

第 6 章 方程求根

姚遥

南京大学智能科学与技术学院

本章目录

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

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（对数、三角函数）
- 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

方程求根 | 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，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

方程求根 | 逐步搜索法

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	-	-	-	+

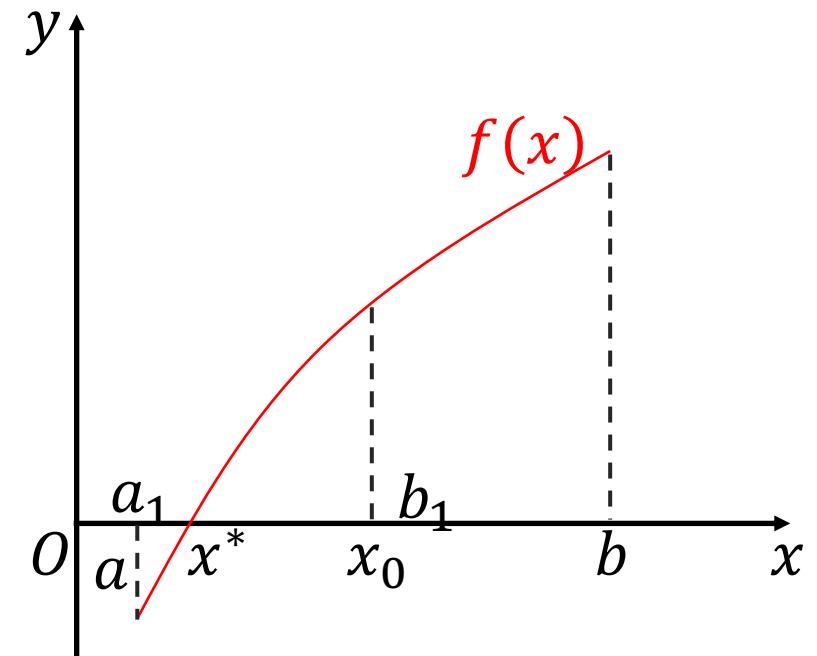
步长 h 的选择是个关键

- h 过大，计算量小，但不精确
- h 过小，精度高，但计算量增大

方程求根 | 二分法

考察有根区间 $[a, b]$ ，取区间中点 $x_0 = \frac{(a+b)}{2}$ 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，则令 $a_1 = a, b_1 = x_0$
- 新有根区间 $[a_1, b_1]$ 的长度仅为 $[a, b]$ 的一半，对压缩了的有根区间 $[a_1, b_1]$ 施行同样的手续



方程求根 | 二分法

考察有根区间 $[a, b]$ ，取区间中点 $x_0 = \frac{(a+b)}{2}$ 分为两半，然后检查 $f(x_0)$ 与 $f(a)$ 是否同号

- 如此反复二分，即可得出一系列有根区间

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

- 每个区间 $[a_k, b_k]$ 都是前一个区间的一半，长度 $b_k - a_k = (b - a)/2^k$ 在 $k \rightarrow \infty$ 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。

方程求根 | 二分法

考察有根区间 $[a, b]$ ，取区间中点 $x_0 = \frac{(a+b)}{2}$ 分为两半，然后检查 $f(x_0)$ 与 $f(a)$ 是否同号

- 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，可以获得一个近似根的序列 $x_0, x_1, \dots, x_k, \dots$ ，该序列必以根 x^* 为极限
- 由于

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

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

方程求根 | 二分法

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，按式(6.1.1)，只需二分 $k = 6$ 次，即可达到预定精度

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

方程求根 | 二分法

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

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	-

方程求根 | 二分法计算步骤

- **准备**：计算 $f(x)$ 在有根区间 $[a, b]$ 端点处的值 $f(a), f(b)$
- **二分**：计算 $f(x)$ 在区间中点 $\frac{a+b}{2}$ 处的值 $f\left(\frac{a+b}{2}\right)$
- **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 $(a, \frac{a+b}{2})$ 内，则以 $(a + b)/2$ 代替 b
 - 若 $f\left(\frac{a+b}{2}\right)$ 与 $f(a)$ 同号，则根位于区间 $(\frac{a+b}{2}, b)$ 内，则以 $(a + b)/2$ 代替 a
- 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

本章目录

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

方程求根 | 迭代法

考察下列形式的方程：

$$x = \varphi(x) \quad (6.2.1)$$

这种方程是**隐式的**，因而不能直接得出它的根

迭代法

- 给出根的猜测值 x_0 ，代入(6.2.1)右端可得 $x_1 = \varphi(x_0)$ ，同样继续可得 $x_2 = \varphi(x_1)$
- 如此反复迭代，如果按公式

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

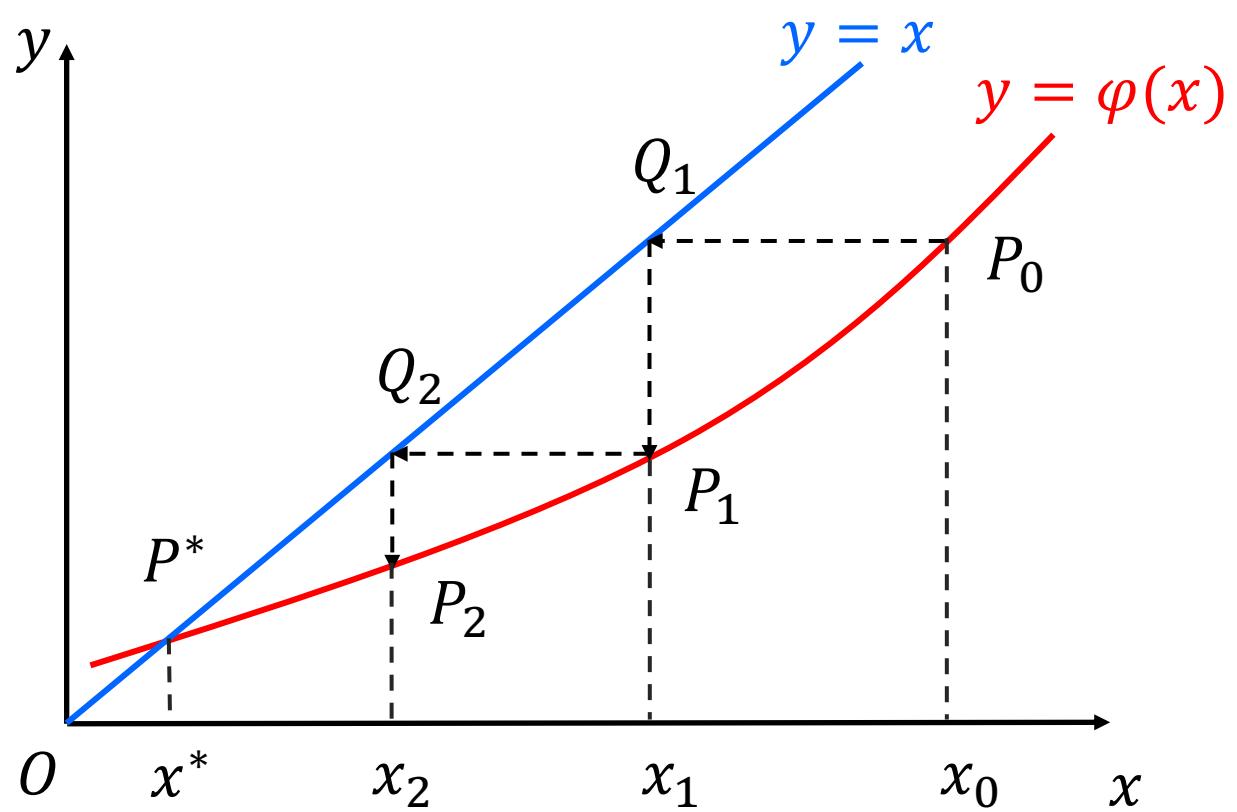
确定的数列 $\{x_k\}$ 有极限 $x^* = \lim_{k \rightarrow \infty} x_k$ ，则称迭代过程收敛，显然 x^* 是方程的根

- 上述迭代法是一种逐次逼近法，其基本思想是将**隐式方程** $x = \varphi(x)$ 归结为一组**显式的计算公式**(6.2.2)

方程求根 | 迭代法

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

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

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

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

- 记录各步迭代的结果，如下表所示

方程求根 | 迭代法

例6.3 求方程

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

在 $x_0 = 1.5$ 附近的根 x^*

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		

- 若仅取 6 位数字， x_7 与 x_8 完全相同，可以认为 x_7 是方程(6.2.3)的根

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

$$x_1 = 2.375, \quad x_2 = 12.39$$

- 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

方程求根 | 迭代法：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

$$|\varphi'(x)| \leq L \quad (6.2.4)$$

- 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， $\xi \in [a, b]$ ，因此利用条件(6.2.4)可得

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

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

- 故当 $k \rightarrow \infty$ 时，迭代值 x_k 将收敛到所求根 x^*

上述推断要求 $x_k \in [a, b]$ ，
进而对于任意 $x \in [a, b]$ ，
要求有 $\varphi(x) \in [a, b]$

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

$$a \leq \varphi(x) \leq b \quad (6.2.5)$$

2. 存在正数 $L < 1$ ，使对于任意 $x \in [a, b]$ ，有

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

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

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

定理6.1证明

- 根据(6.2.6)，有

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

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

- 于是对任意正整数 p ，有

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

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

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

方程求根 | 迭代法：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

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

- 上式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)

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

对于任意正整数 p ，有

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

在上式中令 $p \rightarrow \infty$ ，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

- **准备**：提供迭代初始值 x_0
- **迭代**：计算迭代值 $x_1 = \varphi(x_0)$
- **控制**：检查 $|x_1 - x_0|$
 - 若 $|x_1 - x_0| > \varepsilon$ (ε 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 $|x_1 - x_0| \leq \varepsilon$ 时，终止计算，取 x_1 为所求结果

方程求根 | 迭代法：局部收敛性

在实际应用时，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 : $|x - x^*| \leq \delta$ ，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

方程求根 | 迭代法：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

- 由连续函数性质，存在 x^* 的某个邻域 R : $|x - x^*| \leq \delta$ ，对于任意 $x \in R$ 有

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

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

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

- 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

方程求根 | 迭代法：例子

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， $|e^{-x}'| \approx 0.6 < 1$ ，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
- 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
0	0.5	10	0.5669072
1	0.6065306	11	0.5672772
2	0.5452392	12	0.5670763
3	0.5797031	13	0.5671863
4	0.5600646	14	0.5671188
5	0.5711721	15	0.5671571
6	0.5648629	16	0.5671354
7	0.5684380	17	0.5671477
8	0.5664094	18	0.5671407
9	0.5675596		

方程求根 | 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

方程求根 | 迭代过程的加速

- 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

$$\begin{aligned} x_1 - x^* &\approx L(x_0 - x^*) \\ \Rightarrow x^* &\approx \frac{1}{1-L}x_1 - \frac{L}{1-L}x_0 \end{aligned} \tag{6.2.9}$$

方程求根 | 迭代过程的加速

- 可以期望，按上式右侧求得的

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

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：

- 校正

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

- 改进

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

方程求根 | 迭代过程的加速

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

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

- 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

方程求根 | 迭代过程的加速

例6.5 求解方程 $x = e^{-x}$

- 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，加速效果显著！

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

方程求根 | Aitken (埃特金) 方法

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

解决思路：每次迭代都与 L 相关，可用两次迭代消除 L

- 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 $x_2 = \varphi(x_1)$ ，可得

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

$$x_2 - x^* \approx L(x_1 - x^*)$$

则

$$\frac{x_1 - x^*}{x_2 - x^*} \approx \frac{x_0 - x^*}{x_1 - x^*}$$

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

方程求根 | Aitken (埃特金) 方法

如果将得到改进值作为一步，则计算公式如下

- 校正：

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

- 再校正：

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

- 改进：

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

改进公式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

方程求根 | Aitken (埃特金) 方法

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：

$$x_{k+1} = x_k^3 - 1 \quad (6.2.12)$$

- 现以该公式为基础，形成Aitken算法，即

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

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

方程求根 | Aitken (埃特金) 方法

例6.6 用Aitken方法求解方程(6.2.3)： $f(x) = x^3 - x - 1 = 0$

- 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

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

本章目录

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

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 $\varphi(x) = x + f(x)$ ，这时相应的迭代公式是

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

一般来说，这种迭代公式不一定收敛，或者收敛的速度缓慢

解决方法

- 叠加运用加速技巧

$$\begin{aligned} \bar{x}_{k+1} &= \varphi(x_k) \\ x_{k+1} &= \bar{x}_{k+1} + \frac{L}{1-L}(\bar{x}_{k+1} - x_k) \end{aligned} \quad (6.2.10)$$

方程求根 | Newton法

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 $\varphi(x)$

- 对于迭代过程(6.3.1)，其加速公式为

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

记 $M = L - 1$ ，合并可得

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

- 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

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

方程求根 | Newton法

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 $\varphi(x)$

- 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
- 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

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

方程求根 | Newton法与线性化

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。如果 $f(x)$ 是线性函数，则对它求根是容易的。

- 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

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

- 方程 $f(x) = 0$ 可以近似表示为

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

- 上式为线性方程，记其根为 x_{k+1} ，根的计算公式就是Newton公式

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

方程求根 | Newton法：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

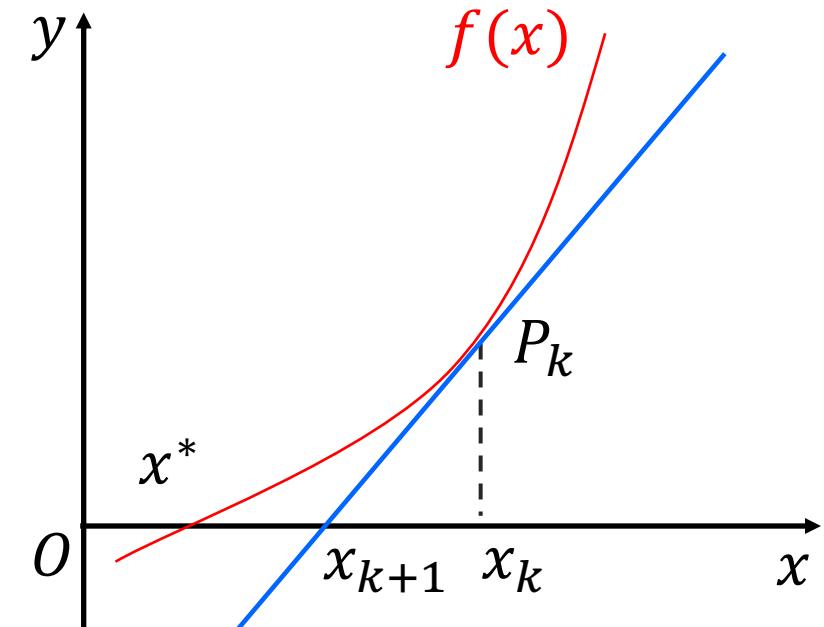
- 过曲线 $y = f(x)$ 上横坐标为 x_k 的点 P_k

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



方程求根 | Newton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 $k \rightarrow \infty$ 时成立下列渐进关系式

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

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

方程求根 | 迭代法收敛速度

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

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

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可断定迭代过程 $x_{k+1} = \varphi(x_k)$ 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

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

方程求根 | 迭代法收敛速度

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

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

则该迭代过程在点 x^* 邻近是 p 阶收敛的

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

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

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

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

对一般迭代方法

- 如果当 $x \in [a, b]$ 时 $\varphi'(x) \neq 0$ ，则迭代过程只是线性收敛的

对Newton公式(6.3.3)，其迭代函数为

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

- 假定 x^* 是 $f(x)$ 的一个单根，即 $f(x^*) = 0, f'(x^*) \neq 0$ ，则可得 $\varphi'(x^*) = 0$
- 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

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

- 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

方程求根 | Newton法：计算步骤

- 准备**：选定初始近似值 x_0 ，计算 $f_0 = f(x_0)$, $f'_0 = f'(x_0)$
- 迭代**：按 $x_1 = x_0 - \frac{f_0}{f'_0}$ 迭代一次得到新的近似值 x_1 ，计算， $f_1 = f(x_1)$, $f'_1 = f'(x_1)$
- 控制**：若满足 $|\delta| < \varepsilon_1$ 或 $|f_1| < \varepsilon_2$ 则终止迭代， x_1 即为所求根；否则转第 4 步

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

其中 C 是取绝对误差或相对误差的控制常数，一般取 $C = 1$

- 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

方程求根 | Newton法：应用举例

开方计算：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

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

- 可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

$$\begin{aligned} x_{k+1} - \sqrt{a} &= \frac{1}{2x_k} (x_k - \sqrt{a})^2 \\ x_{k+1} + \sqrt{a} &= \frac{1}{2x_k} (x_k + \sqrt{a})^2 \end{aligned}$$

方程求根 | Newton法：应用举例

- 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

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

- 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 $k \rightarrow \infty$ 时 $x_k \rightarrow \sqrt{a}$ ，即迭代过程恒收敛

方程求根 | Newton法：应用举例

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

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，故可取确定的初值来写程序，如 $x_0 = 1$
- 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

方程求根 | Newton法：应用举例

除法计算：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

方程求根 | Newton法：应用举例

- 研究误差：

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

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 $r_k \rightarrow 0$ ，因此迭代是收敛的

方程求根 | Newton法：收敛性

- 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

例：用Newton法求方程

$$x^3 - x - 1 = 0 \quad (6.3.9)$$

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

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

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

方程求根 | Newton法：收敛性

- 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，需要想办法对Newton法进行改进

方程求根 | 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

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

- 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 x_k 做适当的**加权平均**，得到新的改进值

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

方程求根 | Newton下山法

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

λ ($0 < \lambda \leq 1$) 称为**下山因子**，我们希望通过挑选下山因子使单调性条件成立

- 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