数值分析与计算软件

第 2 课

刘帆

南京大学工程管理学院

2020 年 2 月 24 日

0. 引言

- 在生产实践和科学技术中，常遇到一些高次代数方程或超越方程，只能用数值方法求出根的近似值。

0. 引言

- 在生产实践和科学技术中，常遇到一些高次代数方程或超越方程，只能用数值方法求出根的近似值。
- 记非线性方程的一般形式为

$$f(x) = 0 \tag{1}$$

方程的解 x^* 称为根，也称为 $f(x)$ 的零点。

0. 引言

- 在生产实践和科学技术中，常遇到一些高次代数方程或超越方程，只能用数值方法求出根的近似值。
- 记非线性方程的一般形式为

$$f(x) = 0 \tag{1}$$

方程的解 x^* 称为根，也称为 $f(x)$ 的零点。

- 当 $f(x) = a_0 + a_1 x + \ldots + a_n x^n$ 时，公式（1）为代数方程。否则，公式（1）称作超越方程。。

0. 引言

- 若 $f(x)$ 可以表示为

$$f(x) = (x - x^*)^m g(x) \tag{2}$$

其中，m 是正整数，且 $g(x^*) \neq 0$，则称 x^* 为 $f(x)$ 的 m 重根。

0. 引言

- 若 $f(x)$ 可以表示为

$$f(x) = (x - x^*)^m g(x) \tag{2}$$

其中，m 是正整数，且 $g(x^*) \neq 0$，则称 x^* 为 $f(x)$ 的 m 重根。

- 若 $f(x) \in C(a, b)$，且 $f(a) * f(b) < 0$，则 (a, b) 上至少存在 $f(x) = 0$ 的一个实根。

0. 引言

数值方法求解方程（1）的解，一般通过以下两步进行逐步搜索

0. 引言

数值方法求解方程（1）的解，一般通过以下两步进行逐步搜索

- 确定根所在的区间，按照某个规则，不停缩小这个有根区间
- 不断重复上一步，有根区间的长度会逐渐趋近到零，这时，区间内的点逐渐逼近方程的根。

1. 二分法

对于在区间 (a, b) 上连续，且满足 $f(a) * f(b) < 0$ 的函数 $f(x)$，通过不断地把函数 $f(x)$ 的零点所在的区间（有根区间）一份为二，使区间的长度不断缩小，从而逐步逼近零点的方法叫做二分法。

二分法的算法

二分法的算法

Step 0： 给定初始有根区间端点 a， b 的值，以及预设精度 e

Step 1： 若 $|b-a| > e$，则计算 $f(x)$ 在区间 (a,b) 端点处的值 $f(a)$ 和 $f(b)$；否则停止，此时 $\frac{a+b}{2}$ 即可作为近似根。

Step 2： 计算 $f(x)$ 在区间中点 $\frac{a+b}{2}$ 处的值 $f(\frac{a+b}{2})$

Step 3： (1) 若 $f(\frac{a+b}{2})=0$，则停止， $\frac{a+b}{2}$ 是根。

(2) 若 $f(\frac{a+b}{2})f(a) < 0$，则令 $b=\frac{a+b}{2}$， 返回 Step 1。

(3) 若 $f(\frac{a+b}{2})f(a) > 0$，则令 $a=\frac{a+b}{2}$， 返回 Step 1。

收敛性

收敛性

上述算法中 Step 2 和 Step 3，每执行一次就把新的区间分成两份，根的范围也缩小一半。记第 k 次二分后得到的区间为 (a_k, b_k)，x_k 为第 k 次循环后的近似解（即区间中点），准确值为 x^*，则

$$|x_k - x^*| = \left|\frac{a_{k-1} + b_{k-1}}{2} - x^*\right| \le \frac{b_{k-1} - a_{k-1}}{2} = \frac{b-a}{2^k} \tag{3}$$

明显，当循环次数 k 趋于无穷大时，近似解的误差限会趋向于 0。

例 1

- 使用二分法求方程 $f(x) = x^3 - x - 1 = 0$ 在区间 $(1, 2)$ 内的实根，设置误差不超过 10^{-2}。

例 1

- 使用二分法求方程 $f(x) = x^3 - x - 1 = 0$ 在区间 $(1, 2)$ 内的实根，设置误差不超过 10^{-2}。解：令 k 为循环次数，误差不超过 10^{-2} 要求

$$\frac{b-a}{2^k} \leq 10^{-2} \tag{4}$$

可以解出 $k \geq 7$。二分法经过 7 次循环的结果，如下表所示：

例 1

结果

k	a_k	b_k	x_k	$f(x_k)$ 的符号
0	1	2	1.5	+
1	1	1.5	1.25	-
2	1.25	1.5	1.375	+
3	1.25	1.375	1.3125	-
4	1.3125	1.375	1.3438	+
5	1.3125	1.3438	1.3281	+
6	1.3125	1.3281	1.3203	-
7	1.3203	1.3281	1.3242	-

总结

1. 二分法

总结

- 简单易用，总是收敛。
- 收敛慢，不能求复根和偶数重根，一次只能求一个根。

总结

- 简单易用，总是收敛。
- 收敛慢，不能求复根和偶数重根，一次只能求一个根。
- 常用于求根的初始近似值, 然后再使用其它的方法求根。

2. 不动点迭代

迭代算法是数值计算方法中一种逐次逼近的方法。

2. 不动点迭代

迭代算法是数值计算方法中一种逐次逼近的方法。

- 构造 $f(x) = 0$ 的一个等价方程：$\phi(x) = x$（一定可以构造出来）。
- $f(x) = 0$ 的根（即 $f(x)$ 的零点），称为 $\phi(x)$ 的不动点。

基本思想

2. 不动点迭代

基本思想

- 给定初始值 x_0，构造如下迭代公式:

$$x_{k+1} = \phi(x_k), k = 0, 1, 2, 3, \cdots . \tag{5}$$

- 若迭代收敛，且 $\lim_{k\to\infty} x_k = x^*$，则 x^* 为不动点，也即是原方程的根。

基本思想

- 给定初始值 x_0，构造如下迭代公式：

$$x_{k+1} = \phi(x_k), k = 0, 1, 2, 3, \cdots . \tag{5}$$

- 若迭代收敛，且 $\lim_{k\to\infty} x_k = x^*$，则 x^* 为不动点，也即是原方程的根。
- 这种方法叫做不动点迭代。

不动点迭代算法

不动点迭代算法

Step 0： 给定迭代初值 x_0 和预设精度 e

Step 1： 计算迭代值 $x_1 = \phi(x_0)$。

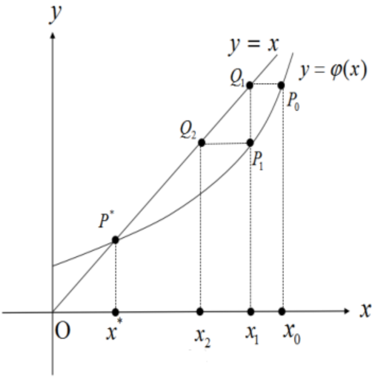
Step 2： （1） 若 $|x_1 - x_0| \geq e$，则令 $x_0 = x_1$，返回 Step 1。

若 $|x_1 - x_0| < e$，则停止，取 x_1 为所求的结果。

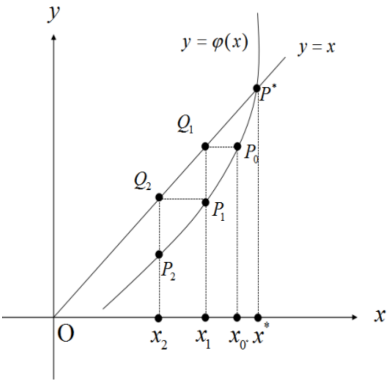
2. 不动点迭代

几何含义：迭代求曲线 $y = \phi(x)$ 与 $y = x$ 的交点

几何含义：迭代求曲线 $y = \phi(x)$ 与 $y = x$ 的交点



2. 不动点迭代



例 2

- 求 $f(x) = x - 10^x + 2 = 0$ 在区间 $(0, 1)$ 中的根，计算结果保留 4 位有效数字

例 2

- 求 $f(x) = x - 10^x + 2 = 0$ 在区间 $(0, 1)$ 中的根，计算结果保留 4 位有效数字

  解 1：因为 $f(0) * f(1) < 0$，所以 $(0, 1)$ 为有根区间。

  构造迭代公式：

  $$x_{k+1} = log_{10}(x_k + 2) \tag{6}$$

  取初始值 $x_0 = 1$，可以逐次算得

  $x_1 = 0.4771$，$x_2 = 0.3939$，$x_3 = 0.3791$，$x_4 = 0.3764$，

  $x_5 = 0.3759$，$x_6 = 0.3758$，......

例 2

解 2：实际上一个方程可以转化的不动点迭代公式是不唯一的。比如，本例子中，也可以构造如下的不动点迭代公式

$$x_{k+1} = 10^{x_k} - 2 \tag{7}$$

你们可以取初值 $x_0 = 1$ 计算，会发现结果开始发散。

这个例子说明，我们在构造不动点迭代的时候，要选取合适的 $\phi(x)$。

收敛性

收敛性

定理

设迭代函数 $\phi(x) \in C^1(a,b)$，如果对任意的 $x \in (a,b)$，满足

(1) $\phi(x) \in (a,b)$；(2) 存在常数 $0 < L < 1$，$|\phi'(x)| \le L < 1$；则 $\phi(x)$ 在 (a,b) 上有唯一不动点，记为 x^*；且对任意初值 $x_0 \in (a,b)$，不动点迭代 $x_{k+1} = \phi(x_k)$ 收敛。其误差估计如下：

$$|x_k - x^*| \le \frac{L}{1-L}|x_k - x_{k-1}| \tag{8}$$

$$|x_k - x^*| \le \frac{L^k}{1-L}|x_1 - x_0| \tag{9}$$

2. 不动点迭代

从该定理中，可以得到

2. 不动点迭代

从该定理中，可以得到

- 只要 $|x_k - x_{k-1}|$ 充分小，就可以保证近似误差 $|x_k - x^*|$ 足够小，因此在算法中用前后两次计算的差值来作为判定终止的条件；
- L 值越小，迭代收敛的越快；

2. 不动点迭代

从该定理中，可以得到

- 只要 $|x_k - x_{k-1}|$ 充分小，就可以保证近似误差 $|x_k - x^*|$ 足够小，因此在算法中用前后两次计算的差值来作为判定终止的条件；
- L 值越小，迭代收敛的越快；
- 在上述定理的条件下，迭代公式在 (a, b) 区间上全局收敛。

从该定理中，可以得到

- 只要 $|x_k - x_{k-1}|$ 充分小，就可以保证近似误差 $|x_k - x^*|$ 足够小，因此在算法中用前后两次计算的差值来作为判定终止的条件；
- L 值越小，迭代收敛的越快；
- 在上述定理的条件下，迭代公式在 (a, b) 区间上全局收敛。
- 若 $\phi(x)$ 不可导，则可用条件：对任意 $x, y \in (a, b)$，都有 $|\phi(x) - \phi(y)| \leq L|x - y|$ 代替，结论仍然成立。

2. 不动点迭代

全局收敛条件比较苛刻，在实际应用时通常只考察不动点迭代在 $f(x) = 0$ 的根的附近是否具有收敛性，即局部收敛性。

全局收敛条件比较苛刻，在实际应用时通常只考察不动点迭代在 $f(x)=0$ 的根的附近是否具有收敛性，即局部收敛性。

定义

设 x^* 是 $\phi(x)$ 的不动点，若存在 x^* 的某个领域 $U=(x^*-\delta, x^*+\delta)$，对任意 $x_0 \in U$，不动点迭代 $x_{k+1}=\phi(x_k)$ 产生的点列都收敛到 x^*，则称之为局部收敛。

2. 不动点迭代

定理

设 x^* *是* $\phi(x)$ *的不动点，若* $\phi'(x)$ *在* x^* *的某个领域内连续，且* $|\phi'(x^*)| < 1$，*则不动点迭代* $x_{k+1} = \phi(x_k)$ *局部收敛。*

例 3

- 求解 $f(x) = x^2 - 3 = 0$ 的正根，在以下三个迭代公式中，哪些是收敛的?

  (1) $\phi(x) = x^2 - 3 + x$;

  (2)$\phi(x) = x - \frac{x^2-3}{4}$;

  (3)$\phi(x) = \frac{1}{2}(x + \frac{3}{x})$

例 3

- 求解 $f(x) = x^2 - 3 = 0$ 的正根，在以下三个迭代公式中，哪些是收敛的?

  (1) $\phi(x) = x^2 - 3 + x$;

  (2)$\phi(x) = x - \frac{x^2-3}{4}$;

  (3)$\phi(x) = \frac{1}{2}(x + \frac{3}{x})$

  解：分别求 $\phi(x)$ 在 $\sqrt{3}$ 处的导数，判断其绝对值是否小于 1 即可。

为了衡量不动点迭代的速度，我们作以下定义

定义

设迭代 $x_{k+1} = \phi(x_k)$ *收敛到* $\phi(x)$ *的不动点* x^**，记* $e_k = x_k - x^*$*，若*

$$\lim_{k \to \infty} \frac{e_{k+1}}{e_k^p} = C \tag{10}$$

其中常数 $C \neq 0$*，则称该迭代为p 阶收敛。特别的，*

(1）当 p=1 时，称为线性收敛，此时要求 $|C| < 1$*；(2）当 p=2 时，称为平方收敛；(3）当 p>1 时，统称为超线性收敛。*

2. 不动点迭代

可以通过以下定理来判断收敛速度

定理

设 x^* 是 $\phi(x)$ 的不动点，若 $\phi^{(p)}(x)$ 在 x^* 的某领域内连续，且

$$\phi'(x^*) = \phi''(x^*) = \cdots = \phi^{(p-1)}(x^*) = 0, \phi^{(p)}(x^*) \neq 0 \qquad (11)$$

则迭代 $x_{k+1} = \phi(x_k)$ 是 p 阶收敛。

例 4

- 求解 $x^3 - x - 1 = 0$ 的正实根 x^3，当前后两次迭代结果相差小于 10^{-5} 时，计算停止。

例 4

- 求解 $x^3 - x - 1 = 0$ 的正实根 x^3，当前后两次迭代结果相差小于 10^{-5} 时，计算停止。

  解：很容易得到原方程只有一个正实根，且在 $(0, 2)$ 之间。我们这里列举两个迭代公式，来比较他们的收敛速度。

  方法一：$\phi(x) = (x+1)^{\frac{1}{3}}$

  方法二：$\phi(x) = \frac{2x^3+1}{3x^2-1}$

  应当是第二种方法收敛更快。

例 4

结果如下:

迭代次数 k	方法一	误差 $\lvert x_k - x_{k-1}\rvert$	方法二	误差 $\lvert x_k - x_{k-1}\rvert$
0	1		1	
1	1.259921	0.259921	1.5	0.500000
2	1.312294	0.052373	1.347826	0.152174
3	1.322354	0.010060	1.325200	0.022626
4	1.324269	0.001915	1.324718	4.82225×10^{-4}
5	1.324633	3.63880×10^{-4}	1.324718	2.16754×10^{-7}
6	1.324702	6.91233×10^{-5}		
7	1.324715	1.31299×10^{-5}		
8	1.324717	2.49399×10^{-6}		

总结

1. 二分法
2. 不动点迭代

谢谢！

Q&A