非线性方程

解法

- 二分法
  - 主要理论：

$$|p_n - p_{n-1}| < \frac{b-a}{2^n}$$

  - 不一定收敛。设置终止条件为 $|p_N - p_{N-1}| < \epsilon$, 可以保证绝对误差较小。
  - 计算问题：
    - 避免向上溢出：$p_n = a_n + \frac{b_n - a_n}{2}$
    - 避免使用乘法：使用符号函数，$signal(f(p_n)) \times signal(f(p_n - 1)) < 0$
- 不动点迭代
  - 主要理论
    - 设 $g(x)$ 为连续函数。
    - 若 $\forall x \in [a,b], g(x) \in [a,b]$, 则区间 $[a,b]$ 中存在不动点。
    - 若 $\exists k < 1\ st. \forall x \in [a,b], g^{'}(x) < k$ 则不动点唯一。
    - 收敛性：在满足前两点的条件下，$p_n = g(p_{n-1})$ 定义的数列收敛到唯一的不动点 p 原理如下：

$$|p_n - p| = |g(p_{n-1}) - g(p)| = |g^{'}(\theta)||p_{n-1} - p| < k \cdot |p_{n-1} - p|$$
$$\implies |p_n - p| < k^{n-1} \cdot |p_1 - p|$$

    进一步的，我们有：（估计值到真实值的距离按k的比例缩减，相邻两次估计点的间距也以k的比例缩减）

$$\begin{cases} |p_n - p| < k^n \times \max(a - p_0, b - p_0); \\ |p_n - p| < \frac{k^n}{1-k} \times |p_1 - p_0| \end{cases}$$

  - 迭代函数收敛的速度与迭代函数的选取有很大关系。收敛速率为 n 的定义：

$$\lim_{x \to \infty} \frac{|p_k - p^*|}{|p_{k-1} - p^*|^n} \approx \mathrm{const}$$

  可以用迭代函数的泰勒展开式计算收敛速率，当迭代函数的 1 到 n-1 阶导数均为0，而 n 阶导数不为零时，迭代函数为 n 阶收敛的：

$$p_k = f(p_{k-1}) = f(p^*) + \frac{f^{(n)}(p^*)}{n!}(p_{k-1} - p^*)^n \implies$$
$$\frac{|p_k - p^*|}{|p_{k-1} - p^*|^n} \approx \frac{f^{(n)}(p^*)}{n!}$$

- 牛顿法：就是迭代法，无重根情况下二次收敛，有重根线性收敛。有两个主要问题，一是要计算导数，二是要选取合适的区间。

$$p_n = p_{n-1} - \frac{f(p_{n-1})}{f'(p_{n-1})}$$

  - 割线方法（牛顿方法的改进，省去导数的计算）：超线性收敛

$$p_n = p_{n-1} - \frac{f(p_{n-1})(p_{n-1} - p_{n-2})}{f(p_{n-1}) - f(p_{n-2})}$$

  - 简化牛顿法：（减少求导的次数，相当于用平行线靠近准确值。）：线性收敛

$$p_k = p_{k-1} - \frac{f(p_{k-1})}{f'(p_0)}$$

  - 修正牛顿法（1）：加快简化牛顿法的收敛，将简化牛顿法中的 m 步合并为一步。
  - 修正牛顿法（2）：解决有重根函数的问题
    - 定义新函数：

$$\mu(x)=\frac{f(x)}{f^{'}(x)}$$

对 $\mu(x)$ 使用牛顿法。

- 牛顿下山法：减小选择区间的约束。
  - 基本思想：选取的 p_{k+1} 始终满足 $f(p_{k+1})<f(p_k)$

$$p_k=p_{k-1}-\lambda\frac{f(p_{k-1})}{f^{'}(p_{k-1})}$$

其中 λ 依次选取 $1,\frac{1}{2},\cdots$ 直到满足下山条件。

- Muller 方法:在局部用三个点确定的二次函数的根来估计下一个点（将一次的切线近似变成二次的近似，加快收敛）

$$x_3=x_2-\frac{2a}{b+sign(b)\sqrt{b^2-4ac}}$$

- Aitken 加速

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

  - 过程一：线性收敛

$$\begin{aligned}p_0,p_1,p_2 &\implies \hat{p_0}\\ p_1,p_2,p_3 &\implies \hat{p_1}\\ &\cdots\end{aligned}$$

数列 $\hat{p_n}$ 构成收敛更快的数列。

  - 过程二：二次收敛

$$\begin{aligned}p_0\to p_1\to p_2 &\implies p_3\\ p_3\to p_4\to p_5 &\implies p_6\\ &\cdots\end{aligned}$$

数值运算中的选择

- 高阶收敛算法
  - 收敛快
  - 计算复杂度高，数值运算不稳定
- 低阶收敛算法
  - 收敛慢
  - 数值计算相对稳定
- 一般选择低阶的算法

单变量求根问题的条件数

- $$cond_{x'}f=\frac{1}{f^{'}(x^{'})}$$
- $$\frac{\text{前向误差}}{\text{后项误差}}=\frac{|p_n-p|}{|f(p_n)-f(p)|}\approx\frac{1}{f^{'}(p)}$$