1 插值法

1.1 引言

定义 1 设函数$f(x)$在$[a,b]$上有定义，且已知$a \le x_0 < x_1 \cdots < x_n \le b$点上的值$y_0, y_1, \cdots, y_n$，若存在简单函数$\phi(x)$，使得

$$\phi(x_i) = y_i \quad i = 0, 1, 2, \cdots n \tag{1.1.1}$$

成立，则称$\phi(x)$ 为$f(x)$ 的插值函数。式1.1.1称为插值条件， $f(x)$ 称为被插值函数，$[a,b]$ 称为插值区间，$x_0, x_1, \cdots, x_n$ 称为插值节点，求$\phi(x)$ 的方法就是插值法。

插值法的研究问题为

1. $\phi(x)$ 是否存在，是否唯一
2. 若存在，如何构造$\phi(x)$
3. 如何估计$\phi(x)$的误差

1.2 Lagrange插值多项式

当$n = 1$ 时，要构造通过两点(x_0, y_0) 和(x_1, y_1) 的不超过一次的多项式$L_1(x)$，使得

$$\begin{cases} L_1(x_0) = y_0 \\ L_1(x_1) = y_1 \end{cases}$$

$L_1(x)$ 的表达式为

$$L_1(x) = \frac{x_1 - x}{x_1 - x_0} y_0 + \frac{x - x_0}{x_1 - x_0} y_1$$

$L_1(x)$ 表达式可以看作是函数值y_0 和y_1 的线性组合，组合的系数记为$l_0(x)$和$l_1(x)$，即

$$\begin{cases} l_0 = \frac{x - x_1}{x_0 - x_1} \\ l_1 = \frac{x - x_0}{x_1 - x_0} \end{cases}$$

系数$l_0(x)$和$l_1(x)$不是常数，而是一次多项式，因此，组合后的结果也是一次多项式。$l_0(x)$和$l_1(x)$称为节点x_0, x_1 上的线性插值基函数。线性插值基函数还需要满足插值条件（见表1)。

根据基函数的表达式可以得知，基函数的构造与函数值无关。

将上述公式推广到一般情形。通过$n+1$ 个节点的n次插值多项式$L_n(x)$，设$L_n(n) = y_0 l_0(x) + y_1 l_1(x) + \cdots + y_n l_x(n)$满足插值条件$L_n(x_j) = y_j$， $j = 0, 1, \cdots, n$。

表 1: interpolate condition

	x_0	x_1
$l_0(x)$	1	0
$l_1(x)$	0	1

定义 2 若n 次多项式$l_k(x)(k=0,1,\cdots,n)$在各节点$x_0<x_1<\cdots<x_n$ 上满足条件，

$$l_k(x_i)=\delta_{ki}=\begin{cases}1 & k=i\\ 0 & k\neq i\end{cases}\quad i,k=0,1,\cdots,n \tag{1.2.1}$$

就称这$n+1$ 个n 次多项式$l_0(x),l_1(x),\cdots,l_n(x)$ 为节点$x_0,x_1,\cdots,x_n$ 上的n次插值基函数。

用类推的方式可以得到n 次插值基函数为

$$l_k(x)=\prod_{j=1,k\neq j}^{n}\frac{x-x_j}{x_k-x_j}$$

于是，插值多项式函数可以表示为

$$L_n(x)=\sum_{k=0}^{n}y_k l_k(x) \tag{1.2.2}$$

形如式1.2.2的插值多项式$L_n(x)$ 称为Lagrange插值多项式。引入记号

$$\omega_{n+1}(x)=(x-x_0)(x-x_1)\cdots(x-x_n)$$

可知

$$\omega'_{n+1}(x_k)=(x_k-x_0)\cdots(x_k-x_{k-1})(x_k-x_{k+1})\cdots(x_k-x_n)$$

于是式1.2.2 可以改写为

$$L_n(x)=\sum_{k=0}^{n}y_k\frac{\omega_{n+1}(x)}{(x-x_k)\omega'_{n+1}(x_k)}$$

若在$[a,b]$ 上用$L_n(x)$ 近似$f(x)$，则其截断误差为$R_n(x)=f(x)-L_n(x)$，$R_n(x)$ 也称为插值多项式的余项或插值余项。

定理 1 设$f^{(n)}(x)$ 在$[a,b]$ 上连续，$f^{(n+1)}(x)$ 在(a,b) 内存在，节点$a\le x_0<x_1<\cdots<x_n\le b$，$L_n(x)$是满足插值条件的插值多项式，则对于任何$x\in[a,b]$，插值余项为

$$R_n(x)=f(x)-L_n(x)=\frac{f^{(n+1)}(\xi)}{(n+1)!}\omega_{n+1}(x) \tag{2.4}$$

其中，$\xi\in(a,b)$且依赖与x 。

需要注意以下几点

1. 余项表达式仅当$f^{n+1}(x)$存在时才能应用，且是唯一的
2. ξ在 (a,b)内的具体位置通常不能给出，因此， $R(x)$不能准确地计算出来，只能估计它的值。
3. n次插值多项式顿时次数不高于n 次的多项式完全精确。
4. 依靠增加节点不一定能减少误差。
5. 插值多项式一般仅用来估计插值区间内点的函数。计算插值区间外点的函数值时，误差可能会很大。

拉格朗日插值的优点：含义直观，形式对称。
缺点：计算量大。

1.3 逐次线性插值法

使用拉格朗日插值多项式$L_n(x)$ 计算函数近似值时，如需增加插值节点，那么原来算出的数据均不能利用，必须重新计算。

现令$I_{i_1,i_2,\cdots,i_n}(x)$表示函数 $f(x)$ 关于节点$x_{i_1},\cdots,x_{i_n}$ 的$n-1$ 次插值多项式，$I_{i_k}(x)$ 是零次多项式，记$I_{i_k}(x)=f(x_{i_k})$，一般情况，两个k 次插值多项式可通过线性插值得到$k+1$ 次插值多项式

$$I_{0,\cdots,k,l}=I_{0,\cdots,k}(x)+\frac{I_{0,\cdots,k-1,l}-I_{0,\cdots,k}(x)}{x_l-x_k}(x-x_k) \tag{1.3.1}$$

这是关于节点$x_0,\cdots,x_k,x_l$ 的插值多项式，显然

$$I_{0,\cdots,k,l}(x_i)=I_{0,\cdots,k}(x_i)=f(x_i)$$

对于$i=0,\cdots,k-1$ 成立。当$x=x_k$ 时，有

$$I_{0,\cdots,k,l}(x_k)=I_{0,\cdots,k}(x_k)=f(x_k)$$

当$x=x_l$ 时，有

$$I_{0,\cdots,k,l}(x_l)=I_{0,\cdots,k}(x_l)+\frac{f(x_l)-I_{0,\cdots,k(x_l)}}{x_l-x_k}(x_l-x_k)=f(x_l)$$

证明式1.3.1的插值多项式满足插值条件，称位Aitken逐次线性插值公式。

式1.3.1 也可以改写为

$$I_{0,\cdots,k,k+1}=I_{0,\cdots,k}(x)+\frac{I_{1,\cdots,k,k+1}-I_{0,\cdots,k}(x)}{x_{k+1}-x_0}(x-x_0) \tag{1.3.2}$$

此时的误差为

$$\frac{I_{0,\cdots,k}-I_{0,\cdots,k-1,k+1}}{x_k-x_{k+1}}(x-x_k)$$

1.4 Newton插值多项式

逐次线性插值法的问题在于没有一个固定的表达式。

1.4.1 差商

定义 3 称$f[x_0,x_k]=\frac{f(x_k)-f(x_0)}{x_k-x_0}$为函数$f(x)$ 关于点x_0,x_k 的一阶差商，称

$$f[x_0,x_1,x_k]=\frac{f[x_0,x_k]-f[x_0,x_1]}{x_k-x_1}$$

为$f(x)$ 关于点x_0,x_1,x_k 的二阶差商，一般地，称

$$f[x_0,x_1,\cdots,x_k]=\frac{f[x_0,x_1,\cdots,x_{k-2},x_k]-f[x_0,x_1,\cdots,x_{k-1}]}{x_k-x_{k-1}} \tag{1.4.1}$$

为$f(x)$ 的k 阶差商。

差商有如下基本性质

1. k 阶差商可表示为函数值$f(x_0),f(x_1),\cdots,f(x_k)$的线性组合，即

$$f[x_0,x_1,\cdots,x_k]=\sum_{j=0}^{k}\frac{f(x_j)}{(x_j-x_0)\cdots(x_j-x_{j-1})(x_j-x_{j+1})\cdots(x_j-x_k)} \tag{1.4.2}$$

2. 由性质(1) 和式1.4.1 可得

$$f[x_0,x_1,\cdots,x_k]=\frac{f[x_1,\cdots,x_k]-f[x_0,x_1,\cdots,x_{k-1}]}{x_k-x_0} \tag{1.4.3}$$

3. 若$f(x)$ 在$[a,b]$ 上存在n 阶导数，且节点$x_0,x_1,\cdots,x_n\in[a,b]$，则$n$ 阶差商于导数的关系为

$$f[x_0,x_1,\cdots,x_n]=\frac{f^{(n)}(\xi)}{n!},\xi\in[a,b] \tag{1.4.4}$$

1.4.2 Newton插值公式

根据差商定义，把x 看作$[a,b]$ 上的一点，可得

$$\begin{aligned}
f(x)&=f(x_0)+f[x,x_0](x-x_0)\\
f[x,x_0]&=f[x_0,x_1]+f[x,x_0,x_1](x-x_1)\\
&\vdots\\
f[x,x_0,\cdots,x_{n-1}]&=f[x_0,x_1,\cdots,x_n]+f[x,x_0,\cdots,x_n](x-x_n)
\end{aligned}$$

后式带入前式可得

$$\begin{aligned} f(x) &= f(x_0) + f[x_0, x_1](x - x_0) + f[x_0, x_1, x_2](x - x_0)(x - x_1) + \cdots \\ &\quad + f[x_0, x_1, \cdots, x_n](x - x_0) \cdots (x - x_{n-1}) + f[x, x_0, \cdots, x_n]\omega_{n+1}(x) \\ &= N_n(x) + R_n(x) \end{aligned}$$

其中

$$\begin{aligned} N_n(x) &= f(x_0) + f[x_0, x_1](x - x_0) + f[x_0, x_1, x_2](x - x_0)(x - x_1) + \cdots \\ &\quad + f[x_0, x_1, \cdots, x_n](x - x_0) \cdots (x - x_{n-1}) \\ R_n(x) &= f(x) - N_n(x_0) = f[x, x_0, \cdots, x_n]\omega_{n+1}(x) \end{aligned} \tag{1.4.5}$$

显然$N_n(x)$ 满足插值条件，且次数不超过n，它式形如$P_n(x) = a_0 + a_1(x - x_0) + \cdots + a_n(x - x_0) \cdots (x - x_{n-1})$的多项式，其系数为

$$a_k = f[x_0, x_1, \cdots, x_k]$$

称$N_n(x)$ 为Newton差商插值多项式。

Newton插值多项式比Lagrange插值多项式节省计算量，便于程序设计。

1.4.3 差分和等距节点插值公式

在实际情况中，可能遇到等距节点的情形，这时插值公式可以进一步简化。

设函数$y = f(x)$ 在等距节点$x_k = x_0 + kh(k = 0, 1, \cdots, n$ 上的值$f_k = f(x_k)$ 为已知，这里h 为常数称为步长。

定义 4 *偏差*

$$\begin{aligned} \Delta f_k &= f_{k+1} - f_k, \\ \nabla f_k &= f_k - f_{k-1}, \\ \delta f_k &= f(x_k + \frac{h}{2}) - f(x_k - \frac{h}{2}) = f_{k+\frac{1}{2}} - f_{k-\frac{1}{2}} \end{aligned}$$

分别称为$f(x)$ 在x_k 处以h为步长的向前差分、向后差分和中心差分。符号Δ, ∇, δ分别称为向前差分算子、向后差分算子和中心差分算子。

利用一阶差分可以定义二阶差分为

$$\Delta^2 f_k = \Delta f_{k+1} - \Delta f_k = f_{k+2} - 2f_{k+1} + f_k$$

一般地，可以定义m 阶差分为

$$\Delta^m = \Delta^{m-1} f_{k+1} - \Delta^{m-1} f_k, \nabla^m f_k = \nabla^{m-1} f_k - \nabla^{m-1} f_{k-1}$$

常用的算子符号由不变算子I 及位移算子E 定义如下

$$If_k = f_k, Ef_k = f_{k+1}$$

于是，可得

$$\Delta = E - I, \nabla = I - E^{-1}, \delta = E^{\frac{1}{2}} - E^{-\frac{1}{2}}$$

根据差分和算子定义，可得到以下性质

1. 各阶差分可用函数值表示

$$\Delta^n f_k = (E-I)^n f_k = \sum_{j=0}^{n} (-1)^j \binom{n}{j} E^{n-j} f_k = \sum_{j=0}^{n} (-1)^j \binom{n}{j} f_{k+n-j}$$

$$\nabla^n f_k = (I-E^{-1})^n f_k = \sum_{j=0}^{n} (-1)^{n-j} \binom{n}{j} E^{j-n} f_k = \sum_{j=0}^{n} (-1)^{n-j} \binom{n}{j} f_{k+j-n}$$

2. 可用各阶差分表示函数值，如，可用向前差分表示f_{n+k} ，即

$$f_{n+k} = E^n f_k = (I+\Delta)^n f_k = \sum_{j=0}^{n} \binom{n}{j} \Delta^j f_k$$

3. 差商和差分有以下关系，对于向前差分有

$$f[x_k, x_{k+1}, \cdots, x_{k+m}] = \frac{1}{m!} \frac{1}{h^m} \Delta^m f_k \tag{1.4.6}$$

对于向后差分有

$$f[x_k, x_{k-1}, \cdots, x_{k-m}] = \frac{1}{m!} \frac{1}{h^m} \nabla^m f_k \tag{1.4.7}$$

将Newton差商插值多项式中各阶差商用相应差分代替，可以得到各种形式的等距节点插值公式。

如果节点$x_k = x_0 + kh$ ，要计算x_0 附近点x 的函数$f(x)$ 的值，可令$x = x_0 + th, 0 \le t \le 1$ ，于是

$$\omega_{k+1}(x) = \prod_{j=0}^{k} (x - x_j) = t(t-1)\cdots(t-k)h^{k+1}$$

将上式与式1.4.6代入式1.4.5中有

$$N_n(x_0+th) = f_0 + t\Delta f_0 + \frac{t(t-1)}{2!}\Delta^2 f_0 + \cdots + \frac{t(t-1)\cdots(t-n+1)}{n!}\Delta^n f_0 \tag{1.4.8}$$

上式称为Newton前插公式，余项为

$$R_n(x)=\frac{t(t-1)\cdots(t-n)}{(n+1)!}h^{n+1}f^{(n+1)}(\xi),\xi\in(x_0,x_n)$$

若要求表示函数在x_n 附近的值，有

$$N_n(x_n+th)=f_n+t\nabla f_n+\frac{t(t-1)}{2!}\nabla^2 f_n+\cdots+\frac{t(t-1)\cdots(t-n+1)}{n!}\nabla^n f_n \tag{1.4.9}$$

称上式为Newton后插公式，余项为

$$R_n(x)=\frac{t(t+1)\cdots(t+n)}{(n+1)!}h^{n+1}f^{(n+1)}(\xi),\xi\in(x_0,x_n)$$

1.5 Hermite插值多项式

一些实际问题不但要求在节点上函数值相等，而且还要求它的导数值相等，甚至要求高阶导数值也相等，满足这种要求的插值多项式就是Hermite插值多项式。

假设函数值和导数值个数相等，设在节点$a\le x_0<x_1<\cdots<x_n\le b$ 上，$y_i=f(x_i),m_j=f'(x_j)$ 要求插值多项式$H(x)$ 满足条件

$$H(x_j)=y_j,H'(x_j)=m_j \tag{1.5.1}$$

这里给出$2n+2$ 个条件，可唯一确定一个次数不超过$2n+1$ 的多项式，其形式为

$$H_{2n+1}(x)=a_0+a_1x+\cdots+a_{2n+1}x^{2n+1}$$

使用Lagrange插值多项式的基函数方法，先求插值基函数$\alpha_j(x)$和 $\beta_j(x)$ ，共有$2n+2$ 个，每个都是$2n+1$ 次多项式，且满足条件

$$\begin{cases}\alpha_j(x_k)=\delta_{jk}\\ \alpha_j'(x_k)=0\\ \beta_j(x_k)=0\\ \beta_j'(x_k)=\delta_{jk}\end{cases} \tag{1.5.2}$$

于是，满足插值条件的插值多项式可写成插值基函数的形式，即

$$H_{2n+1}(x)=\sum_{j=0}^{n}[y_j\alpha_j(x)+m_j\beta_j(x)] \tag{1.5.3}$$

为求基函数$\alpha_j(x)$ 和$\beta_j(x)$ 的表达式，可利用Lagrange插值基函数$l_j(x)$ 。令

$$\alpha_j(x)=(ax+b)l_j^2(x)$$

由条件1.5.2 解得

$$\begin{cases} a = -2l_j'(x_j) \\ b = 1 + 2x_j l_j'(x_j) \end{cases}$$

于是

$$\alpha_j(x) = [1 + 2(x - x_j) \sum_{k=0, k\neq j}^{n} \frac{1}{x_k - x_j}] l_j^2(x)$$

同理，可解得

$$\beta_j(x) = (x - x_j) l_j^2(x)$$

若$f(x)$ 在(a, b) 内的$2n + 2$ 阶导数存在，则其插值余项为

$$R(x) = f(x) - H_{2n+1}(x) = \frac{f^{(2n+2)}(\xi)}{(2n+2)!} \omega_{n+1}^2(x) \tag{1.5.4}$$

1.6 分段低次插值

1.6.1 分段线性插值

高次插值插值多项式不一定好，因为不一定会收敛到$f(x)$，因而，通常不用高次插值，而用低次插值。

分段线性插值就是将插值点用折线段连接起来逼近$f(x)$。设已知节点$a = x_0 < x_1 < \cdots < x_n = b$ 上的函数值$f_0, f_1, \cdots, f_n$，记$h_k = x_{k+1} - x_k, h = \max h_k$，称$I_h(x)$为分段线性插值函数，若满足

1. 记$I_h(x) \in C[a, b]$
2. $I_h(x_k) = f_k (k = 0, 1, \cdots, n)$
3. $I_h(x)$ 在每个区间$[x_k, x_{k+1}]$ 上是线性函数。

则由定义可知，$I_h(x)$ 在每个小区间$[x_k, x_{k+1}]$上可表示为

$$I_h(x) = \frac{x - x_{k+1}}{x_k - x_{k+1}} f_k + \frac{x - x_k}{x_{k+1} - x_k} f_{k+1} \tag{1.6.1}$$

若用插值基函数表示，则$I_h(x)$ 在整个区间$[a, b]$ 上可表示为

$$I_h(x) = \sum_{j=0}^{n} f_j l_j(x) \tag{1.6.2}$$

其中基函数$l_j(x)$ 满足条件$l_j(x_k) = \delta_{jk}$，其形式为

$$l_j(x) = \begin{cases} \frac{x - x_{j-1}}{x_j - x_{j-1}}, x_{j-1} \leq x \leq x_j \\ \frac{x - x_{j+1}}{x_j - x_{j+1}}, x_j \leq x \leq x_{j+1} \\ 0, x \in [a, b], x \notin [x_{j-1}, x_{j+1}] \end{cases} \tag{1.6.3}$$

分段线性插值基函数$l_j(x)$ 只在x_j 附近不为零，在其他定分均为零，这种性质称为局部非零性质。

当$x \in [x_k, x_{k+1}]$ 时

$$l = \sum_{j=0}^{n} l_j(x) = l_k(x) + l_{k+1}(x)$$

$$f(x) = [l_k(x) + l_{k+1}(x)]f(x)$$

此时，有

$$I_h(x) = f_k l_k(x) + f_{k+1} l_{k+1}(x)$$

1.6.2 分段Hermit插值

分段线性插值函数$I_h(x)$ 的导数是间断的。若在节点x_k 上还给出导数值$f'_k = m_k$，就可以构造一个导数连续的分段插值函数$I_h(x)$ ，它满足以下条件

1. $I_h(x) \in C^1[a, b]$
2. $I_h(x_k) = f_k, I'_h(x_k) = f'_k$
3. $I_h(x_k)$ 在每个小区间$[x_k, x_{k+1}]$ 上是三次多项式

若在整个区间$[a, b]$上定义一组分段三次插值基函数$\alpha_j(x)$和$\beta_j(x)$，则$I_h(x)$可表示为

$$I_h(x) = \sum_{j=0}^{n} [f_j \alpha_j(x) + f'_j \beta_j(x)]$$

其中，$\alpha_j(x), \beta_j(x)$ 的表达式分别为

$$\alpha_j(x) = \begin{cases} \left(\frac{x-x_{j-1}}{x_j-x_{j-1}}\right)^2 \left(1 + 2\frac{x-x_j}{x_{j-1}-x_j}\right), x_{j-1} \le x \le x_j \\ \left(\frac{x-x_{j+1}}{x_j-x_{j+1}}\right)^2 \left(1 + 2\frac{x-x_j}{x_{j+1}-x_j}\right), x_j \le x \le x_{j+1} \\ 0 \end{cases}$$

$$\beta_j(x) = \begin{cases} \left(\frac{x-x_{j-1}}{x_j-x_{j-1}}\right)^2 (x - x_j), x_{j-1} \le x \le x_j \\ \left(\frac{x-x_{j+1}}{x_j-x_{j+1}}\right)^2 (x - x_j), x_j \le x \le x_{j+1} \\ 0 \end{cases}$$

对$\alpha_j(x), \beta_j(x)$ 的估计结果为

$$0 \le \alpha_j(x) \le 1$$

$$\begin{cases} |\beta_k(x)| \le \frac{4}{27} h_k \\ |\beta_{k+1}(x)| \le \frac{4}{27} h_k \end{cases}$$

1.7 三次样条插值

分段低次插值函数光滑性较差，无法满足二阶连续性。

定义 5 *若函数*$S(x) \in C^2[a,b]$，*且每个小区间*$[x_j, x_{j+1}]$ *上是三次多项式，其他*$a = x_0 < x_1 < \cdots < x_n = b$ *是给定节点，则称*$S(x)$ *是节点*$x_0, x_1, \cdots, x_n$ *上的三次样条函数。若在节点*x_j *上给定函数值*$y_j = f(x_j)$ *且*

$$S(x_j) = y_j$$

成立，则称$S(x)$ *为三次样条插值函数*

在给定范围$[a,b]$ 上共有n 个小区间，每个小区间是三次样条函数，要确定4 个待定系数，因此，共有$4n$ 个待定系数。 $S(x)$在区间$[a,b]$ 上二阶导数连续，在节点$x_j (j = 1, 2, \cdots, n-1)$ 处应该满足连续性条件

$$\begin{cases} S(x_j - 0) = S(x_j + 0) \\ S'(x_j - 0) = S'(x_j + 0) \\ S''(x_j - 0) = S''(x_j + 0) \end{cases}$$

共有$3n-3$个条件。加上$n+1$ 个插值条件，共有$4n-2$ 个条件。

因此，还需要两个条件才能确定$S(x)$ 。通常可在区间端点上各加一个条件（称为边界条件），边界条件根据实际问题，常见的有以下三种。

1. 已知两端的一阶导数值

$$\begin{cases} S'(x_0) = f'_0 \\ S'(x_n) = f'_n \end{cases} \tag{1.7.1}$$

2. 两端的二阶导数已知

$$\begin{cases} S''(x_0) = f''_0 \\ S''(x_n) = f''_n \end{cases} \tag{1.7.2}$$

3. 当$f(x)$ 是以$x - x_0$ 为周期的周期函数时，则要求$S(x)$ 也是周期函数，这是边界条件应满足

$$\begin{cases} S(x_0 + 0) = S(x_n - 0) \\ S'(x_0 + 0) = S'(x_n - 0) \\ S''(x_0 + 0) = S''(x_n - 0) \end{cases} \tag{1.7.3}$$

这样确定的样条函数$S(x)$ 称为周期样条函数。

1.7.1 三转角方程

现在需要构造满足条件的三次样条函数$S(x)$ 的表达式。若假定$S'(x)$ 在节点x_j 处的值为$S'(x_j)=m_j$，则有分段三次Hermite插值可以得到

$$S(x)=\sum_{j=0}^{n}[y_i\alpha_j(x)+m_j\beta_j(x)]$$

上式的问题在于表达式中的m_j 是未知的，可以利用二阶导数和边界条件来求出m_j。

考虑$S(x)$ 在$[x_j,x_{j+1}]$ 上的表达式，对$S(x)$ 求二阶导数可得

$$S''(x_j+0)=-\frac{4}{h_j}m_j-\frac{2}{h_j}m_{j+1}+\frac{6}{h_j^2}(y_{j+1}-y_j)$$

同理，$S''(x)$ 在区间$[x_{j-1},x_j]$ 上的表达式有

$$S''(x_j-0)=\frac{2}{h_{j-1}}m_{j-1}+\frac{4}{h_{j-1}}m_j-\frac{6}{h_{j-1}^2}(y_j-y_{j-1})$$

有条件$S''(x_j+0)=S''(x_j-0)$ 可得

$$\frac{1}{h_{j-1}}m_{j-1}+2\left(\frac{1}{h_{j-1}}+\frac{1}{h_j}\right)m_j+\frac{1}{h_j}m_{j+1}=3\left(\frac{y_{i+1}-y_i}{h_j^2}+\frac{y_j-y_{j-1}}{h_{j-1}^2}\right)$$

上式可化简为

$$\lambda_j m_{j-1}+2m_j+\mu_j m_{j+1}=g_j$$

其中，$h_j=x_{j+1}-x_j,\lambda_j=\frac{h_j}{h_{j-1}+h_j},\mu_j=\frac{h_{j-1}}{h_{j-1}+h_j},g_j=3(\lambda_j f[x_{j-1},x_j]+\mu_j f[x_j,x_{j+1}])$

上述方程式关于未知数$m_0,m_1,\cdots,m_n$ 得$n-1$ 个方程。

当边界条件为1.7.1，方程只含有$m_1,\cdots,m_{n-1}$ 得$n-1$ 个方程，写成矩阵的形式为

$$\begin{pmatrix}2&\mu_1&0&\cdots&\cdots&0\\ \lambda_2&2&\mu_2&\ddots&&\vdots\\ 0&\lambda_3&2&\mu_3&\ddots&\vdots\\ \vdots&\ddots&\ddots&\ddots&\ddots&0\\ 0&&\ddots&\lambda_{n-2}&2&\mu_{n-2}\\ 0&\cdots&\cdots&0&\lambda_{n-1}&2\end{pmatrix}\begin{pmatrix}m_1\\ m_2\\ m_3\\ \vdots\\ m_{n-2}\\ m_{n-1}\end{pmatrix}=\begin{pmatrix}g_1-\lambda_1 f_0'\\ g_2\\ g_3\\ \vdots\\ g_{n-2}\\ g_{n-1}-\mu_{n-1}f_n'\end{pmatrix}$$

当边界条件为1.7.2 时，方程的矩阵形式为

$$\begin{pmatrix} 2 & 1 & 0 & \cdots & \cdots & 0 \\ \lambda_1 & 2 & \mu_1 & \ddots & & \vdots \\ 0 & \lambda_2 & 2 & \mu_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & & \ddots & \lambda_{n-1} & 2 & \mu_{n-1} \\ 0 & \cdots & \cdots & 0 & 1 & 2 \end{pmatrix} \begin{pmatrix} m_0 \\ m_1 \\ m_2 \\ \vdots \\ m_{n-1} \\ m_n \end{pmatrix} = \begin{pmatrix} g_0 \\ g_1 \\ g_2 \\ \vdots \\ g_{n-1} \\ g_n \end{pmatrix}$$

其中，$g_0 = 3f[x_0, x_1] - \frac{h_0}{2} f_0'', g_n = 3f[x_{n-1}, x_n] + \frac{h_{n-1}}{2} f_n''$。

当边界条件为周期性条件1.7.3时，则方程的矩阵形式为

$$\begin{pmatrix} 2 & \mu_1 & 0 & \cdots & 0 \\ \lambda_2 & 2 & \mu_2 & \ddots & \vdots \\ 0 & \lambda_3 & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & 2 & \mu_{n-1} \\ 0 & \cdots & 0 & \lambda_n & 2 \end{pmatrix} \begin{pmatrix} m_1 \\ m_2 \\ \vdots \\ m_{n-1} \\ m_n \end{pmatrix} = \begin{pmatrix} g_1 \\ g_2 \\ \vdots \\ g_{n-1} \\ g_n \end{pmatrix}$$

上述系数矩阵具有强对角优势，方程组都有唯一解。

三弯矩方程根据二阶导数建立方程组，也可以同样求解插值函数$S(x)$

计算$S(x)$ 的算法步骤为

1. 输入初始数据x_j, y_j及f_0', f_n' 和n
2. j 从0 到$n-1$ 计算$h_j = x_{j+1} - x_j$ 及$f[x_j, x_{j+1}]$
3. j 从1 到$n-1$ 计算λ_j, μ_j, g_j
4. 用追赶法解方程，求出 m_j
5. 计算 $S(x)$ 的系数或计算$S(x)$ 在若干点上的值，并打印结果

2 函数逼近与计算

2.1 引言和预备知识

当用计算机计算一些函数时，若把函数表存入内存进行查表，则占用单元太多，不如直接用公式计算方便，因此，希望求出便于计算且计算量少的公式近似已知函数$f(x)$。

要求在给定精度下求计算次数最少的近似公式，就是函数逼近与计算要解决的问题。问题可叙述为，对于函数类A中给定的函数$f(x)$ ，要求在

另一类较简单的便于计算的函数类B中，求函数$P(x) \in B \subset A$，使得$P(x)$与$f(x)$之差在某种度量意义下最小。函数类A通常是区间$[a,b]$上的连续函数，记作$C[a,b]$；函数类B通常时代数多项式、分式有理函数或三角多项式。

度量的标准由两种

1. 一致逼近或均匀逼近

$$\|f(x)-P(x)\|_\infty = \max_{a\le x\le b}|f(x)-P(x)|$$

2. 均方逼近或平方逼近

$$\|f(x)-P(x)\|_2 = \sqrt{\int_a^b [f(x)-P(x)]^2 \mathbf{d}x}$$

2.1.1 Weierstrass定理

定理 2 *设$f(x) \in C[a,b]$，则对于任何$\epsilon > 0$，总存在一个代数多项式$P(x)$使*

$$\|f(x)-P(x)\|_\infty < \epsilon$$

在$[a,b]$上已知成立。

2.1.2 连续空间

区间$[a,b]$上的所有实连续函数组成一个空间，记作$C[a,b]$。$f \in C[a,b]$的范数定义为

$$\|f\|_\infty = \max_{a\le x\le b}|f(x)|$$

$\|\cdot\|_\infty$称为$\infty-$范数。它满足范数$\|\cdot\|$的三个性质

1. $\|f\| \ge 0$ 当且仅当$f \equiv 0$ 时才有$\|f\| = 0$
2. $\|af\| = |a|\|f\|$对于任何$f \in C[a,b]$ 成立，a 为任意实数。
3. 对于任意$f, g \in C[a,b]$，有

$$\|f+g\| \le \|f\| + \|g\|$$

上式称为三角不等式。

与向量空间类似，当$f, g \in C[a,b]$ 时，定义f 与g 的距离为

$$D(f,g) = \|f-g\|_\infty$$

由性质三可得

$$D(f,g) \le D(f,h) + D(h,g)$$

$$|\|f\|_\infty - \|g\|_\infty| \le \|f-g\|_\infty$$

2.2 最佳一致逼近多项式

记次数不大于n 的多项式集合为H_n ，显然$H_n \subset C[a,b]$。又记$H_n = span\{1, x, \cdots, x^n\}$，其中$1, x, \cdots, x^n$ 是$[a,b]$ 上的一组线性无关的函数组，是H_n 中的一组基，H_n 中的元素$P_n(x)$ 可表示为

$$P_n(x) = a_0 = a_1 x + \cdots + a_n x^n$$

要在H_n 中求$P_n^\star(x)$ 逼近$f(x) \in [a,b]$，使其误差

$$\max_{a\le x\le b} |f(x) - P_n^\star(x)| = \min_{P_n\in H_n} \max_{a\le x\le b} |f(x) - P_n(x)|$$

这就是最佳一致逼近或Chebyshev逼近问题。

定义 6 $P_n(x) \in H_n, f(x) \in C[a,b]$ ，称

$$\Delta(f, P_n) = \|f - P_n\|_\infty = \max_{a\le x\le b} |f(x) - P_n(x)|$$

为$f(x)$ 与$P_n(x)$ 在$[a,b]$ 上的偏差。

$\Delta(f, P_n)$的全体组成一个集合，记作$\{\Delta(f, P_n)\}$，它有下界0。若记集合的下确界为

$$E_n = \inf_{P_n\in H_n} \{\Delta(f, P_n)\} = \inf_{P_n\in H_n} \max_{a\le x\le b} |f(x) - P_n(x)|$$

则称E_n 为$f(x)$在$[a,b]$ 上的最小偏差。

定义 7 假定$f(x) \in C[a,b]$，若存在

$$P_n^\star(x) \in H_n, \Delta(f, P_n^\star) = E_n$$

则称$P_n^\star(x)$ 是$f(x)$ 在$[a,b]$ 上的最佳一致逼近多项式或最小偏差逼近多项式，简称最佳逼近多项式。

定理 3 若$f(x) \in C[a,b]$，则总存在多项式$P_n^\star(x) \in H_n$ ，使得

$$\|f(x) - P_n^\star(x)\|_\infty = E_n$$

2.2.1 Chebyshev定理

定义 8 设$f(x) \in C[a,b], P(x) \in H_n$，若在$x = x_0$ 上有

$$|P(x_0) - f(x_0)| = \max_{a\le x\le b} |P(x) - f(x)| = \mu$$

则称x_0 是$P(x)$ 的偏差点

若$P(x_0)-f(x_0)=\mu$，则称x_0 为正偏差点。
若$P(x_0)-f(x_0)=-\mu$，则称x_0 为负偏差点。
由于$P(x)-f(x)$是连续函数，则$P(x)$的偏差点必定存在。

定理 4 *若$P(x)\in H_n$ 是$f(x)\in C[a,b]$ 的最佳逼近多项式，则$P(x)$ 同时存在正、负偏差点*

这个定理可以使用反证法证明。

从几何上看，表明的多项式在曲线$y=f(x)+E_n$和 $y=f(x)+E_n$之间，且至少与它们各接触一次。若与其中一个函数不接触，则可稍微平移$P_n(x)$ 得到更小的E_n。

定理 5 *$P(x)\in H_n$ 是$f(x)\in C[a,b]$ 的最佳逼近多项式的充分必要条件是$P(x)$ 在$[a,b]$ 上至少有$n+2$各轮流为正、负的偏差点，即有$n+2$ 个点$a\le x_1<x_2<\cdots<x_{n+2}\le b$ 使得*

$$P(x_k)-f(x_k)=(-1)^{-k}\sigma\|P(x)-f(x)\|_\infty,\sigma=\pm1$$

这样的点组称为Chebyshev交错点组。

推论 1 *若$f(x)\in C[a,b]$ ，则在H_n 中存在唯一的最佳逼近多项式*

推论 2 *若$f(x)\in C[a,b]$，则其最佳逼近多项式$P_n^\star\in H_h$ 就是$f(x)$ 的一个Lagrange插值多项式。*

2.2.2 最佳一次逼近多项式

上面的定理值给出了最佳逼近多项式的特性，但是如何求$P(x)$ 却是十分困难的。这里讨论当$n=1$ 的情形。

设$f(x)\in C^2[a,b]$ ，且$f''(x)$ 在(a,b) 内不变号，要求最佳一次逼近多项式$P_1(x)=a_0+a_1x$ 。根据上述定理可知，至少存在三个点$a\le x_1<x_2<x_3\le b$ 使得

$$P_1(x_k)-f(x_k)=(-1)^k\sigma\max_{a\le x\le b}|P_1(x)-f(x)|$$

由于$f''(x)$ 在$[a,b]$ 上不变号，故$f'(x)$ 单调，因此$f'(x)-a_1$在(a,b) 内只有一个零点，记作x_2，因此有$P_1'(x_2)-f'(x_2)=a_1-f'(x_2)=0$，即$f'(x_2)=a_1$。另外两个偏差点必定在区间的端点，即$x_0=a,x_1=b$，且满足

$$P_1(a)-f(a)=P_1(b)-f(b)=-[P_1(x_2)-f(x_2)]$$

因此可以解得

$$\begin{cases} a_0 = \frac{f(a)-f(x_2)}{2} - \frac{f(b)-f(a)}{b-a}\frac{a+x_2}{2} \\ a_1 = \frac{f(b)-f(a)}{b-a} = f'(x_2) \end{cases}$$

因此最佳超出逼近多项式$P_1(x)$ 为

$$P_1(x) = \frac{f(a)-f(x_2)}{2} - \frac{f(b)-f(a)}{b-a}\frac{a+x_2}{2} + \frac{f(b)-f(a)}{b-a}x$$

2.3 最佳平方逼近

2.3.1 内积空间

定义 9 设在区间(a,b) 内，非负函数$\rho(x)$ 满足以下条件，就称$\rho(x)$ 为区间(a,b) 内的权函数。

1. $\int_a^b |x|^n \rho(x)\mathbf{d}x, (n=0,1,\cdots)$ 存在。
2. 对于非负的连续函数$g(x)$，若

$$\int_a^b g(x)\rho(x)\mathbf{d}x = 0$$

则在(a,b) 内$g(x) \equiv 0$。即$\rho(x) \neq 0$

定义 10 设$f(x), g(x) \in C[a,b]$，$\rho(x)$ 是$[a,b]$ 上的权函数，积分

$$(f,g) = \int_a^b \rho(x)f(x)g(x)\mathbf{d}x$$

称为函数$f(x)$ 与$g(x)$ 在$[a,b]$ 上的内积。

内积满足以下四条公理

1. $(f,g) = (g,f)$
2. $(cf,g) = c(f,g)$，c 为常数
3. $(f_1+f_2,g) = (f_1,g)+(f_2,g)$
4. $(f,f) \geq 0$，当且仅当 $f=0$ 时$(f,f)=0$

满足内积定义的函数空间称为内积空间。

定义 11 $f(x) \in C[a,b]$，量

$$\|f\|_2 = \sqrt{\int_a^b \rho(x)f^2(x)\mathbf{d}x} = \sqrt{(f,f)}$$

称为$f(x)$的$Euclid$范数

它同样满足范数的三条性质。

定理 6 对于任何$f, g \in C[a,b]$ 下列结论成立

1. $|(f,g)| \leq \|f\|_2 \|g\|_2$（*Cauchy-Schwarz*不等式）
2. $\|f+g\|_2 \leq \|f\|_2 + \|g\|_2$ （三角不等式）
3. $\|f+g\|_2^2 + \|f-g\|_2^2 = 2(\|f\|_2^2 + \|g\|_2^2)$ （平行四边形定律）

定义 12 若$f(x), g(x) \in C[a,b]$ 满足

$$(f,g) = \int_a^b \rho(x) f(x) g(x) \mathbf{d}x = 0$$

则称f 与g 在$[a,b]$ 上带权$\rho(x)$ 正交

若函数族$\phi_0(x), \phi_1(x), \cdots$满足关系，

$$(\phi_j, \phi_k) = \int_a^b \rho(x)\phi_j(x)\phi_k(x)\mathbf{d}x = \begin{cases} 0, j \neq k \\ A_k > 0, j = k \end{cases}$$

则称$\{\phi_k\}$ 是$[a,b]$ 上带权$\rho(x)$ 的正交函数族。若$A \equiv 1$ 则称$\{\phi_k\}$ 为标准正交函数族。

如三角函数族$1, \cos x, \sin x, \cos 2x, \sin 2x, \cdots$ 就是区间$[-\pi, \pi]$ 上的正交函数族。

定义 13 设$\phi_0(x), \phi_1(x), \cdots, \phi_{n-1}(x)$ 在$[a,b]$ 上连续，如果

$$a_0\phi_0(x) + a_1\phi_1(x) + \cdots + a_{n-1}\phi_{n-1}(x) = 0$$

当且仅当$a_0 = a_1 = \cdots = a_{n-1} = 0$ 时成立，则称$\phi_0, \phi_1, \cdots, \phi_{n-1}$ 在$[a,b]$上时线性无关的。

若函数族$\{\phi_k\}(k = 0, 1, \cdots)$中的任何有限个$\phi_k$ 线性无关，则称$\{\phi_k\}$ 为线性无关函数族。

若$\phi_0(x), \phi_1(x), \cdots, \phi_{n-1}(x)$是$[a,b]$ 上的线性无关函数，且$a_0, a_1, \cdots, a_{n-1}$是任意实数，则

$$S(x) = a_0\phi_0(x) + a_1\phi_1(x) + \cdots + a_{n-1}\phi_{n-1}(x)$$

的全体是$C[a,b]$ 中的一个子集，记作

$$\Phi = span\{\phi_0, \phi_1, \cdots, \phi_{n-1}\}$$

定义 14 $\phi_0(x), \phi_1(x), \cdots, \phi_{n-1}(x)$ 在$[a,b]$ 上线性无关的充分必要条件是它的$Cramer$行列式$G_{n-1} \neq 0$，其中

$$G_{n-1} = G(\phi_0, \phi_1, \cdots, \phi_{n-1}) = \begin{vmatrix} (\phi_0, \phi_0) & (\phi_0, \phi_1) & \cdots & (\phi_0, \phi_{n-1}) \\ (\phi_1, \phi_0) & (\phi_1, \phi_1) & \cdots & (\phi_1, \phi_{n-1}) \\ \vdots & \vdots & & \vdots \\ (\phi_{n-1}, \phi_0) & (\phi_{n-1}, \phi_1) & \cdots & (\phi_{n-1}, \phi_{n-1}) \end{vmatrix}$$

2.3.2 函数的最佳平方逼近

对于$f(x) \in C[a,b]$及$[a,b]$ 中的一个子集$\Phi = span\{\phi_0, \phi_1, \cdots, \phi_n\}$，若存在$S^\star(x) \in \Phi$，使得

$$\|f - S^\star\|_2^2 = \inf_{S \in \phi} \|f - S\|_2^2 = \inf_{S \in \phi} \int_a^b \rho(x)[f(x) - S(x)]^2 \mathbf{d}x$$

则称$S^\star(x)$ 是$f(x)$ 在子集$\Phi \subset C[a,b]$ 中的最佳平方逼近函数。

这个问题等价于求多元函数

$$I(a_0, a_1, \cdots, a_n) = \int_a^b \rho(x)[\sum_{j=0}^n a_j \phi_j(x) - f(x)]^2 \mathbf{d}x$$

的最小值。

I 是关于$a_0, a_1, \cdots, a_n$ 的二次函数，利用多元函数极致的必要条件$\frac{\partial I}{\partial a_k = 0}$，有

$$\frac{\partial I}{\partial a_k} = 2 \int_a^b \rho(x)[\sum_{j=0}^n a_j \phi_j(x) - f(x)]\phi_k(x) \mathbf{d}x$$

可以解得

$$\sum_{j=0}^n (\phi_k, \phi_j) a_j = (f, \phi_k)$$

这是关于$a_0, a_1, \cdots, a_n$ 的线性方程组，称为法方程。由于$\phi_0, \phi_1, \cdots, \phi_k$ 线性无关，所以系数行列是$G(\phi_0, \phi_1, \cdots, \phi_n) \neq 0$ ，所以方程族有唯一解$a_k = a_k^\star$，因此有

$$S^\star(x) = a_0^\star \phi_0(x) + \cdots + a_n^\star \phi_n(x)$$

若令$\delta = f(x) - S^\star(x)$ ，则平方误差为

$$\|\delta\|_2^2 = (f - S^\star, f - S^\star) = (f, f) - (S^\star, f) = \|f\|_2^2 - \sum_{k=0}^n a_k^\star (\phi_k, f)$$

取$\phi_k(x) = x^k, \rho(x) \equiv 1, f(x) \in C[0,1]$，即要在$H_n$ 中求n 次最佳平方逼近多项式。

此时，有

$$(\phi_j, \phi_k) = \int_0^1 x^{k+j} \mathbf{d}x = \frac{1}{k+j+1}$$

$$(f, \phi_k) = \int_0^1 f(x) x^k \mathbf{d}x \equiv d_k$$

用**H** 表示行列式$G_n = G(1, x^2, \cdots, x^n)$，则

$$H = \begin{pmatrix} 1 & \frac{1}{2} & \cdots & \frac{1}{n+1} \\ \frac{1}{2} & \frac{1}{3} & \cdots & \frac{1}{n+2} \\ \vdots & \vdots & & \vdots \\ \frac{1}{n+1} & \frac{1}{n+2} & \cdots & \frac{1}{2n+1} \end{pmatrix}$$

H 称为Hilbert矩阵，即$a = (a_0, a_1, \cdots, a_n)^T, d = (d_0, d_1, \cdots, d_n)^T$ 则方程

$$\mathbf{H}a = d$$

的解$a_k = a_k^\star$ 为所求

2.4 正交多项式

2.4.1 正交化手续

定义 15 设$g_n(x)$ 是首项系数$a_n \neq 0$ 的n 次多项式，如果多项式序列$g_0(x), g_1(x), \cdots$ 满足

$$(g_j, g_k) = \int_a^b \rho(x) g_j(x) g_k(x) \mathbf{d}x = \begin{cases} 0, & j \neq k \\ A_k > 0, & j = k \end{cases}$$

则称多项式序列$g_0(x), g_1(x), \cdots$ 在$[a, b]$ 上带权$\rho(x)$ 正交，并称$g_n(x)$ 是$[a, b]$ 上带权$\rho(x)$ 的n 次正交多项式。

当权函数$\rho(x)$ 及区间$[a, b]$ 给定以后，可以由线性无关的一组基$\{1, x, x^2, \cdots, x^n, \cdots\}$ 并利用正交化的方法构造出正交多项式

$$g_0(x) = 1, g_n(x) = x^n - \sum_{k=0}^{n-1} \frac{(x^n, g_k)}{(g_k, g_k)} g_k(x)$$

这样构造的正交多项式有以下性质

1. $g_n(x)$ 是最高项系数为1的n 次多项式
2. 任一n 次多项式$P_n \in H_n$ 均可表示为$g_0(x), g_1(x), \cdots, g_n(x)$ 的线性组合
3. 当$n \neq m$ 时，$(g_n, g_m) = 0$ 且$g_n(x)$ 与任一次数小于n 的多项式正交。

4. 有递推关系

$$g_{n+1}(x) = (x - \alpha_n)g_n(x) - \beta_n g_{n-1}(x)$$

其中$\alpha_n = \frac{(xg_n, g_n)}{(g_n, g_n)}, \beta_n = \frac{(g_n, g_n)}{(g_{n-1}, g_{n-1})}$

5. 设$g_0(x), g_1(x), \cdots$ 是在$[a, b]$ 上带权的$\rho(x)$ 的正交多项式序列，则$g_n(x)$ 的n 个根都是单重实根，且都在(a, b) 内

2.4.2 Legendre多项式

当区间为$[-1, 1]$ 、权函数$\rho(x) \equiv 1$ 时，由$\{1, x, x^2, \cdots, x^n, \cdots\}$ 正交化得到的多项式就称为Legendre多项式，并用$P_0(x), P_1(x), \cdots, P_n(x), \cdots$ 表示

$$P_0(x) = 1, P_n(x) = \frac{1}{2^n n!}\frac{\mathbf{d}^n}{\mathbf{d}x^n}\{(x^2 - 1)^n\}$$

$P_n(x)$ 的首项x^n 的系数$a_n = \frac{(2n)!}{2^n (n!)^2}$ 。

最高项系数为1的Legendre多项式为

$$\tilde{P}_n(x) = \frac{n!}{(2n)!}\frac{\mathbf{d}^n}{\mathbf{d}x^n}\{(x^2 - 1)^n\}$$

Legendre多项式的性质为

1. 正交性

$$\int_{-1}^{1} P_n(x)P_m(x)\mathbf{d}x = \begin{cases} 0, & m \neq n \\ \frac{2}{2n+1}, & m = n \end{cases}$$

2. 奇偶性

$$P_n(-x) = (-1)^n P_n(x)$$

3. 递推关系

$$(n + 1)P_{n+1}(x) = (2n + 1)xP_n(x) - nP_{n-1}(x)$$

其中，$P_0(x) = 1, P_1(x) = x$

4. 在所有最高项系数为1的n 次多项式中，Legendre多项式$\tilde{P}_n(x)$ 在$[-1, 1]$ 上与零的平方误差最小。

5. $P_n(x)$ 在区间$(-1, 1)$内有n 个不同的实零点

2.4.3 Chebyshev多项式

当权函数$\rho(x)=\frac{1}{\sqrt{1-x^2}}$，区间为$[-1,1]$时，由序列$\{1,x,x^2,\cdots,x^n,\cdots\}$正交化得到的正交多项式就是Chebyshev多项式，它可表示为

$$T_n(x)=\cos(n\arccos x)$$

若令$x=\cos\theta$，则

$$T_n(x)=\cos n\theta$$

Chebyshev多项式的性质为

1. 递推关系

   $$T_{n+1}(x)=2xT_n(x)-T_{n-1}(x)$$

   其中，$T_0(x)=1,T_1(x)=x$。由递推关系可知$T_n(x)$的最高项系数是2^{n-1}

2. $T_n(x)$ 对零的偏差最小。由以下定理

   定义 16 *在区间$[-1,1]$ 上所有最高项系数为1的一切n 次多项式中，$\omega_n(x)=\frac{1}{2^{n-1}}T_n(x)$ 与零的偏差最小，其偏差为$\frac{1}{2^{n-1}}$*

3. Chebyshev多项式$\{T_k(x)\}$ 在区间$[-1,1]$ 上带权$\rho(x)=\frac{1}{\sqrt{1-x^2}}$ 正交，且

   $$\int_{-1}^{1}\frac{T_n(x)T_m(x)}{\sqrt{1-x^2}}\mathbf{d}x=\begin{cases}0, & m\neq n\\ \frac{\pi}{2}, & m=n\neq 0\\ \pi, & m=n=0\end{cases}$$

4. $T_{2k}(x)$ 只含x 的偶次幂，$T_{2k+1}(x)$ 只含x 的奇次幂

5. $T_n(x)$ 在区间$[-1,1]$ 上由n 个零点$x_k=\cos\frac{2k-1}{2n}\pi$

   实际中要求x^n 用$T_0,T_1,\cdots,T_n$ 的线性组合表示，公式为

   $$x^n=2^{1-n}\sum_{k=0}^{\left[\frac{n}{2}\right]}\binom{n}{k}T_{n-2k}(x)$$

2.4.4 其他常用的正交多项式

1. 第二类Chebyshev多项式在区间$[-1,1]$ 上带权$\rho(x)=\sqrt{1-x^2}$ 的正交多项式称为第二类Chebyshev多项式，表达式为

   $$U_n(x)=\frac{\sin[(n+1)\arccos x]}{\sqrt{1-x^2}}$$

由$x=\cos\theta$ 可得

$$\int_{-1}^{1} U_n(x)U_m(x)\sqrt{1-x^2}\mathbf{d}x = \int_0^{\pi} \sin(n+1)\theta\sin(m+1)\theta\mathbf{d}\theta = \begin{cases} 0, & m\neq n \\ \frac{\pi}{2}, & m=n \end{cases}$$

可得到递推关系式为

$$U_{n+1}(x) = 2xU_n(x) - U_{n-1}(x)$$

其中，$U_0(x)=1, U_1(x)=2x$

2. Laguerre多项式在区间$[0,\infty)$ 上带权$\rho(x)=e^{-x}$ 的正交多项式称为Laguerre多项式，表达式为

$$L_n(x) = e^n\frac{\mathbf{d}^n}{\mathbf{d}x^n}(x^ne^{-x})$$

具有正交性

$$\int_0^{\infty} e^{-x}L_n(x)L_m(x)\mathbf{d}x = \begin{cases} 0, & m\neq n \\ (n!)^2, & m=n \end{cases}$$

递推关系为

$$L_{n+1}(x) = (1+2n-x)L_n(x) - n^2L_{n-1}(x)$$

其中，$L_0(x)=1, L_1(x)=1-x$

3. Hermite多项式在区间$(-\infty,\infty)$ 上带权$\rho(x)=e^{-x^2}$ 的正交多项式称为Hermite多项式，表达式为

$$H_n(x) = (-1)^n e^{x^2}\frac{\mathbf{d}^n}{\mathbf{d}x^n}(e^{-x^2})$$

具有正交性

$$\int_{-\infty}^{\infty} e^{-x^2}H_m(x)H_n(x)\mathbf{d}x = \begin{cases} 0, & m\neq n \\ 2^n n!\sqrt{\pi}, & m=n \end{cases}$$

递推关系

$$H_{n+1}(x) = 2xH_n(x) - 2nH_{n-1}(x)$$

其中，$H_0(x)=1, H_1(x)=2x$

2.5 函数按正交多项式展开

设$f(x) \in C[a,b]$ 用正交多项式$\{g_0(x), g_1(x), \cdots, g_n(x)\}$ 作基，求最佳平方逼近多项式

$$S_n(x) = a_0 g_0(x) + a_1 g_1(x) + \cdots + a_n g_n(x)$$

求得系数为$a_k = \frac{(f,g_k)}{(g_k,g_k)}$ 于是，$f(x)$ 的最佳平方逼近多项式为

$$S_n(x) = \sum_{k=0}^{n} \frac{(f,g_k)}{(g_k,g_k)} g_k(x)$$

均方误差为

$$\|\delta_n\|_2 = \|f - S_n\|_2 = \sqrt{\|f\|_2^2 - \sum_{k=0}^{n} \frac{(f,g_k)}{(g_k,g_k)}(f,g_k)}$$

若$f(x)$ 在$[a,b]$ 上按正交多项式$\{g_k(x)\}$ 展开，得到$f(x)$ 的展开式为

$$f(x) \sim \sum_{k=0}^{\infty} a_k g_k(x)$$

上式右端级数称为广义Fourier级数，系数a_k 称为广义Fourier系数。

任何$f(x) \in C[a,b]$ 均可展开为广义Fourier级数，其部分和$S_n(x)$ 是$f(x)$ 的最佳平方逼近。系数a_k 与n 无关。

级数可能不一致收敛于$f(x)$，但在满足一定条件下也可一致收敛到$f(x)$

2.6 曲线拟合的最小二乘法

2.7 Fourier逼近与快速Fourier变换

2.7.1 最佳平方三角逼近与三角插值

设$f(x)$ 是以2π 为周期的平方可积函数，用三角多项式

$$S_n(n) = \frac{1}{2}a_0 + a_1 \cos x + b_1 \sin x + \cdots + a_n \cos nx + b_n \sin nx$$

作最佳平方逼近函数。$f(x)$ 在$[0, 2\pi]$ 上的最小平方三角逼近多项是$S_n(x)$ 的系数是

$$\begin{cases} a_k = \frac{1}{\pi} \int_0^{2\pi} f(x) \cos kx \mathbf{d}x \\ b_k = \frac{1}{\pi} \int_0^{2\pi} f(x) \sin kx \mathbf{d}x \end{cases}$$

a_k, b_k 称为Fourier系数，函数$f(x)$ 按Fourier系数展开得到的级数

$$\frac{1}{2}a_0 + \sum_{k=1}^{\infty} (a_k \cos kx + b_k \sin kx)$$

称为Fourier级数。

当$f'(x)$ 在$[0,2\pi]$ 上分段连续，则上述级数一致收敛到$f(x)$

当$f(x)$ 只在给定的离散点集$\{x_j = \frac{2\pi}{N}j, j = 0,1,\cdots,N-1\}$ 上已知时，则可类似得到在离散点集上的正交性与相应的离散Fourier系数。

给定点集$\{x_j = \frac{2\pi j}{2m+1}\}$ ，对于任何$0 \le k,l \le m$，下式成立

$$\begin{cases} \sum_{j=0}^{2m} \sin lx_j \sin kx_j = \begin{cases} 0, & l \neq k, l = k = 0 \\ \frac{2m+1}{2}, & l = k \neq 0 \end{cases} \\ \sum_{j=0}^{2m} \cos lx_j \cos kx_j = \begin{cases} 0, & l \neq k \\ \frac{2m+1}{2}, & l = k \neq 0 \\ 2m+1, & l = k = 0 \end{cases} \\ \sum_{j=0}^{2m} \cos lx_j \sin kx_j = 0 \end{cases}$$

表明函数族$\{1, \cos x, \sin x, \cdots, \cos mx, \sin mx\}$ 在点集$\{x_j = \frac{2\pi j}{2m+1}\}$ 上正交。

若令$f_j = f(x_j)$，则$f(x)$ 的最小而成三角逼近为

$$S_n(x) = \frac{1}{2}a_0 + \sum_{k=1}^{n}(a_k \cos kx + b_k \sin kx)$$

其中

$$\begin{cases} a_k = \frac{2}{2m+1}\sum_{j=0}^{2m} f_j \cos\frac{2\pi jk}{2m+1} \\ b_k = \frac{2}{2m+1}\sum_{j=0}^{2m} f_j \sin\frac{2\pi jk}{2m+1} \end{cases}$$

假定$f(x)$ 是以2π 为周期的复函数，给定在N 个等分点$x_k = \frac{2\pi}{N}j$上的值$f_k = f(\frac{2\pi}{N}k)$，由于$e^{ijx} = \cos(jx) + i\sin(jx)$ ，函数族$\{1, e^{ix}, \cdots, e^{i(N-1)x}\}$ 在区间$[0,2\pi]$ 上是正交的。

将函数e^{ijx} 在等距点集$x_k = \frac{2\pi}{N}k$ 上的值e^{ijx_k} 组成的向量记作，$\phi_j = (1, e^{ij\frac{2\pi}{N}}, \cdots, e^{ij\frac{2\pi}{N}(N-1)})^T$。

N 个复向量$\phi_0, \phi_1, \cdots, \phi_{N-1}$具有正交性。因此，$f(x)$在 N 个点$\{x_j = \frac{2\pi}{N}j, j = 0,1,\cdots,N-1\}$ 上的最小而成Fourier逼近为

$$S(x) = \sum_{k=0}^{n-1} C_k e^{ikx}$$

其中

$$C_k = \frac{1}{N}\sum_{j=0}^{N-1} f_j e^{ikj\frac{2\pi}{N}}$$

当$n=N$ 时，$S(x)$ 为$f(x)$ 在点x_j 上的插值函数，可得

$$f_j=\sum_{k=0}^{N-1}C_k e^{ikj\frac{2\pi}{N}}$$

2.7.2 快速Fourier变换

设正整数m 除以N 后得商q 及余数r ，则$m=qN+r$， r 称为m 的N同余数，由于$w=e^{i\frac{2\pi}{N}}$ 且$w^N=e^{i2\pi}=1$，因此有$w^m=(w^N)^q w^r=w^r$。因此，计算w^m 时可用同余数r 代替m。

FFT算法利用了这个思想，当 $N=2^3$ 时，由于$0\le k,j\le N-1=7$

3 数值积分与数值微分

3.1 引言

3.1.1 数值求积的基本思想

许多实际问题常常需要计算积分才能求解，但是微积分定理需要知道被积函数的原函数。由于许多被积函数没有初等函数表示的原函数，且许多数值计算给出的是数据表，无法应用微积分定理。

根据积分中值定理，在积分区间内(a,b)存在一点ξ，有下式成立

$$\int_a^b f(x)\mathbf{d}a=(b-a)f(\xi)$$

问题在于点ξ 的位置不知道，无法准确算出$f(\xi)$ 的值。将$f(\xi)$ 称为区间$[a,b]$上的平均高度，此时，只要对平均高度$f(\xi)$ 提供一种算法便可以获得一种数值求积的方法。

一般，可以在区间$[a,b]$ 上适当选取某些节点x_k ，然后用$f(x_k)$ 加权平均得到平均高度$f(\xi)$ 的近似值，构造出的求积公式如下

$$\int_a^b f(x)\mathbf{d}x\approx\sum_{k=0}^{n}A_k f(x_k) \tag{3.1.1}$$

其中，x_k 为求积节点，A_k 为求积系数，也是节点x_k 的权。权的选取仅与节点x_k 的选取有关，而不依赖与被积函数$f(x)$ 的具体形式。

这类积分方法称为机械求积，特点是将积分求值问题归结为函数值的计算。

3.1.2 代数精度的概念

定义 17 如果某个求积公式对于次数不大于m 的多项式均能准确地成立，但对于$m+1$次多项式就不一定准确，则称该求积公式具有m 次代数精度

想要使求积公式3.1.1具有m 次代数精度，只要令它对于$f(x)=1,x,x^2,\cdots,x^m$都能准确成立，需要满足已下要求

$$\begin{cases}\sum A_k = b-a \\ \sum A_k x_k = \frac{1}{2}(b^2-a^2) \\ \vdots \\ \sum A_k x_k^m = \frac{1}{m+1}(b^{m+1}-a^{m+1})\end{cases} \tag{3.1.2}$$

取$m=n$求解方程组3.1.2 即可确定求积系数A_k，从而使3.1.1 至少具有n 次代数精度。

3.1.3 插值型的求积公式

设给定一组节点

$$a \le x_0 < a_1 < \cdots < x_n \le b$$

且已知函数在这些节点上的值，作插值函数$L_n(x)$ 。由于$L_n(x)$ 的原函数容易求出，取$I_n=\int_a^b L_n(x)\mathbf{d}x$作为积分$I=\int_a^b f(x)\mathbf{d}x$的近似值，这样构造出的求积公式

$$I_n = \sum_{k=0}^{n} A_k f(x_k) \tag{3.1.3}$$

是插值型的。其中求积系数A_k 通过插值基函数$l_k(x)$ 的积分

$$A_k = \int_a^b l_k(x)\mathbf{d}x \tag{3.1.4}$$

得出。由插值余项定理可知，对于插值型的求积公式3.1.3 ，余项为

$$R[f] = I - I_n = \int_a^b \frac{f^{n+1}(\xi)}{(n+1)!}\omega(x)\mathbf{d}x \tag{3.1.5}$$

当求积公式3.1.3 是插值型的，则对于次数不大于n 的多项式$f(x)$ ，其余项$R[f]$ 等于零。反之，若求积公式3.1.3 至少具有n 次代数精度，则它必定是插值型的，这时有

$$\int_a^b l_k(x)\mathbf{d}x = \sum_{j=0}^{n} A_j l_k(x_j)$$

综上所述有以下定理

定理 7 *形如3.1.3 的求积公式至少有n 次代数精度的充分必要条件是它是插值型的。*

3.2 Newton-Cotes公式

3.3 Cotes系数

将积分区间$[a,b]$ 划分为n 等分，步长$h = \frac{b-a}{n}$，选取等距节点$x_k = a + kh$ 构造的插值型求积公式

$$I_n = (b-a)\sum_{k=0}^{n} C_k^{(n)} f(x_k) \tag{3.2.1}$$

称为Newton-Cotes公式，其中$C_k^{(n)}$ 称为Cotes系数，按式**??** ，引进变化，则有

$$C_k^{(n)} = \frac{h}{b-a}\int_0^n \prod_{j=0,j\neq k}^{n} \frac{t-j}{k-j}\mathbf{d}t = \frac{(-1)^{n-k}}{nk!(n-k)!}\int_0^n \prod_{j=0,j\neq k}^{n} t-j\mathbf{d}x \tag{3.2.2}$$

当$n=1$ 时，$C_n^{(1)} = C_1^{(1)} = \frac{1}{2}$，此时求积公式为梯形公式。当$n=1$ 时，求积公式时下列Simpson公式

$$S = \frac{b-a}{6}[f(a) + 4f(\frac{a+b}{2}) + f(b)] \tag{3.2.3}$$

当$n=4$ 时，Newton-Cotes公式称为Cotes公式，形式为

$$C = \frac{b-a}{90}[7f(x_0) + 32f(x_1) + 12f(x_2) + 32f(x_3) + 7f(x_4)] \tag{3.2.4}$$

其中$x_k = a + kh, h = \frac{b-a}{4}$ 当$n \geq 8$ 时，Cotes系数有正有负，此时稳定性得不到保证，因此实际计算中不适用高阶的Newton-Cotes公式。

3.3.1 偶阶求积公式的代数精度

定理 8 *当阶n 为偶数时，Newton-Cotes公式至少有$n+1$ 次代数精度。*

3.3.2 几种低阶求积公式的余项

按余项公式3.1.5 ，梯形公式的余项为

$$R_T = I - T = \int_a^b \frac{f''(\xi)}{2}(x-a)(x-b)\mathbf{d}$$

这里积分的核函数$(x-a)(x-b)$ 在区间$[a,b]$ 上保号，应用积分中值定理，在内存(a,b) 内存在一点η 使得

$$R_T = \frac{f''(\eta)}{2}\int_a^b (x-a)(x-b)\mathbf{d}x = -\frac{f''(\eta)}{12}(b-a)^3, \eta \in (a,b) \tag{3.2.5}$$

上面没看懂。

Simpson公式3.2.3 的余项，为此构造次数不大于三的多项式$H(x)$，使之满足

$$\begin{cases} H(a) = f(a) \\ H(b) = f(b) \\ H(c) = f(c) \\ H'(c) = f'(c) \end{cases} \tag{3.2.6}$$

其中$c = \frac{a+b}{2}$

由于Simpson公式具有三次代数精度，对于这样构造出的三次$H(x)$ 是准确的

$$\int_a^b H(x)\mathbf{d}x = \frac{b-a}{6}[H(a) + 4H(c) + H(b)]$$

因此积分余项为

$$R_s = \int_a^b \frac{f^{(4)}(\xi)}{4!}(x-a)(x-c)^2(x-b)\mathbf{d}x = -\frac{b-a}{180}(\frac{b-a}{2})^2 f^{(4)}(\eta) eq : 3.2.7 \tag{3.2.7}$$

Cotes公式3.2.4 的积分余项为

$$R_C = I - C = -\frac{2(b-a)}{945}(\frac{b-a}{4})^6 f^{(6)}(\eta) \tag{3.2.8}$$

3.3.3 复化求积法及其收敛性

为了改善求积的精度，通常采样复化求积法。

设将积分区间$[a,b]$ 划分为n 等分，步长为$h = \frac{b-a}{h}$ ，分点为$x_k = a + kh$ ，所谓复化求积法，就是先用低阶的Newton-Cotes公式得到每个子区间$[x_k, x_{k+1}]$ 上的积分值I_k，然后在求和，利用$\sum_{k=0}^{n-1} I_k$ 作为所求积分I 的近似值。

复化梯形求积公式的形式是

$$T_n = \sum_{k=0}^{n-1} \frac{h}{2}[f(x_k) + f(x_{k+1})] = \frac{h}{2}[f(a) + \sum_{k=1}^{n-1} f(x_k) + f(b)] \tag{3.2.9}$$

其积分余项为

$$I - T_n = \sum_{k=0}^{n-1}[-\frac{h^3}{12}f''(\eta_k)] = -\frac{b-a}{12}h^2 f''(\eta) \tag{3.2.10}$$

记子区间$[x_k, x_{k+1}]$ 的中点为$x_{k+\frac{1}{2}}$ ，则复化Simpson公式为

$$\begin{aligned} S_n &= \sum_{k=0}^{n-1} \frac{h}{6}[f(x_k) + 4f(x_{k+\frac{1}{2}}) + f(x_{k+1})] \\ &= \frac{h}{6}[f(a) + 4\sum_{k=0}^{n-1} f(x_{k+\frac{1}{2}}) + 2\sum_{k=1}^{n-1} f(x_k) + f(b)] \end{aligned} \tag{3.2.11}$$

余项为

$$I - S_n = -\frac{b-a}{180}(\frac{h}{2})^4 f^{(4)}(\eta)$$

如果将每个子区间$[x_k, x_{k+1}]$ 划分为4等分，内分点依次记作$x_{k+\frac{1}{4}}, x_{k+\frac{1}{2}}, x_{k+\frac{3}{4}}$，则复化Cotes公式具有形式

$$C_n = \frac{h}{90}[f(a)+32\sum_{k=0}^{n-1} f(x_{k+\frac{1}{4}})+12\sum_{k=0}^{n-1} f(x_{k+\frac{1}{2}})+32sum_{k=0}^{n-1} f(x_{k+\frac{3}{4}})+14\sum_{k=1}^{n-1} f(x_k)+7f(b)] \tag{4.2.12}$$

余项为

$$I - C_n = -\frac{2(b-a)}{945}(\frac{h}{4})^6 f^{(6)}(\eta)$$

定义 18 *如果一种复化求积公式I_n ，当$h \to 0$ 时成立渐近关系式*

$$\frac{I - I_n}{h^p} \to C$$

则称求积公式I_n 是p 阶收敛的。

其中，C 是不为零的常数。

复化的梯形法、Simpson法和Cotes法分别具有二阶、四阶和六阶收敛精度。

当h 很小的时，对于复化的梯形法、Simpson法和Cotes法分别有下列误差估计式

$$I - T_n \approx -\frac{h^2}{12}[f'(b) - f'(a)] \tag{3.2.13}$$

$$I - S_n \approx -\frac{1}{180}(\frac{h}{2})^4[f'''(b) - f'''(a)] \tag{3.2.14}$$

$$I - C_n \approx -\frac{2}{945}(\frac{h}{4})^6[f^{(5)}(b) - f^{(5)}(a)] \tag{3.2.15}$$

若将步长h 减半，则梯形法、Simpson法与Cotes法的误差分别减到原有误差的$\frac{1}{4}, \frac{1}{16}, \frac{1}{64}$。

Romberg算法

3.3.4 梯形法的递推化

复化求积公式必须预先给出合适的步长，但是事先给出恰当的步长是困难的。在实际中常常采用变步长的计算方案，记在步长逐次分半的过程中，反复利用复化求积公式进行计算，知道所求的积分值满足精度要求为止。

设将求积区间$[a,b]$ 分成n 等分，则一共有$n+1$ 个分点，按复化梯形公式3.2.9 计算积分制T_n ，需要提供$n+1$ 个函数值，如果将求积区间再进行一次二分，则分点增加到$2n+1$ 个，每个子区间$[x_k, x_{k+1}]$ 只增加了一个分点$x_{k+\frac{1}{2}} = \frac{1}{2}(x_k + x_{k+1})$ ，根据复化梯形公式得到该子区间上的积分值为

$$\frac{h}{4}[f(x_k) + 2f(x_{k+\frac{1}{2}}) + f(x_{k+1})$$

这里的h 表示二分前的步长。因此，在整个区间上的积分值为

$$T_{2n} = \frac{h}{4}\sum_{k=0}^{n-1}[f(x_k) + f(x_{k+1})] + \frac{h}{2}\sum_{k=0}^{n-1} f(x_{k+\frac{1}{2}})$$

利用式3.2.9 可得到下列递推公式

$$T_{2n} = \frac{1}{2}T_n + \frac{h}{2}\sum_{k=0}^{n-1} f(x_{k+\frac{1}{2}}) \tag{3.3.1}$$

3.3.5 Romberg公式

梯形法的算法简单，但精度较差，收敛的速度慢。

根据梯形法的误差公式3.2.13，积分值T_n 的截断误差大致与h^2 成正比，当步长二分后，截断误差将减至原有误差的$\frac{1}{4}$ ，即

$$\frac{I - T_{2n}}{I - T_n} \approx \frac{1}{4}$$

移项整理可得

$$I - T_{2n} \approx \frac{1}{3}(T_{2n} - T_n) \tag{3.3.2}$$

根据上式可知，积分近似值T_{2n} 的误差大致等于$\frac{1}{3}(T_{2n} - T_n)$，因此，用这个误差作为$T_{2n}$ 的一种补偿，可以期望得到的结果可能式更好的结果，即

$$\overline{T} = T_{2n} + \frac{1}{3}(T_{2n} - T_n) = \frac{4}{3}T_{2n} - \frac{1}{3}T_n \tag{3.3.3}$$

上述结果中的$\overline{T}$ 实际上是Simpson法的积分值S_n

同理，根据Simpson法的误差公式**??**，可得到Cotes法的积分值C_n ，即

$$C_n = \frac{16}{15}S_{2n} - \frac{1}{15}S_n \tag{3.3.4}$$

依据Cotes法的误差公式**??** ，可得到Romberge公式

$$R_n = \frac{64}{63}C_{2n} - \frac{1}{63}C_n \tag{3.3.5}$$

Richardson外推加速法

定理 9 *设$f(x) \in C^{\infty}[a,b]$，则成立*

$$T(h) = I + \alpha_1 h^2 + \alpha_2 h^4 + \alpha_3 h^6 + \cdots + \alpha_k h^{2k} + \cdots, \tag{3.3.6}$$

其中系数a_k 与h 无关

根据上式，有

$$T(\frac{h}{2}) = I + \frac{\alpha_1}{4}h^2 + \frac{\alpha_2}{16}h^4 + \frac{\alpha_3}{64}h^6 \tag{3.3.7}$$

将式3.3.6和 3.3.7 按以下方式作线性组合，

$$T_1(h) = \frac{4}{3}T(\frac{h}{2}) - \frac{1}{3}T(h) \tag{3.3.8}$$

可以消除误差的主要部分h^2 项，从而得到

$$T_1(h) = I + \beta_1 h^4 + \beta_2 h^6 + \cdots \tag{3.3.9}$$

根据上文可知，构造出的$T_1(h)$ 是Simpson值

同理，有

$$T_1(\frac{h}{2}) = I + \frac{\beta_1}{16}h^4 + \hat{\beta}_2 h^6 + \cdots$$

令

$$T_2(h) = \frac{16}{15}T_1(\frac{h}{2}) - \frac{1}{15}T_1(h)$$

可消去h^4 项，从而有

$$T_2(h) = I + \gamma h^6 + \gamma_2 h^8 + \cdots$$

每加速一次，误差的量级便提高二阶。一般的，将$T_0(h) = T(h)$ 按公式

$$T_m(h) = \frac{4^m}{4^m - 1}T_{m-1}(\frac{h}{2}) - \frac{1}{4^m - 1}T_{m-1}(h) \tag{3.3.10}$$

经过m 次加速后，余项的形式为

$$T_m(h) = I + \delta_1 h^{2(m+1)} + \delta_2 h^{2(m+2)} + \cdots \tag{3.3.11}$$

上述处理方法通常称为Richardson外推加速方法

设以$T_0^{(k)}$ 表示二分k 次后求得的梯形值，且以$T_m^{(k)}$ 表示序列$\{T_0^{(k)}\}$ 的m 次加速值，则依上述递推公式可得

$$T_m^{(k)} = \frac{4^m}{4^m - 1}T_{m-1}^{(k+1)} - \frac{1}{4^m - 1}T_{m-1}^{(k)} \tag{3.3.12}$$

可以逐行构造出下列三角性数表——T数表

$$\begin{array}{cccc} T_0^{(0)} & & & \\ T_0^{(1)} & T_1^{(0)} & & \\ T_0^{(2)} & T_1^{(1)} & T_2^{(0)} & \\ \vdots & \vdots & \vdots & \ddots \end{array}$$

可以证明，如果$f(x)$ 充分光滑，那么，T数表每一列的元素及对角线元素均收敛到所求的积分值I

Romberg算法是在二分过程中逐渐形成T数表的具体方法，步骤如下

1. 准备初值，计算$T_0^{(0)} = \frac{b-a}{2}[f(a)+f(b)]$ ，且令$1 \to k$
2. 求梯形值，按递推公式计算梯形值 $T_0^{(k)}$
3. 求加速值，按加速公式逐个计算出T数表第$k+1$ 行其余各元素$T_j^{(k-j)}$
4. 精度控制，对于指定精度ϵ ，若$|T_k^{(0)} - T_{k-1}^{(0)}| < \epsilon$，则终止计算，并取$T_k^{(0)}$ 作为所求的结果，否则，转步骤2继续计算。

当使用Romberg算法时，需要注意式3.3.6 成立，否则得不到正确的结果。

3.3.6 梯形法的余项展开式

3.4 Gauss公式

机械求积公式**??**中含有$2n+2$ 个待定参数x_k, A_k ，选择适当的参数可能使求积公式具有$2n+1$ 次代数精度，各类公式称为Gauss公式

3.4.1 Gauss公式

Gauss公式的求积点称为Gauss点

定义 19 *如果求积公式*?? *具有*$2n+1$ *次代数精度，则称其节点*x_k *是*$Gauss$*点*

定理 10 *对于插值型求积公式，其节点*x_k *是*$Gauss$*点的充分必要条件是以这些点为零点的多项式*$\omega(x) = \prod_{k=0}^{n}(x - x_k)$ *与任意次数不超过*n *的多项式*$P(x)$ *均正交，即*

$$\int_a^b P(x)\omega(x)\mathbf{d}x = 0 \tag{3.4.1}$$

3.4.2 Gauss-Legendre公式

取$a=-1,b=1$ 考察区间$[-1,1]$ 上的Gauss公式

$$\int_{-1}^{1} f(x)\mathbf{d}x \approx \sum_{k=0}^{n} A_k f_k(x_k) \tag{3.4.2}$$

另，Legendre多项式是区间$[-1,1]$ 上的正交多项式，因此，Legendre多项式$P_{n+1}(x)$ 的零点就是求积公式3.4.2 的Gauss点，形如3.4.2 的Gauss公式特别地称为Gauss-Legendre公式。

求任意区间$[a,b]$ 的Gauss公式，可以通过变换$x=\frac{b-a}{2}t+\frac{a+b}{2}$ 可以化到区间$[-1,1]$ 上，此时$\int_a^b f(x)\mathbf{d}x=\frac{b-a}{2}\int_{-1}^{1} f(\frac{b-a}{2}+\frac{a+b}{2})\mathbf{d}t$。

3.4.3 Gauss公式的余项

定理 11 *对于Gauss公式，其余项为*

$$R(x)=\int_a^b f(x)\mathbf{d}x-\sum_{k=0}^{n} A_k f(x_k)=\frac{f^{(2n+1)}(\xi)}{(2n+2)!}\int_a^b \omega^2(x)\mathbf{d}x \tag{3.4.3}$$

3.4.4 Gauss公式的稳定性

Gauss公式是高精度的，也是数值稳定的。

定理 12 *Gauss公式的求积系数A_k 是全正的。*

在使用求积公式$I_n=\sum_{k=0}^{n} A_k f(x_k)$ 进行实际计算时，通常不一定能够提供准确的数据$f_k=f(x_k)$ ，而只能给出含有误差的数据$f_k^\star$，故，实际求得的积分值为

$$I_n^\star=\sum_{k=0}^{n} A_k f_k^\star$$

由于Gauss公式的求积系数具有非负性，因此

$$\begin{aligned}
|I_n^\star-I_n| &\le \sum_{k=0}^{n} A_k|f_k^\star-f_k| \\
&\le (\sum_{k=0}^{n} A_k)\max_{0\le k\le n}|f_k^\star-f_k| \\
&\le (b-a)\max_{0\le k\le n}|f_k^\star-f_k|
\end{aligned}$$

因此，Gauss公式是稳定的

3.4.5 带权的Gauss公式

考察带权函数的求积公式

$$\int_a^b \rho(x)f(x)\mathbf{d}x \approx \sum_{k=0}^{n} A_k f(x_k)$$

如果它对于任意次数不超过$2n+1$ 的多项式均能准确地成立，则称之为Gauss型的。x_k 是Gauss点的充要条件为$\omega(x)$ 是区间$[a,b]$ 上关于权函数$\rho(x)$ 的正交多项式。

若$a=-1, b=1$ ，且权函数为$\rho(x)=\frac{1}{\sqrt{1-x^2}}$，则建立的高斯公式为

$$\int_{-1}^{1} \frac{f(x)}{\sqrt{1-x^2}}\mathbf{d}x \approx \sum_{k=0}^{n} A_k f(x_k) \tag{3.4.4}$$

称上式为Gauss-Chebyshev公式，Gauss点是$n+1$ 次Chebyshev多项式的零点，即

$$x_k = \cos\left(\frac{2k+1}{2n+2}\pi\right)$$

构造Gauss公式的一般方法是待定系数法。

3.5 数值微分

3.5.1 中点方法

当精度要求不高的时候，可以取差商作为导数的近似值，这便是一种数值微分方法，即

$$f'(a) \approx \frac{f(a+h)-f(a)}{h}$$

类似地，也可以用向后差商或中心差商作近似运算，即

$$f'(a) \approx \frac{f(a)-f(a-h)}{h}$$

$$f'(a) \approx \frac{f(a+h)-f(a-h)}{2h}$$

最后一种数值方法称为中点方法，是前两中方法的算数平均。

上述方法都是将导数的计算归结为计算f 在若干点的函数值，称为机械求导方法。

利用中点公式

$$G(h) = \frac{f(a+h)-f(a-h)}{2h}$$

计算导数$f'(a)$ 的近似值，必须选取合适的步长，为此需要进行误差分析，分别将$f(a\pm h)$在$x=a$ 除作Taylor展开，有

$$f(a\pm h) = f(a) \pm hf'(a) + \frac{h^2}{2!}f''(a) \pm \frac{h^3}{3!}f'''(a) + \cdots$$

带入中点公式得

$$G(h) = f'(a) + \frac{h^2}{3!}f'''(a) + \frac{h^4}{5!}f^{(5)}(a) + \cdots$$

因此，步长越小，计算结果越准确。

另外，当h 很小的时候，$f(a+h)$ 和$f(a-h)$ 很接近，直接相减会造成有效数字的严重损失，因此，步长也不宜太小。

3.5.2 插值型的求导公式

对于$y = f(x)$运用插值原理，可以建立插值多项式$y = P_n(x)$ 作为它的近似，由于多项式求导比较容易，取$P_n'(x)$ 的值作为$f'(x)$ 的近似值，这样建立的数值公式

$$f'(x) \approx P_n'(x) \tag{3.5.1}$$

统称为插值型的求导公式。

即使$f(x)$ 与$P_n(x)$ 的值相差不多，导数的近似值$P_n'(x)$ 与导数的真值$f'(x)$ 仍然可能差别很大，因而在使用求导公式时应特别注意误差的分析。

根据插值余项定理，求导公式3.5.1 的余项为

$$f'(x) - P_n'(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!}\omega_{n+1}'(x) + \frac{\omega_{n+1}(x)}{(n+1)!}\frac{\mathbf{d}}{\mathbf{d}x}f^{(n+1)}(\xi)$$

在余项公式中，ξ 是x 的未知函数，无法对第二项做出进一步的说明，因此，对于随意给出的点x ，误差$f'(x) - P_n'(x)$ 是无法预估的。当在限定求某个节点x_k 上的导出值，第二项为零。

下面仅考察节点除的导数值，假定所给的节点是等距的。

1. 两点公式设给出两个节点x_0, x_1 上的函数值$f(x_0), f(x_1)$，作线性插值公式

   $$P_1(x) = \frac{x - x_1}{x_0 - x_1}f(x_0) + \frac{x - x_0}{x_1 - x_0}f(x_1)$$

   对上式进行求导，即$x_1 - x_0 = h$，有

   $$P_1'(x) = \frac{1}{h}[-f(x_0) + f(x_1)]$$

   此时，带余项的两点公式是

   $$f'(x_0) = \frac{1}{h}[f(x_1) - f(x_0)] - \frac{h}{2}f''(\xi)$$

   $$f'(x_1) = \frac{1}{h}[f(x_1) - f(x_0)] + \frac{h}{2}f''(\xi)$$

2. 三点公式设已给出三个节点$x_0, x_1 = x_0 + h, x_2 = x_0 + 2h$ 上的函数值，作二次插值

$$P_2(x) = \frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)}f(x_0)+\frac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)}f(x_1)+\frac{(x-x_0)(x-x_1)}{(x_2-x_0)(x_2-x_1)}f(x_2)$$

令$x = x_0 + th$ ，上式可表示为

$$P_2(x_0+th) = \frac{1}{2}(t-1)(t-2)f(x_0) - t(t-2)f(x_1) + \frac{1}{2}t(t-1)f(x_2)$$

两端对t 求导，有

$$P_2'(x_0+th) = \frac{1}{2h}[(2t-3)f(x_0)-(4t-4)f(x_1)+(2t-1)f(x_2)] \quad (3.5.3)$$

分别取$t = 0, 1, 2$ 可以得到以下三种三点公式

$$P_2'(x_0) = \frac{1}{2h}[-3f(x_0) + 4f(x_1) - f(x_2)]$$
$$P_2'(x_1) = \frac{1}{2h}[-f(x_0) + f(x_2)]$$
$$P_2'(x_2) = \frac{1}{2h}[f(x_0) - 4f(x_1) + 3f(x_2)]$$

带余项的三点公式为

$$P_2'(x_0) = \frac{1}{2h}[-3f(x_0) + 4f(x_1) - f(x_2)] + \frac{h^2}{3}f'''(\xi)$$
$$P_2'(x_1) = \frac{1}{2h}[-f(x_0) + f(x_2)] - \frac{h^2}{6}f'''(\xi)$$
$$P_2'(x_2) = \frac{1}{2h}[f(x_0) - 4f(x_1) + 3f(x_2)] + \frac{h^2}{3}f'''(\xi)$$

用插值多项式$P_n(x)$ 作为$f(x)$ 的近似函数，还可以建立高阶数值微分公式。

3.5.3 实用的五点公式

设已给出五给节点$x_i = x_0 + ih$ 上的函数值，可以得出以下五点公式的一阶导数

$$m_0 = \frac{1}{12h}[-25f(x_0) + 48f(x_1) - 36f(x_2) + 16f(x_3) - 3f(x_4)]$$
$$m_1 = \frac{1}{12h}[-3f(x_0) - 10f(x_1) + 18f(x_2) - 6f(x_3) + f(x_4)]$$
$$m_2 = \frac{1}{12h}[f(x_0) - 8f(x_1) + 8f(x_3) - f(x_4)]$$
$$m_3 = \frac{1}{12h}[-f(x_0) + 6f(x_1) - 18f(x_2) + 10f(x_3) + 3f(x_4)]$$
$$m_4 = \frac{1}{12h}[3f(x_0) - 16f(x_1) + 36f(x_2) - 48f(x_3) + 25f(x_4)]$$

同理得到二阶导数为

$$M_0 = \frac{1}{12h^2}[35f(x_0) - 104f(x_1) + 114f(x_2) - 56f(x_3) + 11f(x_4)]$$
$$M_1 = \frac{1}{12h^2}[11f(x_0) - 20f(x_1) + 6f(x_2) + 4f(x_3) - f(x_4)]$$
$$M_2 = \frac{1}{12h^2}[-f(x_0) + 16f(x_1) - 30f(x_2) + 16f(x_3) - f(x_4)]$$
$$M_3 = \frac{1}{12h^2}[-f(x_0) + 4f(x_1) + 6f(x_2) - 20f(x_3) + 11f(x_4)]$$
$$M_4 = \frac{1}{12h^2}[11f(x_0) - 56f(x_1) + 114f(x_2) - 104f(x_3) + 35f(x_4)]$$

用五点公式求节点上的导数值往往可以得到满意的结果。

五个相邻节点的选择原则一般是在所考察的节点的两侧各取两个邻近的节点，若一侧的节点数不足两个，则用另一侧的节点补足。

3.5.4 样条求导

样条微分公式可以用来计算插值范围内任何一点上的导出值。

4 常微分方程数值解法

4.1 引言

本节主要考虑一阶方程的初值问题

$$\begin{cases} y' = f(x, y) \\ y(x_0) = y_0 \end{cases} \tag{4.1.1}$$

数值解法就是寻求解$y(x)$ 在一系列离散节点

$$x_1 < x_2 < \cdots < x_n < \cdots$$

上的近似值$y_1, y_2, \cdots, y_n, \cdots$ 。相邻两个节点的间距$h = x_{n+1} - x_n$称为步长。

初值问题4.1.1 的数值解法有个基本特点，它们都采取步进式，即求解过程顺着节点排列次序一步一步地向前推荐。这类算法只要给出递推公式即可，这种计算公式称为差分格式。

4.2 Euler方法

4.2.1 Euler格式

对于初值问题，方程的解$y = y(x)$ 称为它的积分曲线，积分曲线上一点(x, y) 的切线斜率等于函数$f(x, y)$ 的值。

按$f(x,y)$ 在Oxy 平面上建立一个方向场，则积分曲线上每一点的切线方向与方向场在该点的方向以值。我们从初始点$P_0(x_0,y_0)$ 出发，依方向场在该点的方向推进到$x=x_1$ 上的一点P_1 ，再从P_1 依方向场推进到$x=x_2$ 上一点P_2 。依次类推。

显然，P_n 和P_{n+1}有以下关系

$$\frac{y_{n+1}-y_n}{x_{n+1}-x_n}=f(x_n,y_n)$$

即

$$y_{n+1}=y_n+hf(x_n,y_n) \tag{4.2.1}$$

通常采用Taylor展开来分析计算公式的精度。为简化分析，假定y_n 是准确的，即在$y_n=y(x_n)$ 的前提下估计误差$y(x_{n+1})-y_{n+1}$ ，这种误差称为局部截断误差。

Euler格式4.2.1 的局部截断误差为

$$y(x_{n+1})-y_{n+1}=\frac{h^2}{2}y''(\xi)\approx\frac{h^2}{2}y''(x_n) \tag{4.2.2}$$

4.2.2 后退的Euler格式

由于微分方程中含有导数项，因此难以求解。数值解法的关键在于设法消除其导数项，这个步骤称为离散化。由于差分是微分的近似计算，实现离散化的基本途径之一是用差商代替导数。如在Euler格式中就使用来这个方法

对于在点x_{n+1} 列出的方程4.1.1 ，有

$$y'(x_{n+1})=f(x_{n+1},y(x_{n+1}))$$

若使用向后差商$\frac{y(x_{n+1})-y(x_n)}{h}$ 代替导出，则可得

$$\frac{y_{n+1}-y_n}{h}=f(x_{n+1},y_{n+1})$$

即

$$y_{n+1}=y_n+hf(x_{n+1},y_{n+1}) \tag{4.2.3}$$

这个就是后退的Euler格式。

Euler格式是关于y_{n+1}的一个直接计算格式，是显示的。而后退Euler格式右端含有未知的y_{n+1}，是一个关于y_{n+1}的函数方程，是隐式的。

隐式方法更加稳定，但显示方法更加方便。

方程4.2.3 替换使用迭代法求解，迭代过程的实质是逐步显示化。

设用Euler格式$y_{n+1}^{(0)} = y_n + hf(x_n, y_n)$ 给出迭代初值$y_{n+1}^{(0)}$ ，带入隐式方程4.2.3 ，直接计算得

$$y_{n+1}^{(1)} = y_n + hf(x_{n+1}, y_{n+1}^{(0)})$$

然后在用$y_{n+1}^{(0)}$ 带入方程4.2.3 得右端，有

$$y_{n+1}^{(2)} = y_n + hf(x_{n+1}, y_{n+1}^{(1)})$$

如此反复进行迭代得，

$$y_{n+1}^{(k+1)} = y_n + hf(x_{n+1}, y_{n+1}^{(k)})$$

若迭代过程收敛，则极限值$y_{n+1} = \lim_{k\to\infty} y_{n+1}^{(k)}$ 满足隐式方程，从而得到后退Euler方法得解。

后退Euler格式得局部截断误差为

$$y(x_{n+1}) - y_{n+1} \approx -\frac{h^2}{2}y''(x_n) \tag{4.2.4}$$

4.2.3 梯形格式

根据Euler格式和后退Euler的误差公式4.2.2, 4.2.4可知，若将这两种方法进行算数平均，可以消除武昌的主要部分$\pm\frac{h^2}{2}y_n''$，从而得到更高的精度，这种平均化方法通常称为梯形方法，计算格式为

$$y_{n+1} = y_n + \frac{h}{2}[f(x_n, y_n) + f(x_{n+1}, y_{n+1})] \tag{4.2.5}$$

梯形法是隐式的，可用迭代法求解，同后退Euler方法一样，使用Euler方法提供迭代初值，则梯形法的迭代公式为

$$y_{n+1}^{(k+1)} = y_n + \frac{h}{2}[f(x_n, y_n) + f(x_{n+1}, y_{n+1}^{(k)})] \tag{4.2.6}$$

为了分析迭代过程的收敛性，将4.2.5和 4.2.6 相减，得

$$y_{n+1} - y_{n+1}^{(k+1)} = \frac{h}{2}[f(x_{n+1}, y_{n+1}) - f(x_{n+1}, y_{n+1}^{(k)}]$$

因此有

$$|y_{n+1} - y_{n+1}^{(k+1)}| \le \frac{hL}{2}|y_{n+1} - y_{n+1}^{(k)}|$$

其中L 为$f(x, y)$ 关于y 的Lipschitz常数。如果h 充分小，使得$\frac{hL}{2} < 1$ ，则迭代过程是收敛的。

4.2.4 改进的Euler格式

梯形法算法复杂，计算量大。

改进的Euler格式先用Euler格式求得一个初步的近似值$\overline{y}_{n+1}$，称之为预测值，再用梯形公式将它矫正一次，得y_{n+1}，这个结果称为校正值，这一计算格式可表示为

$$y_{n+1} = y_n + \frac{h}{2}[f(x_n, y_n) + f(x_n + h, y_n + hf(x_n, y_n))] \tag{4.2.7}$$

或者是以下形式

$$\begin{cases} y_p = y_n + hf(x_n, y_n) \\ y_c = y_n + hf(x_{n+1}, y_p) \\ y_{n+1} = \frac{1}{2}(y_p + y_c) \end{cases}$$

其实就是将梯形法的迭代次数限制为1。

4.2.5 Euler两步格式

在改进的Euler该格式中，预测公式的精度差和校正公式不匹配，因此，使用中心差商来代替导数值，此时，得到Euler两步格式

$$y_{n+1} = y_{n-1} + 2hf(x_n, y_n) \tag{4.2.8}$$

两步法在计算的时候还需要其他单步法提供一个开始值，才能启动计算公式。

两步法的优点是它调用了两个节点上的已知信息，从而能以较少的计算两获得较高的精度。

使用Euler两步格式与梯形格式相匹配，得到以下预测校正系统

$$\begin{cases} \overline{y}_{n+1} = y_{n-1} + 2hf(x_n, y_n) \\ y_{n+1} = y_n + \frac{h}{2}[f(x_n, y_n) + f(x_{n+1}, \overline{y}_{n+1})] \end{cases} \tag{4.2.9}$$

与改进的Euler格式相比，上述方法的预测公式和矫正公式具有同等精度，因此能方便地估计出截断误差。

假设预测公式中的y_n 和y_{n+1}都是准确的，使用Taylor方法可得去不局部截断误差为

$$y(x_{n+1}) - \overline{y}_{n+1} \approx \frac{h^3}{3} y'''(x_n) \tag{4.2.10}$$

假设矫正公式中的$\overline{y}_{n+1}$ 是准确的，则其局部截断误差为

$$y(x_{n+1}) - \overline{y}_{n+1} \approx \frac{h^3}{3} y'''(x_n) \tag{4.2.11}$$

校正值的误差大约只有预测值的误差的$\frac{1}{4}$ ，即

$$\frac{y(x_{n+1})-y_{n+1}}{y(x_{n+1})-\overline{y}_{n+1}} \approx -\frac{1}{4}$$

由此，可推导出下列事后估计式

$$\begin{cases} y(x_{n+1}-\overline{y}_{n+1} \approx -\frac{4}{5}(\overline{y}_{n+1}-y_{n+1}) \\ y(x_{n+1}-y_{n+1} \approx -\frac{1}{5}(\overline{y}_{n+1}-y_{n+1}) \end{cases} \tag{4.2.12}$$

利用误差作为计算结果的步长，可能改善精度。

设以p_n 和c_n 分别代表第n 步的预测值和校正值，按估计式$refeq:4.2.12$，$p_{n+1}-\frac{4}{5}(p_{n+1}-c_{n+1})$ 和$c_{n+1}+\frac{1}{5}(p_{n+1}-c_{n+1})$ 分别可以取作p_{n+1} 和c_{n+1} 的改进值，在校正值c_{n+1} 尚未计算出之前，可用上一步的偏差值p_n-c_n 代替$p_{n+1}-c_{n+1}$ 来改进预测值p_{n+1}，这个计算方法有以下六步

1. 预测
$$p_{n+1}=y_{n-1}+2hy'_n$$
2. 改进
$$m_{n+1}=p_{n+1}-\frac{4}{5}(p_n-c_n)$$
3. 计算
$$m'_{n+1}=f(x_{n+1},m_{n+1})$$
4. 校正
$$c_{n+1}=y_n+\frac{h}{2}(m'_{n+1}+y'_n)$$
5. 改进
$$y_{n+1}=c_{n+1}+\frac{1}{5}(p_{n+1}-c_{n+1})$$
6. 计算
$$y'_{n+1}=f(x_{n+1},y_{n+1})$$

在启动计算前必须给出开始值y_1 和p_1-c_1，y_1 可用其他单步法计算，p_1-c_1 通常零它为零。

4.3 Runge-Kutta方法

这类方法与Taylor级数法有紧密联系

4.3.1 Taylor级数法

设初值问题的解$y = y(x)$ 可用Taylor展开，有

$$y(x_{n+1}) = y(x_n) + hy'(x_n) + \frac{h^2}{2}y''(x_n) + \frac{h^3}{3!}y'''(x_n) + \cdots \tag{4.3.1}$$

其中$y(x)$ 的各阶导数可用函数f 来表达，具体有

$$\begin{cases} y' = f \\ y'' = \frac{\partial f}{\partial x} + f\frac{\partial f}{\partial y} \\ y''' = \frac{\partial^2 f}{\partial x^2} + 2f\frac{\partial^2 f}{\partial x \partial y} + f^2\frac{\partial^2 f}{\partial y^2} + \frac{\partial f}{\partial y}(\frac{\partial f}{\partial x} + f\frac{\partial f}{\partial y}) \\ \vdots \end{cases} \tag{4.3.2}$$

在展开式4.3.1 右边取若干项，在(x_n, y_n) 按4.3.2 计算$y^{(j)}(x_n)$ 的近似值$y_n^{(j)}$，结果为下列Taylor格式

$$y_{n+1} = y_n + hy'_n + \frac{h^2}{2}y''_n + \cdots + \frac{h^p}{p!}y_n^{(p)} \tag{4.3.3}$$

其中一阶Taylor格式就是Euler格式。

提高Taylor格式的阶p 可提高计算结果的精度，p 阶Taylor格式的局部截断误差为

$$y(x_{n+1}) - y_{n+1} = \frac{h^{p+1}}{(p+1)!}y(p+1)(\xi), x_n < \xi < x_{n+1}$$

定义 20 *如果一种方法的局部截断误差为$O(h^{p+1})$ ，则称该方法具有p 阶精度。*

因此，Taylor格式具有p 阶精度

当阶数提高时，求导过程可能很复杂，因此Taylor级数法通常不直接使用。

4.3.2 Runge-Kutta方法的基本思路

考虑差商$\frac{y(x_{n+1})-y(x_n)}{h}$ ，根据微分中值定理，存在$0 < \theta < 1$ ，使得

$$\frac{y(x_{n+1}) - y(x_n)}{h} = y'(x_n + \theta h)$$

因此，可以得到以下公式

$$y(x_{n+1}) = y(x_n) + hf(x_n + \theta h, y(x_n + \theta h)) \tag{4.3.4}$$

设$K^\star = f(x_n + \theta h, y(x_n + \theta h))$ 为区间$[x_n, x_{n+1}]$ 上的平均斜率。此时，只有对平均斜率提供一种算法，就能根据4.3.4 得出一个计算格式。

Euler格式取点x_n 的斜率值作为平均斜率K。而改进的Euler格式使用x_n和x_{n+1}两个点的斜率值的平均作为平均斜率，其中x_{n+1}处的斜率通过已知信息y_n 来预测。

Runge-Kutta方法的基本思虑时在区间(x_n, x_{n+1}) 内多预测几个点的斜率，然后加权平均作为平均斜率$K^\star$ ，从而构造处更高精度的计算格式。

4.3.3 二阶Runge-Kutta方法

随意考察区间(x_n, x_{n+1}) 内一点$x_{n+p} = x_n + ph, 0 < p \leq 1$ 。希望用x_n和x_{n+p} 两个点的斜率值K_1 和K_2线性组合得到平均斜率$K^\star$，即有

$$y_{n+1} = y_n + (\lambda_1 K_1 + \lambda_2 K_2)h$$

其中，λ_1, λ_2 是待定系数。此时的问题在于如何得到x_{n+p} 处的斜率值K_2

参考改进Euler格式的方法，先用Euler格式得到x_{n+p} 处的预测值，即$y_{n+p} = y_n + phK_1$ ，然后在用预测值y_{n+p} 通过计算f 产生斜率值$K_2 = f(x_{n+p}, y_{n+p})$ ，得到的计算格式如下所示

$$\begin{cases} y_{n+1} = y_n + (\lambda_1 K_1 + \lambda_2 K_2)h \\ K_1 = f(x_n, y_n) \\ K_2 = f(x_{n+p}, y_n + phK_1) \end{cases} \tag{4.3.5}$$

式中包含三个待定系数，希望可以选取这些系数的值使得格式具有二阶精度。

根据4.3.3，二阶Taylor格式为

$$y_{n+1} = y_n + hf_n + \frac{h^2}{2}(f_x + ff_y)_n$$

下标n 表示在x_n 处取值。根据上述定义有$K_1 = f_n, K_2 = f_n + ph(f_x + ff_y)_n + \cdots$ ，将其带入4.3.5 可得

$$y_{n+1} = y_n + (\lambda_1 + \lambda_2)hf_n + \lambda_2 ph^2(f_x + ff_y)_n + \cdots$$

因此，当下式成立时，4.3.5 具有二阶精度

$$\begin{cases} \lambda_1 + \lambda_2 = 1 \\ \lambda_2 p = \frac{1}{2} \end{cases} \tag{4.3.6}$$

满足条件4.3.6 的格式4.3.5 统称为二阶Runge-Kutta格式。

一种特殊的Runge-Kutta格式是所谓的变形的Euler格式，其参数取值为$p = 1, \lambda_2 = 1, \lambda_1 = 0$

4.3.4 三阶Runge-Kutta方法

为了提高精度再额外考察一点$x_{n+q} = x_n + qh, p \leq q \leq 1$，并用三个点$x_n, x_{n+p}, x_{n+q}$的斜率值$K_1, K_2, K_3$线性组合得到平均斜率 $K^\star$，此时计算格式为

$$y_{n+1} = y_n + (\lambda_1 K_1 + \lambda_2 K_2 + \lambda_3 K_3)$$

为了预测x_{n+q} 处的斜率值K_3 ，用K_1 和K_2 线性组合给出区间$[x_n, x_{n+q}]$ 上的平均斜率，从而得到$y(x_{n+q})$ 的预测值$y_{n+q} = y_n + qh(rK_1 + sK_2)$，再得到$K_3$ 为

$$K_3 = f(x_{n+q}, y_{n+q}) = f(x_n + qh, y_n + qh(rK_1 + sK_2))$$

此时计算格式为

$$\begin{cases} y_{n+1} = y_n + h(\lambda_1 K_1 + \lambda_2 K_2 + \lambda_3 K_3) \\ K_1 = f(x_n, y_n) \\ K_2 = f(x_n + ph, y_n + phK_1) \\ K_3 = f(x_n + qh, y_n + qh(rK_1 + sK_2) \end{cases} \tag{4.3.7}$$

当系数满足以下方程组时，上述格式具有三阶精度

$$\begin{cases} \lambda_1 + \lambda 2 + \lambda_3 = 1 \\ r + s = 1 \\ \lambda_2 p + \lambda_3 q = \frac{1}{2} \\ \lambda_2 p^2 + \lambda_3 q^2 = \frac{1}{3} \\ \lambda_3 pqs = \frac{1}{6} \end{cases} \tag{4.3.8}$$

一个特殊的Kutta格式是

$$\begin{cases} y_{n+1} = y_n + \frac{h}{6}(K_1 + 4K_2 + K_3) \\ K_1 = f(x_n, y_n) \\ K_2 = f(x_n + \frac{h}{2}, y_n + \frac{h}{2}K_1) \\ K_3 = f(x_n + h, y_n - hK_1 + 2hK_2) \end{cases}$$

其中$p = \frac{1}{2}, q = 1, r = -1, s = 2, \lambda_1 = \frac{1}{6}, \lambda_2 = \frac{4}{6}, \lambda_3 = \frac{1}{6}$

4.3.5 四阶Runge-Kutta方法

将上述方法进行扩展，可以得到四阶的Runge-Kutta格式。一个常用的

格式如下

$$\begin{cases} y_{n+1} = y_n + \frac{h}{6}(K_1 + 2K_2 + 2K_3 + K_4) \\ K_1 = f(x_n, y_n) \\ K_2 = f(x_n + \frac{h}{2}, y_n + \frac{h}{2}K_1) \\ K_3 = f(x_n + \frac{h}{2}, y_n + \frac{h}{2}K_2) \\ K_4 = f(x_n + h, y_n + hK_3) \end{cases} \tag{4.3.9}$$

Runge-Kutta方法的推到基于Taylor展开方法，因此，它要求解具有较好的光滑性。若阶的光滑性较差，那么四阶Runge-Kutta方法可能不如改进的Euler方法的精度高。因此，实际计算式应针对问题选择合适的算法。

4.3.6 变步长的Runge-Kutta方法

步长越小，精度越高，但会增加计算量，且可能导致舍入误差的积累。因此，步长也需要慎重选择。

对于经典的四阶Runge-Kutta格式4.3.9 ，从节点x_n 出发，以步长h 求出一个近似值$y_{n+1}^{(h)}$ ，截断误差为

$$y(x_{n+1}) - y_{n+1}^{(h)} \approx Ch^5 \tag{4.3.10}$$

将步长折半，从x_n 跨两步到x_{n+1}，再求得一个近似值$y_{n+1}^{(\frac{h}{2})}$ ，每跨一步的截断误差为$C\left(\frac{h}{2}\right)^2$，因此，有

$$y(x_{n+1}) - y_{n+1}^{(\frac{h}{2})} = 2C\left(\frac{h}{2}\right)^2 \tag{4.3.11}$$

可以看到，步长折半后，误差大约减小到$\frac{1}{16}$ ，即

$$\frac{y(x_{n+1}) - y_{n+1}^{(\frac{h}{2})}}{y(x_{n+1}) - y_{n+1}^{(h)}} \approx \frac{1}{16}$$

因此有

$$y(x_{n+1}) - y_{n+1}^{(\frac{h}{2})} \approx \frac{1}{15}(y_{n+1}^{(\frac{h}{2})} - y_{n+1}^{(h)}$$

因此，可以通过检查步长折半前后的两次计算结果的偏差

$$\delta = |y_{n+1}^{(\frac{h}{2})} - y_{n+1}^{(h)}|$$

来判定所选的步长是否合适,ji

1. 对于给定精度ϵ，如果$\delta > \epsilon$ ，应反复将步长折半进行计算，直到$\delta < \epsilon$ 为止，取得到的$y_{n+1}^{(\frac{h}{2})}$ 作为结果

2. 若$\delta < \epsilon$ ，则反复将步长加倍，直到$\delta > \epsilon$ ，这是再将步长折半一次，就得到所要的结果。

4.4 单步法的收敛性和稳定性

定义 21 *若一种数值方法对于任意固定的*$x_n = x_0 + nh$ *，当*$h \to 0$*时有*$y_n \to y(x_n)$ *，则称该方法时收敛的。*

单步法是指在计算y_{n+1}时只用到它前一步的信息y_n ，Taylor级数法、Runge-Kutta方法都是单步法。显示单步法的特征是它们都将y_n 加上某种形式的增量得出y_{n+1}，即计算公式形如

$$y_{n+1} = y_n + h\phi(x_n, y_n, h) \tag{4.4.1}$$

其中$\phi(x, y, h)$ 称为增量函数。

定理 13 *假设单步法*4.4.1 *具有*p *阶精度，且增量函数*$\phi(x, y, h)$*关于*y *满足*$Lipschitz$*条件*

$$|\phi(x, y, h) - \phi(x, \overline{y}, h)| \leq L_\phi(y - \overline{y}) \tag{4.4.2}$$

又设初值y_0 *是准确的，即*$y_0 = y(x_0)$ *，则其整体截断误差*

$$y(x_n) - y_n = O(h^p) \tag{4.4.3}$$

因此，判断单步法的收敛性归结为验证增量函数ϕ 能否满足Lipschitz条件。

4.4.1 单步法的稳定性

定义 22 *若一种数值方法在节点值*y_n *上产生大小为*δ *的扰动，在以后各节点值*$y_m(m > n)$ *上产生的偏差均不超过*δ *，则称该方法是稳定的。*

Euler方法是条件稳定的，其稳定性条件为$h \leq 2\tau$ ，其中，τ是一个具有时间量纲的量。Euler方法的稳定性条件表明，时间常数越小，稳定性对步长h 的限制越苛刻。

后退的Euler方法是恒定的，或称无条件稳定。

4.5 线性多步法

线性多步法的基本思想为，在求解y_{n+1}之前已经求出一系列近似值$y_n, y_{n-1}, \cdots$ ，如果充分利用前面多步的信息来预测y_{n+1}，则可能会获得较高的精度。

构造多步法有许多中途径，其中两种为，基于数值积分的构造方法和基于Taylor展开的构造方法。

4.5.1 基于数值积分的构造方法

将方程$y' = f(x, y)$ 的两端从x_n 到x_{n+1}求积分得

$$y(x_{n+1}) = y(x_n) + \int_{x_n}^{x_{n+1}} f(x, y(x))\mathbf{d}x \tag{4.5.1}$$

为了获得$y(x_{n+1})$得近似值，只要计算出其中的积分项即可。

基于插值原理可以建立一系列数值积分方法，运用这些方法可以导出求解微分方程的一系列计算格式。

设已构造处$f(x, y(x))$ 的插值多项式$P_r(x)$ 那么，计算$\int_{x_n}^{x_{n+1}} P_r(x)\mathbf{d}x$作为积分项的近似值，可以将式4.5.1 离散为得到下列计算公式

$$y_{n+1} = y_n + \int_{x_n}^{x_{n+1}} P_r(x)\mathbf{d}x \tag{4.5.2}$$

4.5.2 Adams显示格式

记$f_k = f(x_k, y_k)$，先用r+1 各数据点$(x_n, f_n), (x_{n-1}, f_{n-1}), \cdots, (x_{n-r}, f_{n-r})$构造插值多项式$P_r(x)$，运用Newton后插公式可写出

$$P_r(x_n + th) = \sum_{j=0}^{r} (-1)^j \binom{-t}{j} \Delta_j f_{n-j}$$

其中$t = \frac{x-x_n}{h}$，Δ^j 表示j 阶的向前差分。将$P_r(x)$ 的的表达式带入4.5.2 中可的Adams显示格式

$$y_{n+1} = y_n + hsum_{j=0}^{r} \alpha_j \Delta^j f_{n-j} \tag{4.5.3}$$

其中$\alpha_j = (-1)^j \int_0^1 \binom{-t}{j}\mathbf{d}t$为不依赖$n$ 和r 的系数。

实际计算式，可以将式4.5.3 中的差分$\Delta^j f_{n-j} = \sum_{i=0}^{r} (-1)^i \binom{j}{i} f_{n-j}$展开，可改写为

$$y_{n+1} = y_n + h\sum_{i=0}^{r} \beta_{ri} f_{n-i}, \beta_{ri} = (-1)^i \sum_{j=i}^{r} \binom{j}{i} \alpha_j \tag{4.5.4}$$

展开部分的推到我不理解，最终结果中$beta_{ri}$ 中包含了ji，这个应该是从j 中选择i 个，应该满足条件 $j \geq i$，但是在βri 的表达式中，$i \geq j$。不理解。

上式是含有参数r 的一族格式，$r + 1$ 为格式步数，一步显示Adams格式为Euler格式，两步显示Adams格式为

$$y_{n+1} = y_n + \frac{h}{2}(3f_n - f_{n-1}) \tag{4.5.5}$$

四步显示Adams格式为

$$y_{n+1} = y_n + \frac{h}{24}(55f_n - 59f_{n-1} + 37f_{n-2} - 9f_{n-3}) \tag{4.5.6}$$

4.5.3 Amams隐式格式

Amams显示方法实际是个外推过程，效果不够理想。为了改进比较效果，可以使用内插过程，即用$x_{n+1}, x_n, \cdots, x_{n-r+1}$ 为插值节点，得到函数$P_r(x)$ ，然后重复上述推导过程得出Adams隐式格式，即

$$y_{n+1} = y_n + h\sum_{j=0}^{r} \alpha_j^\star \Delta^j f_{n-j+1}, \alpha_j^\star = (-1)^j \int_{-1}^{0} \binom{-t}{j} \mathbf{d}t \tag{4.5.7}$$

将差分展开，可写为

$$y_{n+1} = y_n + h\sum_{i=0}^{r} \beta_{rj}^\star f_{n-i+1}, \beta_{ri}^\star (-1)^i \sum_{j=i}^{r} \binom{j}{i} \alpha_j^\star \tag{4.5.8}$$

一步隐式Adams格式为后退的Euler格式，两步隐式Adams格式为梯形格式。而四步隐式Adams为

$$y_{n+1} = y_n + \frac{h}{24}(9f_{n+1} + 19f_n - 5f_{n-1} + f_{n-2}) \tag{4.5.9}$$

4.5.4 Adams预测-校正系统

分析Adams方法的误差方法与分析Euler方法一样。因此，显示格式4.5.6的局部截断误差为

$$y(x_{n+1}) - y_{n+1} \approx \frac{251}{720} h^5 y^{(5)}(x_n) \tag{4.5.10}$$

隐式格式的局部截断误差为

$$y(x_{n+1}) - y_{n+1} \approx -\frac{19}{720} h^5 y^{(5)}(x_n) \tag{4.5.11}$$

显示格式和隐式格式都具有四阶精度，可以匹配成下列Adams预测校正系统

1. 预测

$$\overline{y}_{n+1} = y_n + \frac{h}{24}(55f_n - 59f_{n-1} + 37f_{n-2} - 9f_{n-3})$$
$$\overline{f}_{n+1} = f(x_{n+1}, \overline{y}_{n+1})$$

2. 校正

$$y_{n+1} = y_n + \frac{h}{24}(9\overline{y}_{n+1} + 19f_n - 5f_{n-1} + f_{n-2}$$
$$f_{n+1} = f(x_{n+1}, y_{n+1})$$

这个预测校正法是四步法，计算时需要用到前三步的信息，必须借助某些单步法为它提供初值。

4.5.5 基于Taylor展开的构造方法

一般的线性多步格式具有以下形式

$$y_{n+1} = \sum_{k=0}^{r} \alpha_k y_{n-k} + h \sum_{k=-1}^{r} \beta_k y'_{n-k} \tag{4.5.12}$$

当$\beta_{-1} = 0$时，格式是现实的，否则是隐式的。

设$y_{n-k} = y(x_{n-k}), y'_{n-k} = y'(x_{n-k})$，由Taylor展开得

$$\begin{cases} y_{n-k} = \sum_{j=0}^{p} \frac{(-kh)^j}{j!} y_n^{(j)} + \frac{(-kh)^{p+1}}{(p+1)} y_n^{(p+1)} + \cdots \\ y'_{n-k} = \sum_{j=1}^{p} \frac{(-kh)^{j-1}}{(j-1)!} y_n^{(j)} + \frac{(-kh)^p}{p!} y_n^{(p+1)} \cdots \end{cases}$$

代入4.5.12得

$$\begin{aligned} y_{n+1} = (\sum_{k=0}^{r} \alpha_k) y_n + \sum_{j=1}^{p} \frac{h^j}{j!} [\sum_{k=1}^{r} (-k)^j \alpha_k + j \sum_{k=-1}^{r} (-k)^{j-1} \beta_k] y_n^{(j)} + \\ \frac{h^{p+1}}{(p+1)!} [\sum_{k=1}^{r} (-k)^{p+1} \alpha_k + (p+1) \sum_{k=-1}^{r} (-k)^p \beta_k] y_n^{(p+1)} + \cdots \end{aligned} \tag{4.5.13}$$

要使上述格式为p 阶精度，只要零展开式与$y(x_{n+1})$ 得Taylor展开式符合到h^p 项，因此要求满足以下条件

$$\begin{cases} \sum_{k=0}^{r} \alpha_k = 1 \\ \sum_{k=1}^{r} (-k)^j \alpha_k + j \sum_{k=-1}^{r} (-k)^{j-1} \beta_k = 1 \end{cases} \tag{4.5.14}$$

四阶显示格式为

$$y_{n+1} = \alpha_0 + y_n + \alpha_1 y_{n-1} + \alpha_2 y_{n-2} + h(\beta_0 y'_n + \beta_1 y'_{n-1} + \beta_2 y'_{n-2} + \beta_3 y'_{n-3} \tag{4.5.15}$$

系数应当满足条件

$$\begin{cases} \alpha_0 + \alpha_1 + \alpha_2 = 1 \\ -\alpha_1 - 2\alpha_2 + \beta_0 + \beta_1 + \beta_2 + \beta_3 = 1 \\ \alpha_1 + 4\alpha_2 - 2\beta_1 - 4\beta_2 - 6\beta_3 = 1 \\ -\alpha_1 - 8\alpha_2 + 3\beta_1 + 12\beta_2 + 27\beta_3 = 1 \\ \alpha_1 + 16\alpha_2 - 4\beta_1 - 32\beta_2 - 108\beta_3 = 1 \end{cases} \tag{4.5.16}$$

上述方程组有7 个待定系数，但是只有5 个方程，因此有两个自由度。

令$\alpha_1 = \alpha_2 = 0$ 可解得四阶Adams格式。

四阶隐式格式为

$$y_{n+1} = \alpha_0 + y_n + \alpha_1 y_{n-1} + \alpha_2 y_{n-2} + h(\beta_{-1} y'_{n+1} + \beta_0 y'_n + \beta_1 y'_{n-1} + \beta_2 y'_{n-2}) \tag{4.5.17}$$

系数应当满足条件

$$\begin{cases} \alpha_0+\alpha_1+\alpha_2=1 \\ -\alpha_1-2\alpha_2+\beta_{-1}+\beta_0+\beta_1+\beta_2=1 \\ \alpha_1+4\alpha_2+2\beta_{-1}-2\beta_1-4\beta_2=1 \\ -\alpha_1-8\alpha_2+3\beta_{-1}+3\beta_1+12\beta_2=1 \\ \alpha_1+16\alpha_2+4\beta_{-1}-4\beta_1-32\beta_2=1 \end{cases} \tag{4.5.18}$$

取$\alpha_1=\alpha_2=0$ 可得四阶隐式Adams格式

4.5.6 Milne格式

另一类格式使用y_{n-3} 而不使用y'_{n-3}，其形式为

$$y_{n+1}=\alpha_0y_n+\alpha_1y_{n-1}+\alpha_2y_{n-2}+\alpha_3y_{n-3}+h(\beta_0y'_n+\beta_1y'_{n-1}+\beta_2y'_{n-2}) \tag{4.5.19}$$

运用Taylor展开方法可以导出如上式一类的方法。这类方法中包含Milne格式，即

$$y_{n+1}=y_{n-3}+\frac{4h}{3}(2y'_n-y'_{n-1}+2y'_{n-2}) \tag{4.5.20}$$

其局部截断误差为

$$y(x_{n+1})-y_{n+1}\approx\frac{14}{15}h^5y_n^{(5)} \tag{4.5.21}$$

Milne格式也可以通过数值积分的途径推到出来。

再在形如4.5.17中适当挑选一个与Milne格式相匹配，令$\alpha_1=1,\alpha_2=0$可得到下列四阶格式

$$y_{n+1}=y_{n-1}+\frac{h}{3}(y'_{n+1}+4y'_n+y'_{n-1}) \tag{4.5.22}$$

上述格式通常称为Simpson格式。

用Simpson格式4.5.22 和Milne格式**??** 匹配，构成下列预测校正系统

1. 预测

$$\overline{y}_{n+1}=y_{n-3}+\frac{4h}{3}(2y'_n-y'_{n-1}+2y'_{n-2})$$
$$\overline{y}'_{n+1}=f(x_{n+1},\overline{y}_{n+1})$$

2. 校正

$$y_{n+1}=y_{n-1}+\frac{h}{3}(\overline{y}'_{n+1}+4y'_n+y'_{n-1})$$
$$y'_{n+1}=f(x_{n+1},y_{n+1})$$

4.5.7 Hamming格式

上述预测校正系统的稳定性较差。

考察式?? 中不显含y'_{n-2} 的一类格式

$$y_{n+1} = \alpha_0 y_n + \alpha_1 y_{n-1} + \alpha_2 y_{n-2} + h(\beta_{-1} y'_{n+1} + \beta_0 y'_n + \beta_1 y'_{n-1} \quad (4.5.23)$$

Hamming法线当$\alpha_1 = 0$ 时格式的稳定性较好，即

$$y_{n+1} = \frac{1}{8}(9y_n - y_{n-2}) + \frac{3h}{8}(y'_{n+1} + 2y'_n - y'_{n-1}) \quad (4.5.24)$$

其局部截断误差为

$$y(x_{n+1}) - y_{n+1} \approx -\frac{h^5}{40} y_n^{(5)} \quad (4.5.25)$$

Hamming格式不能用数值积分方法推到出来。

用Milne格式和Hamming格式相匹配，利用误差公式改进计算结果，可建立以下预测-校正系统

1. 预测
$$p_{n+1} = y_{n-3} + \frac{4h}{3}(2y'_n - y'_{n-1} + 2y'_{n-2})$$
2. 改进
$$m_{n+1} = p_{n+1} - \frac{112}{121}(p_n - c_n)$$
3. 计算
$$m'_{n+1} = f(x_{n+1}, m_{n+1})$$
4. 校正
$$c_{n+1} = \frac{1}{8}(9y_n - y_{n-2}) + \frac{3h}{8}(m'_{n+1} + 2y'_n - y'_{n-1})$$
5. 改进
$$y_{n+1} = c_{n+1} + \frac{9}{121}(p_{n+1} - c_{n+1})$$
6. 计算
$$y'_{n+1} = f(x_{n+1}, y_{n+1})$$

4.6 方程组与高阶方程的情形

4.6.1 一阶方程组

将前面研究方法中的y 和f 理解为向量，那么，之前的计算格式可应用到一阶方程组的情形。

考察一阶方程组

$$y_i' = f_i(x, y_1, y_2, \cdots, y_N)$$

的初值问题，初始条件为

$$y_i(x_0) = y_i^0$$

采用向量记号，记$mathbfy = (y_1, y_2, \cdots, y_N)^T, \mathbf{y}^0 = (y_1^0, y_2^0, \cdots, y_N^0)^T, \mathbf{f} = (f_1, f_2, \cdots, f_N)^T$，则上述方程组的初值问题可表示为$\begin{cases} \mathbf{y}' = f(x, \mathbf{y}) \\ \mathbf{y}(x_0) = \mathbf{y}_0 \end{cases}$，求解这一初值问题的四阶Runge-Kutta格式为

$$\mathbf{y}_{n+1} = \mathbf{y}_n + \frac{h}{6}(\mathbf{k}_1 + 2\mathbf{k}_2 + 2\mathbf{k}_3 + \mathbf{k}_4)$$

其中

$$\begin{cases} \mathbf{k}_1 = f(x_n, \mathbf{y}_n \\ \mathbf{k}_2 = f(x_n + \frac{h}{2}, \mathbf{y}_n + \frac{h}{2}\mathbf{k}_1) \\ \mathbf{k}_3 = f(x_n + \frac{h}{2}, \mathbf{y}_n + \frac{h}{2}\mathbf{k}_2) \\ \mathbf{k}_4 = f(x_n + h, \mathbf{y}_n + h\mathbf{k}_3) \end{cases}$$

4.6.2 化高阶方程组为一阶方程组

高阶微分方程的初值问题，可以归结为一阶方程组来求解。如考察以下m 阶微分方程

$$y^{(m)} = f(x, y, y', \cdots, y^{(m-1)} \tag{4.6.1}$$

初始条件为

$$y(x_0) = y_0, y'(x_0) = y_0', \cdots, y^{(m-1)}(x_0) = y_0^{(m-1)} \tag{4.6.2}$$

引进新变量$y_1 = y, y_2 = y', \cdots, y_m = y^{(m-1)}$，可将$m$ 阶方程4.6.1化为如下的一阶方程组

$$\begin{cases} y_1' = y_2 \\ y_2' = y_3 \\ \vdots \\ y_{m-1}' = y_m \\ y_m' = f(x, y_1, y_2, \cdots, y_m) \end{cases} \tag{4.6.3}$$

此时，初始条件变为

$$y_1(x_0) = y_0, y_2(x_0) = y_0', \cdots, y_m(x_0) = y_0^{(m-1)} \tag{4.6.4}$$

4.7 边值问题的数值解法

对于高阶微分方程，定解条件通常有两种给法，一种是之前的初始条件，另一种是给出积分曲线的边界条件。

设在区间$a < x < b$ 上求解方程$y'' = f(x, y, y')$ ，则边界条件可以给为$y(a) = \alpha, y(b) = \beta$

4.7.1 试射法

试射法的基本思想是将边值问题转换为初值问题求解，根据边界条件寻求与它等价的初值条件。

设凭经验提供斜率m的两个预测值m_1, m_2 ，分别按这两个斜率值试设，求解相应的初值问题，从而得到$y(b)$ 的两个结果β_1, β_2 ，若β_1, β_2 均不满足预定的精度，则使用线性插值法校正斜率，即

$$m_3 = m_1 + \frac{m_2 - m_1}{\beta_2 - \beta_1}(\beta - _1)$$

然后在按斜率值m_3 试射，重复上述过程，直到得到满意的结果

4.7.2 差分方程的建立

差分方法的关键在于恰当地选取差商逼近微分方程中的导数。为逼近二阶导数$y''(x)$，一般用二阶差商——向前差商的向后差商，即

$$y''(x) \approx= \frac{y(x+h) - 2y(x) + y(x-h)}{h^2}$$

将积分区间$[a, b]$ 划分为N 等分，步长$h = \frac{b-a}{h}$ ，节点$x_n = x_0 + nh$。用差商代替导数，可将边值问题离散化为

$$\begin{cases} \frac{y_{n+1} - 2y_n + y_{n-1}}{h^2} = f(x_n, y_n, \frac{y_{n+1} - y_{n-1}}{2h}) \\ y_0 = \alpha, y_N = \beta \end{cases} \tag{4.7.1}$$

若所给方程是如下形式的线性方程

$$y'' + p(x)y' + q(x)y = r(x) \tag{4.7.2}$$

则差分方程的形式为

$$\frac{y_{n+1} - 2y_n + y_{n-1}}{h^2} + p_n \frac{y_{n+1} - y_{n-1}}{2h} + q_n y_n + r_n \tag{4.7.3}$$

其中p,q,r 的下标n 表示在节点x_n 取值。利用边界条件消去y_0 和y_N ，整理可得到关于y_n 的方程组

$$\begin{cases}(-2+h^2q_1)y_1+(1+\frac{h}{2}p_1)y_2=h^2r_1-(1-\frac{h}{2}p_1)\alpha\\(1-\frac{h}{2}p_n)y_{n-1}+(-2+h^2q_n)y_n+(1+\frac{h}{2}p_n)y_{n+1}=h^2r_n(2\le n\le N-2\\(1-\frac{h}{2}p_{N-1})y_{N-2}+(-2+h^2q_{N-1})y_{N-1}=h^2r_{N-1}-(1+\frac{h}{2}p_{N-1})\beta\end{cases} \tag{4.7.4}$$

这个方程组是三对角型的，即

$$\begin{pmatrix}-2+h^2q_1 & 1+\frac{h}{2}p_1 & & & \\ 1-\frac{h}{2}p_2 & -2+h^2q_2 & 1+\frac{h}{2}p_2 & & \\ & \ddots & \ddots & \ddots & \\ & & 1-\frac{h}{2}p_{N-2} & -2+h^2q_{N-2} & 1+\frac{h}{2}p_{N-2}\\ & & & 1-\frac{h}{2}p_{N-1} & -2+h^2q_{N-1}\end{pmatrix}$$

上述方程可以用追赶法求解。

4.7.3 差分问题的可解性

定理 14 对于一组不全相等的数y_n，记

$$l(y_n)=\frac{y_{n+1}-2y_n+y_{n-1}}{h^2}-q_ny_n$$

假定$l(y_n)\ge 0$，则y_n 的正的最大值只能是y_0 或y_N ；如果$l(y_n)<0$，则y_n 的负的最小值只能是y_0 或y_N

定理 15 差分问题的解存在并且是唯一的

4.7.4 差分方法的收敛性

定理 16 设y_n 是差分问题的解，而$y(x_n)$ 是边值问题的解$y(x)$ 在节点x_n 的值，则截断误差$e_n=y(x_n)-y_n$ 有下列估计式

$$|e_n|\le\frac{M(b-a)^2}{96}h^2,M=\max_{a\le x\le b}|y^{(4)}(x)| \tag{4.7.5}$$

5 方程求根

5.1 根的搜索

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方程$f(x)=0$在区间(a,b) 内一定有实根，称$[a,b]$ 为方程$f(x)=0$的有根区间。

5.1.1 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，按照某个预定的步长h进行一次根的搜索，即检查$x_k=a+kh$上函数值$f(x_k)$的符号，一旦节点x_k与端点a的函数值异号，则可以确定一个缩小了的有根区间$[x_{k-1},x_k]$。

逐步搜索法的步长h选择是关键，当h足够小，可以求得任意精度的近似跟，但计算量会增大。

5.1.2 二分法

二分法在考察有根区间$[a,b]$，取中点$x_0=\frac{a+b}{2}$将它分为两半，检查$f(x_0)$和$f(a)$是否同号，然后得到新的有根区间，重复上述步骤，即可找到满足精度条件的根$x^\star$。

二分法算法简单且收敛性可以得到保证。

5.2 迭代法

5.2.1 迭代过程的收敛性

考察下列形式的方程

$$x=\phi(x) \tag{5.2.1}$$

这种方程是隐式的，不能直接得出它的根。如果给出根的某个猜测值x_0，将它带入上式右端，可求得$x_1=\phi(x_0)$。有可取x_1作为猜测值，进一步得到$x_2=\phi(x_1)$，如此反复迭代，如果按公式

$$x_{k+1}=\phi(x_k), k=0,1,2,\cdots \tag{5.2.2}$$

确定的数列$\{x_k\}$有极限$x^\star=\lim_{k\to\infty}x_k$，则迭代过程收敛，此时极限值$x^\star$就是方程5.2.1的根

迭代法是一种逐次逼近法，基本思想是将隐式方程5.2.1归结为一组显示的计算公式5.2.2，实质是一个逐步显示化的过程。

定义 23 *假定函数$\phi(x)$满足下列两项条件*

1. *对于任意$x\in[a,b]$，有*

$$a\le\phi(x)\le b \tag{5.2.3}$$

2. *存在正数$L<1$，使对于任意$x\in[a,b]$有*

$$|\phi'(x)|\le L<1 \tag{5.2.4}$$

则迭代过程$x_{k+1}=\phi(x_k)$ 对于任意初值$x_0\in[a,b]$ 均收敛与方程$x=\phi(x)$ 的根$x^\star$，且具有如下的误差估计式

$$|x_k-x^\star|\le\frac{L^k}{1-L}|x_1-x_0| \tag{5.2.5}$$

定义 24 若存在$x^\star$ 的某个领域$R:|x-x^\star|\le\delta$，使得迭代过程$x_{k+1}=\phi(x_k)$ 对于任意初值$x_0\in R$ 均收敛，则称迭代过程$x_{k+1}=\phi(x_k)$ 在根$x^\star$ 邻近具有局部收敛性。

定理 17 设$x^\star$ 为方程$x=\phi(x)$ 的根，$\phi'(x)$ 在$x^\star$ 的邻近连续且$|\phi'(x^\star)|<1$，则迭代过程$x_{k+1}=\phi(x_k)$在$x^\star$ 邻近具有局部收敛性。

5.2.2 迭代公式的加工

若迭代过程收敛，只要迭代足够多次，就可以使结果达到任意精度。但是若迭代过程收敛缓慢，则计算量会变得很大。

设x_0 是根$x^\star$ 的某个预测值，用迭代公式迭代一次的$x_1=\phi(x_0)$，另，根据微分中值公式有

$$x_1-x^\star=\phi'(\xi)(x_0-x^\star)$$

假定$\phi'(x)$ 改变不大，近似取某个近似值L，则可得

$$x^\star=\frac{1}{1-L}x_1-\frac{L}{1-L}x_0 \tag{5.2.6}$$

因此，可以期望上式右端是一个比x_1 更好的近似值。

用$\overline{x}_k$ 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示为

1. 校正，$\overline{x}_{k+1}=\phi(x_k)$
2. 改进，$x_{k+1}=\overline{x}_{k+1}+\frac{L}{1-L}(\overline{x}_{k+1}-x_k)$

上述加速方案中包含了导数相关的信息，实际可能不方便。

仍设$x^\star$ 的某个猜测值为x_0，进行两次迭代有$x_1=\phi(x_0),x_2=\phi(x_1)$，由于$x_2-x^\star\approx L(x_1-x^\star)$ 将它与5.2.6 联立可消去未知的L，由此可得

$$x^\star\approx\frac{x_0x_2-x_1^2}{x_0-2x_1+x_2}=x_2-\frac{(x_2-x_1)^2}{x_0-2x_1+x_2}$$

上述公式中不包含L，但是需要用到两次迭代计算，计算公式如下

1. 校正，$\tilde{x}_{k+1}=\phi(x_k)$
2. 再校正，$\overline{x}_{k+1}=\phi(\tilde{x}_{k+1})$

3. 改进，$x_{k+1}=\overline{x}_{k+1}-\frac{(\overline{x}_{k+1}-\tilde{x}_{k+1})^2}{\overline{x}_{k+1}-2\tilde{x}_{k+1}+x_k}$

上述过程称为Aitken方法。

将发散得迭代公式通过Aitken方法处理后可以获得相当好得收敛性。

5.3 Newton法

5.3.1 Newton公式

对于方程$f(x)=0$，为了使用迭代法，需要针对所给得函数$f(x)$ 构造合适的迭代函数$\phi(x)$ 。

若令$\phi(x)=x+f(x)$，则迭代公式为

$$x_{k+1}=x_k+f(x_k) \tag{5.3.1}$$

运用加速技巧，其加速公式为

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

记$M=L-1$，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，其相应的迭代函数为

$$\phi(x)=x-\frac{f(x)}{M} \tag{5.3.2}$$

其中L 是$\phi'(x)$ 的估计值，而$\phi(x)=x+f(x)$，因此，M 实际上是$f'(x)$ 的估计值，使用$f'(x)$ 替换就可以得到Newton公式

$$x_{k+1}=x_k-\frac{f(x_k)}{f'(x_k)} \tag{5.3.3}$$

5.3.2 Newton法的几何解释

5.3.3 Newton法的局部收敛性

迭代过程的收敛速度指再接近收敛的过程中迭代误差的下降速度.

定义 25 设迭代过程$x_{k+1}=\phi(x_k)$ 收敛于方程$x=\phi(x)$ 的根$x^\star$，如果迭代误差$e_k=x_k-x^\star$ 当$k\to\infty$ 时成立下列渐近关系是

$$\frac{e_{k+1}}{e_k^p}\to C$$

则称该迭代过程是p 阶收敛的，当$p=1$ 时称为线性收敛，当$p>1$ 时称为超线性收敛，$p=2$ 是称为平方收敛。

定理 18 对于迭代过程$x_{k+1} = \phi(x_k)$ ，如果$\phi^{(p)}(x)$ 在所求根$x^\star$邻近连续，且

$$\begin{cases} \phi'(x^\star) = \phi''(x^\star) = \cdots = \phi^{(p-1)}(x^\star) = 0 \\ \phi^{(p)}(x^\star) \neq 0 \end{cases} \tag{5.3.4}$$

则迭代过程在点$x^\star$ 邻近是p 阶收敛的。

根据上述定理可知，迭代过程的收敛速度依赖于迭代函数$\phi(x)$ 的选取。

对于Newton公式5.3.3 ，根$x^\star$ 的邻近是平方收敛的。

Newton法的计算步骤为

1. 准备，选定初始近似值x_0，计算$f_0 = f(x_0), f_0' = f'(x_0)$
2. 迭代，按公式$x_1 = x_0 - f_0/f_0'$ 迭代一次，得新得近似值x_1，计算$f_1 = f(x_1), f_1' = f'(x_1)$
3. 控制，如果x_1 满足$|\delta|\epsilon_1$或$|f_1| < \epsilon_2$ ，则终止迭代，以x_1 作为所求得根，否则转步骤4，其中ϵ_1, ϵ_2 是允许误差，er

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

   其中C 是误差控制常数，一般可取$C = 1$
4. 修改，若迭代次数达到预先指定得次数N 或$f_1' = 0$ 则方法失败，否则，以(x_1, f_1, f') 代替(x_0, f_0, f_0') 转步骤2继续。

5.3.4 Newton下山法

Newton法的收敛性依赖与初值x_0 的选取，如果x_0 偏离所求的根$x^\star$ 比较远，则Newton法可能发散。

为了防止迭代发散，对迭代过程再附加一项要求，即具有单调性

$$|f(x_{k+1}| < f(x_k) \tag{5.3.5}$$

满足这项要求的算法称为下山法。

将Newton法与下山法结合起来，可在下山法保证迭代收敛的前提下，用Newton法加快收敛速度。因此将Newton法的计算结果与前一步的近似值适当加权平均作为新的改进值，即

$$x_{k+1} = \lambda \overline{x}_{k+1} + (1 - \lambda)x_k \tag{5.3.6}$$

其中$\lambda(0 < \lambda \leq 1)$ 称为下山因子。挑选下山因子时希望能使单调性成立。

5.4 弦截法与抛物线法

当函数f 比较复杂时，提供它的导数值往往是困难的。根据插值原理，可以利用迭代过程中的信息$f(x_k), f(x_{k-1}), \cdots$ 来回避导数值$f'(x_k)$ 的计算。

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$P_r(x)$ ，并适当选取$P_r(x) = 0$ 的一个根作为$f(x) = 0$ 的新近似根x_{k+1}。这确定了一个迭代过程，记迭代函数为ϕ，且

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

5.4.1 弦截法

再上述迭代过程中，$r = 1$ 时称为弦截法.

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$P_1(x)$，并利用$P_1(x) = 0$ 的根作为$f(x) = 0$ 的新近似根x_{k+1}。构造的插值多项式为

$$P_1(x) = f(x_k) + \frac{f(x_k) - f(x_{k-1})}{x_k - x_{k-1}}(x - x_k) \tag{5.4.1}$$

因此，有

$$x_{k+1} = x_k - \frac{f(x_k)}{f(x_k) - f(x_{k-1})}(x_k - x_{k-1}) \tag{5.4.2}$$

上述公式相当于Newton公式中的导数$f'(x)$ 用差商代替的结果

上述公式的几何意义是 x_k, x_{k-1}在函数$f(x)$ 上的点P_k, P_{k-1} 的连线与x轴交点的横坐标。

弦截法需要用到前面的两步结果，因此，必须先给出开始的两个值x_0, x_1

定理 19 假设$f(x)$ 在根$x^\star$ 的领域$\Delta : |x - x^\star| \le \delta$ 内具有二阶连续导数，且对于任意$x \in \Delta$，有$f'(x) \neq 0$，又初值$x_0, x_1 \in \Delta$ ，当领域充分小时，弦截法将按阶$p = \frac{1+\sqrt{5}}{2} \approx 1.618$ 收敛到根$x^\star$

弦截法的计算步骤如下

1. 准确，选取初值x_0, x_1，计算函数值$f_0 = f(x_0), f_1 = f(x_1)$

2. 迭代，按公式
$$x_2 = x_1 - \frac{f_1(x_1 - x_0)}{f_1 - f_0}$$
迭代一次得到新的近似值x_2，计算函数值$f_2 = f(x_2)$

3. 控制。若x_2 满足$|\delta| \leq \epsilon_1$ 或$|f_2| \leq \epsilon_2$ ，则迭代过程收敛，终止迭代，否则执行步骤4，其中

$$\delta = \begin{cases} |x_2 - x_1| & |x_2| < C \\ \frac{|x_2 - x_1|}{|x_2|} & |x_2| \geq C \end{cases}$$

其中C 是预先指定的控制数

4. 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，否则转步骤2继续迭代。

5.4.2 抛物线法

当$r = 2$时的迭代方法称为抛物线法。

设已知方程$f(x) = 0$的三个近似根x_0, x_1, x_2 ，以这三个节点构造二次插值多项式$P_2(x)$ ，并适当取$P_2(x)$ 的一个零点x_{k+1}作为新的近似根。

插值多项式为

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

有两个零点

$$x_{k+1} = x_k - \frac{2f(x_k)}{\omega \pm \sqrt{\omega^2 - 4f(x_k)f[x_k, x_{k-1}, x_{k-2}]}} \tag{5.4.3}$$

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

为了确定x_{k+1}的值，假定x_k 更接近根$x^\star$ ，为了保证精度，可选择于x_k 接近的值作为新的近似根x_{k+1}。只需令根式前的符号与ω 的符号相同即可

对于抛物线法，迭代误差有下列渐近关系式

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

抛物线法比弦截法收敛更快。

抛物线法的计算步骤如下

1. 准确，选定初始近似值x_0, x_1, x_2 ，并计算相应的值f_0, f_1, f_2 以及

$$\lambda_2 = \frac{x_2 - x_1}{x_1 - x_0}$$

2. 迭代，计算

$$\begin{cases} \delta_2 = 1 + \lambda_2 \\ a = f_0\lambda_2^2 - f_1\lambda_2\delta_2 + f_2\lambda_2 \\ b = f_0\lambda_2^2 - f_1 + \delta_2^2 + f_2(\lambda_2 + \delta_2) \\ c = f_2\delta_2 \\ \lambda_3 = \frac{-2c}{b \pm \sqrt{b^2 - 4ac}} \end{cases}$$

3. 控制，如果x_3 满足$|\delta| \leq \epsilon_1$ 或$|f_3| < \epsilon_3$，则终止迭代，以x_3 作为所求的根，否则执行步骤4

4. 修改，如果迭代次数达到预先指定的次数N ，则认为迭代过程不收敛，计算失败。否则转步骤2继续迭代。

5.5 代数方程求根

如果$f(x)$ 是多项式，由于多项式的特殊性，可以提供更为有效的算法。

5.5.1 多项式求值的秦九韶算法

设给定多项式

$$f(x) = a_0x^n + a_1 = x^{n-1} + \cdots + a_{n-1}x + a_n$$

其中系数a_i 均为实数。用一次式$x - x_0$ 除$f(x)$ ，商记作$P(x)$ ，余数显然等于$f(x_0)$，即

$$f(x) = f(x_0) + (x - x_0)P(x) \tag{5.5.1}$$

因此需要具体确定$P(x)$ 与$f(x_0)$ 。令

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

带入式5.5.1 ，比较两端同次幂的系数可得

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

从而有

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

进一步考察$f(x)$ 的Taylor展开式

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

按5.5.1 的形式表示，则

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

因此，导数$f'(x_0)$ 可看作是$P(x)$ 用因式$x-x_0$ 相除的余数，即

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

其中$Q(x)$ 是$n-2$ 次多项式，设

$$Q(x)=c_0x^{n-2}+c_1x^{n-3}+\cdots+c_{n-3}x+c_{n-2}$$

系数为

$$\begin{cases}c_0=b_0\\ c_i=b_i+x_0c_{i-1},1\le i\le n-1\\ f'(x_0)=c_{n-1}\end{cases}$$

继续这一过程可以依次求出$f(x)$ 在点x_0 的各阶导数。

5.5.2 代数方程的Newton法

就多项式方程考察Newton公式，根据秦久韶算法可以方便的求出$f(x_k)$和导数值$f'(x_k)$。

5.5.3 劈因子法

若能从多项式中分离出一个二次因式$\omega^\star=x^2+u^\star x+v^\star$ ，就可以获得它的一对共轭复根。劈因子法的基本思想是，从某个近似的二次因子出发，用某种迭代过程使之逐步精确化。

用二次式$\omega(x)$ 除$f(x)$ ，商为$p(x)$ ，是个$n-2$ 次多项式，余式为一次式，记为r_0x+r_1 ，记

$$f(x)=(x^2+ux+v)p(x)+r_0x+r_1 \tag{5.5.2}$$

r_0,r_1 为u,v 的函数，即

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

劈因子法的目的就是逐步修改u,v 的值，使余数r_0,r_1 变得很小。考察方程

$$\begin{cases}r_0(u,v)=0\\ r_1(u,v)=0\end{cases} \tag{5.5.3}$$

这是关于u,v 得非线性方程组，设有解$(u^\star,v^\star)$ ，将方程左端在(u,v) 展开到一阶项有

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

使用Newton法思想将方程组5.5.3 线性化，可得到下列线性方程组

$$\begin{cases} r_0 + \frac{\partial r_0}{\partial u}\Delta u + \frac{\partial r_0}{\partial v}\Delta v = 0 \\ r_1 + \frac{\partial r_1}{\partial u}\Delta u + \frac{\partial r_1}{\partial v}\Delta v = 0 \end{cases} \tag{5.5.4}$$

从方程组5.5.4 解出增量$\Delta u, \Delta v$ ，即可得到改进的二次因式

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

计算方程组的系数的步骤如下

1. 计算r_0, r_1，将

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

带入式5.5.2 ，并比较各次幂的系数，可知

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

可解得

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

2. 计算$\frac{\partial r_0}{\partial v}, \frac{\partial r_1}{\partial v}$，对5.5.2 求导有

$$P(x) = -(x^2 + ux + v)\frac{\partial P}{\partial v} + s_0 x + x_1 \tag{5.5.5}$$

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

3. 计算$\frac{\partial r_0}{\partial u}, \frac{\partial r_1}{\partial u}$，对5.5.2求导有

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

另外，根据式5.5.5有

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

比较上面两式可得，$\frac{\partial r_0}{\partial u} = us_0 - s_1, \frac{\partial r_1}{\partial u} = vs_0$

6 解线性方程组的直接方法

6.1 引言

关于线性方程组的数值解法一般有两类

1. 直接法，经过有限步算数运算即可求得方程组精确解的方法。是解低阶稠密矩阵方程组的有效方法。
2. 迭代法，用某种极限过程取逐步逼近线性方程组精确解的方法。是解大型稀疏矩阵方程组的重要方法。

6.2 Gauss消去法

6.2.1 消元手续

设有线性方程组

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n = b_2 \\ \vdots a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}xn = b_n \end{cases} \tag{6.2.1}$$

上述方程组也可以写成矩阵式$Ax = b$ 其中

$$\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{n1} & an2 & \cdots & a_{nn} \end{pmatrix}, \mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}, \mathbf{b} = \begin{pmatrix} b_1 \\ b_2 \\ \cdots \\ b_n \end{pmatrix}$$

消去法解方程组的基本思想是用逐次消去未知数的方法把原来方程组$\mathbf{Ax} = \mathbf{b}$ 化为与其等价的三角方程组。

设$a_{ii}^{(i)} \neq 0$ ，则三角方程组的求解公式为

$$\begin{cases} x_n = \frac{b_n^{(n)}}{a_{nn}^{(n)}} \\ x_k = (b_k^{(k)} - \sum_{j=k+1}^{n} a_{kj}^{(k)} x_j)/a_{kk}^{(k)} \quad k = n-1, n-2, \cdots \end{cases} \tag{6.5.2}$$

定理 20 *如果A 为n 阶非奇异矩阵，则可通过Gauss消去法将方程组6.2.1化为三角方程组。约化的主元素$a_{ii}^{(i)} \neq 0$的充要条件是矩阵$\mathbf{A}$ 的顺序主子式$D_i \neq 0$ ，即*

$$D_1 = a_{11} \neq 0$$

$$D_i = \begin{vmatrix} a_{11} & \cdots & a_{1k} \\ \vdots & & \vdots \\ a_{i1} & \cdots & a_{ii} \end{vmatrix}$$

如果$\mathbf{A}$ 的顺序主子式$D_k \neq 0$ ，则

$$\begin{cases} a_{11}^{(1)} = D_1 \\ a_{kk}^{(k)} = D_k / D_{k-1} \end{cases}$$

定理 21 *如果n 阶矩阵$\mathbf{A}$ 的所有顺序主子式均不为零，即$D_i \neq 0$ ，则可通过Gauss消去法，将方程组6.2.1 化为三角方程组。*

计算公式如下

1. 消元计算$(k = 1, 2, \cdots, n-1)$

$$\begin{aligned} m_{ik} &= \frac{a_{ik}^{(k)}}{a_{kk}^{(k)}} \\ a_{ij}^{(k+1)} &= a_{ij}^{(k)} - m_{ik} a_{kj}^{(k)} \\ b_i^{(k+1)} &= b_i^{(k)} - m_{ik} b_k^{(k)} \end{aligned}$$

2. 回代计算，求解公式为6.5.2

6.2.2 矩阵的三角分解

设式6.2.1 中$\mathbf{A}$ 的各顺序主子式均不为零，由于对$\mathbf{A}$ 施行行的初等变换相当于用初等矩阵左乘$\mathbf{A}$ ，于是对式6.2.1 施行第一次消元后，$\mathbf{A}^{(1)}$ 化为$\mathbf{A}^{(2)}$，$\mathbf{b}^{(1)}$ 化为$\mathbf{b}^{(2)}$，即

$$\mathbf{L}_1 A^{(1)} = A^{(2)}, \mathbf{L}_1 b^{(1)} = b^{(2)}$$

其中

$$\mathbf{L}_1 = \begin{pmatrix} 1 & & & & \\ -m_{21} & 1 & & & \\ -m_{31} & & 1 & & \\ \vdots & & & \ddots & \\ -m_{n_1} & & & & 1 \end{pmatrix}$$

重复消元过程，最后得到

$$\begin{cases} \mathbf{L}_{n-1} \cdots \mathbf{L}_2 \mathbf{L}_1 \mathbf{A}^{(1)} = \mathbf{A}^{(n)} \\ \mathbf{L}_{n-1} \cdots \mathbf{L}_2 \mathbf{L}_1 \mathbf{b}^{(1)} = \mathbf{b}^{(n)} \end{cases} \tag{6.2.3}$$

其中

$$\mathbf{L}_k = \begin{pmatrix} 1 & & & & \\ & \ddots & & & \\ & & 1 & & \\ & & -m_{k+1,k} & 1 & \\ & & \vdots & & \ddots \\ & & -m_{nk} & & 1 \end{pmatrix}$$

将上三角矩阵$\mathbf{A}^{(n)}$ 记作$\mathbf{U}$，由式6.2.3可得

$$\mathbf{A} = \mathbf{L}_1^{-1}\mathbf{L}_2^{-1}\cdots\mathbf{L}_{n-1}^{-1}\mathbf{U} = \mathbf{L}\mathbf{U}$$

其中

$$L = \begin{pmatrix} 1 & & & \\ m_{21} & 1 & & \\ m_{31} & m_{32} & & \\ \vdots & \vdots & \ddots & \ddots \\ m_{n1} & m_{n2} & \cdots & m_{n,n-1} \end{pmatrix}$$

Gauss消去法实质上产生了一个将$\mathbf{A}$分解为两个三角矩阵相乘的因式分解。

定理 22 设$\mathbf{A}$ 为n 阶矩阵，如果$\mathbf{A}$ 的顺序主子式$D_i \neq 0$ ，则$\mathbf{A}$ 可分解为一个单位下三角矩阵$\mathbf{L}$ 和一个上三角矩阵$\mathbf{U}$ 乘积，且这种分解是唯一的。

6.2.3 计算量

消元过程的计算量，第一步计算乘数m_{i1} 需要$n-1$ 次除法运算，计算$a_{ij}^{(2)}$ 需要$(n-1)^2$ 次乘法运算及$(n-1)^2$ 次加减运算，总计有$n(n-1)(2n-1)/6$次加减运算，$n(n-1)(2n-1)/6$ 次乘法运算，$n(n-1)/2$ 除法运算。

计算$b^{(n)}$ 的计算量，乘除法次数为$n(n-1)/2$ ，加减法次数为$n(n-1)/2$

解$A^{(n)}x = b^{(n)}$ 所需的计算量，乘除法次数为$n(n+1)/2$，加减法次数为$n(n-1)/2$

因此，解方程的总乘除法次数和加减法次数为分别为 $n^3/3 + n^2 - n/3, n(n-1)(2n+5)/6$

6.3 Gauss主元素消去法

在使用Gauss消去法时，消元过程可能会出现$a_{kk}^{(k)}$的情况，此时消去法将无法进行。即使$a_{kk}^{(k)} \neq 0$ ，但很小时，也会导致其他元素数量级的严重增长和舍入误差的扩散，使得计算解不可靠。

在采用Gauss消去法解方程时，应避免采用绝对值小的主元素，最好每一步都选取系数矩阵中绝对值最大的元素作为主元素，使得Gauss消去法具有较好的数值稳定性。

6.3.1 完全主元素消去法

设方程组6.2.1 的增广矩阵为

$$\mathbf{B}=\left[\begin{array}{cccccc|c} a_{11} & a_{12} & \cdots & a1j_1 & \cdots & a_{1n} & b_1 \\ a_{21} & a_{22} & \cdots & a_{2j_1} & \cdots & a_{2n} & b_2 \\ \vdots & \vdots & & \vdots & & \vdots & \vdots \\ a_{i_1 1} & a_{i_1 2} & \cdots & a_{i_1 j_1} & \cdots & a_{i_1 n} & b_{i_1} \\ \vdots & \vdots & & \vdots & & \vdots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nj_1} & \cdots & a_{nn} & b_n \end{array}\right]$$

首先在$\mathbf{A}$ 中选取绝对值最大的元素作为主元素，然后交换$\mathbf{B}$ 的第一行与第i_1行，第一列与j_1 列，经第一次消元计算的$(\mathbf{A},\mathbf{b})\to(\mathbf{A}^{(2)},\mathbf{b}^{(2)})$ 。重复上述过程，可将原方程组化为

$$\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ & a_{22} & \cdots & a_{2n} \\ & & \ddots & \vdots \\ & & & a_{nn} \end{pmatrix}\begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}=\begin{pmatrix} b_1 & b_2 & \vdots & b_n \end{pmatrix}$$

其中$y_1,y_2,\cdots,y_n$ 的次序为未知数$x_1,x_2,\cdots,x_n$ 调换后的次序，回代求解的

$$\begin{cases} y_n=b_n/a_{nn} \\ y_i=(b_i-\sum_{j=i+1}^{n}a_{ij}y_j)/a_{ii} \end{cases}$$

6.3.2 列主元素消去法

列主元素消去法仅考虑依次按列选主元素，然后换行使之变道主元的位置上，在进行消元计算。

列主元素消去法算法步骤为设$\mathbf{Ax}=\mathbf{b}$ ，消元结果冲掉$\mathbf{A}$，乘数m_{ij} 冲掉a_{ij}，计算解$\mathbf{x}$ 冲掉常数项$\mathbf{b}$ ，行列式存放在$det\mathbf{A}$ 。

1. $det\mathbf{A}\leftarrow 1$，对于$k=1,2,\cdots,n-1$，做到步7

2. 按列选主元素$|a_{i_k k}|=\max_{k\le i\le n}|a_{ik}|$。

3. 若$a_{i_k k}=0$，则$det\mathbf{A}\leftarrow 0$，计算停止.

4. 如果$i_k = k$ ，则转步5，否则换行

$$a_{kj} \leftrightarrow a_{i_k j}, b_k \leftrightarrow b_{i_k}, det\mathbf{A} \leftarrow -det\mathbf{A}$$

5. 计算乘数m_{ik}，$a_{ik} \leftarrow m_{ik} a_{ik}/a_{kk}$

6. 消元计算

$$a_{ij} \leftarrow a_{ij} - m_{ik} a_{kj}, b_i < b_i - m_{ik} b_k$$

7. $det\mathbf{A} \leftarrow a_{kk} det\mathbf{A}$

8. 回代求解

$$b_n \leftarrow b_n/a_{nn}, b_i \leftarrow (b_i - \sum_{j=i+1}^{n} a_{ij} b_j)/a_{ii}$$

9. $det\mathbf{A} \leftarrow a_{nn} det\mathbf{A}$

定理 23 如果**A** 为非奇异矩阵，则存在排列矩阵**P** 使

$$\mathbf{PA} = \mathbf{LU}$$

其中**L** 为单位下三角阵，**U**为上三角阵。

L 元素存放在数组**A** 的下三角部分，**U** 元素存放在**A** 的上三角部分。

6.3.3 Gauss-Jordan消去法

Gauss-Jordan消去法使Gauss消去法的一种修正，即消去对角线下方和上方的元素。

在进行第k 计算时，考虑对上述矩阵的第k 行上、下都进行消元计算，具体步骤为

1. 按列选主元素，即确定i_k 使$|a_{i_k k}| = \max_{k \le i \le n} |a_{ik}|$ 。

2. 当$i_k \neq k$ 时，交换$(\mathbf{A}, \mathbf{b})$ 第k 行与第i_k 行元素。

3. 计算乘数$m_{ik} = -a_{ik}/a_{kk}, m_{kk} = 1/a_{kk}$ 。

4. 消元计算

$$a_{ij} \leftarrow a_{ij} + m_{ik} a_{kj}, b_i \leftarrow b_i + m_{ik} b_k$$

5. 计算主行$a_{kj} \leftarrow a_{kj} m_{kk}, b_k \leftarrow b_k m_{kk}$

在上述过程结束后，有

$$(\mathbf{A},\mathbf{b}) \to (\mathbf{A}^{(k+1)},\mathbf{b}^{(k+1)}) = \left[\begin{array}{cccc|c} 1 & & & & \hat{b}_1 \\ & 1 & & & \hat{b}_2 \\ & & \ddots & & \vdots \\ & & & 1 & \hat{b}_n \end{array}\right]$$

Gauss-Jordan消去法将$\mathbf{A}$ 约化为单位矩阵，解就是常数项，计算量大约$n^3/2$次乘除法运算。

定理 24 *设$\mathbf{A}$ 为非奇异矩阵，方程组$\mathbf{AX} = \mathbf{I}_n$的增广矩阵为$C = (A|I_n)$，如果对$C$应用Gauss-Jordan消去法化为 $(I_n|T)$，则 $A^{-1} = T$。*

Gauss-Jordan列主元素方法求逆的步骤为

1. $det\mathbf{A} \leftarrow 1$，对于$k = 1, 2, \cdots, n$ 做到步骤8。
2. 按列选主元素$|a_{i_k k}| = \max_{k\le i\le n}; c_0 \leftarrow a_{i_k k}, Ip(k) \leftarrow i_k$。
3. 如果$c_0 = 0$，则计算停止。
4. 如果$i_k = k$，则转步骤5；否则换行

$$a_{kj} \leftrightarrow a_{i_k j}, det\mathbf{A} \leftarrow -det\mathbf{A}$$

5. $det\mathbf{A} \leftarrow det\mathbf{A}c_0$
6. 计算$h \leftarrow a_{kk} \leftarrow \frac{1}{c_0}$，$a_{ik} \leftarrow m_{ik}h$
7. 消元计算 $a_{ij} \leftarrow a_{ij} + m_{ik}a_{kj}$
8. 计算主行$a_{kj} \leftarrow a_{kj}h$
9. 交换列对于 $k = n-1, n-2, \cdots, 2, 1$
   (a) t = Ip(k)
   (b) 如果$t > k$，换行$a_{ik} \leftrightarrow a_{it}$
   (c) 继续循环

6.4 Gauss消去法的变形

6.4.1 直接三角分解法

将Gauss消去法改写为紧凑形式，可以从矩阵$\mathbf{A}$ 的元素得到计算$\mathbf{L}, \mathbf{U}$元素的递推公式，而不需要任何中间步骤，就是所谓的直接三角法。求解6.2.1 的问题等价与求解以下两个三角方程组

1. $\mathbf{Ly}=\mathbf{y}$，求$\mathbf{y}$

2. $\mathbf{Ux}=\mathbf{y}$，求 $\mathbf{x}$

1. 不选主元的三角分解法设$\mathbf{A}$ 为非奇异矩阵，且有分解式$\mathbf{A}=\mathbf{LU}$，其中$\mathbf{L}$ 为单位下三角阵，$\mathbf{U}$ 为上三角阵，即

$$\mathbf{A}=\begin{pmatrix}1 & & & \\ l_{21} & 1 & & \\ \vdots & \ddots & \ddots & \\ l_{n1} & \cdots & l_{n,n-1} & 1\end{pmatrix}\begin{pmatrix}u_{11} & u_{12} & \cdots & u_{1n} \\ & u_{22} & \cdots & u_{2n} \\ & & \ddots & \vdots \\ & & & u_{nn}\end{pmatrix} \tag{6.4.1}$$

$\mathbf{L},\mathbf{U}$ 的元素可以由n 步直接计算定出，其中，第r 步定出$\mathbf{U}$ 的第r 行和$\mathbf{L}$ 的第r 列元素，由式6.4.1 有

$$a_{1i}=u_{1i}$$

$$l_{i1}=a_{i1}/u_{11}$$

于是可以得到$\mathbf{U}$ 的第一行元素和$\mathbf{L}$ 的第一列元素。已给出$\mathbf{U}$ 的前$r-1$ 行元素和$\mathbf{L}$ 的第$r-1$ 列元素，由式6.4.1 可求得$\mathbf{U}$ 的第r 行和$\mathbf{L}$ 的第r 列。

$$u_{ri}=a_{ri}-\sum_{k=1}^{r-1}l_{rk}u_{ki}$$

$$l_{ir}=(a_{ir}-\sum_{k=1}^{r-1}l_{ik}u_{kr})/u_{rr}$$

算法步骤为

(a) 根据公式计算 $\mathbf{U}$的第一行和$\mathbf{L}$ 的第一列。

(b) 计算$\mathbf{U}$ 的第r 行和$\mathbf{L}$ 的第r 列

(c) 解$\mathbf{Ly}=\mathbf{b}$

$$\begin{cases}y_1=b_1 \\ y_i=b_i-\sum_{k=1}^{i-1}l_{ik}y_k\end{cases} \tag{6.4.2}$$

(d) 求解$\mathbf{Ux}=\mathbf{y}$

$$\begin{cases}x_n=y_n/u_{nn} \\ x_i=(y_i-\sum_{k=i+1}^{n}u_{ik}x_k)/u_{ii}\end{cases} \tag{6.4.3}$$

直接分解法大约需要$\frac{n^3}{3}$ 次乘除运算，和Gauss消去法的计算量基本相似。上述分解公式又称为Doolittle分解公式。

2. 选主元的三角分解法上述方法和Gauss方法有一样的问题，当主元为零或主元的绝对值很小的时候，可能会引入误差的累计。此时可以通过交换行来进行分解。在第r 步分解时，为了避免用小的数u_{rr} 作除数，引进量

$$s_i = a_{ir} - \sum_{k=1}^{r-1} l_{ik}u_{kr}$$

因此有

$$u_{rr} = s_r, l_{ir} = s_i/s_r, \max_{r\le i\le n} |s_i| = |s_{i_r}|$$

用s_{i_r} 作为u_{rr} 交换$\mathbf{A}$ 的r 行与i_r 行元素位置的新元素，然后再进行第r 步分解计算。算法步骤为

(a) 按不选主元算法计算到第4步

(b) 计算$s_i = a_{ir} - \sum_{k=1}^{r-1} l_{ik}u_{kr}$

(c) 选主元，$|s_{i_r} = \max_{r\le i\le n} |s_i|$ ，$Ip(r) \leftarrow i_r$。

(d) 交换 $\mathbf{A}$ 的第r 行与i_r元素$a_{ri} \leftrightarrow a_{i_r i}$

(e) 计算$\mathbf{U}$ 的第r 行元素，$\mathbf{L}$ 的第r 列元素

$$\begin{cases} a_{rr} = u_{rr} = s_r \\ a_{ir} \leftarrow l_{ir} = s_i/u_{rr} = a_{ir}/a_{rr} \\ a_{ri} \leftarrow u_{ri} = a_{ri} - \sum_{k=1}^{r-1} l_{rk}u_{ki} \end{cases}$$

(f) 求解$\mathbf{Ly} = \mathbf{Pb}$ 和$\mathbf{Ux} = \mathbf{y}$

利用上述算法可以实现$\mathbf{PA} = \mathbf{LU}$ 三角分解，可以用于计算$\mathbf{A}$ 的逆矩阵。

6.4.2 平方根法

平方根法利用对称正定矩阵的三角分解而得到的求解对称正定方程组的一种有效方法。设$\mathbf{A}$ 为对称阵，且$\mathbf{A}$ 的所有顺序主子式均不为零，$\mathbf{A}$ 可唯一分解为式6.4.1 的形式。

定理 25 设$\mathbf{A}$ 为n 阶对称帧，且$\mathbf{A}$ 的所有顺序主子式均不为零，则$\mathbf{A}$ 可唯一分解为

$$A = LDL^T$$

其中$\mathbf{L}$ 为单位下三角阵，$\mathbf{D}$ 为对角阵。

由$\mathbf{A}$ 的对称正定的，因此，$d_1 = D_1 > 0, d_i = D_i/D_{i-1} > 0$ ，于是有

$$D = \begin{pmatrix} d & & \\ & \ddots & \\ & & d_n \end{pmatrix} = \begin{pmatrix} \sqrt{d_1} & & \\ & \ddots & \\ & & \sqrt{d_n} \end{pmatrix} \begin{pmatrix} \sqrt{d_1} & & \\ & \ddots & \\ & & \sqrt{d_n} \end{pmatrix} = D^{\frac{1}{2}} D^{\frac{1}{2}}$$

因此可得

$$\mathbf{A} = \mathbf{LDL}^T = \mathbf{LD}^{\frac{1}{2}}\mathbf{D}^{\frac{1}{2}}\mathbf{L}^T = (\mathbf{LD}^{\frac{1}{2}})(\mathbf{LD}^{\frac{1}{2}})^T = \mathbf{L}_1\mathbf{L}_1^T$$

其中$\mathbf{L}_{=}\mathbf{LD}^{\frac{1}{2}}$ 为下三角阵.

定理 26 *如果$\mathbf{A}$ 为n 阶对称正定矩阵，则存在一个实的非奇异下三角阵$\mathbf{L}$ 使得$\mathbf{A} = \mathbf{LL}^T$ ，当限定$\mathbf{L}$ 的对角元素为正时，这种分解时唯一的。*

解对称正定方程组$\mathbf{Ax} = \mathbf{b}$ 的平方根法计算公式步骤为

1. $l_{jj} = (a_{jj} - \sum_{k=1}^{j-} l_{jk}^2)^{\frac{1}{2}}$
2. $l_{ij} = (a_{ij} - \sum_{k=1}^{j-1} l_{ik}l_{jk})/l_{jj}$
3. 方程$\mathbf{Ly} = \mathbf{b}$ 求$\mathbf{y}$
$$y_i = (b_i - \sum_{k=1}^{i-1} l_{ik}y_k)/l_{ii}$$
4. 方程$\mathbf{L^T x} = \mathbf{y}$ 求$\mathbf{x}$
$$x_i = (b_i - \sum_{k=i+1}^{n} l_{ki}x_k)/l_{ii}$$

由第一步可知，$a_{jj} = \sum_{k=1}^{j} l_{jk}^2$ ，所以

$$l_{jk}^2 \le a_{jj} \le \max_{1\le j\le n}\{a_{jj}\}$$

$$\max_{j,k}\{l_{jk}^2\} \le \max_{1\le j\le n}\{a_{jj}\}$$

分解过程中元素l_{jk} 的数量级不会增长且对角元素l_{jj} 恒为正数，因此平方根法是一个数值稳定的方法。

平方根法大约需要$n^3/6$ 次乘除法运算，一般是LU 分解法计算量的一般。

在上述计算中$\mathbf{L}$ 的元素l_{ii} 需要用到开放运算，为了避免开方运算，使用$\mathbf{A} = \mathbf{LDL}^T$ ，按行计算$\mathbf{L}$ 的元素l_{ij} ，由矩阵乘法得

$$a_{ij} = \sum_{k=1}^{n}(\mathbf{LD})_{ik}(\mathbf{L}^T)_{kj} = \sum_{k=1}^{n} l_{ik}d_k l_{jk} = \sum_{k=1}^{j-1} l_{ik}d_k l_{jk} + l_{ij}d_j l_{jj}$$

因此可得到下列计算$\mathbf{L}$ 和$\mathbf{D}$ 的公式

1. $l_{ij} = (a_{ij} - \sum_{k=1}^{j-1} l_{ik} d_k l_{jk})/d_j$
2. $d_i = a_{ii} - \sum_{k=1}^{i-1} l_{ik}^2 d_k$

引入$t_{ij} = l_{ij} d_j$避免重复计算，按行计算$\mathbf{L}, \mathbf{T}$ 的公式为

1. $t_{ij} = a_{ij} - \sum_{k=1}^{j-1} t_{ik} l_{jk}$
2. $l_{ij} = t_{ij}/d_j$
3. $d_i = a_{ii} - \sum_{k=1}^{i-1} t_{ik} l_{ik}$
4. 方程$\mathbf{Ly} = \mathbf{b}$ 求解$\mathbf{y}$

$$\begin{cases} y_1 = b_1 \\ y_i = b_i - \sum_{k=1}^{i-} l_{ik} y_h \end{cases}$$

5. 方程$\mathbf{DL^T x} = \mathbf{y}$ 求解$\mathbf{x}$

$$\begin{cases} x_n = y_n/d_n \\ x_i = y_i/d_i - \sum_{k=i+1}^{n} l_{ki} x_k \end{cases}$$

上述方法称为改进的平方根法。

6.4.3 追赶法

对角占优的三对角方程组

$$\begin{pmatrix} b_1 & c_1 & & & & & \\ a_2 & b_2 & c_2 & & & & \\ & \ddots & \ddots & \ddots & & & \\ & & a_i & b_i & c_i & & \\ & & & \ddots & \ddots & \ddots & \\ & & & & a_{n-1} & b_{n-1} & c_{n-1} \\ & & & & & a_n & b_n \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_i \\ \vdots \\ x_{n-1} \\ x_n \end{pmatrix} = \begin{pmatrix} f_1 \\ f_2 \\ \vdots \\ f_i \\ \vdots \\ f_{n-1} \\ f_n \end{pmatrix}$$

简记$\mathbf{Ax} = \mathbf{f}$ 其中$\mathbf{A}$ 满足下列对角占有条件

1. $|b_1| > |c_1| > 0$
2. $|b_i| \geq |a_i| + |c_i|$
3. $|b_n| > |a_n| > 0$

根据上一阶可知，可将$\mathbf{A}$ 分解为两个三角阵的乘积，

$$\mathbf{A}=\mathbf{LU}=\begin{pmatrix}\alpha_1 & & & & \\ \gamma_2 & \alpha_2 & & & \\ & \gamma_3 & \alpha_3 & & \\ & & \ddots & \ddots & \\ & & & \gamma_n & \alpha_n\end{pmatrix}\begin{pmatrix}1 & \beta_1 & & & \\ & 1 & \beta_2 & & \\ & & \ddots & \ddots & \\ & & & 1 & \beta_{n-1} \\ & & & & 1\end{pmatrix}$$

其中$\alpha_i,\beta_i,\gamma_i$ 为待定系数，比较两端可得

$$\begin{cases}b_1=\alpha_1 \\ c_1=\alpha_1\beta_1 \\ \alpha_i=\gamma_i \\ b_i=\gamma_i\beta_{i-1}+\alpha_i \\ c_i=\alpha_i\beta_i\end{cases} \tag{6.4.6}$$

求解$\mathbf{Ax}=\mathbf{f}$ 等价于解两个三角方程组$\mathbf{Ly}=\mathbf{f}$ 和$\mathbf{Ux}=\mathbf{y}$ ，先后求$\mathbf{y}$ 与$\mathbf{x}$ 从而得到以下解三对角方程组的追赶法公式

1. 计算$\{\beta_i\}$ 的递推公式

$$\beta_1=c_1/b_1,\beta_i=c_i/(b_i-a_i\beta_{i-1})$$

2. 解$\mathbf{Ly}=\mathbf{f}$

$$y_1=f_1/b_1,y_i=(f_i-a_iy_{i-1})/(b_i-a_i\beta_{i-1})$$

3. 解$\mathbf{Ux}=\mathbf{y}$

$$x_n=y_n,x_i=y_i-\beta_ix_{i+1}$$

将计算系数β 和y 的过程称为追的过程，将计算解x 的过程为赶的过程。

定理 27 设有三对角方程组$\mathbf{Ax}=\mathbf{f}$ ，其中$\mathbf{A}$ 满足对角占优条件，则$\mathbf{A}$ 为非奇异矩阵且由追赶法计算公式中$\{\alpha_i\},\{\beta_i\}$ 满足

1. $0<|\beta_i|<1$

2. $0<|c_i|\leq|b_i|-|a_i|<|\alpha_i|<|b_i|+|a_i|$且$0<|b_n|-|a_n|<|\alpha_n|<|b_n|+|a_n|$

追赶法实际上是吧Gauss法应用到求解三对角方程上，计算量为$5n-4$ 次乘除法运算。

上述定理说明追赶法计算公式中不会出现中间结果数量级的巨大增长和舍入误差的严重积累。

6.5 向量和矩阵的范数

定义 26 设$\mathbf{x}=(x_1,x_2,\cdots,x_n)^T,\mathbf{y}=(y_1,y_2,\cdots,y_n)^T\in R^n$ 将实数$(\mathbf{x},\mathbf{y})=\mathbf{y}^T\mathbf{x}=\sum_{i=1}^n x_iy_i$称为向量$\mathbf{x},\mathbf{y}$ 的数量积。将非负实数$\|x\|_2=(\mathbf{x},\mathbf{x})^{\frac{1}{2}}=(\sum_{i=1}^n x_i^2)^{\frac{1}{2}}$称为向量$\mathbf{x}$ 的*Euclid*范数。

定理 28 设$\mathbf{x},\mathbf{y}\in\mathbf{R}^n$，则

1. $(\mathbf{x},\mathbf{x})=0$ 当且仅当$\mathbf{x}=\mathbf{0}$ 时成立。
2. $(\alpha\mathbf{x},\mathbf{y})=\alpha(\mathbf{x},\mathbf{y})$，$\alpha$ 为实数
3. $(\mathbf{x},\mathbf{y})=(\mathbf{x},\mathbf{y})$
4. $(\mathbf{x}_1+\mathbf{x}_2,\mathbf{y})=(\mathbf{x}_1,\mathbf{y})+(\mathbf{x}_2,\mathbf{y})$
5. $|(\mathbf{x},\mathbf{y})|\le\|\mathbf{x}\|_2\|\mathbf{y}\|_2$。当且仅当$\mathbf{x}$ 与$\mathbf{y}$ 线性相关时成立。
6. $\|\mathbf{x}+\mathbf{y}\|_2\le\|\mathbf{x}\|_2+\|\mathbf{y}\|_2$

定义 27 如果向量$\mathbf{x}\in\mathbf{R}^n$ 的某个实值函数$N(\mathbf{x})=\|x\|$ ，满足条件

1. $\|\mathbf{x}\|\ge 0$
2. $\|\alpha x\|=|\alpha|\|\mathbf{x}\|,\forall\alpha\in\mathbf{R}$
3. $\|\mathbf{x}+\mathbf{y}\|\le\|\mathbf{x}\|+\|\mathbf{y}\|$

则称$N(\mathbf{x})$是$\mathbf{R}^n$ 上的一个向量范数

以下是几种常用的范数

1. ∞-范数（最大范数）：$\|\mathbf{x}\|_\infty=\max_{i\le i\le n}|x_i|$
2. 1-范数：$\|\mathbf{x}\|_1=\sum_{i=1}^n|x_i|$
3. 2-范数：$\|\mathbf{x}\|_2=(\mathbf{x},(\mathbf{x})^{\frac{1}{2}}=(\sum_{i=1}^n x_i^2)^{\frac{1}{2}}$
4. p-范数：$\|\mathbf{x}\|_p=(\sum_{i=1}^n|x_i|^p)^{\frac{1}{p}}$，其中$p\in[1,\infty)$

定义 28 设$\{\mathbf{x}^{(k)}\}$ 为$\mathbf{R}^n$ 中一向量序列，$\mathbf{x}^\star\in\mathbf{R}^n$，记$\mathbf{x}^{(k)}=(x_1^{(k)},x_2^{(k)},\cdots,x_n^{(k)})^Tfi\mathbf{x}^\star=(x_1^\star,x_2^\star,\cdots,x_n^\star)^T$ 。若$\lim_{k\to\infty}x_i^{(k)}=x_i^\star$，则称$\mathbf{x}^{(k)}$ 收敛于向量$\mathbf{x}^\star$ ，记为

$$\lim_{k\to\infty}\mathbf{x}^{(k)}=\mathbf{x}^\star$$

定理 29 设非负函数$N(\mathbf{x})=|\mathbf{x}\|$为$\mathbf{R}^n$ 上任一向量范数，则$N(\mathbf{x})$ 是$\mathbf{x}$ 分量$x_1,x_2,\cdots,x_n$的连续函数。

定理 30 设$\|\mathbf{x}\|_s, \|\mathbf{x}\|_t$为$\mathbf{R}^n$ 上向量的任意两种范数，则存在常数$c_1, c_2 > 0$使得

$$c_1\|\mathbf{x}\|_s \leq \|\mathbf{x}\|_t \leq c_2\|\mathbf{x}\|_s$$

定理 31 $\lim_{k\to\infty} \mathbf{x}^{(k)} = \mathbf{x}^\star \leftrightarrow \|\mathbf{x}^{(k)} - \mathbf{x}^\star\| \to 0$

将向量范数推广到矩阵上有，用$\mathbf{R}^{n\times n}$ 表示$n \times n$ 矩阵几何，则由$\mathbf{R}^{n\times n}$上2-范数可以得到$R^{n\times n}$ 中矩阵的一种范数

$$\mathbf{F}(\mathbf{A}) = \|\mathbf{A}\|_F = (\sum_{i,j=1}^{n} a_{ij}^2)^{\frac{1}{2}}$$

称为$\mathbf{A}$ 的Frobenius范数。

定义 29 如果矩阵$\mathbf{A} \in \mathbf{R}^{n\times n}$ 的某个非负的实值函数$N(\mathbf{A}) = \|\mathbf{A}\|$ ，满足条件

1. $\|\mathbf{A}\| \geq 0$
2. $\|c\mathbf{A}\| = |c|\|\mathbf{A}\|$
3. $\|\mathbf{A} + \mathbf{B}\| \leq \|\mathbf{A}\| + \|\mathbf{B}\|$
4. $\|\mathbf{AB}\| \leq \|\mathbf{A}\|\|\mathbf{B}\|$

则称$N(\mathbf{A})$ 是$\mathbf{R}^{n\times n}$ 上的一个矩阵范数。

定义 30 设$\mathbf{x} \in \mathbf{R}^n, \mathbf{A} \in \mathbf{R}^{n\times n}$ ，给出一种向量范数$\|\mathbf{x}\|_v$相应地定义一个矩阵的非负函数

$$\|\mathbf{A}\| = \max_{x\neq 0} \frac{\|\mathbf{Ax}\|_v}{\|\mathbf{x}\|_v}$$

可验证$\|\mathbf{A}\|_v$ 是$\mathbf{R}^{n\times n}$ 上的一个矩阵范数，称为$\mathbf{A}$ 的算子范数。

定理 32 设$\|\mathbf{x}\|_v$ 是$\mathbf{R}^n$ 上的一个向量范数，则$\|\mathbf{A}\|_v$ 是$\mathbf{R}^{n\times n}$ 上的矩阵范数且满足相容条件

$$\|\mathbf{Ax}\|_v \leq \|\mathbf{A}\|_v\|\mathbf{x}\|_v$$

定理 33 设$\mathbf{x} \in \mathbf{R}^n, \mathbf{A} \in \mathbf{R}^{n\times n}$，则

1. $\|\mathbf{A}\|_\infty = \max_{i\leq i\leq n} \sum_{j=1}^{n} |a_{ij}|$称为$\mathbf{A}$ 的行范数。
2. $\|\mathbf{A}\|_1 = \max_{1\leq j\leq n} \sum_{i=1}^{n} |a_{ij}|$称为$\mathbf{A}$ 的列范数。
3. $\|\mathbf{A}\|_2 = \sqrt{\lambda_{max}(\mathbf{A}^T\mathbf{A})}$称为$\mathbf{A}$ 为2-范数。其中$\lambda_{max}(\mathbf{A}^T\mathbf{A})$ 表示$\mathbf{A}^T\mathbf{A}$的最大特征值。

定义 31 设$\mathbf{A} \in \mathbf{R}^{n\times n}$，则$\rho(\mathbf{A}) \leq \|\mathbf{A}\|$，即$\mathbf{A}$ 的谱半径不超过$\mathbf{A}$ 的谱半径。

定理 34 设$\mathbf{A} \in \mathbf{R}^{n\times n}$，则$\rho(\mathbf{A}) \leq \|\mathbf{A}\|$，即$\mathbf{A}$ 的谱半径不超过$\mathbf{A}$ 的任何一种算子范数。

定理 35 如果$\mathbf{A} \in \mathbf{R}^{n\times n}$ 为对称矩阵，则$\|\mathbf{A}\|_2 = \rho(\mathbf{A})$

定理 36 如果$\|\mathbf{B}\| < 1$，则$\mathbf{I} \pm \mathbf{B}$ 为非奇异矩阵，且$\|(\mathbf{I} \pm \mathbf{B})^{-1}\| \leq \frac{1}{1-\|\mathbf{B}\|}$，其中$\|\cdot\|$ 指矩阵的算子范式。

6.6 误差分析

6.6.1 矩阵的条件数

定义 32 如果矩阵$\mathbf{A}$ 或常数项$\mathbf{b}$ 的微小变化，引起方程组$\mathbf{Ax} = \mathbf{b}$ 解的巨大变化，则称此方程组为病态方程组，矩阵$\mathbf{A}$ 称为病态矩阵，否则称方程组为良态方程组，$\mathbf{A}$ 称为良态矩阵。

定理 37 设$\mathbf{A}$ 是非奇异矩阵，$\mathbf{Ax} = \mathbf{b} \neq \mathbf{0}$，且$\mathbf{A}(\mathbf{x} + \delta\mathbf{x}) = \mathbf{b} + \delta\mathbf{b}$，则

$$\frac{\|\delta\mathbf{x}\|}{\|\mathbf{x}\|} \leq \|\mathbf{A}^{-1}\|\|A\|\frac{\|\delta\mathbf{b}\|}{\mathbf{b}}$$

上述定理给出了解的相对误差上界，常数项$\mathbf{b}$ 的相对误差在解中可能放大$\|\mathbf{A}^{-1}\|\|\mathbf{A}\|$倍。

定理 38 设$\mathbf{A}$ 是非奇异矩阵，$\mathbf{Ax} = \mathbf{b} \neq \mathbf{0}$，且$(\mathbf{A} + \delta\mathbf{A})(\mathbf{x} + \delta\mathbf{x}) = \mathbf{b}$，如果$\|\mathbf{A}^{-1}\|\|\delta\mathbf{A}\| < 1$，则

$$\frac{\|\delta\mathbf{x}\|}{\|\mathbf{x}\|} \leq \frac{\|\mathbf{A}^{-1}\|\|\mathbf{A}\|\frac{\|\delta\mathbf{A}\|}{\|\mathbf{A}\|}}{1 - \|\mathbf{A}^{-1}\|\|\mathbf{A}\|\frac{\|\delta\mathbf{A}\|}{\|\mathbf{A}\|}}$$

如果$\delta\mathbf{A}$ 充分小，且满足条件$\|\mathbf{A}^{-1}\|\|\delta\mathbf{A}\| < 1$，上述定理说明矩阵$\mathbf{A}$ 的相对误差在解中可能放大$\|\mathbf{A}^{-1}\|\|\mathbf{A}\|$ 倍

综上所示，量$\|\mathbf{A}^{-1}\|\|\mathbf{A}\|$ 刻画了方程的病态程度。

定义 33 设$\mathbf{A}$ 为非奇异矩阵，称数$cond(\mathbf{A})_v = \|\mathbf{A}^{-1}\|_v\|\mathbf{A}\|_v$为矩阵$\mathbf{A}$ 的条件数。

当$\mathbf{A}$ 的条件数越大，方程组的病态程度越严重，也难得到方程组的准确的解。

通常使用的条件数有

1. $cond(\mathbf{A})_\infty = \|\mathbf{A}^{-1}\|_\infty \|\mathbf{A}\|_\infty$

2. $\mathbf{A}$ 的谱条件数

$$cond(\mathbf{A})_2 = \|\mathbf{A}\|_2 \|\mathbf{A}^{-1}\|_2 = \sqrt{\frac{\lambda_{max}(\mathbf{A}^T\mathbf{A})}{\lambda_{min}(\mathbf{A}^T\mathbf{A})}}$$

条件数有以下性质

1. 对任何非奇异矩阵$\mathbf{A}$ ，都有$cond(\mathbf{A})_v \geq 1$。

2. 设$\mathbf{A}$ 为非奇异矩阵，c 为不等于零的常数，则$cond(c\mathbf{A})_v = cond(\mathbf{A})_v$

3. 如果$\mathbf{A}$ 为正交矩阵，则$cond(\mathbf{A})_2 = 1$ ；如果$\mathbf{A}$ 为非奇异矩阵，$\mathbf{R}$ 为正交矩阵，则

$$cond(\mathbf{RA})_2 = cond(\mathbf{AR})_2 = cond(\mathbf{A})_2$$

要判断矩阵是否病态，需要计算条件数，而计算$\mathbf{A}^{-1}$ 是比较麻烦的，实际计算发现病态情况的方法有以下几种

1. 如果在$\mathbf{A}$ 的三角约化时出现小主元，那么对大多数矩阵来说，$\mathbf{A}$ 是病态矩阵。

2. 如果$\mathbf{A}$ 的最大特征值和最小特征值之比是大的，则$\mathbf{A}$ 是病态的。

3. 如果系数矩阵的行列式值相对来说很小，或系数矩阵某些行进行线性相关，则$\mathbf{A}$ 可能是病态的。

4. 如果系数矩阵$\mathbf{A}$ 元素间数量级相差很大且无一定规则，则$\mathbf{A}$ 可能是病态的。

病态问题通常不能用选主元素的消去法来解决，一般采用高精度算数运算或预处理方法，即将求解$\mathbf{Ax} = \mathbf{b}$ 的问题转换为求解一阶等价方程组

$$\begin{cases} \mathbf{PAQy} = \mathbf{Pb} \\ \mathbf{y} = \mathbf{Q^{-1}x} \end{cases}$$

通过选择非奇异矩阵$\mathbf{P}, \mathbf{Q}$，使$cond(\mathbf{PAQ}) < cond(\mathbf{A})$ ，一般选择$\mathbf{P}, \mathbf{Q}$ 为对角阵或三角阵。

当矩阵$\mathbf{A}$ 的元素大小不均时，在$\mathbf{A}$ 的行中引进适当的比例因子，对$\mathbf{A}$ 的条件数是有影响的，但不能保证$\mathbf{A}$ 的条件数一定得到改善。

定理 39 *1. 设$\mathbf{A}$ 为非奇异矩阵，$\mathbf{x}$ 是精确解，$\mathbf{Ax} = \mathbf{b} \neq \mathbf{0}$;*

2. 设$\overline{\mathbf{x}}$ 是方程组的近似解，$\mathbf{r} = \mathbf{b} - \mathbf{A}\overline{\mathbf{x}}$则

$$\frac{\|\mathbf{x} - \overline{\mathbf{x}}\|}{\|\mathbf{x}\|} \leq cond(\mathbf{A}) \frac{\mathbf{r}}{\mathbf{b}}$$

6.6.2 舍入误差

设$\bar{\mathbf{x}}$为用选主元素Gauss消去法的计算解，$\mathbf{x}$ 为精确解。Wilkinson等人提出向后误差分析方法，基本思想思想是把计算过程中舍入误差对解的影响归结为原始数据变化对解的影响。即计算解$\bar{\mathbf{x}}$ 是下述扰动方程组的精确解$(\mathbf{A}+\delta\mathbf{A})\mathbf{x}=\mathbf{b}$，其中$\delta\mathbf{A}$ 为某个小矩阵

定理 40 *如果*

1. *设*$\mathbf{A}$ *为*n *阶非奇异矩阵。*
2. *用列主元素消去法阶方程。*
3. *记*$a_k=\max_{1\le i,j\le n}|a_{ij}^{(k)}|, a=\max_{1\le i,j\le n}|a_{ij}|, r=\max_{i\le k\le n}a_k/a$
4. t *为计算机字长，矩阵的阶数*n *满足*$n2^{-t}\le 0.01$，*则*
   (a) *选主元素Gauss消去法计算的三角阵*$\mathbf{L},\mathbf{U}$ *满足*$\mathbf{LU}=\mathbf{A}+\mathbf{E}$，*其中*
   $$|(\mathbf{E})_{ij}|\le 2(n-1)ra2^{-t}$$
   (b) *用选主元素Gauss消去法得到的计算解*$\bar{\mathbf{x}}$ *精确满足*$(\mathbf{A}+\delta\mathbf{A})\mathbf{x}=\mathbf{b}$，*其中*
   $$\|\delta\mathbf{A}\|_\infty\le 1.01(n^3+3n^2)\mathbf{r}\|\mathbf{A}\|_\infty 2^{-t}$$
   (c) *计算解精度估计*
   $$\frac{\|\mathbf{x}-\bar{\mathbf{x}}\|_\infty}{\|\mathbf{x}\|_\infty}\frac{cond(\mathbf{A})_\infty}{1-cond(\mathbf{A})_\infty\frac{\|\delta\mathbf{A}\|_\infty}{\|\mathbf{A}\|_\infty}}\times 1.01(n^3+3n^2)r2^{-t}$$

上式说明计算解$\bar{\mathbf{x}}$ 精度估计式说明，$\bar{\mathbf{x}}$ 得到相对误差限依赖于$cond(\mathbf{A})_\infty$、元素的增长因子、方程组阶数、计算机字长等。

7 解线性方程组的迭代法

7.1 引言

对于大型系数矩阵方程组，利用迭代法求解式合适的。下面距离说明迭代法的基本思想。

设需求解以下方程组

$$\begin{cases}8x_1-3x_2+2x_3=20\\4x_1+11x_2-x_3=33\\6x_1+3x_2+12x_3=35\end{cases}\tag{7.1.1}$$

通常简记为$\mathbf{Ax}=\mathbf{b}$ 其中

$$\mathbf{A}=\begin{pmatrix}8 & -3 & 2\\ 4 & 11 & -1\\ 6 & 3 & 12\end{pmatrix},\mathbf{x}=\begin{pmatrix}x_1\\ x_2\\ x_3\end{pmatrix},\mathbf{b}=\begin{pmatrix}20\\ 33\\ 36\end{pmatrix}$$

上述方程组的精确解为$\mathbf{x}^\star=\begin{pmatrix}3 & 2 & 1\end{pmatrix}^T$ 将上式7.1.1改写为以下形式

$$\begin{cases}x_1=\frac{1}{8}(3x_2-2x_3+20)\\ x_2=\frac{1}{11}(-4x_1+x_3+33)\\ x_3=\frac{1}{12}(-6x_1-3x_2+36)\end{cases}\tag{7.1.2}$$

可表示为$\mathbf{x}=\mathbf{B}_0\mathbf{x}+\mathbf{f}$ 其中

$$\mathbf{B}_0=\begin{pmatrix}0 & \frac{3}{8} & -\frac{2}{8}\\ -\frac{4}{11} & 0 & \frac{1}{11}\\ -\frac{6}{12} & -\frac{3}{12} & 0\end{pmatrix}=\mathbf{I}-\mathbf{D^{-1}A},\mathbf{f}=\begin{pmatrix}\frac{20}{8}\\ \frac{33}{11}\\ \frac{36}{12}\end{pmatrix}=\mathbf{D^{-1}b}$$

任取初始值$\mathbf{x}^{(0)}=(0,0,0)^T$，将这值带入式7.1.2 右端，得到新值

$$\mathbf{x}^{(1)}=(x_1^{(1)},x_2^{(1)},x_3^{(1)})^T=(2.5,3,3)^T$$

再将$\mathbf{x}^{(1)}$ 带入7.1.2 得到$\mathbf{x}^{(2)}$，反复利用这个计算公式，可得到迭代公式

$$\begin{cases}x_1^{(k+1)}=(3x_2^{(k)}-2x_3^{(k)+20}/8\\ x_2^{(k+1)}=(-4x_1^{(k)}+x_3^{(k)}+33)/11\\ x_3^{(k+1)}=(-6x_1^{(k)}-3x_2^{(k)}+36)/12\end{cases}\tag{7.1.3}$$

可写为

$$\mathbf{x}^{(k+1)}=\mathbf{B}_0\mathbf{x}^{(k)}+\mathbf{f}\tag{7.1.3}$$

其中k 表示迭代次数。

定义 34 *1.* 对于给定的方程组$\mathbf{x}=\mathbf{Bx}+\mathbf{f}$，使用迭代公式7.1.3 求得近似解的方法称为迭代法。

2. 如果$\lim_{k\to\infty}\mathbf{x}^{(k)}$存在，称此迭代法收敛，显然$\mathbf{x}^\star$ 是方程组的解，否则称此迭代法发散。

引入误差向量

$$\epsilon^{(k+1)}=\mathbf{x}^{(k+1)}-\mathbf{x}^\star$$

由递推公式可得

$$\epsilon^{(k+1)} = \mathbf{B}\epsilon^{(k)}$$

由递推式得

$$\epsilon^{(k)} = \mathbf{B}\epsilon^{(k-1)} = \cdots = \mathbf{B}\epsilon^{(0)}$$

要考察$\{\mathbf{x}^{(k)}\}$ 得收敛性，即要研究$\mathbf{B}$ 在什么条件下有$\epsilon^{(k)} \to \mathbf{0}$ ，即要研究$\mathbf{B}$ 满足什么条件时有$\mathbf{B}^k \to \mathbf{O}$

7.2 Jacobi迭代法与Gauss-Seidel迭代法

7.2.1 Jacobi迭代法

设有方程组

$$\mathbf{Ax} = \mathbf{b} \tag{7.2.1}$$

$\mathbf{A}$ 为非奇异阵且$a_{ij} \neq 0$ ，将$\mathbf{A}$ 分裂为$\mathbf{A} = \mathbf{D} - \mathbf{L} - \mathbf{U}$，其中

$$\mathbf{D} = \begin{pmatrix} a_{11} & & & \\ & a_{22} & & \\ & & \ddots & \\ & & & a_{nn} \end{pmatrix}, \mathbf{L} = -\begin{pmatrix} 0 & & & \\ a_{21} & 0 & & \\ a_{31} & a_{32} & 0 & \\ \vdots & \vdots & \ddots & \ddots \\ a_{n1} & a_{n2} & \cdots & 0 \end{pmatrix}, \mathbf{L} = -\begin{pmatrix} 0 & a_{12} & a_{13} & \cdots & a_{1n} \\ & 0 & a_{23} & \cdots & a_{2n} \\ & & \ddots & \ddots & \vdots \\ & & & 0 & a_{n-1,n} \\ & & & & 0 \end{pmatrix}$$

将式7.2.1第i 个方程用a_{ii} 去除在移项，得到等价方程组，

$$\mathbf{x} = \mathbf{B_0 x} + \mathbf{f} \tag{7.2.2}$$

其中$\mathbf{B_0} = \mathbf{I} - \mathbf{D^{-1}A} = \mathbf{D^{-1}}(\mathbf{L} + \mathbf{U}\text{fi}\mathbf{f} = \mathbf{D^{-1}b}$ 对方程7.2.2 应用迭代法，得到Jacobi迭代公式。

$$\begin{cases} \mathbf{x}^{(0)} \\ \mathbf{x}^{(k+1)} = \mathbf{B_0 x}^{(k)} + \mathbf{f} \end{cases} \tag{7.2.3}$$

其中$\mathbf{x}^{(k)}$ 为第k 次迭代向量。其中$\mathbf{B_0}$ 称为Jacobi方法迭代矩阵。

7.2.2 Gauss-Seidel迭代法

Jacobi方法中，每步都是用$\mathbf{x}^{(k)}$得全部分量来计算$\mathbf{x}^{(k)}$ 的所有分量，在计算第i 个分量$x_i^{(k+1)}$ 时，已经计算的最新分量$x_1^{(k+1)}, x_2^{(k+1)}, \cdots, x_{i-1}^{(k+1)}$没有倍利用。

解方程组的 Gauss-Seidel迭代法（简称为G-S方法）利用最新计算出来的$k+1$ 次近似$\mathbf{x}^{(k+1)}$ 的分量$x_j^{(k+1)}$

$$\begin{cases} \mathbf{x}^{(0)} = (x_1^{(0)}, x_2^{(0)}, \cdots, x_n^{(0)})^T \\ x_i^{(k+1)} = \frac{1}{a_{ii}}(b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k+1)} - \sum_{j=i+1}^{n} a_{ij}x_j^{(k)} \end{cases} \tag{7.2.3}$$

写成矩阵的形式为

$$\mathbf{D}\mathbf{x}^{(k+1)} = \mathbf{b} + \mathbf{L}\mathbf{x}^{(\mathbf{k+1})} + \mathbf{U}\mathbf{x}^{(\mathbf{k})}, (\mathbf{D}-\mathbf{L})^{(\mathbf{k+1})} = \mathbf{b} + \mathbf{U}\mathbf{x}^{(\mathbf{k})}$$

若设$(\mathbf{D}-\mathbf{L})^{-\mathbf{1}}$ 存在，则

$$\mathbf{x}^{(k+1)} = (\mathbf{D}-\mathbf{L})^{-\mathbf{1}}\mathbf{U}\mathbf{x}^{(\mathbf{k})} + (\mathbf{D}-\mathbf{L})^{-\mathbf{1}}\mathbf{b}$$

因此，Gauss-Seidel迭代公式的矩阵形式为

$$\mathbf{x}^{(k+1)} = \mathbf{G}\mathbf{x}^{(\mathbf{k})} + \mathbf{f} \tag{7.2.4}$$

其中$\mathbf{G} = (\mathbf{D}-\mathbf{L})^{-\mathbf{1}}\mathbf{U}, \mathbf{f} = (\mathbf{D}-\mathbf{L})^{-\mathbf{1}}\mathbf{b}$

Gauss-Seidel迭代法就是对方程组$\mathbf{x} = \mathbf{G}\mathbf{x} + \mathbf{f}$ 应用迭代法，$\mathbf{G}$ 称为Gauss-Seidel迭代法的迭代矩阵。

在一定条件下，Gauss-Seidel迭代法比Jacobi迭代法收敛快。

有部分方程组使用Jacobi方法收敛，而Gauss-Seidel迭代法是发散的。

7.3 迭代法的收敛性

定义 35 设有矩阵序列$\mathbf{A}_k = (a_{ij}^{(k)})_{n\times n}$及$\mathbf{A} = (a_{ij})_{n\times n}$，如果

$$\lim_{k\to\infty} a_{ij}^{(k)} = a_{ij}$$

成立，则称$\{\mathbf{A}_k\}$ 收敛于$\mathbf{A}$ ，记作$\lim_{k\to\infty}\mathbf{A}_k = \mathbf{A}$

矩阵序列极限的概念可以用任何矩阵范数来描述

定理 41 $\lim_{k\to\infty}\mathbf{A}_k = \mathbf{A}$的充要条件是$\|\mathbf{A}_k - \mathbf{A} \to 0\|$

定理 42 设$\mathbf{B} = (b_{ij})_{n\times n}$，则$\mathbf{B}^{\mathbf{k}} \to \mathbf{O}$ 的充要条件是$\rho(\mathbf{B}) < 1$

定理 43 设有方程组

$$\mathbf{x} = \mathbf{B}\mathbf{x} + \mathbf{f} \tag{7.3.1}$$

对于任意初始向量$\mathbf{x}^{(0)}$ 以及任意$\mathbf{f}$ ，解次方程组的迭代法收敛的充要条件是$\rho(\mathbf{B}) < 1$。

考察误差向量$\epsilon^{(\mathbf{k})} = \mathbf{x}^{(\mathbf{k})} - \mathbf{x}^{\star} = \mathbf{B}^{\mathbf{k}}\epsilon^{(\mathbf{0})}$。设$\mathbf{B}$ 有n 个线性无关的特征向量$\mathbf{u_1}, \mathbf{u_2}, \cdots, \mathbf{u_n}$ ，相应的特征值为$\lambda_1, \lambda_2, \cdots, \lambda_n$ ，由$\epsilon^{(0)} = \sum_{i=1}^{n} a_i \mathbf{u}_i$得

$$\epsilon^{(k)} = \mathbf{B}^{\mathbf{k}}\epsilon^{(\mathbf{0})} = \sum_{i=1}^{n} a_i \mathbf{B}^{\mathbf{k}}\mathbf{u_i} = \sum_{i=1}^{n} a_i \lambda_i^k \mathbf{u_i}$$

可知，当$\rho(\mathbf{B}) < 1$ 越小，$\epsilon^{(\mathbf{k})} \to \mathbf{0}$越快，因此，可用量$\rho(\mathbf{B})$ 来刻画迭代法收敛快慢。

根据给定精度来确定迭代次数k ，即使

$$[\rho(\mathbf{B})]^k \leq 10^{-s} \tag{7.3.2}$$

取对数得

$$k \geq \frac{s \ln 10}{-\ln \rho(\mathbf{B})}$$

定义 36 称$R(\mathbf{B}) = -\ln \rho(\mathbf{B})$ 为迭代法得收敛速度。

定理 44 如果方程组得迭代公式为$\mathbf{x}^{(k+1)} = \mathbf{B}\mathbf{x}^{(\mathbf{k})} + \mathbf{f}$ 且迭代矩阵得某一种范数$\|B\|_v = q < 1$，则

1. 迭代法收敛
2. $\|\mathbf{x}^{\star} - \mathbf{x}^{(\mathbf{k})}\|_v \leq \frac{q}{1-q}\|\mathbf{x}^{(\mathbf{k})} - \mathbf{x}^{(\mathbf{k-1})}\|_v$
3. $\|\mathbf{x}^{\star} - \mathbf{x}^{(\mathbf{k})}\|_v \leq \frac{q^k}{1-q}\|\mathbf{x}^{(\mathbf{1})} - \mathbf{x}^{(\mathbf{0})}\|_v$

当$\mathbf{B}$ 的某一种范数$\|\mathbf{B}\| < 1$ 时，如果相邻两次迭代$\|\mathbf{x}^{(k)} - \mathbf{x}^{(k-1)}\| < \epsilon_0$，则$\|\mathbf{x}^{\star} - \mathbf{x}^{(\mathbf{k})}\| \leq \frac{1}{1-q}\epsilon_0$，计算时通常利用$\|\mathbf{x}^{\star} - \mathbf{x}^{(\mathbf{k})}\| < \epsilon_0$ 来作为控制迭代的终止条件。

定理 45 解方程组的*Gauss-Seidel*迭代法收敛的充要条件是$\rho(\mathbf{G}) < 1$ ，其中$\mathbf{G}$ 为*Gauss-Seidel*迭代的迭代矩阵。

定义 37 设$\mathbf{A} = (a_{ij})_{n\times n} \in \mathbf{R}^{n\times n}$

1. 如果矩阵$\mathbf{A}$ 满足条件

$$|a_{ii}| > \sum_{j=1, j\neq i}^{n} |a_{ij}| \tag{7.3.3}$$

   则称$\mathbf{A}$ 为严格对角占优阵.

2. 如果$|a_{ii}| \geq \sum_{j=1, j\neq i}^{n} |a_{ij}|$ 且至少由一个不等式严格成立，称$\mathbf{A}$ 为弱对角占优阵。

定义 38 设$\mathbf{A}=(a_{ij})_n \in \mathbf{R}^{n\times n}$，当$n\geq 2$时，如果存在$n$阶置换矩阵$\mathbf{P}$使

$$\mathbf{P^T AP}=\begin{pmatrix}\mathbf{A}_{11} & \mathbf{A}_{12}\\ \mathbf{O} & \mathbf{A}_{22}\end{pmatrix} \tag{7.3.4}$$

成立，其中$\mathbf{A}_{11}$为r阶子矩阵，$\mathbf{A}_{22}$为$n-r$阶子矩阵，则称$\mathbf{A}$使可约矩阵。如果不存在置换矩阵$\mathbf{P}$使上式成立，则称$\mathbf{A}$使不可约矩阵。

$\mathbf{A}$是可约矩阵，意味着$\mathbf{Ax}=\mathbf{b}$可经过若干行列重排化为两个低阶方程组求解。由$\mathbf{Ax}=\mathbf{b}$可化为$\mathbf{P^T AP}(\mathbf{P^T x})=\mathbf{P^T b}$，且记

$$\mathbf{y}=\mathbf{P^T x}=\begin{pmatrix}\mathbf{y}_1\\ \mathbf{y}_2\end{pmatrix}, \mathbf{P^T b}=\begin{pmatrix}\mathbf{d}_1\\ \mathbf{d}_2\end{pmatrix}$$

其中$\mathbf{y}_1, \mathbf{d}_1$为$r$维向量，于是，求解$\mathbf{Ax}=\mathbf{b}$化为求解

$$\begin{cases}\mathbf{A_{11}y_1}+\mathbf{A_{12}y_2}=\mathbf{d_1}\\ \mathbf{A_{22}y_2}=\mathbf{d_2}\end{cases}$$

定理 46 如果$\mathbf{A}=(a_{ij})_n \in \mathbf{R}^{n\times n}$为严格对角占优阵或为不可约弱对角占优阵，则$\mathbf{A}$是非奇异矩阵。

定理 47 如果$\mathbf{A}\in \mathbf{R}^{n\times m}$为严格对角占优阵或为不可约弱对角占优阵，则对于任意的$\mathbf{x}^{(0)}$，则*Jacobi*迭代法和*Gauss-Seidel*迭代法均收敛。

7.4 解线性方程组的超松弛迭代法

逐次超松弛迭代法（SOR方法）是Gauss-Seidel方法的一种加速方法，是解大型稀疏矩阵方程组的有效方法之一。它计算公式简单，程序设计容易，占用内存少，但需要选择好的加速因子。

设有方程组

$$\mathbf{Ax}=\mathbf{b} \tag{7.3.1}$$

其中$\mathbf{A}\in \mathbf{R}^{n\times n}$为非奇异矩阵，且设$a_{ii}\neq 0$，分解$\mathbf{A}$为

$$\mathbf{A}=\mathbf{D}-\mathbf{L}-\mathbf{U} \tag{7.3.2}$$

已知第k次迭代向量$\mathbf{x}^{(k)}$，及第$k+1$次迭代向量$\mathbf{x}^{(k+1)}$的分量$x_j^{(k+1)}(j=1,2,\cdots,i-1)$，要求计算分量$x_i^{(k+1)}$。

首先使用Gauss-Seidel迭代法定义辅助量

$$\tilde{x}_i^{(k+1)}=\frac{1}{a_{ii}}(b_i-\sum_{j=1}^{i-1}a_{ij}x_j^{(k+1)}-\sum_{j=i+1}^{n}a_{ij}x_j^{(k)}) \tag{7.3.3}$$

再把$x_i^{(k+1)}$ 取为$x_i^{(k)}$ 与$\tilde{x}_i^{(k+1)}$ 某个平均值，即

$$x_i^{(k+1)} = (1-\omega)x_i^{(k)} + \omega\tilde{x}_i^{(k+1)} = x_i^{(k)} + \omega(\tilde{x}_i^{(k+1)} - x_i^{(k)}) \tag{7.3.4}$$

将式7.3.3 代入式7.3.4 可得到逐次超松弛迭代公式

$$\begin{cases} x_i^{(k+1)} = x_i^{(k)} + \frac{\omega}{a_{ii}}(b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k+1)} - \sum_{j=i}^{n} a_{ij}x_j^{(k)} \\ x^{(k)} = (x_1^{(k)}, x_2^{(k)}, \cdots, x_n^{(k)})^T \end{cases} \tag{7.3.5}$$

其中ω 称为松弛因子。

当$\omega = 1$ 时，SOR方法就是Gauss-Seidel迭代法。

当$\omega < 1$ 时称为低松弛法；当$\omega > 1$ 时称为超松弛法。

用$\mathbf{A} = \mathbf{D} - \mathbf{L} - \mathbf{U}$ ，可得SOR方法的矩阵形式为

$$\mathbf{x}^{(k+1)} = \mathbf{L}_\omega \mathbf{x}^{(\mathbf{k})} + \mathbf{f} \tag{7.4.6}$$

其中

$$\mathbf{L}_\omega = (\mathbf{D} - \omega\mathbf{L})^{-1}[(1-\omega)\mathbf{D} + \omega\mathbf{U}], \mathbf{f} = \omega(\mathbf{D} - \omega\mathbf{L})^{-1}\mathbf{b}$$

称矩阵$\mathbf{L}_\omega$ 为SOR方法的迭代矩阵。

定理 48 *设有线性方程组且$a_{ii} \neq 0$，则解方程组的SOR方法收敛的充要条件是*

$$\rho(\mathbf{L}_\omega) < 1$$

希望松弛因子ω 使得迭代过程收敛，即应选择因子ω 使$\rho(\mathbf{L}_\omega) = \min_\omega$

设解的SOR方法收敛，则$0 < \omega < 2$

如果$\mathbf{A}$ 为对称正定阵，且$0 < \omega < 2$ ，则解的SOR方法收敛。

最佳松弛因此公式为

$$\omega_{opt} = \frac{2}{1 + \sqrt{1 - \rho^2(\mathbf{B}_0)}}$$

其中$\rho(\mathbf{B}_0)$ 是Jacobi方法迭代矩阵$\mathbf{B}_0$ 的谱半径。但一般来说，实际计算$\rho(\mathbf{B})_\mathbf{0}$ 较困难。

8 矩阵的特征值与特征向量计算

8.1 引言

定理 49 *如果λ_i 是矩阵$\mathbf{A}$ 的特征值，则有*

1. $\sum_{i=1}^{n} \lambda_i = \sum_{i=1}^{n} a_{ii} = tr\mathbf{A}$

2. $det\mathbf{A} = \lambda_1\lambda_2\cdots\lambda_n$

定理 50 设$\mathbf{A}$ 与$\mathbf{B}$ 为相似矩阵，即存在非奇异阵$\mathbf{T}$使$\mathbf{B} = \mathbf{T^{-1}AT}$，则

1. $\mathbf{A}$ 与$\mathbf{B}$ 有相同的特征值

2. 若 x 是$\mathbf{B}$ 的一个特征向量，则$\mathbf{Tx}$ 是$\mathbf{A}$ 的特征向量。

定理 51 设$\mathbf{A} = (a_{ij})_{n\times n}$ ，则$\mathbf{A}$ 的每一个特征值必属于下述某个圆盘中。

$$|\lambda - a_{ii}| \leq \sum_{j=1,j\neq i}^{n} |a_{ij}|$$

上述定理表明$\mathbf{A}$ 的每一个特征值比属于$\mathbf{A}$ 的一个圆盘中，若一个特征向量的第i 个分量最大，则对应的特征值一定属于第i 个圆盘中。

定义 39 设$\mathbf{A}$ 为n 解实对称矩阵，对于任一非零向量$\mathbf{x}$，称$R(\mathbf{x}) = \frac{(\mathbf{Ax},\mathbf{x})}{(\mathbf{x},\mathbf{x})}$为对应于向量$\mathbf{x}$ 的$Rayleigh$商。

定理 52 设$\mathbf{A} \in \mathbf{R}^{n\times n}$ 为对称矩阵，其特征值依次记为$\lambda_1 \geq \lambda_2 \geq \cdots \lambda_n$ ，对应的特征向量为$\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n$组成规范化正交组，则

1. $\lambda_n \leq R(\mathbf{x}) \leq \lambda_1$

2. $\lambda_1 = \max_{xin\mathbf{R}^n, x\neq 0} R(\mathbf{x})$

3. $\lambda_n = \min_{xin\mathbf{R}^n, x\neq 0} R(\mathbf{x})$

8.2 幂法及反幂法

8.2.1 幂法

在一些问题中，通常只需求出矩阵的按模最大的特征值和相应的特征向量，此时适合使用幂法求解。

幂法是一种计算实矩阵$\mathbf{A}$ 的主特征值的一种迭代法，有点实方法简单，对于稀疏矩阵较合适，但有时收敛速度很慢。

设实矩阵$\mathbf{A}$ 有一个完全的特征向量组，其特征值为$\lambda_1, \lambda_2, \cdots, \lambda_n$ ，相应的特征向量为$\mathbf{x_1}, \mathbf{x_2}, \cdots, \mathbf{x_n}$。已知$\mathbf{A}$ 的主特征值实实根，且满足条件

$$|\lambda_1| > |\lambda_2| \geq |\lambda_3| \geq \cdots \geq |\lambda_n| \tag{8.2.1}$$

幂法的基本思想是任取一个非零的初始向量$\mathbf{v_0}$ ，由矩阵$\mathbf{A}$ 构造一向量序列

$$\begin{cases} \mathbf{v_1} = \mathbf{Av_0} \\ \mathbf{v_2} = \mathbf{Av_1} = \mathbf{A^2v_0} \\ \vdots \\ \mathbf{v_{k+1}} = \mathbf{Av_k} = \mathbf{A^{k+1}v_0} \\ \vdots \end{cases} \tag{8.2.2}$$

称为迭代向量。由假设，$\mathbf{v_0}$ 可表示为

$$\mathbf{v_0} = a_1\mathbf{x_1} + a_2\mathbf{x_2} + \cdots + a_n\mathbf{x_n} \tag{8.2.3}$$

因此有

$$\begin{aligned}
\mathbf{v_k} &= \mathbf{A}\mathbf{v_{k-1}} \\
&= \mathbf{A^k}\mathbf{v_0} \\
&= a_1\lambda_1^k\mathbf{x_1} + a_2\lambda_2^k\mathbf{x_2} + \cdots + a_n\lambda_n\mathbf{x_n} \\
&= \lambda_1^k[a_1\mathbf{x_1} + \sum_{i=2}^{n} a_1(\lambda_i/\lambda_1)^k\mathbf{x_i} \\
&= \lambda_1^k(a_1\mathbf{x_1} + \epsilon_\mathbf{k})
\end{aligned}$$

其中$\epsilon_\mathbf{k} = \sum_{i=2}^{n} a_1(\lambda_i/\lambda_1)^k\mathbf{x_i}$ 。

由假设$|\lambda_i/\lambda_1| < 1$ ，故$\epsilon_\mathbf{k} \to \mathbf{0}$ ，从而

$$\lim_{k\to\infty} \frac{\mathbf{v_k}}{\lambda_1^k} = a_1\mathbf{x_1} \tag{8.2.4}$$

当k 充分大的时

$$\mathbf{v}_k \approx a_1\lambda_1^k\mathbf{x_1} \tag{8.2.5}$$

即迭代向量$\mathbf{v}_k$ 为λ_1 的特征向量的近似向量。

用$(\mathbf{v}_k)_i$ 表示$\mathbf{v}_k$的第i 个分量，则

$$\frac{(\mathbf{v}_{k+1})_i}{(\mathbf{v}_k)_i} = \lambda_1\left\{\frac{a_1(\mathbf{x}_1)_i + (\epsilon_{k+1})_i}{a_1(\mathbf{x}_1)_i + (\epsilon_k)_i}\right\} \tag{8.2.6}$$

因此

$$\lim_{k\to\infty} \frac{(\mathbf{v}_{k+1})_i}{(\mathbf{v}_k)_i} = \lambda_1 \tag{8.2.7}$$

即相邻两迭代相邻分量的比值收敛到主特征值。

以上方法就称为幂法，式8.2.6 的收敛速度由比值$r = \frac{\lambda_2}{\lambda_1}$来确定，$r$ 越小收敛越快。

定理 53 设$\mathbf{A} \in \mathbf{R}^{n\times n}$ 有n 个线性无关的特征向量，主特征值λ_1 满足

$$|\lambda_1| > |\lambda_2| \geq |\lambda_3| \geq \cdots \geq |\lambda_n| \tag{8.2.1}$$

则对于任何非零初始向量$\mathbf{v}_0$，式8.2.4、8.2.7 成立。

当$\mathbf{A}$ 的主特征值为实重根时，即$\lambda_1 = \lambda_2 = \cdots = \lambda_r$ ，定理的结论还是正确的。

应用幂法计算$\mathbf{A}$ 的主特征值λ_1 及对应的特征向量时，如果$|\lambda_1| > 1$ 或$|\lambda_1| < 1$ ，迭代向量$\mathbf{v}_k$ 的各个不等于零的分量将随$k \to \infty$而趋于无穷或趋于零。计算时可能会溢出，此时需要将迭代向量加以规范化。

设$\mathbf{A} \in \mathbf{R}^{n\times n}$ 有n 个线性无关的特征向量，主特征值λ_1 满足$|\lambda_1| > |\lambda_2| \geq |\lambda_3| \geq \cdots \geq |\lambda_n|$，则对于翻译非零初始向量$\mathbf{v}_0 = \mathbf{u}_0$，按下述方法构造向量序列

$$\begin{cases} \mathbf{v}_0 = \mathbf{u}_0 \neq \mathbf{0} \\ \mathbf{v}_k = \mathbf{A}\mathbf{u}_{k-1} \\ \mathbf{u}_k = \frac{\mathbf{v}_k}{\max(\mathbf{v}_k)} \end{cases} \tag{8.2.9}$$

有$\lim_{k\to\infty} \mathbf{u}_k = \frac{\mathbf{x}_1}{\max(\mathbf{x}_1)}, \lim_{k\to\infty} \max(\mathbf{v}_k) = \lambda_1$ 其中$\max(\mathbf{v})$ 表示向量$\mathbf{v}$ 的绝对值最大的分量。

8.2.2 加速方法

再应用幂法计算$\mathbf{A}$ 的主特征值时，其收敛速度由比值$r = \frac{\lambda_1}{\lambda_2}$ 来决定，当r 接近1时，收敛可能很慢，此时应该使用加速方法。

1. 原点平移法

   引进矩阵$\mathbf{B} = \mathbf{A} - p\mathbf{I}$ ，其中p 为选择参数。

   设$\mathbf{A}$ 的特征值为$\lambda_1, \lambda_2, \cdots, \lambda_n$ ，则$\mathbf{B}$ 的特征值为$\lambda_1 - p, \lambda_2 - p, \cdots, \lambda_n - p$ ，且$\mathbf{A}, \mathbf{B}$ 的特征向量相同。如果需要计算$\mathbf{A}$ 的主要特征值λ_1 ，就选择适当的p 使$\lambda_1 - p$ 仍是$\mathbf{B}$ 的主特征值，且使

   $$\left|\frac{\lambda_2 - p}{\lambda_1 - p}\right| < \left|\frac{\lambda_2}{\lambda_1}\right|$$

   对$\mathbf{B}$ 应用幂法，使得在计算$\mathbf{B}$ 的主特征值$\lambda_1 - p$ 的过程中得到加速，虽然可以选择有利的p 值，使幂法得到加速，但选择适当的参数p 使困难的。

   设$\mathbf{A}$ 的特征值满足

   $$\lambda_1 > \lambda_2 \geq \cdots \geq \lambda_{n-1} > \lambda_n \tag{8.2.10}$$

   无论p 如何选择，B 的主特征值为$\lambda_1 - p$ 或$\lambda_n - p$ ，当希望计算λ_1 时，选择p 使$|\lambda_1 - p| > |\lambda_n - p|$，且使收敛速度的比值最小，及

   $$\omega = \max\left\{\frac{|\lambda_2 - p|}{|\lambda_1 - p|}, \frac{|\lambda_n - p|}{|\lambda_1 - p|}\right\} = min$$

   当$\lambda_2 - p = -(\lambda_n - p), p = \frac{\lambda_2 + \lambda_n}{2} = p^\star$ 时，收敛素的的比值为$\frac{\lambda_2 - \lambda_n}{2\lambda_1 - \lambda_2 - \lambda_n}$ ，因此，当$\mathbf{A}$ 的特征值满足条件且λ_2, λ_n 能初步估计时，可以确定$p^\star$ 的近似值。

   当希望计算λ_n 时，应选择$p = \frac{\lambda_1 + \lambda_{n-1}}{2} = p^\star$

原点位移的加速方法是一个矩阵变换方法，这种变换容易计算又不会破坏矩阵$\mathbf{A}$ 的稀疏性，但p 的选择依赖于对$\mathbf{A}$ 的特征值分布的大致了解。

2. Rayleigh商加速法

定理 54 设$\mathbf{A} \in \mathbf{R}^{n\times n}$ 为对称阵，特征值满足$|\lambda_1| > |\lambda_2| \geq |\lambda_3| \geq \cdots \geq |\lambda_n|$，对应的特征向量满足$(\mathbf{x}_i, \mathbf{x}_j) = \delta_{ij}$，应用幂法计算$\mathbf{A}$ 的主特征值λ_1，则规范化向量$\mathbf{u}_k$的*Rayleigh*商给出了λ_1 的较好的近似，即

$$\frac{(\mathbf{A}\mathbf{u}_k, \mathbf{u}_k)}{\mathbf{u}_k, \mathbf{u}_k} = \lambda_1 + O\left((\frac{\lambda_2}{1})^{2k}\right)$$

8.2.3 反幂法

反幂法用来计算矩阵按模最小的特征值及其特征向量，及计算对应于一个给定近似特征值的特征向量。

设$\mathbf{A} \in \mathbf{R}^{n\times n}$ 为非奇异矩阵，$\mathbf{A}$ 的特征值次序记作$|\lambda_1| \geq |\lambda_2| \geq \cdots \geq |\lambda_n|$，相应的特征向量为$\mathbf{x_1}, \mathbf{x_2}, \cdots, \mathbf{x_n}$，则$\mathbf{A}^{-1}$ 的特征值为$|\frac{1}{\lambda_n}| \geq |\frac{1}{\lambda_{n-1}}| \geq \cdots \geq |\frac{1}{\lambda_1}|$，对应的特征向量为$\mathbf{x_n}, \mathbf{x_{n-1}}$。

因此，计算$\mathbf{A}$ 按模最小的特征值λ_n 的问题就是计算$\mathbf{A}^{-1}$ 的按模最大的特征值问题。

对$\mathbf{A}^{-1}$ 应用幂法迭代法称为反幂法，可求得主特征值$\frac{1}{\lambda_n}$ ，从而求得λ_n。

定理 55 设

1. $\mathbf{A}$ 有n 个线性无关的特征向量
2. $\mathbf{A}$ 为非奇异矩阵且特征值满足

$$|\lambda_1| \geq |\lambda_2| \geq \cdots \geq |\lambda_{n-1}| > |\lambda_n| > 0$$

则对任何初始非零向量$\mathbf{u_0} = \mathbf{v_0}$ ，由反幂法构造的向量序列$\{\mathbf{v_k}\}, \{\mathbf{u}_k\}$ 满足

$$\lim_{k\to\infty} \mathbf{u}_k = \frac{\mathbf{x_n}}{\max(\mathbf{x_n})}, \lim_{k\to\infty} \max(\mathbf{v}_k) = \frac{1}{\lambda_n}$$

收敛速度的比值为$\left|\frac{\lambda_n}{\lambda_{n-1}}\right|$

在反幂法中也可以用原点平移法来加速迭代过程或求其他特征值及特征向量。

8.3 Householder方法

8.3.1 引言

设$\mathbf{A} \in \mathbf{R}^{n\times n}$，则存在一个正交阵$\mathbf{R}$，使

$$\mathbf{R^T AR} = \begin{pmatrix} T_{11} & T_{12} & \cdots & T_{1s} \\ & T_{22} & \cdots & T_{2s} \\ & & \ddots & \vdots \\ & & & T_{ss} \end{pmatrix}$$

其中对角块为一阶或二阶矩阵，每一个一阶对角块为$\mathbf{A}$ 的实特征值，每一个二阶对角块的两个图特征值是$\mathbf{A}$ 的一对共轭复特征值。

定义 40 一方阵$\mathbf{B}$，如果当$i > j+1$ 时有$b_{ij} = 0$，则称$\mathbf{B}$ 为上*Hessenberg*阵，即

$$\mathbf{B} = \begin{pmatrix} b_{11} & b_{12} & \cdots & b_{1n} \\ b_{21} & b_{22} & \cdots & b_{2n} \\ & \ddots & \ddots & \vdots \\ & & b_{n,n-1} & b_{nn} \end{pmatrix}$$

求原矩阵特征值问题转换为求上Hessenberg阵或对称三角阵的特征值问题。

定义 41 设向量$\mathbf{w}$ 满足$\|w\|_2 = 1$，矩阵$\mathbf{H} = \mathbf{I} - \mathbf{2ww^T}$ 称为初等反射阵，记作$H(\mathbf{w})$，即

$$H(\mathbf{w}) = \begin{pmatrix} 1-2w_1^2 & -2w_1w_2 & \cdots & -2w_1w_n \\ -2w_2w_1 & 1-2w_2^2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & -2w_{n-1}w_n \\ -2w_nw_1 & \cdots & 1-2w_n^2 & \end{pmatrix}$$

定理 56 初等反射阵$\mathbf{H}$ 是对称阵、正交阵和对合阵。

初等反射阵在计算上的意义是它能用来约化矩阵，这种约化矩阵的方法称为Householder方法

定理 57 设$\mathbf{x}, \mathbf{y}$ 为两个不相等的n 维向量，$\|\mathbf{x}\|_2 = \|\mathbf{y}\|_2$，则存在一个初等反射阵$\mathbf{H}$，使$\mathbf{Hx} = \mathbf{y}$。

易知$\mathbf{w} = \frac{\mathbf{x}-\mathbf{y}}{\|\mathbf{x}-\mathbf{y}\|_2}$ 是使$\mathbf{Hx} = \mathbf{y}$ 成立的唯一长度等于1 的向量。

设向量$\mathbf{x} \in \mathbf{R}^n(\mathbf{x} \neq \mathbf{0}, = \pm\|\mathbf{x}\|_2$，且$\mathbf{x} \neq -\sigma\mathbf{e}_1$，则存在一个初等反射阵

$$\mathbf{H} = \mathbf{I} - 2\frac{\mathbf{u}\mathbf{u}^\mathbf{T}}{\|u\|_2^2} = \mathbf{I} - ^{-1}\,\mathbf{u}\mathbf{u}^\mathbf{T}$$

使$\mathbf{H}\mathbf{x} = -\sigma\mathbf{e}_1$ ，其中$\mathbf{u} = \mathbf{x} + \sigma\mathbf{e}_1, \rho^{-1} = \|\mathbf{u}\|_2^2/2$。

设$\mathbf{x} = (\alpha_1, \alpha_2, \cdots, \alpha_n)^T \neq \mathbf{0}, \mathbf{u} = (u_1, u_2, \cdots, u_n)^T$ ，则有 $\mathbf{u} = (\alpha_1 + \sigma, \alpha_2, \cdots, \alpha_n)^T, \rho = (\sigma + \alpha_1)$ ，如果σ 和α_1 异号，那么计算$\alpha_1 + \sigma$ 时有效数字可能损失，因此取σ 和α_1 有相同的符号，即取$= sgn(\alpha_1)\|x\|_2$。

计算的算法如下

1. 计算$\sigma = sgn(\alpha_1)(\sum_{i=1}^n \alpha_i^2)^{\frac{1}{2}}$
2. $\alpha_1 \to u_1 = \alpha_1 + \sigma$
3. $\rho = \sigma u_1$

在上述算法中，可能出现溢出，为了避免溢出，通常将$\mathbf{x}$ 规范化，即

$$\eta = \max_i \|\alpha_i\|, \mathbf{x}' = \frac{\mathbf{x}}{\eta}$$

因此，有$\sigma' = \frac{\sigma}{}, \mathbf{H}' = \mathbf{H}$。

因此，改进的算法为

1. $\eta = \max_i \|\alpha_1\|$
2. $\alpha_i \leftarrow u_i = \frac{\alpha_i}{\eta}$
3. $= sgn(u_1)(\sum_{i=1}^n u_i^2)^{\frac{1}{2}}$
4. $u_1 \leftarrow u_1 + \sigma$
5. $\rho = \sigma u_1$
6. $\sigma = \eta\sigma$

8.3.2 用正交相似变换约化矩阵

如果$\mathbf{A} \in \mathbf{R}^{n\times n}$，则存在初等反射阵$\mathbf{U_1}, \mathbf{U_2}, \cdots, \mathbf{U_{n-2}}$ ，使

$$\mathrm{U}_{n-2}\cdots U_2U_1AU_1U_2\cdots U_{n-2} = \mathbf{C}$$

用初等反射阵正交相似约化$\mathbf{A}$ 为上Hessenberg阵，大约需要$\frac{5}{3}n^3$次乘法运算。

由于$\mathbf{U_k}$ 是正交阵，求$\mathbf{A}$ 的特征值问题转换为求上Hessenberg阵$\mathbf{C}$ 的特征值问题。记$\mathbf{P} = \mathbf{U_{n-2}} \cdots \mathbf{U_2 U_1}$ ，则

$$\mathbf{PAP^T} = \mathbf{C}$$

设$\mathbf{y}$ 是$\mathbf{C}$ 的对应特征值λ 的特征向量，则$\mathbf{P^T y}$ 是$\mathbf{A}$ 的对应特征值λ 的特征向量，且

$$\mathbf{P^T y} = \mathbf{U_1 U_2} \cdots \mathbf{U_{n-2} y} = (\mathbf{I} - \lambda_1^{-1} \mathbf{u_1 u_1^T}) \cdots (\mathbf{I} - \lambda_{n-2}^{-1} \mathbf{u_{n-2} u_{n-2}^T}) \mathbf{y}$$

定理 58 *如果$\mathbf{A} \in \mathbf{R}^{n \times n}$ 为对称阵，则存在初等反射阵$\mathbf{U}_1, \mathbf{U}_2, \cdots, \mathbf{U}_{n-2}$ ，使*

$$\mathbf{U_{n-2}} \cdots \mathbf{U_2 U_1 A U_1 U_2} \cdots \mathbf{U_{n-2}} = \mathbf{A_{n-1}} = \begin{pmatrix} c_1 & b_1 & & & \\ b_1 & c_2 & b_2 & & \\ & \ddots & \ddots & \ddots & \\ & & b_{n-2} & c_{n-1} & b_{n-1} \\ & & & b_{n-1} & c_n \end{pmatrix} = \mathbf{C}$$

算法 TODO

将对称阵$\mathbf{A}$ 用初等反射阵正交相似约化为对称三角阵约需做$\frac{2}{3}n^3$ 次乘法运算。

用正交矩阵进行约化的特定为，构造的$\mathbf{U_k}$ 容易求逆且$\mathbf{U}_k$ 的元素数量级不大，因此算法十分稳定。

8.4 QR算法

8.4.1 引言

QR方法使一种变换方法，是计算中小型矩阵全部特征值问题的最有效方法之一。目前,QR方法主要用来计算

1. 上Hessenberg阵的全部特征值问题

2. 对称三角阵的全部特征值问题

QR方法收敛快，算法稳定。

对于一般矩阵$\mathbf{A} \in \mathbf{R}^{n \times n}$ ，首先用Householder方法将$\mathbf{A}$ 化为上Hessenberg阵$\mathbf{B}$ ，然后在用QR 方法计算$\mathbf{B}$ 的全部特征值问题。

此外，还可以考虑使用如下形式的平面旋转矩阵来约化

$$\mathbf{P}_{ij}=\begin{pmatrix}1&&&&&&&&\\&\ddots&&&&&&&\\&&1&&&&&&\\&&&c&&s&&&\\&&&&1&&&&\\&&&&&\ddots&&&\\&&&&&&1&&\\&&&s&&c&&&\\&&&&&&&1&\\&&&&&&&&\ddots\\&&&&&&&&&\\1&&&&&&&&&\end{pmatrix} \tag{8.4.1}$$

设$\mathbf{x}=(\alpha_1,\alpha_2,\cdots,\alpha_i,\cdots,\alpha_j,\cdots,\alpha_n)^T$ ，其中α_i,α_j 不全为零，则可选一平面旋转矩阵$\mathbf{P}_{ij}$ 使$\mathbf{P}_{ij}\mathbf{x}=\mathbf{y}\equiv(\alpha_1,\alpha_2,\cdots,\alpha_i^{(1)},\cdots,\alpha_j^{(1)},\cdots,\alpha_n)^T$，其中

$$\overset{(1)}{i}=\sqrt{\alpha_i^2+\alpha_j^2} \tag{8.4.2}$$

$$\alpha_j^{(1)}=0 \tag{8.4.3}$$

$$\begin{cases}c=\alpha_i/\sqrt{\alpha_i^2+\alpha_j^2}\\ s=\alpha_j/\sqrt{\alpha_i^2+\alpha_j^2}\end{cases} \tag{8.4.4}$$

设给定$\mathbf{x}=(\alpha,\beta)^T$ ，计算c,s,v 使$\mathbf{P}_{ij}\mathbf{x}=\begin{pmatrix}c&s\\-s&c\end{pmatrix}\begin{pmatrix}\alpha\\\beta\end{pmatrix}=\begin{pmatrix}v\\0\end{pmatrix}$ ，为了防止溢出，将$\mathbf{x}$ 规范化，有

$$\eta\equiv\|\mathbf{x}\|_\infty=\max\{\|\alpha\|,\|\beta\|\}\neq 0$$

$$\mathbf{x}' = \mathbf{x}/\eta = \begin{pmatrix} \frac{\alpha}{\eta} \\ \frac{\beta}{\eta} \end{pmatrix}$$

计算算法如下

1. 计算$\eta = max\{\|\alpha\|, \|\beta\|\}$
2. 如果$\eta = 0$，则$c \leftarrow 1, s \leftarrow 0$，转步骤8
3. $\alpha' = \frac{\alpha}{\eta}$
4. $\beta' = \frac{\beta}{\eta}$
5. $v' = \sqrt{\alpha'^2 + \beta'^2}$
6. $c = \alpha'/v', s = \beta'/v'$
7. $v = \eta v'$
8. 计算终止

定理 59 如果$\mathbf{A}$ 为非奇异矩阵，则存在正交矩阵$\mathbf{P}_1, \mathbf{P}_2, \cdots, \mathbf{P}_{n-1}$ 使

$$\mathbf{P}_{n-1} \cdots \mathbf{P}_2 \mathbf{P}_1 \mathbf{A} = \begin{pmatrix} r_{11} & r_{12} & \cdots r_{1n} \\ & r_{22} & \cdots & r_{2n} \\ & & \ddots & \vdots \\ & & & r_{nn} \end{pmatrix} \tag{8.4.5}$$

且$r_{ii} > 0$

定理 60 如果$\mathbf{A} \in \mathbf{R}^{n \times n}$ 为非奇异矩阵，则$\mathbf{A}$ 可分解为一正交阵$\mathbf{Q}$ 与上三角阵$\mathbf{R}$ 的乘积，即$\mathbf{A} = \mathbf{QR}$ 且当$\mathbf{R}$ 对角元素都为正数时分解唯一

8.4.2 QR算法

设$\mathbf{A} = \mathbf{A_1} \in \mathbf{R}^{n \times n}$且对$\mathbf{A}$ 进行QR分解，即$\mathbf{A} = \mathbf{QR}$，其中$\mathbf{R}$ 为上三角阵，$\mathbf{Q}$ 为正交阵，可得新矩阵

$$\mathbf{B} = \mathbf{RQ} = \mathbf{Q^T A Q}$$

B 是由$\mathbf{A}$ 经过正交相似变换得到，因此$\mathbf{B}$ 与$\mathbf{A}$ 特征值相同。再对$\mathbf{B}$ 进行QR分解，又可得一新的矩阵，重复上述过程可以得到矩阵序列。

QR算法就是利用矩阵的QR分解，按上述递推法则构造矩阵序列$\{\mathbf{A_k}\}$的过程，只要$\mathbf{A}$ 为非奇异矩阵，则由QR算法就完全确定$\{\mathbf{A}_k\}$

定理 61 设$\mathbf{A}=\mathbf{A_1}\in\mathbf{R}^{n\times n}$，$QR$算法为

$$\begin{cases}\mathbf{A_k}=\mathbf{Q_k}\mathbf{R_k}\\ \mathbf{A_{k+1}}=\mathbf{R_k}\mathbf{Q_k}\end{cases}\tag{8.4.8}$$

且$\tilde{\mathbf{Q}}_k\equiv\mathbf{Q_1}\mathbf{Q_2}\cdots\mathbf{Q_k},\tilde{\mathbf{R_k}}=\mathbf{R_k}\cdots\mathbf{R_2}\mathbf{R_1}$ 则有

1. $\mathbf{A_{k+1}}$ 相似于$\mathbf{A_k}$ ，即$\mathbf{A_{k+1}}=\mathbf{Q_k^T}\mathbf{A_k}\mathbf{Q_k}$
2. $\mathbf{A}_{k+1}=(\mathbf{Q_1}\mathbf{Q_2}\cdots\mathbf{Q_k})^TA_1(\mathbf{Q_1}\mathbf{Q_2}\cdots\mathbf{Q_k})=\tilde{\mathbf{Q_k^T}}\mathbf{A_1}\tilde{\mathbf{Q_k}}$
3. $\mathbf{A^k}$ 的QR分解式为$\mathbf{A}^k=\tilde{\mathbf{Q}}_k\tilde{\mathbf{R}}_k$

设$\mathbf{M}_k=\mathbf{Q}_k\mathbf{R}_k$ ，其中$\mathbf{Q}_k$ 为正交阵，$\mathbf{R}_k$ 为具有正交对角元素的上三角阵，如果$\mathbf{M}_k\to\mathbf{I}$，则$\mathbf{Q}_k\to\mathbf{I}$ 及$\mathbf{R}_k\to\mathbf{I}$。

定理 62 设$\mathbf{A}\in\mathbf{R}^{n\times n}$

1. 如果$\mathbf{A}$ 的特征值满足$|\lambda_1|>|\lambda_2|>\cdots>|\lambda_n|>0$
2. $\mathbf{A}$ 有标准形$\mathbf{A}=\mathbf{XDX^{-1}}$，其中$\mathbf{D}=diag(\lambda_1,\lambda_2,\cdots,\lambda_n)$ ，且设$\mathbf{X}^{-1}$有三角分解$\mathbf{X}^{-1}=\mathbf{LU}$，则由$QR$算$afaijug$的$\{\mathbf{A}_k\}$ 本质上收敛于上三角阵

定理 63 如果对称阵$\mathbf{A}$满足上述定理条件，则由QR算法产生的$\{\mathbf{A_k}\}$ 收敛于对角阵。

8.4.3 带原点位移的QR方法

为了加速收敛，选择数列$\{s_k\}$ ，按下述方法构造矩阵序列$\{\mathbf{A}_k\}$ 称为带原点位移的QR算法。

1. 设$\mathbf{A}=\mathbf{A}_1\in\mathbf{R}^{n\times n}$
2. 将$\mathbf{A_k}-s_k\mathbf{I}$ 进行QR分解，即$\mathbf{A}_k-s_k\mathbf{I}=\mathbf{Q}_k\mathbf{R}_k$
3. 构造新矩阵$\mathbf{A}_{k+1}=\mathbf{R}_k\mathbf{Q}_k+s_k\mathbf{I}=\mathbf{Q}_k^T\mathbf{A}_k\mathbf{Q}_k$
4. $\mathbf{A}_{k+1}=\tilde{\mathbf{Q}}_k^T\mathbf{A}\tilde{\mathbf{Q}}_k$，其中$\tilde{\mathbf{Q}}_k=\mathbf{Q_1}\mathbf{Q_2}\cdots\mathbf{Q_k},\tilde{\mathbf{R}}_k=\mathbf{R_k}\cdots\mathbf{R_2}\mathbf{R_1}$
5. 矩阵$(\mathbf{A}-s_1\mathbf{I})(\mathbf{A}-s_2)\cdots(\mathbf{A}-s_k\mathbf{I})\equiv\phi(\mathbf{A})$ 有QR分解式$\phi(\mathbf{A})=\tilde{\mathbf{Q}}_k\tilde{\mathbf{R}}_k$
6. 带位移QR方法变换一步的计算：首先用正交变换将$\mathbf{A}_k-s_k\mathbf{I}$ 化为上三角阵，即

$$\mathbf{P_{n-1}}\cdots\mathbf{P_2}\mathbf{P_1}(\mathbf{A}_k-s_k\mathbf{I})=\mathbf{R}_k$$

其中$\mathbf{Q}_k^T = \mathbf{P_{n-1}} \cdots \mathbf{P_2 P_1}$为一系列平面旋转矩阵的乘积，于是

$$\mathbf{A}_{k+1} = \mathbf{P_{n-1}} \cdots \mathbf{P_2 P_1} (\mathbf{A_k} - s_k \mathbf{I}) \mathbf{P_1^T P_2^T} \cdots \mathbf{P_{n-1}^T} + s_k \mathbf{I}$$

若$\mathbf{A}$ 为上Hessenberg阵，则用QR算法禅古的 $\mathbf{A}_2, \mathbf{A}_3, \cdots, \mathbf{A}_k$ 也是上Hessenberg阵，