第十章 非线性方程及非线性方程组的解法

单根：若 $f(x^*)=0$，$f'(x^*)\neq 0$，则 x^*是$f(x)=0$的单根.

重根：若 $f^{(i)}(x^*)=0$ $(i=0,1,\cdots,k)$，$f^{(k+1)}(x^*)\neq 0$，

则 x^*是 $f(x)=0$ 的 $k+1$ 重根.

有根区间：包含至少一个根的区间.

收敛阶：设$\{x_n\}$收敛到x^*，若$\exists\, p\geq 1$，$c\neq 0$，st.

$$\lim_{n\to\infty}\frac{|x_{n+1}-x^*|}{|x_n-x^*|^p}=C,$$

则称$\{x_n\}$ p阶收敛，C为渐近误差常数。

$p=1$时，称$\{x_n\}$线性收敛；

$p=2$时，称$\{x_n\}$平方收敛。

10.1 二分算法

1° 二分算法

设区间$[a,b]$上有且仅有一个根，且 $f(a)f(b)<0$.

记 $a_1=a$，$b_1=b$，$x_n=\frac{1}{2}(a_n+b_n)$

若 $f(x_n)f(a_n)>0$，则令 $\begin{cases}a_{n+1}=x_n\\ b_{n+1}=b_n\end{cases}$

若 $f(x_n)f(b_n)>0$，则令 $\begin{cases}a_{n+1}=a_n\\ b_{n+1}=x_n\end{cases}$

注：二分算法一定收敛，但收敛慢。

2° 线性插值方法

$$x_{n+1}=b_n-f(b_n)\frac{b_n-x_n}{f(b_n)-f(x_n)}$$

注：若$f''(x)\geq 0$，$(x\in[a,b])$，则线性插值方法收敛。

一定条件下，线性插值方法比二分算法快。

10.2 迭代法

1° 不动点迭代法

若 $f(x)=0 \iff x=\varphi(x)$，

则 $f(x)=0$ 的求根问题转换为 $\varphi(x)$ 的不动点问题。

取初始值 x_0，令 $x_{n+1}=\varphi(x_n)$.

2° 不动点迭代法的收敛条件.

定理1：① 设 $\varphi\in C^0[a,b]$，且 $\forall x\in[a,b]$：$\varphi(x)\in[a,b]$.

则 φ 在 $[a,b]$ 有不动点.

② 设 $\varphi\in C^1[a,b]$，且 $\exists L\in(0,1)$，$\forall x\in[a,b]$：

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

则 φ 在 $[a,b]$ 的不动点是唯一的.

证明：① 若 $\varphi(a)=a$ 或 $\varphi(b)=0$，

则不动点存在

若 $\varphi(a)>a$，且 $\varphi(b)<b$.

则令 $\psi(x)=\varphi(x)-x$

则 $\psi(a)>0$，$\psi(b)<0$，且 $\psi(x)\in C[a,b]$

由连续函数的介值定理，$\exists x^*\in(a,b)$：

$\psi(x^*)=0$.

$\therefore$ x^* 是 $\varphi(x)$ 的不动点. □.

② 若 x_1，x_2 均为 $\varphi(x)$ 在 $[a,b]$ 上的不动点，

$$
\begin{aligned}
\text{则 } |x_1-x_2| &= |\varphi(x_1)-\varphi(x_2)| \\
&= |\varphi'(\xi)|\cdot|x_1-x_2| \quad \text{(微分中值定理)} \\
&\le L|x_1-x_2| \\
&< |x_1-x_2|
\end{aligned}
$$

矛盾. □

定理2：设 $\varphi\in C^1[a,b]$ 且 $\forall x\in[a,b]$；$\varphi(x)\in[a,b]$，

且 $\exists L\in(0,1)$，$\forall x\in[a,b]$：$|\varphi'(x)|\le L$，

则不动点迭代法的序列 $\{x_n\}$ 收敛.

证明：设 x^* 为 $[a,b]$ 上的唯一不动点.

$$
\begin{aligned}
\text{则 } |x_n-x^*| &= |\varphi(x_{n-1})-\varphi(x^*)| \\
&= |\varphi'(\xi)|\cdot|x_{n-1}-x^*| \\
&\le L|x_{n-1}-x^*| \\
&\cdots \\
&\le L^n|x_0-x^*|
\end{aligned}
$$

$\therefore$ $\{x_n\}$ 收敛于 x^*.

注：1° 在进行不动点迭代前，先算 L 的值，L 越小，收敛越快。

2° [redacted] 线性收敛，

10.3 迭代法的加速方法。

1° 松弛法

$$\begin{cases} \omega_n = \dfrac{1}{1-\varphi'(x_n)} \\ x_{n+1} = (1-\omega_n)x_n + \omega_n \varphi(x_n) \end{cases}$$

注：1° 要求一阶导数；2° 平方收敛

2° Aitken 加速法

$$\begin{cases} x_{n+1}^{(1)} = \varphi(x_n) \\ x_{n+1}^{(2)} = \varphi(x_{n+1}^{(1)}) \\ x_{n+1} = x_{n+1}^{(2)} - \dfrac{(x_{n+1}^{(2)} - x_{n+1}^{(1)})^2}{x_{n+1}^{(2)} - 2x_{n+1}^{(1)} + x_n} \end{cases}$$

注：1° 不用求导数

2° 平方收敛

3° 可改善收敛性质

10.4 牛顿法（切线法）

将 $f(x)$ 在 x_k 点附近展开，得

$$f(x) = f(x_k) + \text{[redacted]}\ (x-x_k)f'(x_k) + h.o.t$$

线性近似解为

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)} \quad (f'(x_k) \neq 0)$$

注：1° 二阶近似解法称为柯西迭代法

2° $x_0 \approx x^*$ 时，牛顿迭代法平方收敛

3° 简化牛顿法：$x_{k+1} = x_k - \dfrac{f(x_k)}{f'(x_0)}$

导数只用算 1 次；

线性收敛。

4° 牛顿下山法：$x_{k+1} = x_k - \lambda \dfrac{f(x_k)}{f'(x_k)}$，

其中 λ 为下山因子，且 $|f(x_{k+1})| < |f(x_k)|$

收敛性质可能改善；

$\lambda \neq 1$ 时，线性收敛

3° $f(x)$ 为多项式时，用秦九韶算法算函数值和导数值.（可减少计算量）

$$f(c) = (\cdots(a_0 c + a_1)c + a_2)c + \cdots + a_{n-1})c + a_n$$

（导数的算法对应系数不同）.

10.5 弦位法

（拟牛顿法：差商代替导数）

双点弦位法：
$$x_{k+1} = x_k - f(x_k)\frac{x_k - x_{k-1}}{f(x_k) - f(x_{k-1})}$$

单点弦位法：
$$x_{k+1} = x_k - f(x_k)\frac{x_k - x_0}{f(x_k) - f(x_0)}$$

注：1° (x_0, x_1) 充分靠近 x^* 时，双点弦位法收敛阶数 $p=1.618$

2° 单点弦位法本质上是简单迭代法，一般为线性收敛。

3° 弦位法不用求导数，效率高.

4° 用三项的方法称为抛物线法（$p=1.84$）

10.6 非线性方程组的迭代解法

非线性方程组
$$\begin{cases} f_1(x_1, x_2, \cdots, x_n) = 0 \\ f_2(x_1, x_2, \cdots, x_n) = 0 \\ \cdots \\ f_n(x_1, x_2, \cdots, x_n) = 0 \end{cases}$$

1° 简单迭代法

将方程组写为同解方程组 $x_i = \varphi_i(x_1, \cdots, x_n)$ $(i=1,\cdots,n)$

迭代 $x_i^{(k+1)} = \varphi_i(x_1^{(k)}, \cdots, x_n^{(k)})$，$(i=1,\cdots,n)$

向量形式：取初始向量 $\vec{x}^{(0)}$，

迭代 $\vec{x}^{(k+1)} = \Phi(\vec{x}^{(k)})$

收敛性：Jocobi 行列式 $\Phi'(\vec{x}) = \left(\frac{\partial \varphi_i}{\partial x_j}\right)_{n\times n}$

当 $|\Phi'(\vec{x})| \le L < 1$ 时，简单迭代法收敛。

2° 牛顿迭代法

方程组 $f(\vec{x})$ 在 $\vec{x}^{(k)}$ 处泰勒展开

$$f(\vec{x}) = f(\vec{x}^{(k)}) + F(\vec{x}^{(k)})(\vec{x} - \vec{x}^{(k)}) + h.o.t.$$

取线性项作迭代

$$f(\vec{x}^{(k)}) + F(\vec{x}^{(k)})(\vec{x}^{(k+1)} - \vec{x}^{(k)}) = 0$$

即 $\vec{x}^{(k+1)} = \vec{x}^{(k)} - [F(\vec{x}^{(k)})]^{-1} f(\vec{x}^{(k)})$

其中 $F(\vec{x})$ 为Jacobi矩阵，$F(\vec{x}) = \left(\frac{\partial f_i(\vec{x})}{\partial x_j}\right)_{n\times n}$.

令 $F(\vec{x}^{(k)}) = L^{(k)}U^{(k)}$

$\vec{\delta}^{(k)} = [F(\vec{x}^{(k)})]^{-1} f(\vec{x}^{(k)})$

$U^{(k)}\vec{\delta}^{(k)} = \vec{y}^{(k)}$

得到迭代格式：

$$\begin{cases} F(\vec{x}^{(k)}) = L^{(k)}U^{(k)} \\ L^{(k)}\vec{y}^{(k)} = f(\vec{x}^{(k)}) \\ U^{(k)}\vec{\delta}^{(k)} = \vec{y}^{(k)} \\ \vec{x}^{(k+1)} = \vec{x}^{(k)} - \vec{\delta}^{(k)} \end{cases}$$

注：1° 简化牛顿法：取 $F(\vec{x}^{(k)}) = F(\vec{x}^{(0)})$，从而省去了第一步。

2° 拟牛顿法（弦位法）：用差商代替偏导数。

3° 导数计算量大或无法求得时，用弦位法或简单迭代法。

4° 可用二分法或最速下降法求初值。