$x = \varphi(x)$的求根问题在O_{xy}平面上就是要确定曲线$y = \varphi(x)$与直线$y = x$的交点P^*
  - 从x_0确定P_0
  - 从P_0确定Q_1
  - 从Q_1确定P_1
  - 从P_1确定Q_2
  - 从Q_2确定P_2



  - 如果点列$\{P_k\}$趋于点P^*，则x_k收敛到所求的根x^*

例6.3

- 求方程

$$f(x) = x^3 - x - 1 = 0 \qquad (6.2.3)$$

在$x_0 = 1.5$附近的根x^*

- 将方程(6.2.3)改写成$x = \sqrt[3]{x+1}$的形式，据此建立迭代公式

$$x_{k+1} = \sqrt[3]{x_k + 1}, \qquad k = 0,1,2,\cdots$$

- 记录各步迭代的结果，如下表所示

例6.3（续）

- 若仅取6位数字，那么x_7与x_8完全相同，这时可以认为x_7已经满足方程(6.2.3)，即为所求的跟

k	x_k	k	x_k
0	1.5	5	1.32476
1	1.35721	6	1.32473
2	1.33086	7	1.32472
3	1.32588	8	1.32472
4	1.32494		

发散的迭代

- 迭代法并不总能取得满意的效果
  - 按照方程(6.2.3)的另一等价形式：$x = x^3 - 1$，建立迭代公式
    $$x_{k+1} = x_k^3 - 1$$
  - 迭代初值仍取$x_0 = 1.5$，则有
    $$x_1 = 2.375, \qquad x_2 = 12.39$$
  - 继续迭代下去已经没有必要，因为结果显然会越来越大，不可能趋于某个极限
- 这种不收敛的迭代过程是发散的
  - 纵使进行了千百次迭代， 其结果也是毫无价值的

收敛条件

- 设方程$x=\varphi(x)$在区间(a,b)内有根x^*，则保证迭代过程$x_{k+1}=\varphi(x_k)$收敛的条件为：存在定数$0<L<1$，使得对任意$x\in[a,b]$，有

$$|\varphi'(x)|\leq L \tag{6.2.4}$$

  - 由微分中值定理

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

    其中ξ是x^*与x_k之间某一点

  - 当$x_k\in[a,b]$时，$\xi\in[a,b]$

收敛条件（续）

- 因此利用条件(6.2.4)可断定

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

- 据此反复递推，有

$$|x_k-x^*| \le L^k|x_0-x^*|$$

  - ✓ 具体的值并不知道，因为x^*是未知的
- 故当$k\to\infty$时，迭代值x_k将收敛到所求根x^*

- 在上述论证过程中，应当保证一切迭代值x_k落在区间(a,b)内
  - 为此要求，对于任意$x\in[a,b]$，总有$\varphi(x)\in[a,b]$

收敛条件（续）

- **定理6.1** 假定函数$\varphi(x)$满足下列两项条件

1. 对于任意$x \in [a, b]$，有

$$a \leq \varphi(x) \leq b \tag{6.2.5}$$

2. 存在正数$L < 1$，使对于任意$x \in [a, b]$，有

$$|\varphi'(x)| \leq L < 1 \tag{6.2.6}$$

则迭代过程$x_{k+1} = \varphi(x_k)$对于任意初值$x_0 \in [a, b]$均收敛于方程$x = \varphi(x)$的根x^*。

并且有如下误差估计式

$$|x_k - x^*| \leq \frac{L^k}{1 - L}|x_1 - x_0| \tag{6.2.7}$$

定理6.1（证明）

$$|\varphi'(x)| \le L < 1 \quad (6.2.6)$$

- 根据(6.2.6)，有

$$|x_{k+1} - x_k| = |\varphi(x_k) - \varphi(x_{k-1})| \le L|x_k - x_{k-1}| \qquad (6.2.8)$$

- 据此反复递推，得

$$|x_{k+1} - x_k| \le L^k |x_1 - x_0|$$

- 于是对任意正整数p，有

$$\begin{aligned}|x_{k+p} - x_k| &\le |x_{k+p} - x_{k+p-1}| + |x_{k+p-1} - x_{k+p-2}| + \cdots + |x_{k+1} - x_k| \\ &\le \left(L^{k+p-1} + L^{k+p-2} + \cdots + L^k\right)|x_1 - x_0| \le \frac{L^k}{1-L}|x_1 - x_0|\end{aligned}$$

- 在上式中令$p \to \infty$，注意到$\lim\limits_{p\to\infty} x_{k+p} = x^*$，即得式(6.2.7)

误差估计

- 在用迭代法进行实际计算时，必须按精度要求控制迭代次数

$$|x_k - x^*| \le \frac{L^k}{1-L}|x_1 - x_0| \qquad (6.2.7)$$

  - 误差估计式(6. 2. 7) 原则上可用来确定迭代次数，但它由于含有信息L而不便于实际应用

- 根据式(6.2.8)，

$$|x_{k+1} - x_k| = |\varphi(x_k) - \varphi(x_{k-1})| \le L|x_k - x_{k-1}| \qquad (6.2.8)$$

对于任意正整数p，有

$$|x_{k+p} - x_k| \le \left(L^{p-1} + L^{p-2} + \cdots + 1\right)|x_{k+1} - x_k| \le \frac{1}{1-L}|x_{k+1} - x_k|$$

误差估计（续）

- 在上式中令$p \to \infty$，得

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

- 由此可见，只要相邻两次计算结果的偏差$|x_{k+1} - x_k|$足够小，即可保证近似值x_k有足够的精度
  - 因此，可以通过检查$|x_{k+1} - x_k|$来判断迭代过程是否应终止

计算步骤

1. 准备　提供迭代初始值x_0

2. 迭代　计算迭代值$x_1 = \varphi(x_0)$

3. 控制　检查$|x_1 - x_0|$

- 若$|x_1 - x_0| > \varepsilon$（$\varepsilon$为预先指定的精度），则以$x_1$替换$x_0$，继续步骤2的迭代
- 当$|x_1 - x_0| \le \varepsilon$时，终止计算，取$x_1$为所求结果

局部收敛性

- 在实际应用迭代法时，通常在所求的根x^*的邻近进行考察，而研究所谓局部收敛性

- **定义6.1** 若存在x^*的某个邻域R: $|x-x^*|\le\delta$，使迭代过程$x_{k+1}=\varphi(x_k)$对于任意初值$x_0\in R$均收敛，则称迭代过程$x_{k+1}=\varphi(x_k)$在根x^*邻近具有局部收敛性

- **定理6.2** 设x^*为方程$x=\varphi(x)$的根，$\varphi'(x)$在x^*的邻近连续且$|\varphi'(x^*)|<1$，则迭代过程$x_{k+1}=\varphi(x_k)$在x^*邻近具有局部收敛性

定理6.2（证明）

1. 由连续函数的性质，存在x^*的某个邻域$R: |x - x^*| \le \delta$，使对于任意$x \in R$成立

$$|\varphi'(x)| \le L < 1$$

2. 对任意$x \in R$，总有$\varphi(x) \in R$，这是因为

$$|\varphi(x) - x^*| = |\varphi(x) - \varphi(x^*)| \le L|x - x^*| \le |x - x^*|$$

- 依据定理6.1，可断定迭代过程$x_{k+1} = \varphi(x_k)$对于任意初值$x_0 \in R$均收敛

例6.4

- 求方程$x = e^{-x}$在$x = 0.5$附近的一个根，要求精度$\varepsilon = 10^{-5}$
  - 过$x = 0.5$以$h = 0.1$为步长搜索一次，可发现所求根在区间$(0.5, 0.6)$以内
  - 由于在根的附近，$|(e^{-x})'| \approx 0.6$，该值小于1，因此迭代公式$x_{k+1} = e^{-x_k}$对于初值$x_0 = 0.5$是收敛的
  - 按照$x_{k+1} = e^{-x_k}$迭代，记录结果

例6.4（续）

- 比较相邻的两次迭代值，迭代18次得所求根0.56714

k	x_k	k	x_k	k	x_k
0	0.5	7	0.5684380	14	0.5671188
1	0.6065306	8	0.5664094	15	0.5671571
2	0.5452392	9	0.5675596	16	0.5671354
3	0.5797031	10	0.5669072	17	0.5671477
4	0.5600646	11	0.5672772	18	0.5671407
5	0.5711721	12	0.5670763		
6	0.5648629	13	0.5671863		

迭代过程的加速

- 对于收敛的迭代过程，只要迭代足够多次，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

- 因此迭代过程的加速是个重要的课题

迭代公式的加工

- 设x_0是根x^*的某个预测值，用迭代公式校正一次，得$x_1=\varphi(x_0)$
- 由微分中值定理得

$$x_1-x^*=\varphi'(\xi)(x_0-x^*)$$

其中ξ介于x^*与x_0之间

- 假定$\varphi'(x)$改变不大，近似地取某近似值L，则由

$$x_1-x^*\approx L(x_0-x^*)$$

$$\Rightarrow x^*\approx\frac{1}{1-L}x_1-\frac{L}{1-L}x_0 \qquad (6.2.9)$$

迭代公式的加工（续）

- 可以期望，按上式右侧求得的

$$x_2 = \frac{1}{1-L}x_1 - \frac{L}{1-L}x_0 = x_1 + \frac{L}{1-L}(x_1 - x_0)$$

是比x_1更好的近似值

- 将每得到一次改进值算作一步，并用$\bar{x}_k$和x_k分别表示第k步的校正值和改进值，则加速迭代计算方案可表述为：

1. 校正：

$$\bar{x}_{k+1} = \varphi(x_k)$$

2. 改进：

$$x_{k+1} = \bar{x}_{k+1} + \frac{L}{1-L}(\bar{x}_{k+1} - x_k) \qquad (6.2.10)$$

例6.5

- 求解方程$x = e^{-x}$
  - 由于在$x_0 = 0.5$附近，有$(e^{-x})' \approx -0.6$，故上述计算公式的具体形式是

$$\begin{cases} \bar{x}_{k+1} = \mathrm{e}^{-x_k} \\ x_{k+1} = \bar{x}_{k+1} - \dfrac{0.6}{1.6}(\bar{x}_{k+1} - x_k) \end{cases}$$

  - 下表展示了计算结果

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

例6.5（续）

- 例6.4迭代18次得到精度10^{-5}的结果$x=0.56714$

$$x_{k+1}=e^{-x_k}$$

- 例6.5中只需迭代3次即可得出相同结果，可见加速效果显著

$$\begin{cases} \bar{x}_{k+1}=\mathrm{e}^{-x_k} \\ x_{k+1}=\bar{x}_{k+1}-\dfrac{0.6}{1.6}(\bar{x}_{k+1}-x_k) \end{cases}$$

Aitken（埃特金）方法

- 上述加速方案有个缺点，由于其中含有导数$\varphi'(x)$的有关信息L，实际使用不便

- 仍设已知x^*的某个猜测值为x_0，将校正值$x_1=\varphi(x_0)$再校正一次，又得$x_2=\varphi(x_1)$
  - 显然
$$x_1 - x^* \approx L(x_0 - x^*)$$
$$x_2 - x^* \approx L(x_1 - x^*)$$
  - 可得
$$\frac{x_1 - x^*}{x_2 - x^*} \approx \frac{x_0 - x^*}{x_1 - x^*}$$

Aitken方法（续）

- 由此推知

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

- 这样构造出的改进公式确实不再含有关于导数的信息，但是它需要用两次迭代值进行加工

□ 如果将得到改进值作为一步，则计算公式如下

1. 校正： $\tilde{x}_{k+1} = \varphi(x_k)$

2. 再校正： $\bar{x}_{k+1} = \varphi(\tilde{x}_{k+1})$

3. 改进： $$x_{k+1} = \bar{x}_{k+1} - \frac{(\bar{x}_{k+1} - \tilde{x}_{k+1})^2}{\bar{x}_{k+1} - 2\tilde{x}_{k+1} + x_k}$$

例6.6

- 用Aitken方法求解方程(6.2.3)：$f(x) = x^3 - x - 1 = 0$
  - 前面曾经指出，求解该方程的下述迭代公式是发散的：

$$x_{k+1} = x_k^3 - 1 \tag{6.2.12}$$

  - 现以该公式为基础，形成Aitken算法，即

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

$$x_{k+1} = \bar{x}_{k+1} - \frac{(\bar{x}_{k+1} - \tilde{x}_{k+1})^2}{\bar{x}_{k+1} - 2\tilde{x}_{k+1} + x_k}$$

例6.6（续）

- 仍取$x_0 = 1.5$，计算结果如下表所示

k	$\tilde{x}_k$	$\bar{x}_k$	x_k
0			1.5
1	2.37500	12.3965	1.41629
2	1.84092	5.23888	1.35565
3	1.49140	2.31728	1.32895
4	1.34710	1.44435	1.32480
5	1.32518	1.32714	1.32472

- 可以看到，将发散的迭代公式(6.2.12)通过Aitken方法处理后，竟获得了相当好的收敛性

目录

- 根的搜索
- 迭代法
- Newton法
- 弦截法与抛物线法
- 代数方程求根

Newton公式

- 对于方程$f(x)=0$，为要应用迭代法，必须先将它改写成$x=\varphi(x)$的形式，即需要针对所给的函数$f(x)$构造合适的迭代函数$\varphi(x)$
  - $\varphi(x)$可以有很多种
  - 例如，令$\varphi(x)=x+f(x)$，这时相应的迭代公式是

$$x_{k+1}=x_k+f(x_k) \qquad (6.3.1)$$

  - 一般来说，这种迭代公式不一定收敛，或者收敛的速度缓慢

Newton公式（续）

- 运用加速技巧，$\bar{x}_{k+1} = \varphi(x_k)$

$$x_{k+1} = \bar{x}_{k+1} + \frac{L}{1-L}(\bar{x}_{k+1} - x_k) \qquad (6.2.10)$$

- 对于迭代过程(6.3.1)，其加速公式为

$$\begin{cases} \bar{x}_{k+1} = x_k + f(x_k) \\ x_{k+1} = \bar{x}_{k+1} + \dfrac{L}{1-L}(\bar{x}_{k+1} - x_k) \end{cases}$$

  - 记$M = L - 1$，上面两个式子可以合并写成

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

Newton公式（续）

- 这种迭代公式通常称为简化的Newton公式，其相应的迭代函数是

$$\varphi(x) = x - \frac{f(x)}{M} \qquad (6.3.2)$$

  - 需要计算$M = L - 1$

- 注意到L是$\varphi'(x)$的估计值
  - $\varphi(x) = x + f(x)$，因此$\varphi'(x) = 1 + f'(x)$
  - 意味着，$M = L - 1$实际上是$f'(x)$的估计值

Newton公式（续）

- 如果用$f'(x)$代替式(6.3.2)中的M，则得如下迭代函数：

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

- 其相应的迭代公式为Newton公式

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)} \qquad (6.3.3)$$

牛顿法与线性化

- 对于方程$f(x) = 0$，如果$f(x)$是线性函数，则对它求根是容易的

- Newton法实际上是一种线性化方法，其基本思想是将非线性方程$f(x) = 0$逐步归结为某种线性方程来求解
  - 设已知方程$f(x) = 0$有近似根x_k，将函数$f(x)$在点x_k展开，有

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

牛顿法与线性化（续）

- 于是方程$f(x)=0$可以近似表示为

$$f(x_k)+f'(x_k)(x-x_k)=0 \qquad (6.3.4)$$

- (6.3.4)是个线性方程，记其根为x_{k+1}，则x_{k+1}的计算公式就是Newton公式(6.3.3)

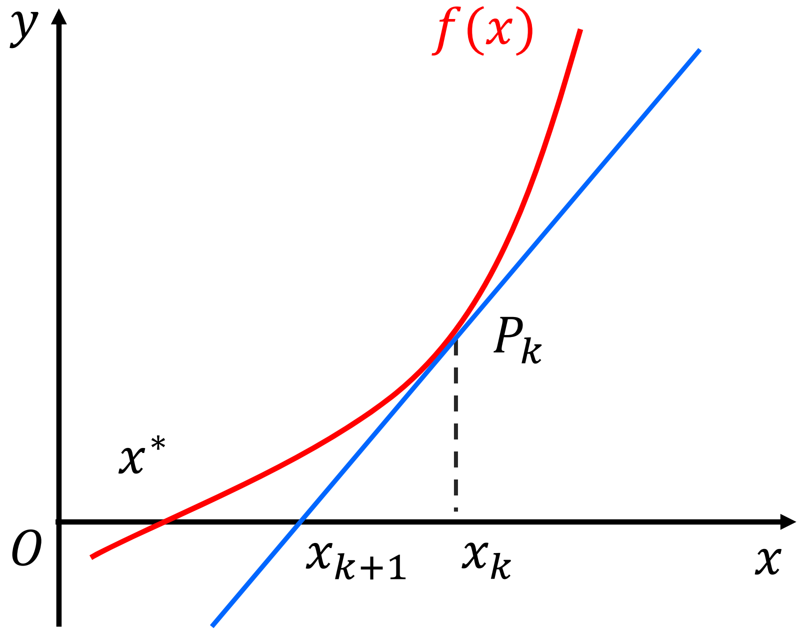
$$x_{k+1}=x_k-\frac{f(x_k)}{f'(x_k)} \qquad (6.3.3)$$

Newton法的几何解释

- 方程$f(x)=0$的根x^*可解释为曲线$y=f(x)$与x轴的交点的横坐标
  - 过曲线$y=f(x)$上横坐标为x_k的点P_k引切线，注意到切线方程为

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

  - 将该切线与x轴的交点的横坐标x_{k+1}作为x^*的新的近似值
  - x_{k+1}满足式(6.3.4)
  - Newton法又称**切线法**



收敛速度

- 对于一种迭代过程，为了保证它是有效的，需要肯定它的收敛性，同时考察它的收敛速度，指在接近收敛的过程中迭代误差的下降速度
- 定义**6.2** 设迭代过程$x_{k+1} = \varphi(x_k)$收敛于方程$x = \varphi(x)$的根x^*，若迭代误差$e_k = x_k - x^*$在$k \to \infty$时成立下列渐进关系式

$$\frac{e_{k+1}}{e_k^p} \to C \quad （C \neq 0为常数）$$

  则称该迭代过程是p阶收敛的
  - $p = 1$时称为线性收敛，$p > 1$时称为超线性收敛，$p = 2$时称为平方收敛

收敛性定理

- **定理6.3** 对于迭代过程$x_{k+1}=\varphi(x_k)$，若$\varphi^{(p)}(x)$在所求根x^*的邻近连续，并且

$$\begin{cases}\varphi'(x^*)=\varphi''(x^*)=\cdots=\varphi^{(p-1)}(x^*)=0\\ \varphi^{(p)}(x^*)\neq 0\end{cases}\qquad(6.3.5)$$

  则该迭代过程在点x^*邻近是p阶收敛的

  - 由于$\varphi'(x^*)=0$，据定理6.2立即可以断定迭代过程$x_{k+1}=\varphi(x_k)$具有局部收敛性
  - 再将$\varphi(x_k)$在根x^*处展开，利用条件(6.3.5)，则有

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

定理6.3（证明）

- 注意到

$$\varphi(x_k) = x_{k+1}, \qquad \varphi(x^*) = x^*$$

- 由上式得

$$x_{k+1} - x^* = \frac{\varphi^{(p)}(\zeta)}{p!}(x_k - x^*)^p$$

- 因此对于迭代误差， 有

$$\frac{e_{k+1}}{e_k^p} \to \frac{\varphi^{(p)}(x^*)}{p!}$$

这表明迭代过程$x_{k+1} = \varphi(x_k)$确实是p阶收敛的

Newton法的局部收敛性

- 由上述定理知，迭代过程的收敛速度依赖于迭代函数$\varphi(x)$的选取
  - 如果当$x \in [a,b]$时，$\varphi'(x) \neq 0$，则该迭代过程只可能是线性收敛的
- 对Newton公式(6.3.3)，其迭代函数为

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

  - 假定x^*是$f(x)$的一个单根，即$f(x^*) = 0, f'(x^*) \neq 0$，则由上式知$\varphi'(x^*) = 0$，于是由定理6.3可知，Newton法在根$x^*$的邻近是平方收敛的

例6.7

□ 用Newton法解方程

$$xe^x - 1 = 0 \qquad (6.3.6)$$

- 此方程对应的Newton公式为 $x_{k+1} = x_k - \frac{x_k - e^{-x_k}}{1+x_k}$ ，取迭代初值$x_0 = 0.5$，迭代结果见下表

k	0	1	2	3
x_k	0.5	0.57102	0.56716	0.56714

- 所给方程(6.3.6)实际上是方程$x = e^{-x}$的等价形式，比较例6.7与例6.5的计算结果可以看出Newton法的收敛速度是很快的

Newton法的计算步骤

1. 准备　选定初始近似值x_0，计算

$$f_0 = f(x_0), \quad f_0' = f'(x_0)$$

2. 迭代　按公式$x_1 = x_0 - \frac{f_0}{f_0'}$迭代一次，得到新的近似值$x_1$，计算

$$f_1 = f(x_1), \qquad f_1' = f'(x_1)$$

3. 控制　若x_1满足$|\delta| < \varepsilon_1$或$|f_1| < \varepsilon_2$，则终止迭代，x_1即为所求根；否则转第4步

Newton法的计算步骤（续）

此处$\varepsilon_1,\varepsilon_2$是允许误差，且有

$$\delta=\begin{cases}|x_1-x_0|, & |x_1|<C\\ \dfrac{|x_1-x_0|}{|x_1|}, & |x_1|\ge C\end{cases}$$

其中C是取绝对误差或相对误差的控制常数，一般取$C=1$

4. 修改　若迭代次数达到预定值N或$f_1'=0$，则方法失败；否则以(x_1,f_1,f_1')代替 (x_0,f_0,f_0')转第2步继续迭代

Newton法的应用举例

- 对于给定正数a，应用Newton法解二次方程$x^2-a=0$，可导出开方值$\sqrt{a}$的计算程序

$$x_{k+1}=\frac{1}{2}\left(x_k+\frac{a}{x_k}\right) \tag{6.3.7}$$

  - 下面证明这种迭代公式对于任意初始值$x_0>0$都是收敛的（并不局限在x^*的邻近）
  - 对式(6.3.7)配方，易知

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

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

Newton法的应用举例（续）

- 将以上两式相除，得

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

- 据此反复递推，有

$$\frac{x_k-\sqrt{a}}{x_k+\sqrt{a}}=\left(\frac{x_0-\sqrt{a}}{x_0+\sqrt{a}}\right)^{2^k} \qquad (6.3.8)$$

- 记$q=\frac{x_0-\sqrt{a}}{x_0+\sqrt{a}}$，整理式(6.3.8)，得

$$x_k-\sqrt{a}=2\sqrt{a}\frac{q^{2^k}}{1-q^{2^k}}$$

- 对于任意$x_0>0$，总有$|q|<1$，故由上式推知，当$k\to\infty$时$x_k\to\sqrt{a}$，即迭代过程恒收敛

$$x_{k+1}=\frac{1}{2}\left(x_k+\frac{a}{x_k}\right) \quad (6.3.7)$$

例6.8

- 求$\sqrt{115}$
  - 取初值$x_0=10$，对$a=115$按式(6.3.7)迭代3次，便得到精度为10^{-6}的结果，见下表

k	0	1	2	3	4
x_k	10	10.750000	10.723837	10.723805	10.723805

  - 由于式(6.3.7)对于任意初值$x_0>0$均收敛，且收敛速度快，故可取确定的初值，如$x_0=1$，来编写通用的程序
  - 用该通用程序求$\sqrt{115}$，也只需迭代7次，即可得到以上结果10.723805

举例

- 对于给定的正数a，对方程$\frac{1}{x}-a=0$应用Newton法，可导出求 $\frac{1}{a}$ 而不用除法的计算程序，也即

$$x_{k+1}=x_k(2-ax_k)$$

  - 这个算法有实际意义，早期设计电子计算机时，为节省硬件设备，曾运用这种技术避开除法操作

  - 该算法在初值x_0满足$0<x_0<\frac{2}{a}$时收敛

举例（续）

- 由于

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

因此，对$r_k=1-ax_k$，有递推公式

$$r_{k+1}=r_k^2$$

- 据此反复递推，有

$$r_k=r_0^{2^k}$$

- 若初值满足$0<x_0<\frac{2}{a}$，则对$r_0=1-ax_0$有$|r_0|<1$。这时有$r_k\to 0$，因此迭代是收敛的

Newton法的收敛问题

- 一般来说，Newton法的收敛性依赖于初值x_0的选取
  - 若x_0偏离所求根x^*比较远，Newton法可能发散
- 用Newton法求方程

$$x^3 - x - 1 = 0 \tag{6.3.9}$$

在$x = 1.5$附近的一个根x^*

  - 设取迭代初值$x_0 = 1.5$，用Newton公式

$$x_{k+1} = x_k - \frac{x_k^3 - x_k - 1}{3x_k^2 - 1} \tag{6.3.10}$$

计算得$x_1 = 1.34783, x_2 = 1.32520, x_3 = 1.32472$

Newton法的收敛问题（续）

- $x_0 = 1.5$时，迭代三次得到的结果x_3有六位有效数字

- 如果改用$x_0 = 0.6$作为迭代初值，依Newton公式(6.3.10)迭代一次，得

$$x_1 = 17.9$$

比$x_0 = 0.6$更偏离所求根$x^* = 1.32472$

□ 因此，需要对Newton法进行改进

Newton下山法

- 为防止迭代发散，对迭代过程再附加一项要求，即具有单调性

$$|f(x_{k+1})| < |f(x_k)| \tag{6.3.11}$$

  - 满足该要求的算法称为下山法

- 将Newton法与下山法结合起来使用，即可在下山法保证函数值稳定下降的前提下，用Newton法加快收敛速度

  - 将Newton法的计算结果$\bar{x}_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$与前一步的近似值$x_k$适当加权平均，得到新的改进值

Newton下山法（续）

$$x_{k+1} = \lambda \bar{x}_{k+1} + (1-\lambda)x_k \qquad (6.3.12)$$

- 其中λ $(0 < \lambda \leq 1)$称为下山因子。在挑选下山因子时，希望使单调性条件(6.3.11)成立

下山因子的选择是一个逐步探索的过程

- 设从$\lambda = 1$开始反复将λ减半进行试算，如果能定出值λ使单调性条件(6.3.11)成立，则称“下山成功”
- 与此相反，如果在上述过程中找不到使条件(6.3.11)成立的下山因子λ，则称“下山失败”，这时需另选初值x_0重算