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 | 弦截法与抛物线法：研究动机

用Newton公式 求 x_{k+1} 时，不但要给出函数值 $f(x_k)$，而且要提供导数值 $f'(x_k)$

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

- 当函数f比较复杂时，提供它的导数值往往是有困难的

弦截法与抛物线法

- 设法利用迭代过程中的函数值“老信息” $f(x_k),\ f(x_{k-1}),\cdots,f(x_{k-r})$ 来回避导数值$f'(x_k)$ 的计算
- 导出这类求根方法的基础是插值原理

方程求根 | 弦截法与抛物线法

设 $x_k, x_{k-1}, \cdots, x_{k-r}$ 是 $f(x) = 0$ 的一组近似根

- 利用函数值 $f(x_k), f(x_{k-1}), \cdots f(x_{k-r})$ 构造插值多项式 $P_r(x)$
- 适当选取 $P_r(x) = 0$ 的一个根作为 $f(x) = 0$ 的新近似根 x_{k+1}

这就确定了一个迭代过程，记迭代函数为 φ，则

$$x_{k+1} = \varphi(x_k, x_{k-1}, \cdots, x_{k-r})$$

下面具体考察 $r = 1$（弦截法）和 $r = 2$（抛物线法）两种情形

方程求根 | 弦截法

设 x_k, x_{k-1} 是 $f(x)=0$ 的近似根，利用 $f(x_k),\ f(x_{k-1})$ 构造一次插值多项式 $P_1(x)$，并用 $P_1(x)=0$ 的根作为 $f(x)=0$ 的新近似根 x_{k+1}

- 易得

$$P_1(x)=f(x_k)+\frac{f(x_k)-f(x_{k-1})}{x_k-x_{k-1}}(x-x_k) \qquad (6.4.1)$$

- 因此有

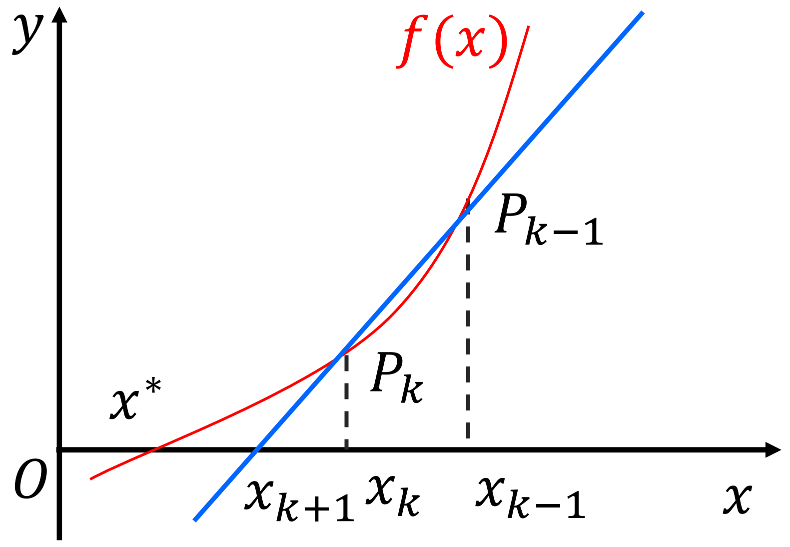
$$x_{k+1}=x_k-\frac{f(x_k)}{f(x_k)-f(x_{k-1})}(x_k-x_{k-1}) \qquad (6.4.2)$$

- 导出的迭代公式(6.4.2)可看作Newton公式中的导数 $f'(x_k)$ 用差商 $\frac{f(x_k)-f(x_{k-1})}{x_k-x_{k-1}}$ 取代的结果

方程求根 | 弦截法：几何意义

曲线 $y=f(x)$ 上横坐标为 x_k, x_{k-1} 的点分别记作 P_k, P_{k-1}，则弦线 P_kP_{k-1} 的斜率等于差商值 $\frac{f(x_k)-f(x_{k-1})}{x_k-x_{k-1}}$，其方程是

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



- x_{k+1} 是弦线 P_kP_{k-1} 与 x 轴交点的横坐标
- 因此，被称为弦截法

方程求根 | 弦截法 vs Newton法

弦截法与Newton法都是线性化方法，但有本质区别

Newton法计算 x_{k+1} 时只用到前一步的值 x_k

弦截法求 x_{k+1} 时要用到前面两步的结果 x_k, x_{k-1} ，因此必须给出两个开始值 x_0, x_1

方程求根 | 弦截法：收敛性

定理6.4 假设 $f(x)$ 在根 x^* 的邻域 Δ: $|x-x^*|\le\delta$ 内具有二阶连续导数，且对于任意 $x\in\Delta$，有 $f'(x)\neq 0$，又设初值 $x_0, x_1\in\Delta$，那么当邻域 Δ 充分小时，弦截法(6.4.2)将按阶 $p=\frac{1+\sqrt{5}}{2}\approx 1.618$ 收敛到根 x^*

- 弦截法具有超线性的收敛性（ $p=1.618$ ）
- Newton法在根 x^* 的邻近是平方收敛的

例6.9 用弦截法解方程：$f(x) = xe^x - 1 = 0$

- 设取 $x_0 = 0.5, x_1 = 0.6$ 作为开始值，用弦截法求得的结果如下表所示

k	0	1	2	3	4
x_k	0.5	0.6	0.56532	0.56709	0.56714

- 比较例6.7 用Newton法的计算结果，可见弦截法的收敛速度也是相当快的

k	0	1	2	3
x_k	0.5	0.57102	0.56716	0.56714

方程求根 | 弦截法：计算步骤

- **准备**：选取初始近似值 x_0, x_1，计算相应的函数值 $f_0 = f(x_0)$，$f_1 = f(x_1)$
- **迭代**：按公式 $x_2 = x_1 - f_1 / \frac{f_1 - f_0}{x_1 - x_0}$，迭代一次，得新的近似值 x_2，计算 $f_2 = f(x_2)$
- **控制**：如果 x_2 满足 $|\delta| \le \varepsilon_1$ 或 $|f_2| \le \varepsilon_2$，则认为过程收敛，终止迭代并输出 x_2 为所求根；否则执行步骤 4

$$\delta = \begin{cases} |x_2 - x_1|, & |x_2| < C \\ \dfrac{|x_2 - x_1|}{|x_2|}, & |x_2| \ge C \end{cases}$$

此处 $\varepsilon_1, \varepsilon_2$ 是允许误差，其中 C 是预先指定的控制常数

- **修改**：若迭代次数达到预先指定的次数 N，则认为过程不收敛，计算失败；否则以 $(x_1, f_1), (x_2, f_2)$ 分别代替 $(x_0, f_0), (x_1, f_1)$，继续进行第 2 步的迭代

方程求根 | Aitken加速公式：弦截法视角

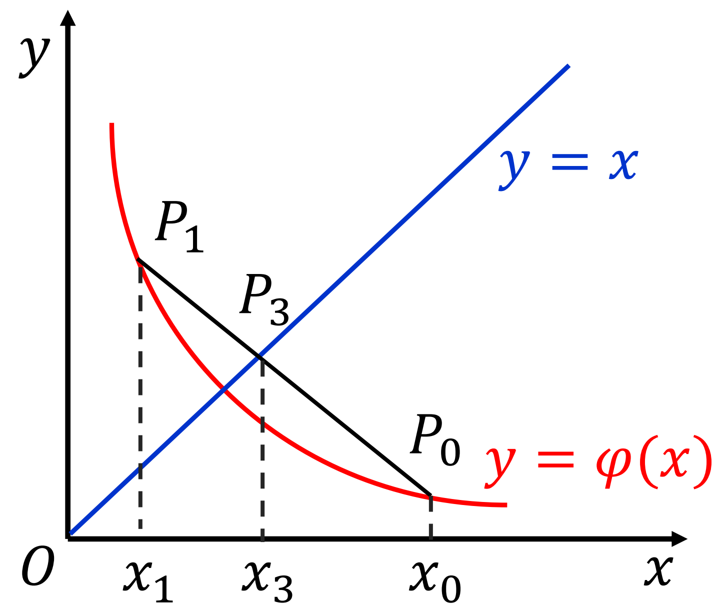
用弦截法求解形如 $x=\varphi(x)$ 的方程（构造 $x_3=\varphi(x_2,x_1)$）

- 设 x_0 为方程 $x=\varphi(x)$ 的一个近似根，依据迭代值 $x_1=\varphi(x_0)$， $x_2=\varphi(x_1)$ 在曲线 $y=\varphi(x)$ 上定出两点 P_0和 P_1
- 引弦线 P_0P_1，设与直线 $y=x$ 交于一点 P_3
- 则点P_3的坐标x_3满足

$$x_3=x_1+\frac{x_2-x_1}{x_1-x_0}(x_3-x_0)$$

- 由此解出Aitken加速公式

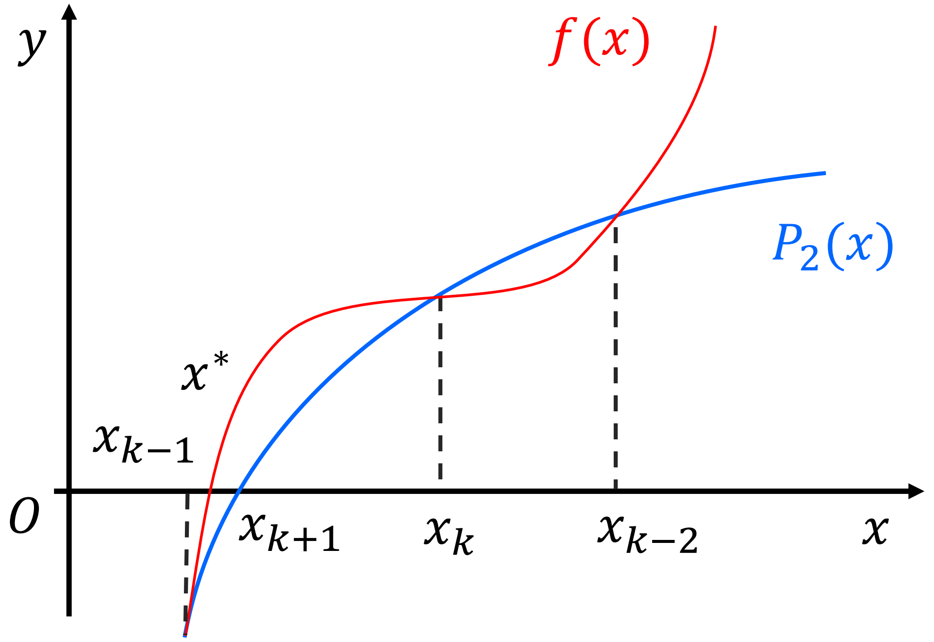
$$x_3=\frac{x_0x_2-x_1^2}{x_0-2x_1+x_2}$$



方程求根 | 抛物线法

已知方程 $f(x) = 0$ 的三个近似根 x_k, x_{k-1}, x_{k-2}，以这三点为节点构造二次插值多项式 $P_2(x)$，并适当选取 $P_2(x)$ 的一个零点 x_{k+1} 作为新的近似根，这样确定的迭代过程称为抛物线法

- 用抛物线 $y = P_2(x)$ 与 x 轴的交点 x_{k+1} 作为所求根 x^* 的近似位置



方程求根 | 抛物线法：计算公式

- 根据插值多项式

$$P_2(x) = f(x_k) + f[x_k, x_{k-1}](x - x_k) \\ + f[x_k, x_{k-1}, x_{k-2}](x - x_k)(x - x_{k-1})$$

可得两个零点

$$x_{k+1} = x_k - \frac{2f(x_k)}{\omega \pm \sqrt{\omega^2 - 4f(x_k)f[x_k, x_{k-1}, x_{k-2}]}} \qquad (6.4.3)$$

$$\omega = f[x_k, x_{k-1}] + f[x_k, x_{k-1}, x_{k-2}](x_k - x_{k-1})$$

- 为了从式(6.4.3)中确定一个值 x_{k+1}，需要讨论根式前的正负号
  - 在 x_k, x_{k-1}, x_{k-2} 三个近似根中，假定 x_k 更接近 x^*，故选接近 x_k 的 x_{k+1}
  - 因此只要令根式前的符号与 ω 的符号相同（分母更大）

方程求根 | 抛物线法：收敛速率

在一定条件下可以证明，对于抛物线法，迭代误差有下列渐近关系式：

$$\frac{|e_{k+1}|}{|e_k|^{1.840}} \to \left|\frac{f'''(x^*)}{6f'(x^*)}\right|^{0.42}$$

- **抛物线法**也是超线性收敛的（$p = 1.840$），收敛速率比弦截法更接近Newton法
- **弦截法**具有超线性的收敛性（ $p = 1.618$ ）
- **Newton法**在根 x^* 的邻近是平方收敛的（ $p = 2$ ）

方程求根 | 抛物线法：例子

例6.10 用抛物线法求解方程 $f(x) = xe^x - 1 = 0$

- 设用表6.9中的前三个值 $x_0 = 0.5, x_1 = 0.6, x_2 = 0.56532$ 作为开始值，计算得

$$
\begin{aligned}
&f(x_0) = -0.175639, &\quad& f(x_1) = -0.093271,\\
&f(x_2) = -0.005031, && f[x_1, x_0] = 2.68910,\\
&f[x_2, x_1] = 2.83373, && f[x_2, x_1, x_0] = 2.21418,
\end{aligned}
$$

- 故 $\omega = f[x_2, x_1] + f[x_2, x_1, x_0](x_2 - x_1) = 2.75694$

- 代入式(6.4.3)，求得

$$x_3 = x_2 - \frac{2f(x_2)}{\omega + \sqrt{\omega^2 - 4f(x_2)f[x_2, x_1, x_0]}} = 0.56714$$

方程求根 | 抛物线法：计算步骤

- **准备**：选定初始近似值 x_0, x_1, x_2，计算对应函数值 f_0, f_1, f_2
- **迭代**：计算下面，并取 $\pm$ 号让分母最大

$$\lambda_2 = \frac{x_2 - x_1}{x_1 - x_0}, \qquad \delta_2 = 1 + \lambda_2, \qquad a = f_0\lambda_2^2 - f_1\lambda_2\delta_2 + f_2\lambda_2,$$

$$b = f_0\lambda_2^2 - f_1\delta_2^2 + f_2(\lambda_2 + \delta_2), \qquad c = f_2\delta_2, \qquad \lambda_3 = \frac{-2c}{b \pm \sqrt{b^2 - 4ac}}$$

得新的近似值 $x_3 = x_2 + \lambda_3(x_2 - x_1)$，再计算 $f_3 = f(x_3)$

- **控制**：若x_3满足 $|\delta| \le \varepsilon_1$ 或 $|f_3| \le \varepsilon_2$（$\varepsilon_1, \varepsilon_2, \delta$ 的意义与弦截法的计算步骤中的相同），则终止迭代，输出 x_3 为所求根；否则执行第4步
- **修改** 如果迭代次数达到预先设定的次数 N，则认为过程不收敛，输出计算失效标志；否则以 $(x_1, x_2, x_3, f_1, f_2, f_3)$ 分别代替 $(x_0, x_1, x_2, f_0, f_1, f_2)$ 转第2步继续迭代

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 | 代数方程求根

如果 $f(x)$ 是多项式，则 $f(x)=0$ 特别地称为代数方程

- 前面介绍的求根方法原则上也适用于解代数方程 ✓
- 由于多项式的特殊性，可以针对其特点提供更为有效的算法

多项式求值的秦九韶算法

- 多项式求值、求导很方便

代数方程的Newton法

劈因子法

方程求根 | 多项式求值的秦九韶算法

设给定系数 $a_i\ (0 \le i \le n)$ 均为实数的 n 次多项式 $f(x)$ ，如何快速计算函数值 $f(x_0)$ 及其各阶导数？

$$f(x) = a_0x^n + a_1x^{n-1} + \cdots + a_{n-1}x + a_n$$

- 根据泰勒展开易得

$$f(x) = f(x_0) + f'(x_0)(x - x_0) + \frac{f''(x_0)}{2!}(x - x_0)^2 + \cdots + \frac{f^{(n)}(x_0)}{n!}(x - x_0)^n$$

- 可得

$$f(x) = f(x_0) + (x - x_0)P(x) \qquad (6.5.1)$$

其中 $P(x)$ 为 $n-1$ 次多项式

$$P(x) = b_0x^{n-1} + b_1x^{n-2} + \cdots + b_{n-2}x + b_{n-1}$$

方程求根 | 多项式求值的秦九韶算法

- 得到两种写法

$$f(x) = a_0x^n + a_1x^{n-1} + \cdots + a_{n-1}x + a_n$$

$$f(x) = f(x_0) + (x - x_0)(b_0x^{n-1} + b_1x^{n-2} + \cdots + b_{n-2}x + b_{n-1})$$

- 比较两端同次幂的系数，得

$$\begin{cases} a_0 = b_0 \\ a_i = b_i + x_0b_{i-1}, \qquad 1 \le i \le n-1 \\ a_n = f(x_0) - x_0b_{n-1} \end{cases}$$

- 从而有

$$\begin{cases} b_0 = a_0 \\ b_i = a_i + x_0b_{i-1}, \qquad 1 \le i \le n \\ f(x_0) = b_n \end{cases} \qquad (6.5.2)$$

方程求根 | 多项式求值的秦九韶算法

这里提供的一种计算函数值 $f(x_0)$ 的有效算法称为秦九韶法

- 外国文献中通常称为Horner算法，其实比秦九韶晚了五六个世纪
- 这种算法的优点是计算量小，结构紧凑，容易编制计算程序
- 在计算 $f(x_0)$ 的同时，还得到了 $P(x)$ 的系数

方程求根 | 多项式求值的秦九韶算法

继续讨论如何求解 $f'(x_0)$

- 进一步考察 $f(x)$ 的Taylor展开式

$$f(x) = f(x_0) + f'(x_0)(x - x_0) + \frac{f''(x_0)}{2!}(x - x_0)^2 + \cdots + \frac{f^{(n)}(x_0)}{n!}(x - x_0)^n$$

- 对比式(6.5.1)

$$f(x) = f(x_0) + (x - x_0)P(x) \tag{6.5.1}$$

可知

$$P(x) = f'(x_0) + \frac{f''(x_0)}{2!}(x - x_0) + \cdots + \frac{f^{(n)}(x_0)}{n!}(x - x_0)^{n-1}$$

方程求根 | 多项式求值的秦九韶算法

- 由此可见，导数 $f'(x_0)$ 又可看作 $P(x)$ 用因式 $x-x_0$ 相除得出的余数，即有

$$P(x)=f'(x_0)+(x-x_0)Q(x)$$

其中 $Q(x)$ 是 $n-2$ 次多项式：$Q(x)=c_0x^{n-2}+c_1x^{n-3}+\cdots+c_{n-3}x+c_{n-2}$

- 注意到 $P(x)$ 的具体形式在计算 $f(x_0)$ 时已经得到，因此可以针对上式重复使用秦九韶算法，即可得到 $f'(x_0)$ 和 $Q(x)$

$$\begin{cases} b_0=a_0 \\ b_i=a_i+x_0b_{i-1}, 1\le i\le n \\ f(x_0)=b_n \end{cases} \quad (6.5.2) \qquad \begin{cases} c_0=b_0 \\ c_i=b_i+x_0c_{i-1}, 1\le i\le n-1 \\ f'(x_0)=c_{n-1} \end{cases} \quad (6.5.3)$$

- **继续这一过程，可以依次求出 $f(x)$ 在点 x_0 处的各阶导数**

方程求根 | 代数方程求根：Newton法

再就多项式方程

$$f(x) = a_0x^n + a_1x^{n-1} + \cdots + a_{n-1}x + a_n = 0$$

考察Newton公式

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)} \tag{6.5.4}$$

- 根据式前几页的推导，上式中的函数值 $f(x_k)$ 和导数值 $f'(x_k)$ 均可方便地求出

$$\begin{cases} b_0 = a_0 \\ b_i = a_i + x_0 b_{i-1}, 1 \le i \le n \\ f(x_0) = b_n \end{cases} \quad (6.5.2) \qquad \begin{cases} c_0 = b_0 \\ c_i = b_i + x_0 c_{i-1}, 1 \le i \le n-1 \\ f'(x_0) = c_{n-1} \end{cases} \quad (6.5.3)$$

方程求根 | 代数方程求根：劈因子法

用二次式 $\omega(x)$ 除 $f(x)$，商是一个 $n-2$ 次多项式 $P(x)$ ，余式为一次式 r_0x+r_1

$$f(x)=(x^2+ux+v)P(x)+r_0x+r_1 \tag{6.5.7}$$

其中 r_0,r_1 均为 u,v 的函数： $r_0=r_0(u,v)$，$r_1=r_1(u,v)$

劈因子法的目的是逐步修改 u,v 的值，使余数 r_0,r_1 变得很小，则可从 $f(x)$ 中分离出二次式 $\omega(x)$， $\omega(x)$ 的即为 $f(x)$ 的根

方程求根 | 代数方程求根：劈因子法

- 考察非线性方程

$$\begin{cases} r_0(u,v)=0 \\ r_1(u,v)=0 \end{cases} \tag{6.5.8}$$

- 设有解 (u^*,v^*)，将 $r_0(u^*,v^*)=0$, $r_1(u^*,v^*)=0$ 的左端在 (u,v) 展开到一阶项

$$\begin{cases} r_0+\dfrac{\partial r_0}{\partial u}(u^*-u)+\dfrac{\partial r_0}{\partial v}(v^*-v)\approx 0 \\ r_1+\dfrac{\partial r_1}{\partial u}(u^*-u)+\dfrac{\partial r_1}{\partial v}(v^*-v)\approx 0 \end{cases}$$

- 用Newton法的处理思想，将非线性方程组线性化，归结得到下列线性方程组

$$\begin{cases} r_0+\dfrac{\partial r_0}{\partial u}\Delta u+\dfrac{\partial r_0}{\partial v}\Delta v=0 \\ r_1+\dfrac{\partial r_1}{\partial u}\Delta u+\dfrac{\partial r_1}{\partial v}\Delta v=0 \end{cases} \tag{6.5.9}$$

- 从方程组(6.5.9)解出增量 $\Delta u, \Delta v$，即可得改进后的二次因式

$$\omega(x) = x^2 + (u + \Delta u)x + v + \Delta v$$

关键问题：如何得到方程组(6.5.9)的各个系数

$$\begin{cases} r_0 + \dfrac{\partial r_0}{\partial u}\Delta u + \dfrac{\partial r_0}{\partial v}\Delta v = 0 \\ r_1 + \dfrac{\partial r_1}{\partial u}\Delta u + \dfrac{\partial r_1}{\partial v}\Delta v = 0 \end{cases} \qquad (6.5.9)$$

$$f(x) = (x^2 + ux + v)P(x) + r_0 x + r_1 \qquad (6.5.7)$$

1. 计算 r_0 和 r_1：将

$$P(x) = b_0x^{n-2} + b_1x^{n-3} + \cdots + b_{n-3}x + b_{n-2}$$

代入式(6.5.7)

$$f(x) = (x^2 + ux + v)P(x) + r_0x + r_1 \qquad (6.5.7)$$

$$f(x) = a_0x^n + a_1x^{n-1} + \cdots + a_{n-1}x + a_n$$

- 比较各次幂的系数，易知

$$\begin{cases} a_0 = b_0 \\ a_1 = b_1 + ub_0 \\ a_i = b_i + ub_{i-1} + vb_{i-2}, \qquad 2 \leq i \leq n-2 \\ a_{n-1} = ub_{n-2} + vb_{n-3} + r_0 \\ a_n = vb_{n-2} + r_1 \end{cases}$$

方程求根 | 代数方程求根：劈因子法

- 可得 r_0, r_1 的计算公式

$$\begin{cases} b_0 = a_0 \\ b_1 = a_1 - ub_0 \\ b_i = a_i - ub_{i-1} - vb_{i-2}, \qquad 2 \le i \le n \\ r_0 = b_{n-1} \\ r_1 = b_n + ub_{n-1} \end{cases} \tag{6.5.10}$$

- 注意到，在计算r_0, r_1的同时，我们还得到了$P(x)$的系数

$$P(x) = b_0 x^{n-2} + b_1 x^{n-3} + \cdots + b_{n-3} x + b_{n-2}$$

2. 计算 $\frac{\partial r_0}{\partial v}, \frac{\partial r_1}{\partial v}$：对式(6.5.7)关于 v 求导

$$f(x) = (x^2 + ux + v)P(x) + r_0 x + r_1 \tag{6.5.7}$$

得

$$P(x) = -(x^2 + ux + v)\frac{\partial P}{\partial v} + s_0 x + s_1 \tag{6.5.11}$$

$$s_0 = -\frac{\partial r_0}{\partial v}, s_1 = -\frac{\partial r_1}{\partial v} \tag{6.5.12}$$

- 由(6.5.11)可知，用 $x^2 + ux + v$ 除 $P(x)$，作为余式可得 $s_0 x + s_1$
- 由于$P(x)$是 $n-2$ 次多项式，这里商 $\frac{\partial P}{\partial v}$ 是 $n-4$ 次多项式

方程求根 | 代数方程求根：劈因子法

$$P(x) = -(x^2 + ux + v)\frac{\partial P}{\partial v} + s_0 x + s_1 \qquad (6.5.11)$$

- 记

$$\frac{\partial P}{\partial v} = c_0 x^{n-4} + c_1 x^{n-5} + \cdots + c_{n-5} x + c_{n-4}$$

- 注意到，$P(x)$ 已经在第一步计算得到，因此可以继续模仿(6.5.10)的计算过程（两种表达的系数相等），来计算 s_0 和 s_1

$$\begin{cases} b_0 = a_0 \\ b_1 = a_1 - ub_0 \\ b_i = a_i - ub_{i-1} - vb_{i-2}, \qquad 2 \le i \le n \\ r_0 = b_{n-1} \\ r_1 = b_n + ub_{n-1} \end{cases} \qquad (6.5.10)$$

方程求根 | 代数方程求根：劈因子法

- 得到

$$\begin{cases} c_0 = b_0 \\ c_1 = b_1 - ub_0 \\ c_i = b_i - uc_{i-1} - vc_{i-2}, \qquad 2 \le i \le n-2 \\ s_0 = c_{n-3} \\ s_1 = c_{n-2} + uc_{n-3} \end{cases}$$

- 可知

$$\frac{\partial r_0}{\partial v} = -s_0, \frac{\partial r_1}{\partial v} = -s_1$$

3. 计算 $\frac{\partial r_0}{\partial v}, \frac{\partial r_1}{\partial u}$：对式(6.5.7)关于 u 求导

$$f(x) = (x^2 + ux + v)P(x) + r_0 x + r_1 \tag{6.5.7}$$

可得

$$xP(x) = -(x^2 + ux + v)\frac{\partial P}{\partial u} - \frac{\partial r_0}{\partial u}x - \frac{\partial r_1}{\partial u}$$

- 另外，由式(6.5.11)有

$$\begin{aligned} xP(x) &= -(x^2 + ux + v)x\frac{\partial P}{\partial v} + (s_0 x + s_1)x \\ &= -(x^2 + ux + v)\left(x\frac{\partial P}{\partial v} - s_0\right) - (us_0 - s_1)x - vs_0 \end{aligned}$$

- 可得

$$\frac{\partial r_0}{\partial u} = us_0 - s_1, \frac{\partial r_1}{\partial u} = vs_0$$

方程求根 | 总结

6.1 根的搜索

- 逐步搜索法、二分法、二分法的收敛性

6.2 迭代法

- 收敛条件、误差估计、局部收敛性
- 迭代公式的加工、Aitken方法

6.3 Newton法

- Newton公式、几何解释
- 局部收敛性、Newton下山法

方程求根 | 总结

6.4 弦截法与抛物线法

- 弦截法、几何意义、收敛性
- 抛物线法、几何意义、收敛性

6.5 代数方程求根

- 多项式求值的秦九韶算法
- 代数方程的Newton法
- 劈因子法

第六章

习题 3(1)(3)，7(2)(3)，8，13，15