第八章　插值与样条函数的并行计算

在数值计算或工程设计中，常会遇到插值问题，即通过曲线 $Y=f(x)$ 上一组离散点 (x_i, y_i) $(i=0, 1, \cdots, n)$，求作一条简单的代数曲线 $Y=p(x)$ 作为曲线 $Y=f(x)$ 的近似。由于高次插值的计算过程不稳定，实用时通常先将插值区间划分为若干段，然后再在每个分段上采用低次插值（即所谓分段插值法）。本章首先讨论分段插值的并行计算。

不少实际问题，不但要求插值函数 $p(x)$ 在节点处的值与 $f(x)$ 在相应节点的值相等，而且要求 $p(x)$，$f(x)$ 在节点处的一阶导数值也相等。本章也将介绍分段三次埃尔米特(Hermit)插值的并行计算。

分段插值法的不足之处是：虽然插值曲线在各个分段是衔接的，但在联结点处不能保证曲线的光滑性。样条插值，对比于其它分段插值的主要优点是：它保证了直到二阶导数的连续性，因此，光滑度较高。样条函数是现代函数逼近的一个十分活跃的分支，是计算方法的重要基础，它在飞机、船舶、汽车等外形设计中都有重要应用。本章将简略介绍样条函数的一些基本方法的并行算法。

8.1　分段插值的并行计算

高次多项式插值不稳定现象是由于局部误差在全实轴上传播所引起的，采用分段低次多项式插值可以限制局部误差的传播范围，从而保证计算的稳定性。本节以分段三次拉格朗日插值为例来分析分段插值的并行计算。

8.1.1 分段三次拉格朗日插值公式的向量表示

假定在区间〔a，b〕上给定节点 $a=x_0<x_1<x_2<\cdots<x_{n-1}<x_n=b$，它们所对应的函数值分别为 y_0，y_1，…，y_n。其中节点的间距 $h=\dfrac{b-a}{n}$。过四个点 (x_{i-1}, y_{i-1})，(x_i, y_i)，(x_{i+1}, y_{i+1})，(x_{i+2}, y_{i+2}) 作三次插值多项式：

$$p(x)=c_{i-1}y_{i-1}+c_iy_i+c_{i+1}y_{i+1}+c_{i+2}y_{i+2} \tag{8-1}$$

其中

$$c_{i-1}=\frac{(x-x_i)(x-x_{i+1})(x-x_{i+2})}{(x_{i-1}-x_i)(x_{i-1}-x_{i+1})(x_{i-1}-x_{i+2})}$$

$$c_i=\frac{(x-x_{i-1})(x-x_{i+1})(x-x_{i+2})}{(x_i-x_{i-1})(x_i-x_{i+1})(x_i-x_{i+2})}$$

$$c_{i+1}=\frac{(x-x_{i-1})(x-x_i)(x-x_{i+2})}{(x_{i+1}-x_{i-1})(x_{i+1}-x_i)(x_{i+1}-x_{i+2})}$$

$$c_{i+2}=\frac{(x-x_{i-1})(x-x_i)(x-x_{i+1})}{(x_{i+2}-x_{i-1})(x_{i+2}-x_i)(x_{i+2}-x_{i+1})}$$

令

$$x=x_i+ht$$

则

$$t=\frac{1}{h}(x-x_i)=\frac{1}{h}(x-a)-i \tag{8-2}$$

节点下标可按下列方式选取：

先计算 $\dfrac{x-a}{h}$ 的整数部分，即 $i^*=\left[\dfrac{x-a}{h}\right]$，然后取

$$i=\begin{cases}1, & i^*\leqslant 1\\ i^*, & 1<i^*<n-2\\ n-2, & i^*\geqslant n-2\end{cases}$$

则有

$$p(x)=-\frac{1}{6}t(t-1)(t-2)y_{i-1}+\frac{1}{2}(t+1)\times(t-1)(t-2)y_i-\frac{1}{2}t(t+1)(t-2)y_{i+1}+\frac{1}{6}t(t+1)(t-1)y_{i+2} \tag{8-3}$$

$p(x)$的系数可表示成

$$[c_{i-1},\ c_i,\ c_{i+1},\ c_{i+2}]=[1,\ t,\ t^2,\ t^3]$$

$$\times\begin{bmatrix} 0 & 1 & 0 & 0 \\ -\frac{1}{3} & -\frac{1}{2} & 1 & -\frac{1}{6} \\ \frac{1}{2} & -1 & \frac{1}{2} & 0 \\ -\frac{1}{6} & \frac{1}{2} & -\frac{1}{2} & \frac{1}{6} \end{bmatrix} \tag{8-4}$$

故

$$p(x)=[c_{i-1},\ c_i,\ c_{i+1},\ c_{i+2}]\begin{pmatrix} y_{i-1} \\ y_i \\ y_{i+1} \\ y_{i+2} \end{pmatrix} \tag{8-5}$$

8.1.2 分段三次拉格朗日插值公式的并行计算

一、一个自变量每次插出一个函数值

设节点等距，并令 $x_0=a$，$x_1=a+h$，$x_2=a+2h$，…，$x_n=a+nh=b$，其中 $h=\frac{b-a}{n}$，对应的函数值分别为 y_0，y_1，…，y_n。

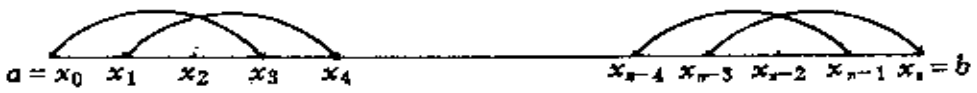


图 8-1

按图 8-1 的划分，在区间〔a，b〕上插值，最多需计算 $n-2$ 个不同的一元三次插值多项式。这组多项式可以写成下列向量形式：

$$\boldsymbol{P}=\boldsymbol{A}t^3+\boldsymbol{B}t^2+\boldsymbol{C}t+\boldsymbol{D}$$

其中 t 的表示式同式（8-2）。由式（8-4）、（8-5）可知

$$\boldsymbol{A}=\begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_{n-2} \end{bmatrix}=-\frac{1}{6}\begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_{n-3} \end{bmatrix}+\frac{1}{2}\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_{n-2} \end{bmatrix}-\frac{1}{2}\begin{bmatrix} y_2 \\ y_3 \\ \vdots \\ y_{n-1} \end{bmatrix}+\frac{1}{6}\begin{bmatrix} y_3 \\ y_4 \\ \vdots \\ y_n \end{bmatrix}$$

$$\boldsymbol{B}=\begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_{n-2} \end{bmatrix}=\frac{1}{2}\begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_{n-3} \end{bmatrix}-\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_{n-2} \end{bmatrix}+\frac{1}{2}\begin{bmatrix} y_2 \\ y_3 \\ \vdots \\ y_{n-1} \end{bmatrix}$$

$$\boldsymbol{C}=\begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_{n-2} \end{bmatrix}=-\frac{1}{3}\begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_{n-3} \end{bmatrix}-\frac{1}{2}\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_{n-2} \end{bmatrix}+\begin{bmatrix} y_2 \\ y_3 \\ \vdots \\ y_{n-1} \end{bmatrix}-\frac{1}{6}\begin{bmatrix} y_3 \\ y_4 \\ \vdots \\ y_n \end{bmatrix}$$

$$\boldsymbol{D}=\begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_{n-2} \end{bmatrix}=\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_{n-2} \end{bmatrix}$$

向量 $\boldsymbol{A}$、$\boldsymbol{B}$、$\boldsymbol{C}$、$\boldsymbol{D}$ 是适合于向量运算的，它们的 $n-2$ 个分量在向量机上可以并行计算。

事实上，向量 $\boldsymbol{A}$ 需执行四次向量乘法，三次向量加法。向量 $\boldsymbol{B}$ 需执行二次向量乘法，二次向量加法。而向量 $\boldsymbol{C}$ 的计算也只需三次向量乘法和三次向量加法即可完成。计算出系数 a_i、b_i、c_i、d_i（$i=1,\ 2,\ \cdots,\ n-2$）后，对于单个自变量使用下列公式计算出函数值 $p_i(x)$

$$\begin{aligned} p_i(x)&=y_i(x)=a_it^3+b_it^2+c_it+d_i \\ &=((a_it+b_i)\ t\ +c_i)\ t\ +d_i \\ &\qquad i=1,2,\cdots,n-2 \end{aligned}$$

采用上述格式的优点在于向量 $\boldsymbol{A}$、$\boldsymbol{B}$、$\boldsymbol{C}$、$\boldsymbol{D}$ 的运算只涉及由函数值 y_i 所构成的一维向量。它们之间只相差一个向量单元，

可以通过双位移向量指令来形成所需的函数值，而不必每次都从内存中反复调用，这样可大大提高向量运算速度。

二、一组自变量每次插出一组函数值

我们仍假设节点等距，并令 $x_0=a$，$x_1=a+h$，…，$x_n=a+nh=b$，其中 $h=\dfrac{b-a}{n}$，对应的函数值分别为 y_0，y_1，…，y_n。对于给定的一组自变量 $\boldsymbol{X}=[x^{(1)}, x^{(2)}, \cdots, x^{(n)}]^T$ 插出一组函数值的并行计算步骤如下：

（1）并行计算

$$\boldsymbol{I}^*=\begin{bmatrix} i_1^* \\ i_2^* \\ \vdots \\ i_n^* \end{bmatrix}$$

其中分量 $i_j^*=\left[\dfrac{x^{(j)}-a}{h}\right]$（$j=1, 2, \cdots, n$），是适合向量运算的。如果注意运用向量计算机的测试指令的功能，则效率可进一步提高。

（2）并行计算

$$\boldsymbol{I}=\begin{bmatrix} i_1 \\ i_2 \\ \vdots \\ i_n \end{bmatrix}$$

其中每个分量

$$i_j=\begin{cases} 1, & i_j^*\leqslant 1 \\ i_j^*, & 1<i_j^*<n-2 \quad j=1, 2, \cdots, n \\ n-2, & i_j^*\geqslant n-2 \end{cases}$$

如果注意运用向量计算机的位串指令和测试指令功能，则不难实现并行计算。

（3）并行计算

$$\boldsymbol{t}=\frac{1}{h}\cdot(\boldsymbol{X}-a)-\boldsymbol{I}$$

其中

$$\boldsymbol{t}=(t_1,\ t_2,\ \cdots,\ t_n)^T$$
$$\boldsymbol{X}=(x^{(1)},\ x^{(2)},\ \cdots,\ x^{(n)})^T$$
$$\boldsymbol{I}=(i_1,\ i_2,\ \cdots,\ i_n)^T$$

（4）并行计算

$$\begin{aligned}\boldsymbol{P}&=\boldsymbol{A}t^3+\boldsymbol{B}t^2+\boldsymbol{C}t+\boldsymbol{D}\\&=((\boldsymbol{A}t+\boldsymbol{B})t+\boldsymbol{C})t+\boldsymbol{D}\end{aligned}$$

其中

$$\boldsymbol{A}=\frac{1}{6}(\boldsymbol{Y}_{t+2}-\boldsymbol{Y}_{t-1})+\frac{1}{2}(\boldsymbol{Y}_t-\boldsymbol{Y}_{t+1})$$
$$=\frac{1}{6}\left(\begin{bmatrix}y_{i_1+2}\\y_{i_2+2}\\\vdots\\y_{i_n+2}\end{bmatrix}-\begin{bmatrix}y_{i_1-1}\\y_{i_2-1}\\\vdots\\y_{i_n-1}\end{bmatrix}\right)+\frac{1}{2}\left(\begin{bmatrix}y_{i_1}\\y_{i_2}\\\vdots\\y_{i_n}\end{bmatrix}-\begin{bmatrix}y_{i_1+1}\\y_{i_2+1}\\\vdots\\y_{i_n+1}\end{bmatrix}\right)$$

$$\boldsymbol{B}=\frac{1}{2}(\boldsymbol{Y}_{t-1}+\boldsymbol{Y}_{t+1})-\boldsymbol{Y}_t$$
$$=\frac{1}{2}\left(\begin{bmatrix}y_{i_1-1}\\y_{i_2-1}\\\vdots\\y_{i_n-1}\end{bmatrix}+\begin{bmatrix}y_{i_1+1}\\y_{i_2+1}\\\vdots\\y_{i_n+1}\end{bmatrix}\right)-\begin{bmatrix}y_{i_1}\\y_{i_2}\\\vdots\\y_{i_n}\end{bmatrix}$$

$$\boldsymbol{C}=\boldsymbol{Y}_{t+1}-\frac{1}{3}\boldsymbol{Y}_{t-1}-\frac{1}{2}\boldsymbol{Y}_t-\frac{1}{6}\boldsymbol{Y}_{t+2}$$
$$=\begin{bmatrix}y_{i_1+1}\\y_{i_2+1}\\\vdots\\y_{i_n+1}\end{bmatrix}-\frac{1}{3}\begin{bmatrix}y_{i_1-1}\\y_{i_2-1}\\\vdots\\y_{i_n-1}\end{bmatrix}-\frac{1}{2}\begin{bmatrix}y_{i_1}\\y_{i_2}\\\vdots\\y_{i_n}\end{bmatrix}-\frac{1}{6}\begin{bmatrix}y_{i_1+2}\\y_{i_2+2}\\\vdots\\y_{i_n+2}\end{bmatrix}$$

$$\boldsymbol{D}=\boldsymbol{Y}_t=\begin{bmatrix}y_{i_1}\\y_{i_2}\\\vdots\\y_{i_n}\end{bmatrix}$$

这样通过上述四个计算步骤，对于区间〔a，b〕上的任意 n 个自变量插出了 n 个函数值 $p(x^{(1)})$、$p(x^{(2)})$、…、$p(x^{(n)})$。

三、一个自变量同时插出s个函数值

设 $x_0=a$，$x_1=a+h$，…，$x_n=a+nh=b$，相应的函数值为

$$\begin{matrix} y_{10}, & y_{11}, & \cdots, & y_{1n} \\ y_{20}, & y_{21}, & \cdots, & y_{2n} \\ \cdots & & & \\ y_{s0}, & y_{s1}, & \cdots, & y_{sn} \end{matrix}$$

对于给定的自变量 x，由三次分段插值多项式计算公式(8-2)～(8-5)，插出 s 个函数值的并行计算格式表示如下：

$$\begin{bmatrix} p_1(x) \\ p_2(x) \\ \vdots \\ p_s(x) \end{bmatrix} = \begin{bmatrix} y_{1,i-1} & y_{1,i} & y_{1,i+1} & y_{1,i+2} \\ y_{2,i-1} & y_{2,i} & y_{2,i+1} & y_{2,i+2} \\ \cdots & & & \\ y_{s,i-1} & y_{s,i} & y_{s,i+1} & y_{s,i+2} \end{bmatrix} \begin{bmatrix} c_{i-1} \\ c_i \\ c_{i+1} \\ c_{i+2} \end{bmatrix} \tag{8-6}$$

该格式的并行效率较高。

8.2 分段三次埃尔米特插值的并行计算

8.2.1 分段三次埃尔米特插值公式

上一节提到的插值法，仅要求插值多项式 $p(x)$ 在插值节点处与被插值函数 $f(x)$ 有相同的函数值。这种多项式往往不能反映给定函数的变化趋势。本节讨论的插值问题，要求构造三次多项式 $H(x)$，使它与函数 $f(x)$ 在插值节点处不仅有相同的函数值而且还有相同的一阶导数值。

设 $a=x_0<x_1<x_2<\cdots<x_n=b$ 是 $n+1$ 个互不相同的节点，要求在每个子区间〔x_i，x_{i+1}〕上用其两端点的函数值及导数值作三次多项式

$$H(x)=a_0+a_1x+a_2x^2+a_3x^3 \tag{8-7}$$

满足

$$H(x_i)=f_i,\quad H'(x_i)=f'_i,\quad H(x_{i+1})=f_{i+1},\quad H'(x_{i+1})=f'_{i+1} \tag{8-8}$$

这里 f_i 与 f'_i 分别表示 $f(x_i)$ 与 $f'(x_i)$（$i=0,\ 1,\ \cdots,\ n-1$）式(8-7)的四个系数 a_0，a_1，a_2，a_3 都线性地依赖于 f_i，f'_i，f_{i+1}，f'_{i+1}。不难推出分段三次埃尔米特插值的计算公式为

$$H(x)=f_i\alpha_i(x)+f'_i\beta_i(x)+f_{i+1}\alpha_{i+1}(x)+f'_{i+1}\beta_{i+1}(x) \tag{8-9}$$

其中 $x_i\leqslant x\leqslant x_{i+1}\quad i=0,\ 1,\ \cdots,\ n-1$

这里函数 $\alpha_i(x)$，$\beta_i(x)$ 可以通过分段线性插值的基函数

$$l_i(x)=\frac{x-x_{i+1}}{x_i-x_{i+1}},\qquad l_{i+1}=\frac{x-x_i}{x_{i+1}-x_i}$$

表达。即

$$\begin{aligned}\alpha_i(x)&=l_i^2(1+2l_{i+1})=3l_i^2-2l_i^3\\ \alpha_{i+1}(x)&=l_{i+1}^2(1+2l_i)=3l_{i+1}^2-2l_{i+1}^3\\ \beta_i(x)&=h_il_i^2l_{i+1}=h_i(l_i^2-l_i^3)\\ \beta_{i+1}(x)&=-h_il_{i+1}^2l_i=-h_i(l_{i+1}^2-l_{i+1}^3)\end{aligned}$$

这里

$$h_i=x_{i+1}-x_i$$

不难验证

$$\alpha_i(x_j)=\delta_{ij},\quad \alpha'_i(x_j)=0$$
$$\beta_i(x_j)=0,\quad \beta'_i(x_j)=\delta_{ij}$$

其中

$$\delta_{ij}=\begin{cases}1 & \text{当 } i=j \text{ 时}\\ 0 & \text{当 } i\neq j \text{ 时}\end{cases}$$

8.2.2 分段三次埃尔米特插值公式的向量表示

设 $a=x_0<x_1<x_2<\cdots<x_n=b$，对于

$x_i\leqslant x\leqslant x_{i+1}$ 引入参数 $t=\dfrac{x-x_i}{h}$

其中 $$h=\frac{b-a}{n}$$

分段三次埃尔米特插值公式可表示为

$$H(x)=f_iF_i(t)+f_{i+1}F_{i+1}(t)+h[f'_iG_i(t)+f'_{i+1}G_{i+1}(t)] \tag{8-10}$$

这里混合函数

$$\begin{cases} F_i(t)=(1-t)^2(1+2t), & F_{i+1}(t)=t^2(3-2t) \\ G_i(t)=t(1-t)^2, & G_{i+1}(t)=-t^2(1-t) \end{cases} \tag{8-11}$$

写成矩阵形式为

$$[F_i,\ F_{i+1},\ G_i,\ G_{i+1}]=[1,\ t,\ t^2,\ t^3]M \tag{8-12}$$

其中

$$M=\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -3 & 3 & -2 & -1 \\ 2 & -2 & 1 & 1 \end{pmatrix}$$

8.2.3 分段三次埃尔米特插值公式的并行计算

由式（8-10）和（8-11）可推知

$$H(x)=t^3(2f_i-2f_{i+1}+hf'_i+hf'_{i+1})+t^2(-3f_i+3f_{i+1}-2hf'_i-hf'_{i+1})+t(hf'_i)+f_i,$$
$$i=0,\ 1,\ \cdots,\ n-1 \tag{8-13}$$

写成向量形式为

$$H(x)=At^3+Bt^2+Ct+D \tag{8-14}$$

其中

$$A=\begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a \end{pmatrix}=2\left(\begin{bmatrix} f_0 \\ f_1 \\ \vdots \\ f_{n-1} \end{bmatrix}-\begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_n \end{bmatrix}\right)+h\left(\begin{bmatrix} f'_0 \\ f'_1 \\ \vdots \\ f'_{n-1} \end{bmatrix}+\begin{bmatrix} f'_1 \\ f'_2 \\ \vdots \\ f'_n \end{bmatrix}\right) \tag{8-15}$$

$$B=\begin{pmatrix} b_0 \\ b_1 \\ \vdots \\ b_{n-1} \end{pmatrix}=\cdots 3\left(\begin{bmatrix} f_0 \\ f_1 \\ \vdots \\ f_{n-1} \end{bmatrix}-\begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_n \end{bmatrix}\right)-2h\begin{bmatrix} f'_0 \\ f'_1 \\ \vdots \\ f'_{n-1} \end{bmatrix}-h\begin{bmatrix} f'_1 \\ f'_2 \\ \vdots \\ f'_n \end{bmatrix} \tag{8-16}$$

$$\boldsymbol{C}=\begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ c_{n-1} \end{bmatrix}=h\begin{bmatrix} f'_0 \\ f'_1 \\ \vdots \\ f'_{n-1} \end{bmatrix} \tag{8-17}$$

$$\boldsymbol{D}=\begin{bmatrix} d_0 \\ d_1 \\ \vdots \\ d_{n-1} \end{bmatrix}=\begin{bmatrix} f_0 \\ f_1 \\ \vdots \\ f_{n-1} \end{bmatrix} \tag{8-18}$$

向量$\boldsymbol{A}$、$\boldsymbol{B}$、$\boldsymbol{C}$是适合于向量计算的。在求得系数a_i、b_i、c_i、d_i后，对于单个自变量，使用下列公式计算函数值

$$H_i(x)=((a_it+b_i)\,t+c_i)\,t+d_i$$

若给定一组自变量$x_i<x^{(i)}<x_{i+1}$　$i=0,\ 1,\ \cdots,\ n-1$插出一组对应函数值的计算步骤如下：

（1）由公式（8-15)～(8-17）并行计算向量$\boldsymbol{A}$、$\boldsymbol{B}$和$\boldsymbol{C}$。向量$\boldsymbol{A}$、$\boldsymbol{B}$、$\boldsymbol{C}$、$\boldsymbol{D}$的运算只涉及由函数值f_i和一阶导数值f'_i（$i=0,\ 1,\ \cdots,\ n$）所构成的一维向量，它们之间只相差一个向量单元，可通过双位移向量指令来形成所需要的函数值或导数值，而不必每次都从内存中反复调用。

（2）并行计算向量　$t=[t_0,\ t_1,\ t_2,\ \cdots,\ t_{n-1}]^T$

$$t=\frac{1}{h}\left(\begin{bmatrix} x^{(0)} \\ x^{(1)} \\ \vdots \\ x^{(n-1)} \end{bmatrix}-\begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_{n-1} \end{bmatrix}\right)$$

（3）并行计算

$$H=At^3+Bt^2+Ct+D=((At+B)t+C)t+D$$

8.3 三次插值样条的其他计算

8.3.1 三次样条函数的定义和插值问题的提法

设在$X-Y$平面上给定$n+1$个有序型值点，即

$$(x_0, y_0), (x_1, y_1), \cdots, (x_n, y_n) \tag{8-19}$$

区间$[a, b]$上给定的分割δ记为：$a=x_0<x_1<\cdots<x_n=b$。要求构造一函数$s(x)$，使其满足条件：

（1）$s(x_i)=y_i(i=0, 1, \cdots, n)$；

（2）在区间$[x_0, x_n]$上，$s(x)$具有二阶连续导数；

（3）在每个小区间$[x_{i-1}, x_i]$上，$s(x)$是x的三次多项式。

这样的函数$s(x)$称为关于有序型值点（8-19）的三次样条函数。

假定$s(x)$在$[x_{i-1}, x_i]$中的三次多项式是

$$s(x)=A_ix^3+B_ix^2+C_ix+D_i \quad (x_{i-1}\leqslant x\leqslant x_i)$$

它可由四个系数决定，故在区间$[x_0, x_n]$上，$s(x)$共有$4n$个系数。但因在每个内节点上满足三个条件：函数、一阶导数、二阶导数连续。这样共有$3(n-1)$个条件，加上节点处的$n+1$个插值条件，条件总数为$4n-2$。因此只需再加二个端点条件，插值问题的解就是唯一的。

8.3.2 三次样条函数的矩阵表示式

一、按照函数值及二阶导数决定的表示式

记$s(x)$在节点x_i处的函数值、一阶导数和二阶导数为

$$s(x_i)=y_i,\ s'(x_i)=m_i,\ s''(x_i)=M_i,$$
$$i=0, 1, \cdots, n \tag{8-20}$$

在每个小区间$[x_{i-1}, x_i](i=1, 2, \cdots, n)$上，$s(x)=s_i(x)$，$s(x)$的二阶导数是线性的，所以

$$s_i''(x)=M_{i-1}\frac{x_i-x}{h_i}+M_i\frac{x-x_{i-1}}{h_i}\quad x_{i-1}\leqslant x\leqslant x_i \tag{8-21}$$

其中 $h_i=x_i-x_{i-1}$ 表示小区间的长度。将式（8-21)积分二次，并由插值条件（8-20）得到

$$s_i'(x)=-M_{i-1}\frac{(x_i-x)^2}{2h_i}+M_i\frac{(x-x_{i-1})^2}{2h_i}+\frac{y_i-y_{i-1}}{h_i}-\frac{h_i(M_i-M_{i-1})}{6}\quad x_{i-1}\leqslant x\leqslant x_i \tag{8-22}$$

$$s_i(x)=M_{i-1}\frac{(x_i-x)^3}{6h_i}+M_i\frac{(x-x_{i-1})^3}{6h_i}+\left(\frac{y_{i-1}}{h_i}-\frac{h_iM_{i-1}}{6}\right)(x_i-x)+\left(\frac{y_i}{h_i}-\frac{h_iM_i}{6}\right)\times(x-x_{i-1})\quad x_{i-1}\leqslant x\leqslant x_i \tag{8-23}$$

从式（8-22）得到

$$\begin{cases} s_i'(x_i-0)=\dfrac{h_i}{6}M_{i-1}+\dfrac{h_i}{3}M_i+\dfrac{y_i-y_{i-1}}{h_i} \\ s_i'(x_i+0)=-\dfrac{h_{i+1}}{3}M_i-\dfrac{h_{i+1}}{6}M_{i+1}+\dfrac{y_{i+1}-y_i}{h_{i+1}} \end{cases} \tag{8-24}$$

由于在每个内节点 x_i 处一阶导数连续，即

$$s'(x_i-0)=s'(x_i+0)$$

由此可得

$$\mu_iM_{i-1}+2M_i+\lambda_iM_{i+1}=d_i\quad i=1,\ 2,\ \cdots,\ n-1 \tag{8-25}$$

其中

$$\begin{cases} \lambda_i=\dfrac{h_{i+1}}{h_i+h_{i+1}}, \\ \mu_i=\dfrac{h_i}{h_i+h_{i+1}}=1-\lambda_i \\ d_i=\dfrac{6}{h_i+h_{i+1}}\left(\dfrac{y_{i+1}-y_i}{h_{i+1}}-\dfrac{y_i-y_{i-1}}{h_i}\right) \end{cases} \tag{8-26}$$

方程组（8-25）对一切内节点即 x_1，x_2，$\cdots x_{n-1}$ 都成立，所

以一共有 $n-1$ 个等式。它们称为三次样条的M关系式。上述节点关系式是关于 $n+1$ 个未知数 M_0，M_1，…，M_n 的 $n-1$ 个方程，要唯一确定这 $n+1$ 个未知数，还必须补充两个条件。这样的条件通常是在端点（即边界点）x_0 及 x_n 处给出的，称为端点条件（或边界条件）。端点条件形式很多，常见的有以下两种：

（1）给定端点的二阶导数（弯矩）

$$M_0=y_0'',\quad M_n=y_n'' \tag{8-27}$$

特别情形，给定 $M_0=M_n=0$ 时，称为自然样条。

（2）给定端点的一阶导数（斜率）

$$s'(x_0)=y_0',\quad s'(x_n)=y_n'$$

由式（8-22）可知，相当于给定了两个方程，即

$$\begin{cases} 2M_0+M_1=\dfrac{6}{h_1}\left(\dfrac{y_1-y_0}{h_1}-y_0'\right) \\ M_{n-1}+2M_n=\dfrac{6}{h_n}\left(y_n'-\dfrac{y_n-y_{n-1}}{h_n}\right) \end{cases} \tag{8-28}$$

端点条件（8-27）和（8-28）可以写成统一的形式为

$$\begin{cases} 2M_0+\lambda_0M_1=d_0 \\ \mu_nM_{n-1}+2M_n=d_n \end{cases} \tag{8-29}$$

其中

$$d_0=\frac{6\lambda_0}{h_1}\left(\frac{y_1-y_0}{h_1}-y_0'\right)+(2-\lambda_0)y_0''$$

$$d_n=\frac{6\mu_n}{h_n}\left(y_n'-\frac{y_n-y_{n-1}}{h_n}\right)+2(1-\mu_n)y_n''$$

当 $\lambda_0=\mu_n=0$ 时，即为式（8-27）。$\lambda_0=\mu_n=1$ 时为式（8-28）。将式（8-29）和（8-25）合在一起即得下列方程组：

$$\begin{cases} 2M_0+\lambda_0M_1 & =d_0 \\ \mu_1M_0+2M_1+\lambda_1M_2 & =d_1 \\ \quad\mu_2M_1+2M_2+\lambda_2M_3 & =d_2 \\ \quad\cdots\cdots & \\ \quad\mu_{n-1}M_{n-2}+2M_{n-1}+\lambda_{n-1}M_n & =d_{n-1} \\ \quad\mu_nM_{n-1}+2M_n & =d_n \end{cases} \tag{8-30}$$

写成矩阵形式为

$$\begin{pmatrix} 2 & \lambda_0 & & & 0 & & \\ \mu_1 & 2 & \lambda_1 & & & & \\ & \mu_2 & 2 & \lambda_2 & & & \\ & \cdots\cdots & & & & & \\ & 0 & & \mu_{n-1} & 2 & \lambda_{n-1} & \\ & & & & \mu_n & 2 & \end{pmatrix} \begin{pmatrix} M_0 \\ M_1 \\ M_2 \\ \vdots \\ M_{n-1} \\ M_n \end{pmatrix} = \begin{pmatrix} d_0 \\ d_1 \\ d_2 \\ \vdots \\ d_{n-1} \\ d_n \end{pmatrix} \tag{8-31}$$

式（8-31）的系数矩阵除了主三对角元素外，其余元素都为零。三对角线方程组（8-31）具有严格的主对角线优势，此方程组有唯一确定的解。它可以用第五章所介绍的并行算法求解。

解方程组（8-31），求出 M_0，M_1，M_2，…，M_n，将其代入式（8-23）即得到所用节点处的二阶导数表示系数的三次样条函数表达式。

为便于在向量机上计算，可把式（8-23）写成下列向量形式：

$$s_i(x)=[1,\ (x-x_{i-1}),\ (x-x_{i-1})^2,\ (x-x_{i-1})^3]\begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} \tag{8-32}$$

其中

$$x_{i-1}\leqslant x\leqslant x_i,\quad i=1,\ 2,\ \cdots,\ n$$

$$\begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ -\dfrac{1}{h_i} & \dfrac{1}{h_i} & -\dfrac{h_i}{3} & -\dfrac{h_i}{6} \\ 0 & 0 & -\dfrac{1}{2} & 0 \\ 0 & 0 & -\dfrac{1}{6h_i} & \dfrac{1}{6h_i} \end{pmatrix} \begin{pmatrix} y_{i-1} \\ y_i \\ M_{i-1} \\ M_i \end{pmatrix} \tag{8-33}$$

二、按照函数值及一阶导数决定的表示式

由埃尔米特插值公式知道，在区间$[x_{i-1},\ x_i]$上 s（x）及其

一、二阶导函数可以写成

$$s(x)=s_i(x)=m_{i-1}\frac{(x_i-x)^2(x-x_{i-1})}{h_i^2}-m_i\frac{(x-x_{i-1})^2(x_i-x)}{h_i^2}+y_{i-1}\frac{(x_i-x)^2[2(x-x_{i-1})+h_i]}{h_i^3}+y_i\frac{(x-x_{i-1})^2[2(x_i-x)+h_i]}{h_i^3} \tag{8-34}$$

$$s'(x)=s_i'(x)=m_{i-1}\frac{(x_i-x)(2x_{i-1}+x_i-3x)}{h_i^2}-m_i\frac{(x-x_{i-1})(2x_i+x_{i-1}-3x)}{h_i^2}+6\frac{y_i-y_{i-1}}{h_i^3}(x_i-x)(x-x_{i-1}) \tag{8-35}$$

$$s''(x)=s_i''(x)=2m_{i-1}\frac{3x-2x_i-x_{i-1}}{h_i^2}+2m_i\frac{3x-x_i-2x_{i-1}}{h_i^2}+6\frac{y_i-y_{i-1}}{h_i^3}(x_i+x_{i-1}-2x) \tag{8-36}$$

$s_i''(x)$ 在 x_i 点的左、右极限分别是

$$\begin{cases} s_i''(x_i-0)=\dfrac{2m_{i-1}}{h_i}+\dfrac{4m_i}{h_i}-6\dfrac{y_i-y_{i-1}}{h_i^2} \\ s_i''(x_i+0)=-\dfrac{4m_i}{h_{i+1}}-\dfrac{2m_{i+1}}{h_{i+1}}+6\dfrac{y_{i+1}-y_i}{h_{i+1}^2} \end{cases}$$

由于在每个内节点 x_i 处二阶导数连续，故有

$$s_i''(x_i-0)=s_i''(x_i+0)$$

于是得到

$$\lambda_i m_{i-1}+2m_i+\mu_i m_{i+1}=c_i \quad i=1,\ 2,\ \cdots,\ n-1 \tag{8-37}$$

式中 λ_i 和 μ_i 取决于式 (8-26)，而

$$c_i=3\left(\lambda_i\frac{y_i-y_{i-1}}{h_i}+\mu_i\frac{y_{i+1}-y_i}{h_{i+1}}\right) \quad i=1,\ 2,\ \cdots,\ n-1 \tag{8-38}$$

式（8-37）称为三次样条的m关系式。它是关于$n+1$个未知数m_0，m_1，…，m_n的$n-1$个方程。为了唯一确定这$n+1$个未知数，亦需给定端点条件。常给定以下两种：

（1）给定端点的二阶导数

$$s''(x_0)=y_0'',\quad s''(x_n)=y_n''$$

这相当于给定了两个方程，即

$$\begin{cases} 2m_0+m_1=3\left[\dfrac{y_1-y_0}{h_1}-\dfrac{h_1}{6}y_0''\right] \\ m_{n-1}+2m_n=3\left[\dfrac{y_n-y_{n-1}}{h_n}+\dfrac{h_n}{6}y_n''\right] \end{cases} \tag{8-39}$$

（2）给定端点处的一阶导数

$$m_0=y_0',\quad m_n=y_n' \tag{8-40}$$

两种端点条件可以统一写作

$$\begin{cases} 2m_0+\mu_0m_1=c_0 \\ \lambda_nm_{n-1}+2m_n=c_n \end{cases} \tag{8-41}$$

其中

$$c_0=(4-\mu_0)\mu_0\left(\frac{y_1-y_0}{h_1}-\frac{h_1}{6}y_0''\right)+2(1-\mu_0)y_0'$$

$$c_n=(4-\lambda_n)\lambda_n\left(\frac{y_n-y_{n-1}}{h_n}+\frac{h_n}{6}y_n''\right)+2(1-\lambda_n)y_n'$$

当$\mu_0=\lambda_n=1$时为第一种端点条件；当$\mu_0=\lambda_n=0$时，为第二种端点条件。

将式（8-41）与（8-37）合在一起得方程组

$$\begin{Bmatrix} 2 & \mu_0 & \cdots & & & 0 \\ \lambda_1 & 2 & \mu_1 & \cdots & & 0 \\ 0 & \lambda_2 & 2 & \mu_2 & \cdots & 0 \\ & \cdots\cdots & & & & \\ 0 & \cdots & & \lambda_{n-1} & 2 & \mu_{n-1} \\ 0 & \cdots & & & \lambda_n & 2 \end{Bmatrix} \begin{Bmatrix} m_0 \\ m_1 \\ m_2 \\ \vdots \\ m_{n-1} \\ m_n \end{Bmatrix} = \begin{Bmatrix} c_0 \\ c_1 \\ c_2 \\ \vdots \\ c_{n-1} \\ c_n \end{Bmatrix} \tag{8-42}$$

这也是具有严格的主对角线优势的三对角线方程组，存在唯一的一组解。

并行解方程组（8-42），求出 m_0，m_1，…，m_n。将其代入式（8-34）即得到所用节点处的一阶导数表示系数的三次样条函数表达式。

为便于在向量机上计算，亦可把式（8-34）写成向量形式

$$s(x)=s_i(x)=[1,\ (x-x_{i-1}),\ (x-x_{i-1})^2,\ (x-x_{i-1})^3]\times\begin{bmatrix}\alpha_0\\ \alpha_1\\ \alpha_2\\ \alpha_3\end{bmatrix} \tag{8-43}$$

其中

$$x_{i-1}\leqslant x\leqslant x_i,\quad i=1,\ 2,\ \cdots,\ n$$

$$\begin{pmatrix}\alpha_0\\ \alpha_1\\ \alpha_2\\ \alpha_3\end{pmatrix}=\begin{pmatrix}1 & 0 & 0 & 0\\ 0 & 0 & 1 & 0\\ -\dfrac{3}{h_i^2} & \dfrac{3}{h_i^2} & -\dfrac{2}{h_i^2} & -\dfrac{1}{h_i}\\ \dfrac{2}{h_i^3} & -\dfrac{2}{h_i^3} & \dfrac{1}{h_i^2} & \dfrac{1}{h_i^2}\end{pmatrix}\begin{pmatrix}y_{i-1}\\ y_i\\ m_{i-1}\\ m_i\end{pmatrix} \tag{8-44}$$

8.3.3 三次插值样条的并行计算

我们对三次插值样条公式的向量形式(8-43）作适当的变形，就可以同时计算 n 个三次插值样条函数。

若给定一组插值自变量$x_1^{(0)}$，$x_2^{(0)}$，…，$x_n^{(0)}$，其中

$$x_{i-1}<x_i^{(0)}<x_i\quad i=1,\ 2,\ \cdots,\ n$$

则并行计算三次插值样条函数的步骤如下：

首先考虑按照函数值及二阶导数决定的表示式：

（1）并行解M连续性方程组（8-30）；

（2）令$\mathbf{X}^0=(x_1^{(0)},\ x_2^{(0)},\ \cdots,\ x_n^{(0)})^T$，将 $x_1^{(0)}$，$x_2^{(0)}$，…，$x_n^{(0)}$ 分别代入式（8-32）经过整理后得向量表示式

$$\begin{aligned}\boldsymbol{S}&=\boldsymbol{\alpha}_0+(\boldsymbol{X}^{(0)}-\boldsymbol{X})\boldsymbol{\alpha}_1+(\boldsymbol{X}^{(0)}-\boldsymbol{X})^2\boldsymbol{\alpha}_2+(\boldsymbol{X}^{(0)}-\boldsymbol{X})^3\boldsymbol{\alpha}_3\\&=(((\boldsymbol{X}^{(0)}-\boldsymbol{X})\boldsymbol{\alpha}_3+\boldsymbol{\alpha}_2)(\boldsymbol{X}^{(0)}-\boldsymbol{X})+\boldsymbol{\alpha}_1)\\&\quad\times(\boldsymbol{X}^{(0)}-\boldsymbol{X})+\boldsymbol{\alpha}_0\end{aligned}\tag{8-45}$$

其中

$$\boldsymbol{X}=(x_0,\ x_1,\ \cdots,\ x_{n-1})^T$$

$$\boldsymbol{\alpha}_0=(y_0,\ y_1,\ \cdots,\ y_{n-1})^T$$

$$\boldsymbol{\alpha}_1=\begin{pmatrix}\frac{1}{h_1}&&&\\&\frac{1}{h_2}&&\\&&\ddots&\\&&&\frac{1}{h_n}\end{pmatrix}\left(\begin{pmatrix}y_1\\y_2\\\vdots\\y_n\end{pmatrix}-\begin{pmatrix}y_0\\y_1\\\vdots\\y_{n-1}\end{pmatrix}\right)-\frac{1}{6}\begin{pmatrix}h_1&&&\\&h_2&&\\&&\ddots&\\&&&h_n\end{pmatrix}\left(\begin{pmatrix}M_1\\M_2\\\vdots\\M_n\end{pmatrix}+2\begin{pmatrix}M_0\\M_1\\\vdots\\M_{n-1}\end{pmatrix}\right)$$

$$\boldsymbol{\alpha}_2=\frac{1}{2}(M_0,\ M_1,\ \cdots,\ M_{n-1})^T$$

$$\boldsymbol{\alpha}_3=\frac{1}{6}\begin{pmatrix}\frac{1}{h_1}&&&\\&\frac{1}{h_2}&&\\&&\ddots&\\&&&\frac{1}{h_n}\end{pmatrix}\left(\begin{pmatrix}M_1\\M_2\\\vdots\\M_n\end{pmatrix}-\begin{pmatrix}M_0\\M_1\\\vdots\\M_{n-1}\end{pmatrix}\right)$$

而

$$\boldsymbol{h}=\begin{bmatrix}h_1\\h_2\\\vdots\\h_n\end{bmatrix}=\begin{bmatrix}x_1\\x_2\\\vdots\\x_n\end{bmatrix}-\begin{bmatrix}x_0\\x_1\\\vdots\\x_{n-1}\end{bmatrix}$$

若向量机的等效并行台数为 n，按照计算公式（8-45）显然可以在向量机上同时计算 n 个三次插值样条函数。计算向量 $\boldsymbol{\alpha}_1$ 和 $\boldsymbol{\alpha}_3$ 共需五个向量加法，五个向量乘法。计算 n 个三次插值样条函数共需九个向量加法、八个向量乘法。同时，我们看到函数值 y_i 和 $M_i(i=0,\ 1,\ \cdots,\ n)$ 在参与向量运算时，只相差一

个向量单元，可通过双位移的向量指令来形成所需要的函数值，而不必重新从内存再度调出，这样就提高了向量运算的速度。

其次，我们可按照函数值及一阶导数决定的表示式，列出三次插值样条函数的并行计算步骤：

（1）用解三对角系统问题的高效并行算法求解m连续性方程组（8-42）；

（2）对给定的一组自变量$x_1^{(0)}$，$x_2^{(0)}$，…，$x_n^{(0)}$并行计算三次插值样条函数。

其中

$$x_{i-1}<x_i^{(0)}<x_i \quad i=1,\ 2,\ \cdots,\ n$$

三次插值样条函数可按下列向量公式计算：

$$\boldsymbol{S}=\boldsymbol{\alpha}_0+(\boldsymbol{X}^{(0)}-\boldsymbol{X})\boldsymbol{\alpha}_1+(\boldsymbol{X}^{(0)}-\boldsymbol{X})^2\boldsymbol{\alpha}_2+(\boldsymbol{X}^{(0)}-\boldsymbol{X})^3\boldsymbol{\alpha}_3$$

$$=(((\boldsymbol{X}^{(0)}-\boldsymbol{X})\boldsymbol{\alpha}_3+\boldsymbol{\alpha}_2)(\boldsymbol{X}^{(0)}-\boldsymbol{X})+\boldsymbol{\alpha}_1)(\boldsymbol{X}^{(0)}-\boldsymbol{X})+\boldsymbol{\alpha}_0$$

其中

$$\boldsymbol{X}^{(0)}=(x_1^{(0)},\ x_2^{(0)},\ \cdots,\ x_n^{(0)})^T$$

$$\boldsymbol{X}=(x_0,\ x_1,\ \cdots,\ x_{n-1})^T$$

$$\boldsymbol{\alpha}_0=(y_0,\ y_1,\ \cdots,\ y_{n-1})^T$$

$$\boldsymbol{\alpha}_1=(m_0,\ m_1,\ \cdots,\ m_{n-1})^T$$

$$\boldsymbol{\alpha}_2=\begin{pmatrix}\frac{3}{h_1^2}\\ \frac{3}{h_2^2}\\ \vdots\\ \frac{3}{h_n^2}\end{pmatrix}\left(\begin{pmatrix}y_1\\ y_2\\ \vdots\\ y_n\end{pmatrix}-\begin{pmatrix}y_0\\ y_1\\ \vdots\\ y_{n-1}\end{pmatrix}\right)-\begin{pmatrix}\frac{2}{h_1^2}\\ \frac{2}{h_2^2}\\ \vdots\\ \frac{2}{h_n^2}\end{pmatrix}\begin{pmatrix}m_0\\ m_1\\ \vdots\\ m_{n-1}\end{pmatrix}-\begin{pmatrix}\frac{1}{h_1}\\ \frac{1}{h_2}\\ \vdots\\ \frac{1}{h_n}\end{pmatrix}\begin{pmatrix}m_1\\ m_2\\ \vdots\\ m_n\end{pmatrix}$$

$$\boldsymbol{\alpha}_3=\begin{pmatrix}\frac{2}{h_1^3}\\ \frac{2}{h_2^3}\\ \vdots\\ \frac{2}{h_n^3}\end{pmatrix}\left(\begin{pmatrix}y_0\\ y_1\\ \vdots\\ y_{n-1}\end{pmatrix}-\begin{pmatrix}y_1\\ y_2\\ \vdots\\ y_n\end{pmatrix}\right)+\begin{pmatrix}\frac{1}{h_1^2}\\ \frac{1}{h_2^2}\\ \vdots\\ \frac{1}{h_n^2}\end{pmatrix}\left(\begin{pmatrix}m_0\\ m_1\\ \vdots\\ m_{n-1}\end{pmatrix}+\begin{pmatrix}m_1\\ m_2\\ \vdots\\ m_n\end{pmatrix}\right)$$

8.4 双三次样条函数的并行计算

8.4.1 双三次样条函数的定义和插值问题的提法

设在X-Y平面上的矩形区域R：$a \leqslant x \leqslant b$，$c \leqslant y \leqslant d$上给定一个矩形网格分割

$$\Delta x:\quad a = x_0 < x_1 < \cdots < x_n = b$$
$$\Delta y:\quad c = y_0 < y_1 < \cdots < y_n = d$$

网格上所有节点的集合记为$\Pi = \{\xi_{ij}\}(i = 0, 1, \cdots, n);(j = 0, 1, \cdots, m)$。

$S_\Pi(x, y)$称为R上的双三次样条，是指它具有以下二个性质：

（1）在每个子网格R_{ij}内$S(x, y)$是x，y的双三次多项式；

（2）$S_\Pi \in C_2^2(R)$即S_Π在R上有直至四阶的混合偏导数(其中包括不超过二阶的单向偏导数）连续。

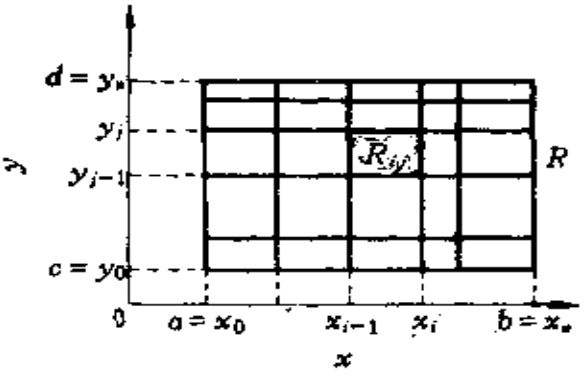


图 8-2

若假定$S(x, y)$除满足上述条件（1）、（2）外，它在$(n+1)(m+1)$个节点$(x_i, y_j)(i = 0, 1, \cdots, n;\ j = 0, 1, \cdots, m)$处满足插值条件

$$S(x_i, y_j) = u_{ij}\quad i = 0, 1, \cdots, n;\quad j = 0, 1, \cdots, m \tag{8-46}$$

则称$S(x, y)$为插值双三次样条函数。

由性质（1）的假定，双三次样条函数在R_{ij}内有如下表达式：

$$S(x, y) = \sum_{k,\ l=1}^{4} a_{ijkl}(x - x_i)^{k-1}(y - y_j)^{l-1} \tag{8-47}$$

$$x_i \leqslant x \leqslant x_{i+1},\ y_j \leqslant y \leqslant y_{j+1} \quad i = 1,\ 2,\ \cdots,\ n-1;$$
$$j = 1,\ 2,\ \cdots,\ m-1$$

不难证明，矩形域R上的双三次样条的自由度等于二方向一维三次样条自由度乘积$(n+3)(m+3)$。对于插值问题，节点上给出$(n+1)(m+1)$个值$S(x_i, y_j)=u_{ij}$($i=0,1,\cdots,n$；$j=0,1,\cdots,m$)。剩下的$2(n+m+4)$个自由度由边界条件确定。

8.4.2 双三次样条函数的矩阵表示形式

下面我们以第一类边界条件为例，即

$$p_{ij}=S_x(x_i, y_j) \quad j=0,1,\cdots,m;\ i=0,n \tag{8-48}$$

$$q_{ij}=S_y(x_i, y_j) \quad j=0,m;\ i=0,1,\cdots,n \tag{8-49}$$

$$r_{ij}=S_{xy}(x_i, y_j) \quad i=0,n;\ j=0,m \tag{8-50}$$

来说明双三次样条函数的具体求法。

一、双三次样条函数所需要的节点导数值的求法

应用一维三次样条公式可算得所有节点上的p_{ij}，q_{ij}与r_{ij}。

1. 计算p_{ij}

先固定$y=y_j$，应用三次样条函数的m连续性方程(8-37)，我们可得到

$$\lambda_i p_{i-1,j}+2p_{ij}+\mu_i p_{i+1,j}=3\left[\lambda_i\frac{u_{ij}-u_{i-1,j}}{h_i}+\mu_i\frac{u_{i+1,j}-u_{ij}}{h_{i+1}}\right]$$
$$i=1,\ 2,\ \cdots,\ n-1 \tag{8-51}$$

其中

$$h_i=x_i-x_{i-1},\quad \lambda_i=\frac{h_{i+1}}{h_i+h_{i+1}},\quad \mu_i=\frac{h_i}{h_i+h_{i+1}}$$

由边界条件(8-48)可知，式(8-51)的边界切向p_{0j}和p_{nj}是已知的，由此从式(8-51)可解出p_{ij}($i=0,1,\cdots,n$)。这样我们分别取$j=0,1,\cdots,m$，便求出R上所有节点(x_i, y_j)处沿x方向的一阶偏导数p_{ij}($i=0,1,\cdots,n$；$j=0,1,\cdots,m$)。

2．计算 q_{ij}

先固定 $x=x_i$，应用三次样条函数的 m 连续性方程(8-37)，我们可得到

$$\lambda_j^* q_{i,j-1}+2q_{ij}+\mu_j^* q_{i,j+1}=3\left[\lambda_j^*\frac{u_{ij}-u_{i,j-1}}{g_j}+\mu_j^*\frac{u_{i,j+1}-u_{ij}}{g_{j+1}}\right]$$
$$j=1,\ 2,\ \cdots,\ m-1 \qquad (8\text{-}52)$$

其中

$$\lambda_j^*=\frac{g_{j+1}}{g_j+g_{j+1}},\quad \mu_j^*=\frac{g_j}{g_j+g_{j+1}},\quad j=1,\ 2,\ \cdots,m-1$$

而

$$g_j=y_j-y_{j-1}\quad j=1,\ 2,\ \cdots,\ m$$

其边界切向 q_{i0} 和 q_{im} 可从式（8-49）得到。由此从式（8-52)可解出 $q_{ij}(j=0,\ 1,\ \cdots,\ m)$。这样我们分别取 $i=0,1,\cdots,n$，便求出 R 上所有节点 $(x_i,\ y_j)$处沿 y 方向的一阶偏导数 q_{ij} $(i=0,\ 1,\ \cdots,\ n;\ j=0,\ 1,\ \cdots,\ m)$。

3．计算 r_{ij}

先固定 $x=x_i$，因为函数 $S_x'(x_i,\ y)$ 在各节点 y_j 处的函数值 $S_x'(x_i,\ y_j)=p_{ij}$ 已从式（8-51）求得。因此得到关于变元 y 的三次样条函数的 m 连续性方程

$$\lambda_j^* r_{i,j-1}+2r_{ij}+\mu_j^* r_{i,j+1}=3\left[\lambda_j^*\frac{p_{ij}-p_{i,j-1}}{g_j}+\mu_j^*\frac{p_{i,j+1}-p_{ij}}{g_{j+1}}\right]$$
$$j=1,\ 2,\ \cdots,\ m-1 \qquad (8\text{-}53)$$

若给定了连续性方程（8-53）的边界切向 r_{i0} 和 r_{im}，那么就解得 $r_{ij}(j=0,\ 1,\ \cdots,\ m)$，再改变 x_i，R 上所有节点 $(x_i,\ y_j)$ 处的二阶混合偏导数 $r_{ij}(i=0,\ 1,\ \cdots,\ n;\ j=0,1,\ \cdots,\ m)$ 就全部确定了。

式（8-53)的边界切向 r_{i0} 和 r_{im} 可以按下述方法求得：

沿 R 的两条边界 $y=y_0$ 与 $y=y_m$，函数 $S_y'(x,y_0)$ 与 $S_y'(x,y_m)$ 在各节点 x_i 处的函数值 $S_y'(x_i,\ y_0)=q_{i0}$ 和 $S_y'(x_i,y_m)=q_{im}$ 可以从式（8-52）得到。于是我们有关于变元 x 的 m 连续性方程

$$\lambda_i r_{i-1,\beta}+2r_{i\beta}+\mu_i r_{i+1,\beta}=3\left[\lambda_i\frac{q_{i\beta}-q_{i-1,\beta}}{h_i}+\mu_i\frac{q_{i+1,\beta}-q_{i\beta}}{h_{i+1}}\right]$$

$$i=1,\ 2,\ \cdots,\ n-1;\ \beta=0,\ m \qquad (8\text{-}54)$$

它们的边界条件由式（8-50）提供。由此解出 r_{i0} 和 r_{im}（$i=0$，1，…，n）。至此，求出了所有节点上的导数值 p_{ij}，q_{ij} 和 r_{ij}。

二、双三次样条函数的矩阵表示式

当所有节点上的导数值 p_{ij}，q_{ij} 与 r_{ij} 求得以后，在每个子矩形内就确定了四阶方阵 K_{ij}，即

$$K_{ij}=\begin{bmatrix} u_{ij} & q_{ij} & u_{i,j+1} & q_{i,j+1} \\ p_{ij} & r_{ij} & p_{i,j+1} & r_{i,j+1} \\ u_{i+1,j} & q_{i+1,j} & u_{i+1,j+1} & q_{i+1,j+1} \\ p_{i+1,j} & r_{i+1,j} & p_{i+1,j+1} & r_{i+1,j+1} \end{bmatrix}$$

容易验证，R_{ij} 上的双三次样条函数可以唯一地用矩阵形式表示。

$$S(x,\ y)=[(x-x_{i-1})][B(h_i)]K_{ij}[B(g_j)]^T\cdot[(y-y_{j-1})]^T \qquad (8\text{-}55)$$

$$(x,\ y)\in R_{ij}$$

其中记号 $[(x-x_{i-1})]$ 代表关于变元 $(x-x_{i-1})$ 的 1×4 阶矩阵。$[(x-x_{i-1})]=[(x-x_{i-1})^3,\ (x-x_{i-1})^2,\ (x-x_{i-1}),\ 1]$。$[B(h_i)]$ 代表 4×4 阶矩阵。

$$[B(h_i)]=\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ -\frac{3}{h_i^2} & -\frac{2}{h_i} & \frac{3}{h_i^2} & -\frac{1}{h_i} \\ \frac{2}{h_i^3} & \frac{1}{h_i^2} & -\frac{2}{h_i^3} & \frac{1}{h_i^2} \end{bmatrix}$$

事实上，只要将 R_{ij} 的四个顶点坐标 $(x_{i-1},\ y_{j-1})$、$(x_{i-1},\ y_j)$、$(x_i,\ y_{j-1})$、$(x_i,\ y_j)$ 代入式（8-55），然后再分别将式（8-55）两边对 x 求偏导数、对 y 求偏导数和对 x、y 求混合偏

导数。这样从得到的四个式子的两边立即可看出矩阵 K_{ij} 中16个元素分别具有前面所述的含意。

从式（8-55）可以看出，双三次样条函数 $S(x, y)$ 在子矩形区域 R_{ij} 上的表示式完全决定于矩阵 K_{ij}。而双三次样条函数 $S(x, y)$ 的系数 $A_{ij}=(a_{ijkl})(k, l=1, 2, 3, 4)$ 则可以表示为

$$A_{ij}=[B(x_i-x_{i-1})]K_{ij}[B(y_j-y_{j-1})]^T \tag{8-56}$$

8.4.3 双三次样条函数的并行计算

由8.4.2可知，双三次样条函数 $S(x, y)$ 的计算步骤可归结为：

（1）给定插值条件（8-46），边界条件（8-48）、(8-49）和(8-50)；

（2）应用一维三次样条公式依次解三对角方程组（8-51)、(8-52)、(8-54）和（8-53)，求得所有节点上的 p_{ij}，q_{ij} 与 r_{ij} $(i=0, 1, \cdots, n; j=0, 1, \cdots, m)$。这些方程组共有 $2n+m+5$ 个。它们是具有严格主对角线优势的三对角线方程组；

（3）为了求 R 中任意一点 (x, y) 所对应的函数值 $S(x, y)$，只要找出整数下标 i 和 j，使得 $(x, y)\in R_{ij}$，然后用矩阵表达式（8-55）计算。

对某些实际问题，常要求在每个子网格 $R_{i+1,j+1}$ 内任意一点 $(x_i^{(j)}, y_j^{(j)})$ 上计算所对应的函数值 $S(x_i^{(j)}, y_j^{(j)})$。

其中

$$x_i<x_i^{(j)}<x_{i+1},\ y_j<y_j^{(j)}<y_{j+1}$$

$$(x_i^{(j)}, y_j^{(j)})\in R_{i+1,j+1}\quad i=0, 1, \cdots, n-1;\ j=0, 1, \cdots, m-1$$

我们可以分层处理，来实现并行计算。首先固定 j，计算在子网格 $R_{i+1,j+1}$ 内任意一点 $(x_i^{(j)}, y_j^{(j)})$ 上所对应的函数值 $S(x_i^{(j)}, y_j^{(j)})$，$(i=0, 1, \cdots, n-1)$。下面将公式(8-55)化为

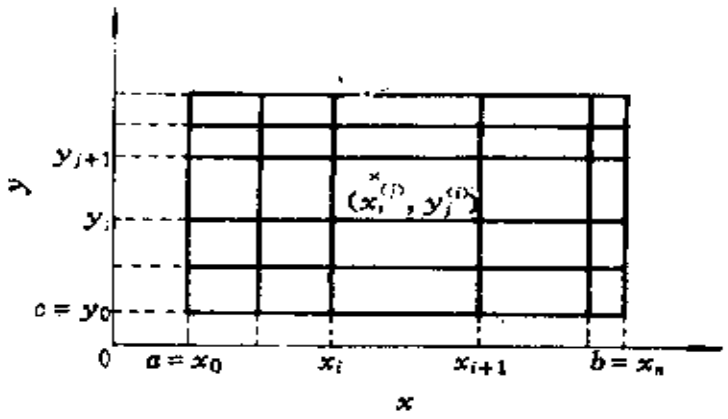


图 8-3

便于向量运算的形式。

由式（8-56）可知，双三次样条函数 $S(x, y)$ 的系数 $A_{ij}=(a_{ijkl})$，（$k, l=1, 2, 3, 4$）可表示为

$$A_{ij}=[B(x_i-x_{i-1})]K_{ij}[B(y_j-y_{j-1})]^T$$

不难验证：

$$a_{ij11}=u_{ij}$$

$$a_{ij12}=q_{ij}$$

$$a_{ij13}=\frac{3}{g_i^2}(u_{i,j+1}-u_{ij})-\frac{1}{g_i}(2q_{ij}+q_{i,j+1})$$

$$a_{ij14}=\frac{2}{g_i^3}(u_{ij}-u_{i,j+1})+\frac{1}{g_i^2}(q_{ij}+q_{i,j+1})$$

$$a_{ij21}=p_{ij}$$

$$a_{ij22}=r_{ij}$$

$$a_{ij23}=\frac{3}{g_i^2}(p_{i,j+1}-p_{ij})-\frac{1}{g_i}(r_{i,j+1}+2r_{ij})$$

$$a_{ij24}=\frac{2}{g_i^3}(p_{ij}-p_{i,j+1})+\frac{1}{g_i^2}(r_{ij}+r_{i,j+1})$$

$$a_{ij31}=\frac{3}{h_i^2}(u_{i+1,j}-u_{ij})-\frac{1}{h_i}(p_{i+1,j}+2p_{ij})$$

$$a_{ij32}=\frac{3}{h_i^2}(q_{i+1,j}-q_{ij})-\frac{1}{h_i}(r_{i+1,j}+2r_{ij})$$

$$a_{ij33}=-\frac{3}{g_j^2}a_{ij31}-\frac{2}{g_j}a_{ij32}+\frac{3}{g_j^2}\left(-\frac{3}{h_i^2}u_{i,j+1}-\frac{2}{h_i}p_{i,j+1}\right.$$
$$\left.+\frac{3}{h_i^2}u_{i+1,j+1}-\frac{1}{h_i}p_{i+1,j+1}\right)-\frac{1}{g_j}\left(-\frac{3}{h_i^2}q_{i,j+1}\right.$$
$$\left.-\frac{2}{h_i}r_{i,j+1}+\frac{3}{h_i^2}q_{i+1,j+1}-\frac{1}{h_i}r_{i+1,j+1}\right)$$

$$a_{ij34}=-\frac{2}{g_j^3}a_{ij31}+\frac{1}{g_j^2}a_{ij32}-\frac{2}{g_j^3}\left(-\frac{3}{h_i^2}u_{i,j+1}-\frac{2}{h_i}p_{i,j+1}\right.$$
$$\left.+\frac{3}{h_i^2}u_{i+1,j+1}-\frac{1}{h_i}p_{i+1,j+1}\right)+\frac{1}{g_j^2}\left(-\frac{3}{h_i^2}q_{i,j+1}\right.$$
$$\left.-\frac{2}{h_i}r_{i,j+1}+\frac{3}{h_i^2}q_{i+1,j+1}-\frac{1}{h_i}r_{i+1,j+1}\right)$$

$$a_{ij41}=\frac{2}{h_i^3}(u_{ij}-u_{i+1,j})+\frac{1}{h_i^2}(p_{ij}+p_{i+1,j})$$

$$a_{ij42}=-\frac{2}{h_i^3}(q_{ij}-q_{i+1,j})+\frac{1}{h_i^2}(r_{ij}+r_{i+1,j})$$

$$a_{ij43}=-\frac{3}{g_j^2}a_{ij41}-\frac{2}{g_j}a_{ij42}+\frac{3}{g_j^2}\left(\frac{2}{h_i^3}u_{i,j+1}+\frac{1}{h_i^2}p_{i,j+1}\right.$$
$$\left.-\frac{2}{h_i^3}u_{i+1,j+1}+\frac{1}{h_i^2}p_{i+1,j+1}\right)-\frac{1}{g_j}\left(\frac{2}{h_i^3}q_{i,j+1}\right.$$
$$\left.+\frac{1}{h_i^2}r_{i,j+1}-\frac{2}{h_i^3}q_{i+1,j+1}+\frac{1}{h_i^2}r_{i+1,j+1}\right)$$

$$a_{ij44}=\frac{2}{g_j^3}a_{ij41}+\frac{1}{g_j^2}a_{ij42}-\frac{2}{g_j^3}\left(\frac{2}{h_i^3}u_{i,j+1}+\frac{1}{h_i^2}p_{i,j+1}\right.$$
$$\left.-\frac{2}{h_i^3}u_{i+1,j+1}+\frac{1}{h_i^2}p_{i+1,j+1}\right)-\frac{1}{g_j}\left(\frac{2}{h_i^3}q_{i,j+1}\right.$$
$$\left.+\frac{1}{h_i^2}r_{i,j+1}-\frac{2}{h_i^3}q_{i+1,j+1}+\frac{1}{h_i^2}r_{i+1,j+1}\right)$$

可以看出，对固定的下标 j，$a_{ij11}\sim a_{ij44}$ 的表示式均可分别组织对下标 i 的向量运算，其中 $i=0,1,\cdots,n-1$。

并行计算出系数 $a_{ij11}\sim a_{ij44}(i=0,1,\cdots,n-1)$后，双三次样条函数可按下面的向量公式计算：

$$S(x_i^{(j)},\ y_i^{(j)})=(x_i^{(j)}-x_i)^3[a_{ij11}(y_i^{(j)}-y_j)^3+a_{ij12}(y_i^{(j)}-y_j)^2+a_{ij13}(y_i^{(j)}-y_j)+a_{ij14}]+(x_i^{(j)}-x_i)^2[a_{ij21}(y_i^{(j)}-y_j)^3+a_{ij22}(y_i^{(j)}-y_j)^2+a_{ij23}(y_i^{(j)}-y_j)+a_{ij24}]+(x_i^{(j)}-x_i)[a_{ij31}(y_i^{(j)}-y_j)^3+a_{ij32}(y_i^{(j)}-y_j)^2+a_{ij33}(y_i^{(j)}-y_j)+a_{ij34}]+a_{ij41}(y_i^{(j)}-y_j)^3+a_{ij42}(y_i^{(j)}-y_j)^2+a_{ij43}(y_i^{(j)}-y_j)+a_{ij44} \tag{8-57}$$

$$i=0,\ 1,\ \cdots,\ n-1$$

其中

$(x_i^{(j)},\ y_i^{(j)})$ 是 j 层网格的一组点（$i=0,\ 1,\cdots,n-1$）；

$(x_i^{(j)},\ y_i^{(j)})\in R_{i+1,j+1}\quad i=0,\ 1,\ \cdots,\ n-1$。

而 $$x_i<x_i^{(j)}<x_{i+1},\ y_j\leqslant y_i^{(j)}\leqslant y_{j+1}$$

显然从式（8-57）可以看出，对固定的下标 j，可组织对下标 i（$i=0,\ 1,\ \cdots,\ n-1$）的并行计算。

用 $a_{ij11}\sim a_{ij44}$ 的表达式与式（8-57）就可以逐层地计算出每个网格内点上的双三次样条函数值。

8.5 累加弦长三次参数样条曲线的并行计算

理论分析和实际应用表明，在小挠度情况下用三次样条函数插值的曲线和用木样条直接画出来的曲线非常相近，效果很好。但是在大挠度情况下（即$|y'|\gg 1$），y''与曲率有相当大的偏差，三次样条函数的光顺性可能变坏，甚至不能使用。另外，在给定了型值点的位置和边界切向后，用三次样条函数表示的插值曲线依赖于坐标系的选择，因而缺乏几何不变性，和曲线的几何特征脱节。

本节将讨论累加弦长三次参数样条曲线及其并行计算。实践表明，累加弦长三次参数样条曲线对于大挠度曲线的拟合效果是很好的。

8.5.1 累加弦长三次参数样条曲线的矩阵表示

一、累加弦长三次参数样条曲线的定义

我们讨论参数表达的平面曲线 $x=x(t)$，$y=y(t)$。当

然，最理想的是以弧长作为参数。因为这是曲线自身的内在坐标。设在直角坐标平面上给定 $n+1$ 个型值点 $p_i(x_i, y_i)$，由于弧长本身还不知道，因此可以采用“累加弦长”的办法。

我们取参数轴 t 上的一个分割 Δ：$0=t_0<t_1<\cdots<t_n$，其中 $t_i=\sum_{j=1}^{i} l_j$（$i=1, 2, \cdots, n$）。可以看出，t 轴上的每个节点 t_i 具有累加弦长的几何意义。

对于这个分割 Δ，设想以累加弦长为自变量，于是两个函数 $x(t)$ 及 $y(t)$ 在节点 $t=t_1, t_2, \cdots, t_n$ 处的样点值为 $x(t_i)=x_i$，$y(t_i)=y_i$（$i=1, 2, \cdots, n$）。

在区间 $0=t_0<t_1<\cdots<t_n$ 上分别作两个三次样条插值函数 $x(t)$、$y(t)$。然后把它们合并起来，即得过型值点的插值曲线 $p(t)=(x(t), y(t))$。我们称 $p(t)$ 为累加弦长三次参数样条曲线。在节点处，每个分量保持一阶和二阶的导数连续。

二、累加弦长三次参数样条曲线的矩阵表示

设相邻两个型值点 p_{i-1} 和 p_i 之间第 i 段曲线方程为 $p(t)=At^3+Bt^2+Ct+D$，$(t_{i-1}\leqslant t\leqslant t_i)$。其中弦长 $l_i=\overline{p_{i-1}p_i}$，参数轴上的节点坐标 $t_i=\sum_{j=1}^{i} l_j$（$i=1, 2, \cdots, n$），因为相邻两段曲线在型值点 p_i 处保持一阶和二阶导数向量连续，并记一阶和二阶导数向量分别为 $\boldsymbol{m}_i$ 和 $\boldsymbol{M}_i$。

则有

$$\boldsymbol{P}'(t_i-0)=\boldsymbol{P}'(t_i+0)\equiv\boldsymbol{m}_i \qquad i=1, 2, \cdots, n-1$$

$$\boldsymbol{P}''(t_i-0)=\boldsymbol{P}''(t_i+0)\equiv\boldsymbol{M}_i$$

类似于三次样条函数，可得到 m 连续性方程

$$\lambda_i\boldsymbol{m}_{i-1}+2\boldsymbol{m}_i+\mu_i\boldsymbol{m}_{i+1}=3(\lambda_i\boldsymbol{e}_i+\mu_i\boldsymbol{e}_{i+1}) \qquad (8\text{-}58)$$

$$i=1, 2, \cdots, n-1$$

其中

$$\lambda_i=\frac{l_{i+1}}{l_i+l_{i+1}},\quad \mu_i=\frac{l_i}{l_i+l_{i+1}},\quad e_i=\frac{1}{l_i}\overrightarrow{p_{i-1}p_i},$$

e_i 表示相邻两型值点 P_{i-1} 和 P_i 所连弦方向上的单位向 量。

相应地，可得到M连续性方程

$$\mu_i M_{i-1}+2M_i+\lambda_i M_{i+1}=6\frac{e_{i+1}-e_i}{l_i+l_{i+1}} \tag{8-59}$$

$$i=1,\ 2,\ \cdots,\ n-1$$

连续性方程（8-58）或（8-59)添上两个适当的边界条件后，即构成完整的连续性方程。

假定参数 曲 线$p(t)=(x(t),\ y(t))$，其 中 $x(t)$ 和 $y(t)$ 为两个插值三次样条函数。我们可分别对三次样条 函 数 $x(t)$ 和$y(t)$ 求m连续性方程或M连续性方程。

由于m连续性方程和M连续性方程的系数矩阵均为 三 对 角阵，完全可以采用第五章介绍的并行算法求解。

为便于在向量机上计算，类似于 式 (8-32)、(8-43)，可 把点 p_{i-1} 与 p_i 之间的第 i 段曲线方程表示为向量形式

$$p(t)=[1,\ (t-t_{i-1}),\ (t-t_{i-1})^2,\ (t-t_{i-1})^3]\begin{bmatrix}\alpha_0\\ \alpha_1\\ \alpha_2\\ \alpha_3\end{bmatrix} \tag{8-60}$$

$$t_{i-1}\leqslant t\leqslant t_i,\quad i=1,\ 2,\ \cdots,\ n$$

其中

$$\begin{pmatrix}\alpha_0\\ \alpha_1\\ \alpha_2\\ \alpha_3\end{pmatrix}=\begin{pmatrix}1 & 0 & 0 & 0\\ 0 & 0 & 1 & 0\\ -\frac{3}{l_i^2} & \frac{3}{l_i^2} & -\frac{2}{l_i^2} & -\frac{1}{l_i}\\ \frac{2}{l_i^3} & -\frac{2}{l_i^3} & \frac{1}{l_i^2} & \frac{1}{l_i^2}\end{pmatrix}\begin{pmatrix}p_{i-1}\\ p_i\\ m_{i-1}\\ m_i\end{pmatrix}$$

$$=\begin{pmatrix} 1 & 0 & 0 & 0 \\ -\frac{1}{l_i} & \frac{1}{l_i} & -\frac{l_i}{3} & -\frac{l_i}{6} \\ 0 & 0 & \frac{1}{2} & 0 \\ 0 & 0 & -\frac{1}{6l_i} & \frac{1}{6l_i} \end{pmatrix} \begin{pmatrix} \boldsymbol{p}_{i-1} \\ \boldsymbol{p}_i \\ \boldsymbol{M}_{i-1} \\ \boldsymbol{M}_i \end{pmatrix}$$

这里

$$\boldsymbol{p}(t)=(x(t),\ y(t))$$

$$\boldsymbol{\alpha}_i=(\alpha_{1i},\ \alpha_{2i}) \quad i=0,\ 1,\ 2,\ 3$$

$$\boldsymbol{p}_i=(x(t_i),\ y(t_i)) \quad i=1,\ 2,\ \cdots,\ n$$

$$\boldsymbol{m}_i=(m_{x_i},\ m_{y_i}) \quad i=0,\ 1,\ \cdots,\ n$$

$$\boldsymbol{M}_i=(M_{x_i},\ M_{y_i})$$

$$s'(x_i)=m_{x_i},\ s''(x_i)=M_{x_i}$$

$$s'(y_i)=m_{y_i},\ s''(y_i)=M_{y_i}$$

8.5.2 累加弦长三次参数样条插值公式的并行计算

若给定一组参数$t_i^{(0)}(i=1,\ 2,\ \cdots,\ n)$，并令$\boldsymbol{t}^{(0)}=(t_1^{(0)},\ t_2^{(0)},\ \cdots,\ t_n^{(0)})$，其中$t_{i-1}<t_i^{(0)}<t_i$。下面的问题是要求出对应于累加弦长三次参数样条曲线上的n个位置向量$\boldsymbol{p}_i^{(0)}=(x_i^{(0)},\ y_i^{(0)})(i=1,\ 2,\ \cdots,\ n)$。类似于公式（8-45）的推导，由式（8-60）可推得插值样条函数$x(t)$的向量表示式

$$\begin{aligned}\boldsymbol{x}(t)&=\boldsymbol{\beta}_{10}+(t^{(0)}-t)\boldsymbol{\beta}_{11}+(t^{(0)}-t)^2\boldsymbol{\beta}_{12}+(t^{(0)}-t)^3\boldsymbol{\beta}_{13}\\&=(((t^{(0)}-t)\boldsymbol{\beta}_{13}+\boldsymbol{\beta}_{12})(t^{(0)}-t)+\boldsymbol{\beta}_{11})(t^{(0)}-t)+\boldsymbol{\beta}_{10}\end{aligned} \tag{8-61}$$

其中

$$\boldsymbol{x}=(x_1^{(0)},\ x_2^{(0)},\ \cdots,\ x_n^{(0)})^T$$

$$\boldsymbol{t}=(t_0,\ t_1,\ \cdots,\ t_{n-1})^T$$

$$\boldsymbol{\beta}_0=(x_0,\ x_1,\ \cdots,\ x_{n-1})^T$$

这里$x_i(i=0,\ 1,\ \cdots,\ n-1)$是位置向量$\boldsymbol{p}_i$的$x$分量。

$$\boldsymbol{\beta}_{11}=(m_{x_0},\ m_{x_1},\ \cdots,\ m_{x_{n-1}})^T$$

$$\boldsymbol{\beta}_{12}=-\begin{pmatrix}\frac{3}{l_1^2}\\ \frac{3}{l_2^2}\\ \vdots\\ \frac{3}{l_n^2}\end{pmatrix}\left[\begin{pmatrix}x_0\\ x_1\\ \vdots\\ x_{n-1}\end{pmatrix}-\begin{pmatrix}x_1\\ x_2\\ \vdots\\ x_n\end{pmatrix}\right]-\begin{pmatrix}\frac{2}{l_1^2}\\ \frac{2}{l_2^2}\\ \vdots\\ \frac{2}{l_n^2}\end{pmatrix}\begin{pmatrix}m_{x_0}\\ m_{x_1}\\ \vdots\\ m_{x_{n-1}}\end{pmatrix}-\begin{pmatrix}\frac{1}{l_1}\\ \frac{1}{l_2}\\ \vdots\\ \frac{1}{l_n}\end{pmatrix}\begin{pmatrix}m_{x_1}\\ m_{x_2}\\ \vdots\\ m_{x_n}\end{pmatrix}$$

$$\boldsymbol{\beta}_{13}=\begin{pmatrix}\frac{2}{l_1^3}\\ \frac{2}{l_2^3}\\ \vdots\\ \frac{2}{l_n^3}\end{pmatrix}\left[\begin{pmatrix}x_0\\ x_1\\ \vdots\\ x_{n-1}\end{pmatrix}-\begin{pmatrix}x_1\\ x_2\\ \vdots\\ x_n\end{pmatrix}\right]+\begin{pmatrix}\frac{1}{l_1^2}\\ \frac{1}{l_2^2}\\ \vdots\\ \frac{1}{l_n^2}\end{pmatrix}\left[\begin{pmatrix}m_{x_0}\\ m_{x_1}\\ \vdots\\ m_{x_{n-1}}\end{pmatrix}+\begin{pmatrix}m_{x_1}\\ m_{x_2}\\ \vdots\\ m_{x_n}\end{pmatrix}\right]$$

完全类似地，可以推得插值样条函数$y(t)$的向量表示式

$$\begin{aligned}\boldsymbol{y}(t)&=\boldsymbol{\beta}_{20}+(t^{(0)}-t)\boldsymbol{\beta}_{21}+(t^{(0)}-t)^2\boldsymbol{\beta}_{22}+(t^{(0)}-t)^3\boldsymbol{\beta}_{23}\\&=(((t^{(0)}-t)\boldsymbol{\beta}_{23}+\boldsymbol{\beta}_{22})(t^{(0)}-t)+\boldsymbol{\beta}_{21})(t^{(0)}-t)+\boldsymbol{\beta}_{20}\end{aligned}\tag{8-62}$$

其中

$$\boldsymbol{y}=(y_1^{(0)},\ y_2^{(0)},\ \cdots,\ y_n^{(0)})^T$$

$\boldsymbol{\beta}_{20}=(y_0,\ y_1,\ \cdots,\ y_{n-1})^T$，这里$y_i(i=0,\ 1,\cdots,\ n-1)$则是位置向量$\boldsymbol{p}_i$的$y$分量。

$$\boldsymbol{\beta}_{21}=(m_{y_0},\ m_{y_1},\ \cdots,\ m_{y_{n-1}})^T$$

$$\boldsymbol{\beta}_{22}=-\begin{pmatrix}\frac{3}{l_1^2}\\ \frac{3}{l_2^2}\\ \vdots\\ \frac{3}{l_n^2}\end{pmatrix}\left[\begin{pmatrix}y_0\\ y_1\\ \vdots\\ y_{n-1}\end{pmatrix}-\begin{pmatrix}y_1\\ y_2\\ \vdots\\ y_n\end{pmatrix}\right]-\begin{pmatrix}\frac{2}{l_1^2}\\ \frac{2}{l_2^2}\\ \vdots\\ \frac{2}{l_n^2}\end{pmatrix}\begin{pmatrix}m_{y_0}\\ m_{y_1}\\ \vdots\\ m_{y_{n-1}}\end{pmatrix}-\begin{pmatrix}\frac{1}{l_1}\\ \frac{1}{l_2}\\ \vdots\\ \frac{1}{l_n}\end{pmatrix}\begin{pmatrix}m_{y_1}\\ m_{y_2}\\ \vdots\\ m_{y_n}\end{pmatrix}$$

$$\beta_{23}=\begin{bmatrix}\frac{2}{l_1^3}\\ \frac{2}{l_2^3}\\ \vdots\\ \frac{2}{l_n^3}\end{bmatrix}\left(\begin{bmatrix}y_0\\ y_1\\ \vdots\\ y_{n-1}\end{bmatrix}-\begin{bmatrix}y_1\\ y_2\\ \vdots\\ y_n\end{bmatrix}\right)+\begin{bmatrix}\frac{1}{l_1^2}\\ \frac{1}{l_2^2}\\ \vdots\\ \frac{1}{l_n^2}\end{bmatrix}\left(\begin{bmatrix}m_{y_0}\\ m_{y_1}\\ \vdots\\ m_{y_{n-1}}\end{bmatrix}+\begin{bmatrix}m_{y_1}\\ m_{y_2}\\ \vdots\\ m_{y_n}\end{bmatrix}\right)$$

从式（8-61）、（8-62）可以看出，只需17次向量加法和16次向量乘法即可得到对应于累加弦长三次参数样条曲线上的 n 个位置向量。

8.6 样条磨光方法的并行计算

8.6.1 样条磨光方法

样条磨光方法是一类适应保凸要求的曲线拟合法，它是解决某些外形设计和数据处理的一个行之有效的方法。

基本样条函数（K次B样条）$\Omega_K(x)$对研究样条函数具有重要的作用。基本样条函数可以通过 δ 函数作若干次积分，再作同样次数的差分而得到

$$\Omega_K(x)=\delta^{K+1}\left(\frac{x^K}{K!}\right)=\frac{\sum_{j=0}^{K+1}(-1)^j\binom{K+1}{j}\left(x+\frac{K+1}{2}-j\right)^K_+}{K!}$$

$\Omega_K(x)$ 是一个分段K次多项式，K次导数的间断点为 $x_j=j-\frac{K+1}{2}$，$j=0,1,\cdots,K+1$。x_j 为它的样条节点。运用B样条函数可以求出阶梯函数和折线函数的统一表达式，然后研究它们的磨光函数，经几次磨光后函数越来越光滑。

阶梯函数 $f(x)$ 可表成 $\Omega_0\left(\frac{x-x_j}{h}\right)=\Omega_0\left(\frac{x-x_0}{h}-j\right)$（$j$

$=0,1,\cdots,n$）的线性组合。即 $f(x)=\sum_{j=0}^{n} y_j\Omega_0\left(\frac{x-x_0}{h}-j\right)$。

对 $f(x)$ 进行磨光，归结为对每一函数 $\Omega_0\left(\frac{x-x_i}{h}\right)$作磨光。

不难推出，阶梯函数的一次磨光函数为折线函数，即

$$S_1(x)=\sum_{j=0}^{n} y_j\Omega_1\left(\frac{x-x_0}{h}-j\right)$$

依次类推，$f(x)$ 的K次磨光函数为

$$S_K(x)=\sum_{j=0}^{K} y_j\Omega_K\left(\frac{x-x_j}{h}\right)$$

就一般问题来说，通常采用折线函数的一次磨光函数和二次磨光函数来处理能满足光滑性要求。

在工程问题中，常常要用到大量的实验曲线，这些曲线一般都是光滑的，用计算机解题时，希望寻求一种方法使构造的曲线也是光滑的，而在给定离散点的型值和原型值有较好的近似且保持其凸凹性，整体误差也小，这样就能进一步提高问题的解的精度。通常采用的方法是分段低次多项式插值，它具有稳定、收敛、准确通过型值点等优点，但是在分段节点处光滑性差。如离散点选取不当，在插值节点处可能偏离原曲线较远而造成计算结果的不合理。采用盈亏修正的样条磨光方法可以避免上述弊病，为了提高磨光函数逼近原型值的精度，而且又使凸性不受破坏，盈亏修正的次数不宜过多，通常以一、二次为宜。

8.6.2 盈亏修正的样条磨光方法的并行计算

一、对给定的一组型值点，采用盈亏修正的样条磨光方法构造拟合曲线

设二次磨光函数$S_2(x)=\sum_{j=0}^{n} y_j\Omega_2\left(\frac{x-x_j}{h}\right)$。对不带边界条件的问题，盈亏修正的迭代过程为

$$\begin{cases} S^{(m-1)}(x)=\sum_{j=0}^{n} y_j^{(m-1)}\Omega_2\left(\dfrac{x-x_j}{h}\right) \\ y_i^{(m)}=y_i^{(m-1)}-S_2^{(m-1)}(x_i)+y_i \\ y_i^{(0)}=y_i \end{cases} \tag{8-63}$$

$$i=0,\ 1,\ \cdots,\ n$$

$$m=1,\ 2,\ \cdots$$

其中$y_i^{(0)}$为原始型值，$y_i^{(m)}$为m次盈亏型值，$S(x)$为磨光函数。

为便于在向量计算机上计算，可将上述过程改写为

$$\mathbf{Y}^{(m)}=\mathbf{Y}^{(m-1)}-A_2\mathbf{Y}^{(m-1)}+\mathbf{Z} \tag{8-64}$$

其中

$$\mathbf{Y}^{(\nu)}=(y_0^{(\nu)},\ y_1^{(\nu)},\ \cdots,\ y_n^{(\nu)})^T$$

$$\mathbf{Y}^{(0)}=\mathbf{Z}$$

$$A_2=\begin{pmatrix} \frac{3}{4} & \frac{1}{8} & & & 0 \\ \frac{1}{8} & \frac{3}{4} & \frac{1}{8} & & \\ & \ddots & \ddots & \ddots & \\ & & & & \frac{1}{8} \\ 0 & & \frac{1}{8} & & \frac{3}{4} \end{pmatrix}$$

式（8-64）是易于组织并行计算的，其主要工作量是要完成$A\mathbf{Y}$型的计算。通常矩阵与向量的乘法有关系式

$$\boldsymbol{B}=A\mathbf{Y}=\sum_{K=1}^{n} Y_K A_K \tag{8-65}$$

其中 $\mathbf{Y}=(y_1,\ y_2,\ \cdots,\ y_n)^T$，$A_K$为$A$的第$K$列向量。按公式（8-65）计算一次，$A\mathbf{Y}$需要完成$n$个向量乘法和$n$个向量加法。

二（三）次样条磨光函数其系数矩阵A是三对角的且是对称阵。

令

$$A=\begin{bmatrix} a_{11} & a_{12} & \cdots & 0 \\ a_{21} & a_{22} & a_{23}\cdots & 0 \\ \vdots & & & \\ 0 & \cdots & & a_{n-1,n} \\ 0 & \cdots & a_{n,n-1} & a_{nn} \end{bmatrix}$$

我们可得

$$\boldsymbol{AY}=\begin{pmatrix} a_{11} \\ a_{22} \\ \vdots \\ a_{n-1,n-1} \\ a_{n,n} \end{pmatrix}\begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_{n-1} \\ y_n \end{pmatrix}+\begin{pmatrix} a_{12} \\ a_{23} \\ \vdots \\ a_{n-1,n} \\ 0 \end{pmatrix}\begin{pmatrix} y_2 \\ y_3 \\ \vdots \\ y_n \\ 0 \end{pmatrix}+\begin{pmatrix} 0 \\ a_{21} \\ \vdots \\ a_{n-1,n-2} \\ a_{n,n-1} \end{pmatrix}\begin{pmatrix} 0 \\ y_1 \\ \vdots \\ y_{n-2} \\ y_{n-1} \end{pmatrix}$$

$$=\begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_{n-1} \\ b_n \end{pmatrix}=\boldsymbol{B} \qquad (8\text{-}66)$$

由式（8-66）可以看出，只需三次向量乘法和二次向量加法即可完成AY型的计算。一般地，AY的计算可推广到$L>1$的情形（L为矩阵A的带宽）。完成AY的计算需$2L+1$次向量乘法和$2L+1$次向量加法（$n \gg L$）。

计算步骤如下：

步 1　$0 \Rightarrow \boldsymbol{B}$

步 2　$b_{K+j}+a_{K+j,j} * y_j \Rightarrow b_{K+j}$

$K=1,\ 2,\ \cdots,\ L;\ j=1,\ 2,\ \cdots,\ n-K$

步 3　$b_j+a_{j,K+j} * y_{K+j} \Rightarrow b_j$

$K=0,\ 1,\ \cdots,\ L;\ j=1,\ 2,\ \cdots,\ n-K$

步 2 和步 3 对下标K均可组织对下标j的向量运算。这种类型的计算格式在样条函数方法的并行计算中是经常用到的。

在串行计算机上公式（8-64）只能步进的进行运算，在同一时间只能执行一种操作。计算一点的磨光函数值需三个乘法，二

个加法。计算一点的盈亏修正值需二个加法。对给定的 $n+1$ 个点，经二次盈亏修正得到对应的磨光拟合函数值共需乘法 $9(n+1)$ 次、加法 $10(n+1)$ 次。而用向量计算机计算只需 9 次向量乘法和10次向量加法即可完成。

在实际计算中，曲线常以表格函数的形式给出，但造表是否合理，离散点的选取是否得当，可采用盈亏修正的样条磨光方法加以检验。在对原型值作了一（二）次盈亏修正后，检验磨光函数值与对应原型值之差，以及盈亏型值和原型值之差能否控制在允许的误差范围内。即考察

$$\left|\frac{S_2^*(x_i)-y_i^{(0)}}{y_i^{(0)}}\right|\sim\varepsilon_1,\quad \left|\frac{y_i^{(*)}-y_i^{(0)}}{y_i^{(0)}}\right|\sim\varepsilon_2$$

$$i=1,\ 2,\ \cdots,\ n$$

其中 $y_i^*(i=1,\ 2,\ \cdots,\ n)$ 为经一（二）次盈亏修正得到的在各结点上的盈亏型值。$S_2^*(x_i)$ 是利用盈亏型值 $y_i^{(*)}$ 作出的相应的磨光函数值。$y_i^{(0)}$ 为原始型值。

若所有各点未能控制在允许的误差范围内，则需对原节点作必要的调整。

二、经拟合的型值点确定后，二次样条磨光公式可作为插值公式使用

二次样条磨光公式可表示为

$$\begin{cases} S_2(x)=\displaystyle\sum_{j=0}^{n+1} y_j\Omega_2\left(\frac{x-x_j}{h}\right)=\sum_{j=0}^{n+1} y_j\Omega_2\left(\frac{x-x_0}{h}-j\right) \\ \Omega_2(x)=\dfrac{\displaystyle\sum_{j=0}^{3}(-1)^j\binom{3}{j}\left(x+\frac{3}{2}-j\right)^2}{2!} \end{cases} \tag{8-67}$$

当需要插值的是一组点时，可用公式（8-67）组织向量运算。事实上，只要注意到 $\Omega_K(x)$ 的基本性质，即当 $|x|>\frac{K+1}{2}$ 时，$\Omega_K(x)=0$。可把公式（8-67）改写为

$$S_2=\mathbf{Y}_{I-1}\Omega_2\left(\frac{\mathbf{X}-x_0}{h}-(\boldsymbol{I}-1)\right)+\mathbf{Y}_I\,\Omega_2\left(\frac{\mathbf{X}-x_0}{h}-\boldsymbol{I}\right)$$
$$+\mathbf{Y}_{I+1}\Omega_2\left(\frac{\mathbf{X}-x_0}{h}-(\boldsymbol{I}+1)\right) \qquad (8\text{-}68)$$

其中

$$\boldsymbol{I}=\begin{bmatrix} i_1 \\ i_2 \\ \vdots \\ i_n \end{bmatrix},\quad \mathbf{X}=\begin{bmatrix} x^{(1)} \\ x^{(2)} \\ \vdots \\ x^{(n)} \end{bmatrix},\quad \mathbf{Y}_{I-1}=\begin{bmatrix} y_{i_1-1} \\ y_{i_2-1} \\ \vdots \\ y_{i_n-1} \end{bmatrix},$$

$$\mathbf{Y}_I=\begin{bmatrix} y_{i_1} \\ y_{i_2} \\ \vdots \\ y_{i_n} \end{bmatrix},\quad \mathbf{Y}_{I+1}=\begin{bmatrix} y_{i_1+1} \\ y_{i_2+1} \\ \vdots \\ y_{i_n+1} \end{bmatrix}$$

公式（8-68）的并行计算步骤如下：

（1）并行计算

$$\boldsymbol{I}^*=\begin{bmatrix} i_1^* \\ i_2^* \\ \vdots \\ i_n^* \end{bmatrix}$$

其中分量 $i_j^*=\left[\frac{x^{(j)}-x_0}{h}+\frac{1}{2}\right]\quad j=1,\ 2,\ \cdots,\ n$

（2）并行计算

$$\boldsymbol{I}=\begin{bmatrix} i_1 \\ i_2 \\ \vdots \\ i_n \end{bmatrix}$$

其中分量

$$i_j=\begin{cases} 1 & i_j^*\leqslant 1 \\ i_j^* & 1<i_j^*<n-1 \\ n-1 & i_j^*\geqslant n-1 \end{cases}\quad j=1,\ 2,\ \cdots,\ n$$

（3）并行计算

$$\Omega_2\left(\frac{X-x_0}{h}-(I-1)\right),\ \Omega_2\left(\frac{X-x_0}{h}-I\right)$$

$$\Omega_2\left(\frac{X-x_0}{h}-(I+1)\right)$$

只要我们注意流水线向量机的特点并充分利用测试指令、双位移指令、向量归并指令和向量位串寄存器是不难实现向量运算的。同时由于减少了数据传输和内外存交换因而使得效率提高。

第九章　非线性方程组并行求解法

9.1　牛顿迭代法

在本书上册的第五章中利用二分法已给出一种对非线性方程

$$f(x)=0$$

行之有效的并行求解方法。在谢德勒（G.S.SHEDLER）关于求方程根的并行数值方法的文章中还给出利用割线法、切线法及抛物线法求方程近似根的并行计算方法以及这些方法的效能分析，此处不拟重复讨论。本章将重点讨论具有 n 个变量 n 个方程的方程组

$$\begin{cases} f_1(x_1,\ x_2,\ \cdots,\ x_n)=0 \\ f_2(x_1,\ x_2,\ \cdots,\ x_n)=0 \\ \quad\cdots\cdots \\ f_n(x_1,\ x_2,\ \cdots,\ x_n)=0 \end{cases} \tag{9-1}$$

的并行求解问题。

若引进映射 F：$D\subset R^n\to R^n$，则方程组（9-1）可简写为

$$F(x)=0 \tag{9-2}$$

此时，$f_i(i=1,\ 2,\ \cdots,\ n)$ 为 F 的分量。

在后面的讨论中，我们常用到下列条件：

（1）映射 F 于开凸集 $D\subset R^n$ 连续可微；

（2）存在 $x^*\in D$，使 $F(x^*)=0$，且 $F(x)$ 的雅可比（Jacobi）矩阵在 x^* 处非奇异；

（3）$F'(x)$ 于 D 内是李普希兹（Lipschitz）连续，即有常数 L 使

$$\|F'(x)-F'(y)\|\leqslant L\|x-y\|,\qquad x,\ y\in D$$

成立。

不难证明，若映射F满足条件（1），（2），则x^*是方程(9-2）的局部唯一解。

和线性方程组的情形不同，解非线性方程组的直接方法通常只用于形式很特殊的小型方程组。因而，我们的讨论将局限于迭代方法。

在迭代方法中最熟悉的方法是牛顿迭代法，即若$x^{(0)}$为x^*的一个初始近似，则牛顿迭代法的一般形式为

$$x^{(k+1)}=x^{(k)}-F'(x^{(k)})^{-1}F(x^{(k)}),$$
$$k=0,\ 1,\ \cdots \tag{9-3}$$

在实际计算中可按以下顺序进行：

（1）若$x^{(k)}$为已知，则并行计算$F(x^{(k)})$；若满足精度要求，则$x^{(k)}$即为所求。否则，计算$F'(x^{(k)})$。

（2）若记$s^{(k)}=x^{(k+1)}-x^{(k)}$，则可用第五章的方法求解关于$s^{(k)}$的线性方程组

$$F'(x^{(k)})s^{(k)}=-F(x^{(k)}) \tag{9-4}$$

关于此算法的收敛性有以下熟知的结果。

定理9-1 设F，$D\subset R^n\to R^n$满足条件（1），（2）。则存在包含x^*的邻域$s\subset D$，使对任何$x^{(0)}\in s$，牛顿迭代式（9-3）在s中适定，且收敛到x^*，同时，存在收敛到零的序列$\{\alpha_k\}$满足

$$\|x^{(k+1)}-x^*\|\leqslant\alpha_k\|x^{(k)}-x^*\|\quad k=0,\ 1,\ \cdots \tag{9-5}$$

此外，若F还满足（3），则存在常数β使

$$\|x^{(k+1)}-x^*\|\leqslant\beta\|x^{(k)}-x^*\|^2\quad k=0,\ 1,\ \cdots \tag{9-6}$$

由定理9-1可见牛顿迭代法具有很快的收敛速度，但由于它在每一步迭代中均需计算导数$F'(x^{(k)})$，无疑就给具体计算带来困难。改进办法之一就是以$F'(x^{(k)})$的离散逼近β_k代替$F'(x^{(k)})$。即β_k满足

$$\beta_k e_j=[F(x^{(k)}+h_k e_j)-F(x^{(k)})]/h_k$$
$$j=1,\ 2,\ \cdots,\ n \tag{9-7}$$

其中，$e_j=(0,\ \cdots,\ 1,\ \cdots,\ 0)^T\in R^n$；$h_k$是步长，$h_k$可如

下选择：

设 $\gamma > 0$ 为常数，则

$$h_k \leqslant \gamma \|F(x^{(k)})\| \quad k = 0, 1, \cdots \tag{9-8}$$

或

$$h_k \leqslant \gamma \|x^{(k)} - x^{(k-1)}\| \quad k = 1, 2, \cdots \tag{9-9}$$

这样得到的迭代

$$x^{(k+1)} = x^{(k)} - \beta_k^{-1} F(x^{(k)}) \quad k = 0, 1, \cdots \tag{9-10}$$

称为离散牛顿迭代法。

离散牛顿法只是消除了计算导数的困难，而在计算量方面与牛顿法一样，每次迭代均需计算 $n+1$ 个向量函数值。因此，上述两种方法虽然收敛速度较快，但由于计算量较大因而限制了求解问题时的效率（效率的严格定义将在9.2.2中给出）。为了减少计算量而提高效率，可考虑采用部分修正的方法。

9.2 部分修正方法

9.2.1 部分修正算法

令 $N = \{1, 2, \cdots, n\}$，任意给定 $m \in N$，并设

$$M(k) = \{j \mid j \equiv ((k-1)m + p) \bmod(n), \ p = 1, \cdots, m\}$$

相应于牛顿法，称下述迭代

$$\begin{cases} A_0 = F'(x^{(0)}) \\ x^{(k+1)} = x^{(k)} - A_k^{-1} F(x^{(k)}) \quad k = 0, 1, \cdots \\ A_{k+1} \in R^{n \times n} \\ A_{k+1} e_j = A_k e_j \quad j \in N - M(k+1) \\ A_{k+1} e_j = F(x^{(k+1)}) e_j \quad j \in M(k+1) \end{cases} \tag{9-11}$$

为解 $F(x) = 0$ 的部分修正牛顿型方法。相应于离散牛顿法，称迭代

$$\begin{cases} A_0 = B_0 \\ x^{(k+1)} = x^{(k)} - A_k^{-1} F(x^{(k)}) \quad k = 0, 1, \cdots \\ A_{k+1} \in R^{n \times n} \\ A_{k+1} e_j = A_k \cdot e_j \quad j \in N - M(k+1) \\ A_{k+1} e_j = B_{k+1} e_j \quad j \in M(k+1) \end{cases} \tag{9-12}$$

为解$F(x)=0$的部分修正离散牛顿型方法。

上述两种方法中A_{k+1}仅有m列与A_k不同，即只需修改A_k的m列而得到A_{k+1}，每步迭代计算$m+1$个向量值函数。当$m=n$时，式（9-11）、（9-12）分别是牛顿法和离散牛顿法，这时，A_{k+1}可视为A_k的所有n列被修改的结果。

用〔x〕表示不小于x的最小整数。τ_q表示多项式$t^{q+1}-t^q-1$的唯一正根，则$\tau_q \in (1, 2)$，$\tau_q > \tau_{q+1}$且$\lim\limits_{q \to +\infty} \tau_q = 1$。关于式（9-11）和（9-12）的收敛性和收敛速度，我们有定理9-2。

定理9-2 设$F(x)$满足条件（1）、（2）、（3），则

1．存在x^*的邻域$S_1 = \{x \mid \|x - x^*\| < \delta_1\}$，对于任意$x^{(0)} \in S_1$，部分修正牛顿型方法（9-11）产生的序列$\{x^{(k)}\}$收敛于$x^*$，且根一收敛阶至少是$\tau_{\left[\frac{n}{m}\right]-1}$。

2．存在x^*的邻域$S_2 = \{x \mid \|x - x^*\| < \delta_2\}$，对于任意$x^{(0)} \in S_2$，按式（9-8）选取步长$h_k$的部分修正离散牛顿型方法（9-12）产生的序列$\{x^{(k)}\}$收敛于$x^*$，且根一收敛阶至少是$\tau_{\left[\frac{n}{m}\right]-1}$。

3．存在x^*的邻域$S_3 = \{x \mid \|x - x^*\| < \delta_3\}$，对于任意$x^{(0)} \in S_3$，按式（9-9）选取步长$h_k$的部分修正离散牛顿型方法（9-12）产生的序列$\{x^{(k)}\}$收敛于$x^*$，且根一收敛阶至少是$\tau_{\left[\frac{n}{m}\right]}$。

证明 我们仅证“3”，“1”、“2”的证明相仿。

为方便起见，范数$\|\cdot\|$表示1-范数。令$\beta \geqslant \|F'(x^*)^{-1}\|$，$\delta_3$、

δ_4为充分小的正数，且满足

$$\beta L\left(4\delta_3+\frac{3}{2}\delta_4\right)\leqslant 1$$

$$\delta_4\leqslant 2\gamma\delta_3$$

$$\beta L\left(\frac{9\gamma}{4}+2\right)\delta_3\leqslant 1$$

$$\overline{S}=\{x\mid \|x-x^*\|\leqslant(1+2\gamma\delta_3\}\subset D$$

任意给定$x^{(0)}\in S_3$，$h_0\in[-\delta_4,\ \delta_4]-\{0\}$，易知$x^{(0)}+h_0e_j\in D$，$j\in N$。从而$A_0$有意义。由$A_0=B_0$及1－范数的定义，存在$j_0\in N$，有

$$\|A_0-F'(x^*)\|=\|(B_0-F'(x^*))e_{j_0}\|$$

$$=\left\|\frac{F(x^{(0)}+h_0e_{j_0})-F(x^{(0)})}{h_0}-F'(x^*)e_{j_0}\right\|$$

$$=\frac{1}{|h_0|}\|F(x^{(0)}+h_0e_{j_0})-F(x^{(0)})-F'(x^*)h_0e_{j_0}\|$$

因$F'(x)$满足李普西兹条件，上式最后一个等号右边不超过

$$\frac{1}{|h_0|}\cdot\frac{L}{2}(\|x^{(0)}+h_0e_{j_0}-x^{(0)}\|+\|x^{(0)}-x^*\|)\|h_0e_{j_0}\|,$$

因而有

$$\|A_0-F'(x^*)\|\leqslant\frac{L}{2}(|h_0|+2\|x^{(0)}-x^*\|)$$

$$\leqslant\frac{L}{2}(2\delta_3+\delta_4)$$

而

$$\frac{\beta L}{2}(2\delta_3+\delta_4)<1$$

根据扰动定理知，A_0非奇异，且

$$\|A_0^{-1}\|\leqslant\frac{\beta}{1-\beta\|A_0-F'(x^*)\|}$$

于是

$$\|x^{(1)}-x^*\|=\|x^{(0)}-x^*-A_0^{-1}F(x^{(0)})\|$$

$$\leqslant\|A_0^{-1}\|(\|A_0-F'(x^*)\|\cdot\|x^{(0)}-x^*\|$$

$$+\|F(x^{(0)})-F(x^*)-F'(x^*)(x^{(0)}-x^*)\|$$

$$\leqslant\|A_0^{-1}\|\left(\|A_0-F'(x^{(0)})\|+\frac{L}{2}\|x^{(0)}-x^*\|\right)\|x^{(0)}-x^*\|$$

$$\leqslant\frac{\beta L/2}{1-\frac{\beta L}{2}(2\delta_3+\delta_4)}(3\delta_3+\delta_4)\|x^{(0)}-x^*\|$$

$$\leqslant\frac{1}{2}\|x^{(0)}-x^*\|$$

现用归纳法证明

$$\|x^{(k)}-x^*\|\leqslant\frac{1}{2}\|x^{(k-1)}-x^*\|\quad k=1,\ 2,\ \cdots$$

$k=1$ 已证其成立。设 k 从 1 到 p 成立。易知$x^{(0)}$、…、$x^{(p)}$和 $x^{(p)}+h_pe_i(h_p\leqslant\gamma\|x^{(p)}-x^{(p-1)}\|)$诸点均在$S_3$中，从而$A_p$有意义，且当 $p<\left[\frac{n}{m}\right]$时

$$A_p=\left(\sum_{i=1}^{p}\sum_{j\in M(i)}B_ie_je_i^T\right)+\left(\sum_{j\in N-\bigcup_{s=1}^{k}M(s)}B_0e_je_j^T\right)$$

当 $p\geqslant\left[\frac{n}{m}\right]$时

$$A_p=\left(\sum_{i=p-\left[\frac{n}{m}\right]+2}^{p}\sum_{j\in M(i)}B_ie_je_i^T\right)$$

$$+\left(\sum_{j\in\left(M\left(p-\left[\frac{n}{m}\right]+1\right)-M(p)\right)}B_{p-\left[\frac{n}{m}\right]+1}e_je_i^T\right)$$

因而，与估计$\|A_0-F'(x^*)\|$类似可得

$$\|A_p-F'(x^*)\|\leqslant\frac{L}{2}(h_{J_p}+2\|x_{J_p}-x^*\|)$$

其中

$$0 \leqslant j_p \leqslant p \qquad \text{当} \quad p < \left\lceil \frac{n}{m} \right\rceil$$

$$p - \left\lceil \frac{n}{m} \right\rceil + 1 \leqslant j_p \leqslant p \qquad \text{当} \quad p \geqslant \left\lceil \frac{n}{m} \right\rceil$$

注意到$h_{j_p} \leqslant \gamma \|x_{j_p} - x_{j_p-1}\|$以及归纳假设，当$j_p = 0$时，

$$\|A_p - F'(x^*)\| \leqslant \frac{L}{2}(2\delta_3 + \delta_4) \tag{9-13}$$

当$j_p \geqslant 1$时，

$$\|A_p - F'(x^*)\| \leqslant \frac{L}{2}\left(\frac{3\gamma}{2} + 2\right)\left\| x_{p-\left\lceil \frac{n}{m} \right\rceil} - x^* \right\| \tag{9-14}$$

这里约定$p < \left\lceil \frac{n}{m} \right\rceil$时，$x_{p-\left\lceil \frac{n}{m} \right\rceil} = x^{(0)}$。由扰动定理，$A_p$非奇异，且

$$\|A_p^{-1}\| \leqslant \frac{\beta}{1 - \beta \|A_p - F'(x^*)\|}$$

类似$\|x^{(1)} - x^*\|$的估计可得

$$\|x^{(p+1)} - x^*\| \leqslant \frac{\beta}{1 - \beta \|A_p - F'(x^*)\|}\left(\|A_p - F'(x^*)\| + \frac{L}{2}\|x_p - x^*\|\right) \cdot \|x_p - x^*\| \tag{9-15}$$

因$\|x_p - x^*\| \leqslant \frac{1}{2}\delta_3$，$\| x_{p-\left\lceil \frac{n}{m} \right\rceil} - x^* \| \leqslant \delta_3$，无论将式(9-13)或(9-14)代入式(9-15)，总有

$$\|x_{p+1} - x^*\| \leqslant \frac{1}{2}\|x_p - x^*\|$$

归纳法完成。所证明的不等式保证了$\{x^{(k)}\}$的收敛性。

当$p \geqslant \left\lceil \frac{n}{m} \right\rceil$时，$j_p \geqslant 1$，所以式(9-14)成立，将其代入式(9-15)得

$$\|x^{(p+1)}-x^*\|\leqslant(c_1\|x_{p-\left\lceil\frac{n}{m}\right\rceil}-x^*\|+c_2\|x_p-x^*\|)\|x_p-x^*\|$$

其中

$$c_1=\frac{\beta L/2}{1-\frac{\beta L}{2}\left(\frac{3\gamma}{2}+1\right)\delta_s}\left(\frac{3\gamma}{2}+1\right)$$

$$c_2=\frac{c_1}{\frac{2\gamma}{2}+1}$$

根据文献〔13〕部分修正离散牛顿型方法当步长按式（9-9）选取时，根一收敛阶至少是 $\tau_{\left\lceil\frac{n}{m}\right\rceil}$。

证毕。

下面转向部分修正方法的并行计算。

算法1（部分修正牛顿型方法的并行计算）

（1）给定$x^{(1)}\in D$，$m\in N$，$\varepsilon>0$，$\gamma>0$，

$0\to k$；

（2）并行计算$F'(x^{(0)})e_j$、$F(x^{(0)})$、$j\in N$；

（3）若$\|F(x^{(0)})\|\leqslant\varepsilon$，结束；

（4）并行求解线性方程组$A_kS_k=F(x^{(k)})$；

（5）并行计算$x^{(k+1)}=S_k+x^{(k)}$；

（6）并行计算$F'(x^{(k+1)})e_j$、$F(x^{(k+1)})$ $j\in M(k+1)$；

（7）若$\|F(x^{(k+1)})\|\leqslant\varepsilon$，结束。否则$k+1\to k$，转（4）。

第（4）、（5）两步通过解线性方程组计算$x^{(k+1)}$。亦可通过计算A_k^{-1}求$x^{(k+1)}$。用E_k表示单位矩阵的$n\times m$子块，它由单位矩阵的第j列（$j\in M(k)$）构成。

$$C_k=F'(x_k)E_k,\quad D_k=A_{k-1}E_k$$

则式（9-11）中A_k可表示为

$$A_k=A_{k-1}+(C_k-D_k)E_k^T$$

根据西蒙-莫里森-伍德伯里（Sherman-Morison-Woodbury）公式

$$A_k^{-1}=A_{k-1}^{-1}+(E_k-A_{k-1}^{-1}C_k)(E_k^TA_{k-1}^{-1}C_k)^{-1}E_k^TA_{k-1}^{-1} \qquad (9-16)$$

其中$E_k^TA_{k-1}^{-1}$为A_{k-1}^{-1}的$m\times n$阶子块，它由A_k^{-1}的m行构成。利用式(9-16)求A_k^{-1}逆，只需计算一个$m\times m$阶矩阵的逆，它比起直接求$n\times n$阶矩阵的逆工作量较小。公式(9-16)的缺点是稳定性差，用时要小心。

算法2（部分修正离散牛顿型方法的并行计算）

（1）给定$x^{(0)}\in D$，$m\in N$，$e>0$，$\gamma>0$，$0\to k$，$h_0\to h$；

（2）并行计算$F(x^{(0)}+he_j)$、$F(x^{(0)})$　$j\in N$；

（3）若$\|F(x^{(0)})\|\leqslant e$，结束；

（4）并行计算$[F(x^{(0)}+h_0e_{j_0})-F(x^{(0)})]/h$　$j\in N$；

（5）并行求解线性方程组$A_kS_k=F(x^{(k)})$；

（6）并行计算$x^{(k+1)}=S_k+x^{(k)}$；

（7）若$h>\gamma\|x^{(k+1)}-x^{(k)}\|$，则$\gamma\|x^{(k+1)}-x^{(k)}\|\to h$；

（8）并行计算$F(x^{(k+1)}+he_j)$、$F(x^{(k+1)})$ $j\in M(k+1)$；

（9）若$\|F(x^{(k+1)})\|\leqslant e$，结束；

（10）并行计算$[F(x^{(k+1)}+he_j)-F(x^{(k+1)})]/h$　$j\in M(k+1)$；

（11）$k+1\to k$，转（5）。

从第（7）步可知，算法2是按式(9-9)选取步长h_k，式(9-8)的步长不适合并行计算，因若按式(9-8)，$F(x^{(k+1)})$要求在诸$F(x^{(k+1)}+h_ke_j)$之前计算，不能实现第（8）步的并行化。

与算法1一样，算法2中的第（5）、（6）步可利用西蒙-莫里森-伍德伯里公式求A_k^{-1}实现。

9.2.2 部分修正方法并行计算的复杂性

如果用某个迭代过程产生R^n中收敛于x^*的序列$\{x^{(k)}\}$，从时间上来说，迭代过程使误差$\|x^{(k)}-x^*\|$逐渐减小的快慢程度不仅依赖于$\{x^{(k)}\}$的收敛速度，而且依赖于产生$x^{(k)}$所需的时间。效率就是对迭代过程这种特性的度量。从定理2可知，由部分修正方

法产生的$\{x^{(k)}\}$具有超线性收敛性，对于这种超线性收敛的序列$\{x^{(k)}\}$，其效率一般定义为

$$E=\log\rho/T$$

其中ρ是根一收敛阶，T是在第k步迭代产生$x^{(k)}$的时间T_k的渐近稳定值，即$\lim\limits_{k\to+\infty} T_k=T$。对于两个不同的过程，若要求它们都将$\|x^{(k)}-x^*\|$减小至一很小的正数，那么具有效率$E_1$的过程所需时间是具有效率$E_2$的过程所需时间的$E_2/E_1$倍。效率愈高的过程，所费时间愈少。

根据定理9-2，部分修正方法算法1、算法2的根收敛阶ρ_1、ρ_2分别至少是$\tau^{\lceil\frac{n}{m}\rceil-1}$和$\tau^{\lceil\frac{n}{m}\rceil}$。每步迭代一般来说计算向量函数值所需的时间与其它运算比较是主要的，因此，可作如下简化性的假定：

（1）每步迭代所需时间只计向量函数值计算时间，其它时间忽略不计；

（2）一个向量函数值求值由一台处理机执行；

（3）任何处理机计算一个向量函数值的时间相等，且为一个时间单位。

若有p台处理机可供使用，根据上述假定和定理9-2，算法1、算法2的效率分别是

$$E_1(n,\ p,\ m)=\frac{\log\tau^{\lceil\frac{n}{m}\rceil-1}}{\lceil(m+1)/p\rceil}$$

$$E_2(n,\ p,\ m)=\frac{\log\tau^{\lceil\frac{n}{m}\rceil}}{\lceil(m+1)/p\rceil}$$

选择$m_p^{(1)}$、$m_p^{(2)}$，分别使

$$E_1(n,\ p,\ m_p^{(1)})=\max_{m\in N} E_1(n,\ p,\ m)$$

$$E_2(n,\ p,\ m_p^{(2)})=\max_{m\in N} E_2(n,\ p,\ m)$$

相应参数 $m_p^{(1)}$，$m_p^{(2)}$的算法 1 、算法 2 分别为部分修正牛顿型和离散牛顿型最优并行算法。记为算法(n， p， $m_p^{(1)}$)、算法(n， p， $m_p^{(2)}$)。当 $p=1$ 时，部分修正牛顿型最优串行算法为算法(n， 1， $m_1^{(1)}$)，其效率为 $E(n,\ 1,\ m_1^{(1)})$。至于部分修正离散牛顿型最优串行算法，我们按式 (9-8) 选取步长，使其效率

$$E(n,\ m)=\frac{\log^{\tau}\left\lceil\frac{n}{m}\right\rceil-1}{m+1}$$

达到最大 $E(n,\ m^*)=\max\limits_{m\in N}E(n,\ m)$。

当 n， p 给定，我们定义算法 (n， p， $m_p^{(1)}$) 的加速率为

$$S_1(n,\ p)=\frac{E_1(n,\ p,\ m_p^{(1)})}{E_1(n,\ 1,\ m_1^{(1)})}$$

算法 (n， p， $m_p^{(2)}$) 的加速率为

$$S_2(n,\ p)=\frac{E_2(n,\ p,\ m_p^{(2)})}{E(n,\ m^*)}$$

把误差 $\|x^{(k)}-x^*\|$ 减至同样一个很小的正数时，算法 (n， p， $m_p^{(1)}$) 所需的时间是最优串行算法的 $1/S_1(n,\ p)$。算法 (n， p， $m_p^{(2)}$) 所需的时间是最优串行算法的 $1/S_2(n,\ p)$。

算法 (n， p， $m_p^{(i)}$)($i=1$， 2) 的并行效率定义为

$$e_i=\frac{S_i(n,\ p)}{p}$$

它反映了所论并行算法在获得并行加速时对机器的利用率。应当区分效率和并行效率这两个不同的概念。

当 $n=10$，100，p 取不同值时，表 9-1 列出了算法(n，p，$m_p^{(1)}$)、算法 (n， p， $m_p^{(2)}$) 的最优修正列数 $m_p^{(1)}$、$m_p^{(2)}$、加速率和并行效率。随着处理机台数的增加，加速愈来愈大，但并行效率亦愈来愈低。

表 9-1

维数 n	p	$m_p^{(1)}$	$m_p^{(2)}$	$S_1(n,p)$	$S_2(n,p)$	$e_1(n,p)$	$e_2(n,p)$
10	2	1	1	1.92	1.70	0.96	0.85
	6	5	5	5.14	4.08	0.86	0.69
	10	5	5	5.14	4.08	0.51	0.41
	11	10	10	7.40	5.14	0.67	0.47
100	2	3	3	1.99	1.95	1.00	0.98
	6	5	5	5.84	5.65	0.97	0.94
	10	9	9	8.31	7.87	0.83	0.79
	11	10	10	9.39	8.81	0.85	0.80
	50	34	34	19.91	16.79	0.40	0.34
	51	50	50	25.06	19.91	0.49	0.39
	100	50	50	25.06	19.91	0.25	0.20
	101	100	100	36.10	25.06	0.36	0.25

9.3 并行牛顿——SOR 迭代方法

考虑下面的非线性问题

$$F(x)=0 \tag{9-17}$$

$$F:\ |R^N \to |R^N$$

$F'(x)=(F_{i,j}(x))$为M——矩阵。

奥特加（Ortega）和莱因博尔特（Rheinboldt）详细讨论了式（9-17）的串行算法，西蒙（Shermon，1978）通过将雅可比矩阵$F'(x)$分裂后构造了一种牛顿迭代方法，证明了收敛性，而且得到了收敛速度的估计。本节我们从线代数问题的多分裂方法着手，给出了式（9-17）的并行牛顿——SOR迭代法，证明了相应的收敛性。

9.3.1 线代数问题的多分裂并行算法

定义9-1 $(B_k,\ C_k,\ D_k)$ 称为 A 的多分裂，当且仅当

（1）$A=B_k-C_k$，B_k^{-1}存在，$k=1,\ 2,\ \cdots,\ K$，

（2）$\sum_{k=1}^{K} D_k=I$，D_k为非负对角矩阵。

定义9-2 设 A，A_k，E_k 及 D_k 均为方阵，$(A_k,\ E_k,\ D_k)$ 称为 A 的收敛性分裂，当且仅当

（1）$A=\sum_{k=1}^{K} A_k$，

（2）$E_k,\ D_k\geqslant 0$ 为对角矩阵；

（3）$(B_k,\ C_k,\ D_k)$ 为 A 的多分裂，这里

$$B_k=A_k+E_k,\ C_k=E_k-\sum_{j\neq k} A_j;$$

（4）$P(H)<1$，这里$H=\sum_{k=1}^{K} D_kB_k^{-1}C_k$。

特别，当$K=1$，M—矩阵的弱正则分裂为收敛性分裂。

假设线代数方程组

$$AX=f\in\mathbb{R}^N \tag{9-18}$$

的解唯一存在。$(B_k,\ C_k,\ D_k)$ 为 A 的多分裂，由多分裂可构造式（9-18）的一种并行迭代算法

$$\begin{aligned}x^{n+1}&=\left(\sum_{k=1}^{K} D_kB_k^{-1}C_k\right)x^n+\left(\sum_{k=1}^{K} D_kB_k^{-1}\right)f\\&=Hx^n+Gf\end{aligned} \tag{9-19}$$

这里$H=\sum_{k=1}^{K} D_kB_k^{-1}C_k$，$G=\sum_{k=1}^{K} D_kB_k^{-1}$。

$$\text{因}H-I=\sum_{k=1}^{K}D_kB_k^{-1}C_k-I=\sum_{k=1}^{K}D_kB_k^{-1}(C_k-B_k)$$

$$=-\sum_{k=1}^{K}D_kB_k^{-1}A=-GA$$

故式（9-19）等价于

$$x^{n+1}=x^n-G(Ax^n-f) \tag{9-20}$$

另一方面，我们可用 SOR 方法来加速收敛，

$$x^{n+1}=(1-\omega)x^n+\omega(Hx^n+Gf)$$

或

$$x^{n+1}=x^n-\omega G(Ax^n-f) \tag{9-21}$$

其中ω为松弛参数。

利锐（Leary）和怀特（White）对三种类型的矩阵（M—矩阵、对称正定阵、规范阵）给出了保证$P(H)<1$的条件，对M—矩阵的情形有下述结果：

若A为M—矩阵且

$A=B_k-C_k,\quad k=1,\ \cdots,\ K$

为A的正则分裂，则

$P(H)<1$。

借助于上述结论，不难得到线代数方程多分裂并行迭代法的收敛性。

9.3.2 非线性问题的并行牛顿——SOR 方法

设

$$F(x)=\sum_{k=1}^{K}F^k(x)$$

$F'(x)=B_k(x)-C_k(x)\quad k=1,\ \cdots,\ K$ 为一弱正则分裂。

$B_k(x)$ 可定义为

$$B_k(x)=(F^k(x)+E_kx)'$$

令 $0\leqslant m\leqslant M(m)-1$，例如可取$M(m)=1$，或$m+1$或$2^m$。

若记

$$B_{k,n}=B_k(x^n)$$

$$C_{k,n}=C_k(x^n)$$

$$f_n=F(x^n)-(B_{k,n}-C_{k,n})x^n=F(x^n)-F'(x^n)x^n$$

$$F_n(x)=(B_{k,n}-C_{k,n})x+f_n=F(x^n)+F'(x^n)(x-x^n)$$

则（9-17）的并行牛顿——SOR 算法如下：

$$x^{n-1,0}=x^n \tag{9-22}$$

$$x^{n+1,m+1}=x^{n+1,m}-\omega\sum_{k=1}^{K}D_kB_{k,n}^{-1}F_n(x^{n+1,m}) \tag{9-23}$$

$$x^{n+1}=x^{n+1,M(n)} \qquad 0\leqslant m\leqslant M(m)-1 \tag{9-24}$$

$$1\leqslant\omega<2。$$

算法中关于m的迭代称为内迭代，而对n的迭代称为外迭代。

特别，当$M(n)=1$，$F(x)=Ax-f$ 时，式（9-24）即为式（9-21）。

下面我们来研究并行牛顿——SOR 方法的收敛性及收敛速度。

设 F：$S\to\mathbb{R}^N$，$S\subset\mathbb{R}^N$ 为$\mathbb{R}^N$ 中一有界集，$x^*\in S$ 且 $F(x^*)=0$，假定

（1）F在 $S_0=\{x:\|x-x^*\|<r_0\}$ 上可微；

（2）$F'(x)$ 在 $x=x^*$非奇异；

（3）存在$L>0$ 使对 $x\in S_0$ 成立

$$\|F'(x)-F'(x^*)\|\leqslant L\|x-x^*\|$$

（4）$F'(x)=\Sigma A_k(x)=B_k(x)-C_k(x)$

$$B_k(x)=A_k(x)+E_k \quad C_k(x)=-\sum_{j\neq k}A_j(x)+E_k$$

（5）对 $k=1$，…，K，$B_k(x)$ 在点 $x=x^*$非奇异；

（6）$(A_k(x^*),\ E_k,\ D_k)$ 为 $F'(x^*)$ 的收敛性分裂；

（7）对 $k=1$，…K存在 $L_k>0$ 及 $r_0>0$ 使对

$$x\in S_0=\{x:\|x-x^*\|<r_0\}$$

成立

$$\|B_k(x)-B_k(x^*)\|\leqslant L_k\|x-x^*\|$$

和西蒙证明串行牛顿迭代方法的收敛性的方法类似，可证明下述收敛定理。

定理9-3 若假设（1）~（7）成立，$F(x^*)=0$ 设$M(n)$为正整数，定义

$$m^*=\max\left\{\{1,\ M(0)\}\cup\left\{M(n)-\sum_{l=0}^{n-1}M(l):n\right.\right.$$

$$\left.\left.=1,\ 2,\ \cdots\right\}\right\},$$

若$m^*<\infty$，则存在 $r>0$ 及 $c<1$ 使对$x^0\in S_r=\{\|x-x^*\|<r\}$，$\omega=1$ 及由式（9-24）给出的 x^{n+1} 有

$$\|x^{n+1}-x^*\|\leqslant C^{M(n)}\|x^n-x^*\|。$$

证明方法和串行的情形类似，这里仅将并行格式化为和串行相似的形式。

令：

$$H_n=\sum_{k=1}^{K}D_kB_{k,n}^{-1}C_{k,n},\quad H_{k,n}=B_{k,n}^{-1}C_{k,n},$$

$$G_n=\sum_{k=1}^{K}D_kB_{k,n}^{-1}$$

由式（9-23）若 $0\leqslant m\leqslant M(n)$，则

$$\begin{aligned}x^{n+1,m}&=x^{n+1,m-1}-G_n((B_{k,n}-C_{k,n})x^{n+1,m-1}+f_n)\\&=x^{n+1,m-1}-\sum_{k=1}^{K}D_k(x^{n+1,m-1}-H_{k,n}x^{n+1,m-1}+B_k^{-1}f_n)\\&=H_nx^{n+1,m-1}-G_nf_n\\&=H_n(H_nx^{n+1,m-2}-G_nf_n)-G_nf_n\\&\ \ \vdots\end{aligned}$$

$$=H_n^m x^n-\sum_{l=0}^{m-1} H_n^l G_n f_n$$

令 $m=M(n)$，则

$$x^{n+1,M(n)}=H_n^{M(n)}x^n-\sum_{l=0}^{M(n)-1} H_n^l G_n f_n$$

$$=x^n-\sum_{l=0}^{M(n)-1} H_n^l[(I-H_n)x^n+G_n f_n]$$

$$=x^n-\sum_{l=0}^{M(n)-1} H_n^l\left[\sum_{k=1}^{K} D_k(I-B_{k,n}^{-1}C_{k,n})x^n\right.$$

$$\left.+\sum_{k=1}^{K} D_k B_{k,n}^{-1}(F(x^n)-(B_{k,n}-C_{k,n})x^n)\right]$$

$$=x^n-\sum_{l=0}^{M(n)-1} H_n^l\left[\sum_{k=1}^{K} D_k B_{k,n}^{-1}F(x^n\right.$$

$$=x^n-\sum_{l=0}^{M(n)-1} H_n^l G_n F(x^n)$$

余下的证明和串行的情形类似，详细证明留给读者。

例 1：半线性偏微分方程

$$-(k_1u_x)_x-(k_1u_y)_y=-ge^u \qquad (x,\ y)\in\Omega \quad (9\text{-}25)$$

$$u=x^2+y^2 \qquad (x,\ y)\in\partial\Omega$$

选取 $k_1=1+x^2+y^2$，$k_2=1+e^x+e^y$

$$g=2(2+3x^2+y^2+e^x+(1+y)e^y)a^{-x2-y2}$$

使 $u=x^2+y^2$ 为式（9-25）的唯一解；$\Omega=(0,1)\times(0,1)$，怀特用差分方法离散式（9-25），在 Denelor HE_p 多处理机上用牛顿—SOR 方法计算式（9-25），收敛准则定义为网格点的相对误差小于 10^{-4}，得到下述结果（见表 9-2）。

表 9-2

K	内迭代次数	PAI	SU	PAI-SU
1	57	1.0	0.999	0.001
2	58	1.769	1.617	0.152
4	63	2.931	2.337	0.594
8	69	4.461	2.915	1.546
16	70	7.329	2.919	4.410

其中 PAI 为由下式定义的并行指标

$$\mathrm{PAI}\equiv(I_1\times U_1)/(I_k\times U_k)$$

I_k= k 等块分裂算法收敛所需的迭代次数

U_k=每等块所含未知数个数

SU 为加速比

$$\mathrm{SU}\equiv E_0/E_k$$

E_0：为串行算法执行的时间，

E_k：k 块并行算法执行时间。

9.4 一类特殊非线性方程组的并行算法

设 $x\in|\mathrm{R}^N$，B 为 $N\times N$ 非负矩阵，

$$\gamma(x)=(\gamma_i(x_i))\in|\mathrm{R}^N \qquad \gamma_i: R\to|R$$

A 为 $N\times N$ M—矩阵

$$\beta(x)=(\beta_i(x_i))\in|\mathrm{R}^N \quad \beta_i: R\to R$$

$$\eta\in|\mathrm{R}^N$$

考虑下述非线性方程组的求解

$$B\gamma(x)+A\beta(x)=\eta \tag{9-26}$$

研究式（9-26）最初的兴趣在于研究抛物型方程的不定边界问题，或斯蒂分（Stefan）问题。然而式（9-26）有其丰富的实际背景和广泛的概括性。

（1）当 $B=0$，$\beta_i(x_i)=x_i$ 时，式（9-26）退化为线代数

问题。

（2）当$B=I$，$\gamma_i(x_i)$连续不减且$\beta_i(x_i)=x_i$时，式（9-26）为半线性问题，来源于半线性椭圆型偏微分方程的数值求解。

（3）当$B=1$，$\gamma_i(x_i)=x_i$且$\beta_i(x_i)$有不连续导数，式（9-26）来源于斯蒂分问题，本节末将给出一个应用实例。

（4）当$B\neq 1$，$\gamma_i(x_i)=x_i$且$\beta_i(x_i)$有不连续导数，式（9-26）来源于斯蒂分问题的有限元离散，B对应于有限元中的“刚度”或“质量”矩阵。

9.4.1 特殊方程的串行算法

在研究特殊方程的并行算法之前，让我们先研究特殊方程的串行算法。

设$A=D-L-U$，这里D为对角阵，L为严格下三角阵，U为严格上三角阵。

设$A=(a_{ij})_{N\times N}$为M—矩阵，S_k为结点块集，$k=1,\cdots,K$。定义严格下三角子矩阵

$$L_k=\begin{cases} -a_{ij} & j<i\text{ 且 }i,\ j\in S_k \\ 0 & \text{其他} \end{cases}$$

设$A=(D-L_k)-(U+L-L_k)=B_k-C_k$

显然，上述分裂为A的正则分裂。特别，$K=1$，$L_K=L$，则为高斯-塞德尔分裂；$K=1$，$L_K=0$，则为雅可比分裂。

设D_k为对角矩阵，d_i^k为D_k的第i个对角元，且$d_i^k=0$，$i\notin S_k$，$\Sigma_k d_i^k=1$。

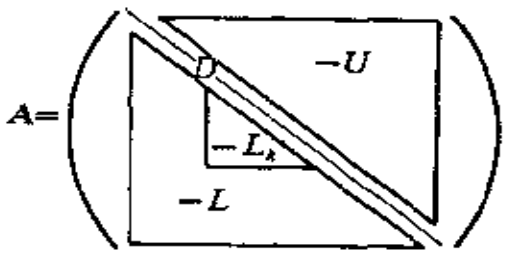


图 9-1

设b_{ij}为实数，用下列记号分别代表几种不同的和式：

$\Sigma^0 b_{ij}$——b_{ij}对$j\neq i$且$j\notin S_k$求和，

$\Sigma^- b_{ij}$——b_{ij}对$j<i$且$j\notin S_k$求和，

$\Sigma^+ b_{ij}$——b_{ij} 对 $j > i$ 且 $j \in S_k$ 求和，

Σd^k——d^k 对 $k = 1$ 到 K 求和。

借助于这些和式，9.3 节中关于线代数方程组的并行算法可等价地写为

$$x_i^{k,n+1} = [f_i - \Sigma^0 a_{ij} x_j^n - \Sigma^- a_{ij} x_j^{k,n+1} - \Sigma^+ a_{ij} x_j^n] / a_{ii}$$

$$x_j^{n+1} = \Sigma_k d_j^k x_j^{k,n+1}$$

设：$B = (b_{ij})$，$A = (a_{ij})$，$1 \leqslant i$，$j \leqslant N$；

b_{ii}，$a_{ii} > 0$

β_i，γ_i：$\mathbb{R} \to \mathbb{R}$ 为连续不减映射；

γ_i 严格增加；

定义

$$R_i(z) = b_{ii}\gamma_i(z) + a_{ii}\beta_i(z)$$

则式（9-26）的分量形式为

$$R_i(x_i) = \eta_i - \sum_{j \leqslant i} (a_{ij}\beta_j(x_j) + b_{ij}\gamma_j(x_j)) - \sum_{j > i} (a_{ij}\beta_j(x_j) + b_{ij}\gamma_j(x_j)) \tag{9-27}$$

设 $R_i^{-1}[W] = Z$ 为 $R_i(Z) = W$ 的逆。

若 R_i^{-1} 存在，则求解式（9-26）的串行算法为

$$x_i^{k+1} = R_i^{-1}\left[\eta_i - \sum_{j \leqslant i} (a_{ij}\beta_j(x_j^{k+1}) + b_{ij}\gamma_j(x_j^{k+1})) - \sum_{j > i} (a_{ij}\beta_j(x_j^k) + b_{ij}\gamma_j(x_j^k)) \right] \tag{9-28}$$

定理9-4 若下列条件成立：

（1）β_i：$\mathbb{R} \to \mathbb{R}$ 连续不减；

（2）$B \geqslant 0$ 且 $b_{ii} > 0$；

（3）A 为 M—矩阵；

（4）$b_{ij}/b_{ii} \leqslant |a_{ij}|/a_{ii}$；

（5）γ_i：$\mathbb{R} \to \mathbb{R}$ 连续，且对任意 x，$y \in \mathbb{R}$ $\gamma_i(x) -$

$\gamma_i(y)\geqslant\varepsilon_i(x-y)>0$，$\varepsilon_i>0$；

则特殊方程有唯一解，且对任意的 x_i^0，算法（9-28）收敛于式（9-26）的唯一解。

证明详见怀特。

9.4.2 特殊方程的并行算法

设 $S_k\subset\{1,\cdots,N\}$ $\bigcup_k S_k=\{1,\cdots,N\}$

$$d_i^k\geqslant 0;\quad d_i^k=0,\quad i\notin S_k;\ \Sigma_k d_i^k=1$$

若 $i\in S_k$，则求解特殊方程的并行格式如下：

$$\begin{aligned}x_i^{k,n+1}=R_i^{-1}[&\eta_i-\Sigma^1(a_{ij}\beta_j(x_j^n)+b_{ij}\gamma_j(x_j^n))\\&-\Sigma^-(a_{ij}\beta_j(x_j^{k,n+1})+b_{ij}\gamma_j(x_j^{k,n+1}))\\&-\Sigma^+(a_{ij}\beta_j(x_j^n)+b_{ij}\gamma_j(x_j^n))]\end{aligned}$$

$$x_i^{n+1}=\Sigma_k d_i^k E_i^{k,n+1}\qquad(9\text{-}29)$$

设 $A=(D-L_k)-(L-L_k+U)$

$$B=(\underline{D}-\underline{L}_k)-(\underline{L}-\underline{L}_k+\underline{U})$$

因 $0\leqslant L_k\leqslant L$，故 $C_k=L-L_k+U\geqslant 0$

$0\leqslant\underline{L}_k\leqslant\underline{L}$，故 $\underline{C}_k=\underline{L}-\underline{L}_k+\underline{U}\geqslant 0$

若证

$[v]_i=v\in\mathbb{R}^N$ 的第 i 个分量，

$(U_i)=\mathbb{R}^N$ 中的向量，且 $v_i\in\mathbb{R}$，

则式（9-29）有如下矩阵形式

$$\begin{aligned}R_i(x_i^{k,n+1})=[&\eta_i+C_k\beta(x^n)+\underline{C}_k\gamma(x^n)\\&+L_k\beta(x^{k,n+1})+\underline{L}_k\gamma(x^{k,n+1})]_i\end{aligned}$$

$$x_i^{n+1}=\Sigma_k D_k E^{k,n+1}\qquad(9\text{-}30)$$

这里 D_k 为对角阵，其对角元为 d_i^k。

定理9-5 若定理 9-4 的条件（1）—（5）仍成立，则并行算法（9-30）对任意初值 x_i^0 收敛到特殊方程组的唯一解。

为证明定理 9-5，需要几个引理。

引理9-1 设 $A=(D-L_k)-(L-L_k+U)$ 为一非异 M—矩阵，$0\leqslant L_k\leqslant L$，$D_k\geqslant 0$ 为对角矩阵且 $\Sigma D_k=1$，则存在单

调范数 $\|\cdot\|$ 使对所有满足要求的 D_k 和 L_k 成立

$$\|H\|\leqslant\|D^{-1}(L+U)\|<1$$

证明 因A为非异M—矩阵，故存在对角阵 D_1，其对角元为正，且 $D_1^{-1}AD_1$ 严格对角伏势。故对 $e=(1,\cdots 1)^T$ 有

$$D_1^{-1}(L+U)D_1e<D_1^{-1}DD_1e$$

定义范数 $$\|x\|=\|D_1^{-1}x\|_\infty$$

则

$$\|D^{-1}(L+U)\|=\sup_{\|x\|\neq 0}\frac{\|D_1^{-1}D^{-1}(L+U)X\|_\infty}{\|D_1^{-1}x\|_\infty}$$

$$\leqslant\|D_1^{-1}D^{-1}(L+U)D_1\|_\infty<1$$

又

$$D_1^{-1}HD_1=D_1^{-1}(\Sigma_kD_kB_k^{-1}C_k)D_1$$

$$=\Sigma_kD_1^{-1}D_kB_k^{-1}C_kD_1$$

$$=\Sigma_kD_k(D_1^{-1}B_k^{-1}C_kD_1)$$

$$D_1^{-1}D^{-1}(L+U)D_1=\Sigma_kD_kD_1^{-1}D^{-1}(L+U)D_1$$

$$0\leqslant L_k\leqslant L\quad B_k^{-1}\geqslant 0\quad C_k\geqslant 0$$

故只需证

$$D_1^{-1}B_k^{-1}C_kD_1e\leqslant D_1^{-1}D^{-1}(L+U)D_1e$$

记

$$\tilde{L}_k=D_1D^{-1}L_kD_1\quad\tilde{L}=D_1^{-1}D^{-1}LD_1,$$

$$\tilde{U}=D_1^{-1}D^{-1}UD_1$$

直接验证可得

$$D_1^{-1}B_k^{-1}C_kD_1=(I-\tilde{L}_k)^{-1}(\tilde{L}-\tilde{L}_k+\tilde{U})$$

因 $$\tilde{L}_k\geqslant 0\quad(I-\tilde{L}_k)^{-1}\geqslant 0$$

若

$$(\tilde{L}-\tilde{L}_k+\tilde{U})e\leqslant(I-\tilde{L}_k)(\tilde{L}+\tilde{U})e$$

$$\leqslant(\tilde{L}+\tilde{U})e-\tilde{L}_k(L+U)e$$

(9-31)

则有

$$(I-\tilde{L}_k)^{-1}(\tilde{L}-\tilde{L}_k+\tilde{U})e\leqslant(\tilde{L}+\tilde{U})e$$

式（9-31）等价于

$$(\tilde{L}-\tilde{L}_k)e\leqslant\tilde{L}e-\tilde{L}_k(\tilde{L}+\tilde{U})e$$

即

$$\tilde{L}_k(\tilde{L}+\tilde{U})e\leqslant\tilde{L}_k e$$

因 $(\tilde{L}+\tilde{U})e<e$，故式（9-31）成立

引理 9-2 令 $a_1\cdots a_k$，$C\in|R$，$\boldsymbol{A}=(a_1,\cdots,a_k)$ 为一 K 维行向量；若 $\max a_k\geqslant C\geqslant\min a_k$，则存在列向量 $\boldsymbol{\delta}=(\delta_1,\cdots,\delta_k)^T$ 使

（1）$\boldsymbol{A}\cdot\boldsymbol{\delta}=C$

（2）$\Sigma_k\delta_k=1$

（3）$\delta_k\geqslant 0\quad k=1,\cdots,K$

证明 若 $a_1=a_2=\cdots=a_k$，则结论显然。

若 $a_{k_1}=\max a_k$，$a_{k_2}=\min a_k$，

$a_{k_2}\leqslant C\leqslant a_{k_1}$

取 $\boldsymbol{\delta}$ 使

$$\delta_k=\begin{cases}(C-a_{k_2})/(a_{k_2}-a_{k_1}) & k=k_1\\ (a_{k_1}-C)/(a_{k_1}-a_{k_2}) & k=k_2\\ 0 & k\neq k_1\ \text{或}\ k_2\end{cases}$$

则 $\boldsymbol{\delta}$ 满足要求。

定理 9-5 的证明

由定理 9-4 知特殊方程组的解 x 唯一存在，故有

$$R_i(x_i)=[\eta_i+C_k\beta(x)+\underline{C}_k\gamma(x)+L_k\beta(x)+\underline{L}_k\gamma(x)]_i \tag{9-32}$$

由条件（2），（3）得 b_{ii}，$a_{ii}>0$，故由（11），（4）知 R_i^{-1} 存在，故由 $R_i(z)$ 的定义有

$$\begin{aligned}\varepsilon_0 b_0|z-z^{\cdot}|&\leqslant b_{ii}|\gamma_i(z)-\gamma_i(z^{\cdot})|+a_{ii}|\beta_i(z)-\beta_i(z^{\cdot})|\\&=|R_i(z)-R_i(z^{\cdot})|\end{aligned}\tag{9-33}$$

这里 $\varepsilon_0=\min\varepsilon_i$，$b_0=\min b_{ii}$

式（9-30）和（9-32）相减得

$$
\begin{aligned}
R_i^- &= |R_i(x_i) - R_i(x_i^{k,n+1})| \\
&= |[C_k(\beta(x)-\beta(x^n)) + \underline{C}_k(\gamma(x)+\gamma(x^n)) \\
&\quad + L_k(\beta(x)-\beta(x^{k,n+1})) + \underline{L}_k(\gamma(x) \\
&\quad - \gamma(x^{k,n+1}))]_i| \\
&\leqslant [C_k|\beta(x)-\beta(x^n)|]_i + [|\underline{C}_k||\gamma(x) \\
&\quad - \gamma(x^n)|]_i \\
&\quad + [L_k|\beta(x)-\beta(x^{k,n+1})|]_i + [|\underline{L}_k||\gamma(x) \\
&\quad - \gamma(x^{k,n+1})|]_i \qquad (9\text{-}34)
\end{aligned}
$$

式（9-33）中分别取 $z=x_i$，$z^-=x_i^{k,n+1}$ 及 $z=x_i$，$z^-=x_i^n$ 得

$$
\begin{aligned}
D|\beta(x)-\beta(x^{k,n+1})| &= |R(x)-R(x^{k,n+1})| - \underline{D}|\gamma(x) \\
&\quad - \gamma(x^{k,n+1})| \qquad (9\text{-}35)
\end{aligned}
$$

$$
\begin{aligned}
D|\beta(x)-\beta(x^n)| &= |R(x)-R(x^n)| - \underline{D}|\gamma(x) \\
&\quad - \gamma(x^n)| \qquad (9\text{-}36)
\end{aligned}
$$

式（9-35），（9-36）分别乘 L_kD^{-1} 及 C_kD^{-1} 代入式（9-34）得

$$
\begin{aligned}
R_i^- \leqslant\ & [C_kD^{-1}|R(x)-R(x^n)|]_i - [C_kD^{-1}\underline{D}|\gamma(x)-\gamma(x^n)|]_i \\
& + [|\underline{C}_k||\gamma(x)-\gamma(x^n)|]_i + [L_kD^{-1}|R(x) \\
& - R(x^{k,n+1})|]_i - [L_kD^{-1}\underline{D}|\gamma(x) \\
& - \gamma(x^{k,n+1})|]_i + [|\underline{L}_k||\gamma(x)-\gamma(x^{k,n+1})|]_i \qquad (9\text{-}37)
\end{aligned}
$$

由条件（4）得

$$|\underline{C}_k| - C_kD^{-1}\underline{D} \leqslant 0,\quad |\underline{L}_k| - L_kD^{-1}\underline{D} \leqslant 0,$$

故由式（9-37）对 $i \in S_k$ 有

$$[(I - L_kD^{-1})(R_i^-)]_i \leqslant [C_kD^{-1}|R(x)-R(x^n)|]_i \qquad (9\text{-}38)$$

因 $0 \leqslant (I - L_kD^{-1})^{-1} = D(D-L_k)^{-1} = DB_k^{-1}$，由式（9-38）对 $i \in S_k$ 有

$$R_i^- \leqslant [DB_k^{-1}C_kD^{-1}|R(x)-R(x^n)|]_i \qquad (9\text{-}39)$$

令

$$a_k = \begin{cases} R_i(x_i^{k,n+1}) & \text{若 } i \in S_k; \\ 0 & \text{其他} \end{cases}$$

$$C=R_i(\Sigma_k d_i^{k,n+1}x_i^{k,n+1})=R_i(x_i^{n+1})$$

则由引理 9-2，对每一 i 存在 $\delta_k=\delta_i^k$ 使

$$\Sigma_k\delta_i^k R_i(x_i^{k,n+1})=R_i(\Sigma_k d_i^k x_i^{k,n+1})$$

因

$$\begin{aligned}|R_i(x_i)-R_i(x_i^{n+1})|&=|R_i(x_i)-\Sigma_k\delta_i^k R_i(x_i^{k,n+1})|\\&=\Sigma_k\delta_i^k|R_i(x_i)-R_i(x_i^{k,n+1})|\\&=\Sigma_k\delta_i^k R_i^-\\&\leqslant\Sigma_k\delta_i^k[DB_k^{-1}C_kD^{-1}|R(x)\\&\quad-R(x^n)|]_i\\&=[\Sigma_k D_{k,n+1}DB_k^{-1}C_kD^{-1}|R(x)\\&\quad-R(x^n)|]_i\end{aligned}\tag{9-40}$$

这里

$$D_{k,n+1}=diag(d_i^{k,n+1})$$

$$d_i^{k,n+1}=\begin{cases}0, & i\notin S_k\\ \delta_i^k, & i\in S_k\end{cases}$$

δ_i^k 与 $n+1$ 有关。

令 $H_{n+1}=\Sigma_k D_{k,n+1}B_k^{-1}C_k\geqslant 0$，因 $D_{k,n+1}$ 与 D 可交换，故由式（9-40）得

$$D^{-1}|R(x)-R(x^{n+1})|\leqslant H_{n+1}D^{-1}|R(x)-R(x^n)|\tag{9-41}$$

由引理 9-1，存在一单调范数使

$$\|H_{n+1}\|\leqslant\|D^{-1}(L+U)\|=\rho<1$$

故由式（9-41）得

$$\begin{aligned}\|D^{-1}|R(x)-R(x^{n+1})|\|&\leqslant\|H_{n+1}D^{-1}|R(x)-R(x^n)|\|\\&\leqslant\rho\|D^{-1}|R(x)-R(x^n)|\|\\&\vdots\\&\leqslant\rho^{n+1}\|D^{-1}|R(x)-R(x^0)|\|\end{aligned}\tag{9-42}$$

式（9-33）中令 $z=x_n$，$z^{\wedge}=x_i^{n+1}$，令 $C=\varepsilon_0 b_0/\max a_{ii}$，则

$$C|x-x^{n+1}|\leqslant\varepsilon_0 b_3 D^{-1}|x-x^{n+1}|\leqslant D^{-1}|R(x)-R(x^{n+1})|$$

上式取单调范数，结合式（9-42）得

$$C\|x-x^{n+1}\|\leqslant\rho^{n+1}\|D^{-1}\|R(x)-R(x^0)\|\|$$

因$\rho<1$，故由上式导出所需的收敛性。

例2（斯蒂分问题）

设u表温度，$H(u)$为热焓。

$$H(u)=\int_0^u(\rho c)(u^-)du^-$$

其中 $(\rho c)(u)$——热熔关于温度的函数，

$$G(u)=\int_0^u K(u^-)du^-$$

其中 $K(u)$——热传导关于温度的函数。

f＝内热源

则非线性热焓方程为

$$H_t(u)-\Delta G(u)=f \tag{9-43}$$

令$v=G(u)$，

$$H(u)=H(G^{-1}(v))=E,$$

$\beta(E)=H(G^{-1}(v))$ 的左逆，

则下述三角形区域上的热焓方程（9-43）的定解问题为

$$\begin{cases} E_t-\Delta\beta(E)=f \\ \dfrac{d\beta(E)}{dn}=0 \qquad x=y,\ 0\leqslant y<3.95;\ 0\leqslant x<3.95, \\ \qquad\qquad y=0 \\ \beta(E)=G(g(t)) \quad x=3.95,\ 0\leqslant y\leqslant 3.95 \end{cases} \tag{9-44}$$

n为边界外法向，$g(t)$为边界温度函数。见图9-2。

初始条件为

$$E(x,\ 0)=E^0(x)$$

式（9-44）时间离散后，在每一时间层生成一椭圆型方程

$$E-\Delta t\cdot\Delta\beta(E)=E^-+\Delta t\cdot f=F \tag{9-45}$$

E^-为前一时间层的热焓。

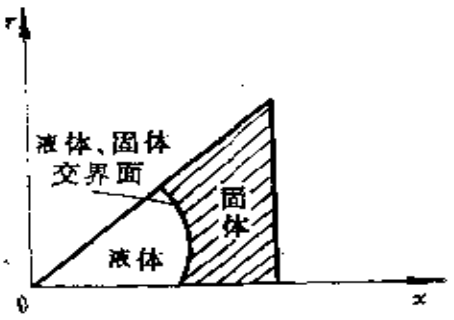


图 9-2

式（9-45）的弱形式为

$$\int E\cdot\phi d\Omega+\Delta t\int(\beta_x(E)\phi_x+\beta_y(E)\phi_y)d\Omega=\int F\phi d\Omega \tag{9-46}$$

对所有 $\phi\in H_1(\Omega)$，$\phi=0$，$x=3.95$，$0\leqslant y\leqslant 3.95$

取 $\phi=\phi_i$——结点 i 的线性形状函数，

$$E=\Sigma E_j\phi_j$$

$$\beta(E)\doteq\Sigma\beta_j(E_j)\phi_j$$

$$F=\Sigma F_j\phi_j$$

则式（9-46）的伽勒金（Galeikin）有限元格式为

$$BE+A\beta(E)=\eta$$

这里

$$B=\left(\int\phi_i\phi_j d\Omega\right)$$

$$A=\Delta t\int(\phi_{ix}\phi_{jx}+\phi_{iy}\phi_{jy})d\Omega \tag{9-47}$$

$$\eta=BF,\quad F=(F_j)$$

式（9-47）恰为特殊方程组形式。

怀特在下述初始条件下：

$$g(t)=273.0-0.050\,t/60$$

$$u(x,\ 0)=273.1,$$

$$E(x,\ 0)=E^0(x)=H(273.1)$$

且假定 $f\equiv 0$

对式（9-47）用并行算法式（9-30）计算了固体-液体交界面，所得结果与物理实验相吻合，见表 9-3。表 9-4 给出了在不同分块下并行计算的有关指标。

表 9-3

时 间	分界面在对角线 $x=y$ 的位置	时 间	分界面在对角线 $x=y$ 的位置
1.500	$11.0h$ $h=3.95/12$	9.000	$4.0h$
3.000	$9.5h$	10.500	$3.0h$
4.500	$8.0h$	12.000	$1.0h$
6.000	$7.0h$	12.300	$0.5h$
7.500	$5.5h$	12.600	$0.0h$

表 9-4

K	U_k	I_k	PAI
1	91	34.3	1.0
4	28	36.3	3.1
16	10	39.5	7.9

表中 K——分块数；

U_k——每块未知量个数；

I_k——收敛迭代次数；

$\mathrm{PAI}=\dfrac{I_1\times U_1}{I_k\times U_k}$：并行算法指标。

第十章　常微分方程初值问题的并行数值计算方法

工程和科学的各个领域中的许多问题都可以用常微分方程来描述，或可以化成常微分方程进行求解。根据常微分方程定解条件给出的不同形式，可得到常微分方程初值问题和边值问题。考虑一阶常微分方程组

$$\frac{dy}{dt}=f(t,\ y)$$

其中y，$f(t,\ y)$都是n维列向量。为确定这个微分方程组的解$y(t)$，需要给出n个定解条件，如果n个定解条件都给定在一个点t_*上（如$y(t_*)=y_*$），称得到的问题为常微分方程初值问题。通常取t_*为运动的初始时刻或终结时刻。如果n个定解条件给定在不同的点t_1，t_2，…，t_J上，则将问题称作常微分方程边值问题。当$J=2$时，称作两点边值问题；而当$J>2$时，称作多点边值问题。

本章研究数值求解常微分方程初值问题的并行算法。关于数值求解边值问题的并行算法将在下一章（第十一章）中讨论。

构造常微分方程初值问题的并行算法，可以从两个方面来进行。一方面是将传统的数值方法向量化，或者再增加方法本身处理的信息量，使之能放大数值积分的步长。另外一个方面是构造新的算法，使之能达到下述几个目的：（1）若干个右函数可并行计算；（2）在若干个子区间上可同时求解同一个常微分方程组；（3）可将整个常微分方程组分解成若干个子组，每个子组可独立求解。只要达到上述的一个目的，在并行处理机上就能提高方法的处理速度。我们着重讨论构造新的算法。

10.1 传统方法的向量化处理

对许多常微分方程组初值问题的数值方法来说，若不考虑右函数的计算，本身就是向量化的。由于许多计算机系统对于并行计算右函数尚有一定的困难，以及用户编制向量化的右函数也没有达到普遍化的程度，因此研究传统方法的向量化实现具有一定的意义。

一阶常微分方程组初值问题的向量形式为

$$y' = f(t, y) \qquad y(0) = y_0 \tag{10-1}$$

其中 $y = (y^1, y^2, \cdots, y^n)^T$，$f = (f^1, f^2, \cdots, f^n)^T$均是 n 维列向量，$f(t, y)$ 是 t 和 y 的向量值函数，不作特别说明时，假定它对其变元具有所需要的连续导数，t 为独立变量。

对步长 $h > 0$，考虑格点$t_i = ih$，$i = 0, 1, 2, \cdots$，计算的目的是要确定问题（10-1）的精确解 $y(t)$ 在格点 t_i 上的值 $y(t_i)$ 的近似值$y_i = (y_i^1, y_i^2, \cdots, y_i^n)^T$。下面我们列出三个计算$y_i$的传统积分公式，即四阶显式龙格-库塔（Runge-Kutta）公式，显式亚当斯-巴希福兹（Adams-Bashforth）公式和隐式亚当斯-莫尔顿（Moulton）公式。

已知y_i，计算y_{i+1}的四阶显式龙格-库塔公式为

$$\begin{aligned}
K_1 &= hf(t_i, y_i) \\
K_2 &= hf\left(t_i + \frac{h}{2}, y_i + \frac{K_1}{2}\right) \\
K_3 &= hf\left(t_i + \frac{h}{2}, y_i + \frac{K_2}{2}\right) \\
K_4 &= hf(t_i + h, y_i + K_3) \\
y_{i+1} &= y_i + \frac{1}{6}(K_1 + 2K_2 + 2K_3 + K_4)
\end{aligned} \tag{10-2}$$

其中若不管右函数 $f(t, y)$ 是如何计算的，算法本身的计算几乎都是向量运算。例如K_1，K_2，K_3，K_4都是标量 h 与向量 f 的乘法，而y_{i+1}是由y_i和K_i通过标量与向量的乘法和向量与向量

的加法得到的。

若已知y_i和f_i，f_{i-1}，…，f_{i-k+1}，其中

$$f_{i-j}=f(t_{i-j},\ y_{i-j})$$

则确定y_{i+1}的亚当斯-巴希福兹公式和亚当斯-莫尔顿公式分别为

$$y_{i+1}=y_i+h\sum_{j=1}^{K}\beta_j f_{i-j+1} \tag{10-3}$$

和

$$y_{i+1}=y_i+h\beta_0^* f(t_{i+1},\ y_{i+1})+h\sum_{j=1}^{K-1}\beta_j^* f_{i-j+1} \tag{10-4}$$

其中系数β_j和β_j^*分别由表10-1和表10-2给出。由式(10-3)、(10-4)可看出，若不管右函数是如何计算的，式(10-3)、(10-4)都可由标量与向量的乘法和向量与向量的加法来实现，公式（10-3）是显式的，而式（10-4）是隐式的，在实现时需求解非线性方程组。但除函数$f(t_{i+1},\ y_{i+1})$的计算外，主要是向量运算，在流水线向量机或阵列机上都可以并行计算，从而提高计算速度，其它传统的方法可进行类似的处理。

表10-1　亚当斯-巴希福兹公式的系数

K \ j		1	2	3	4	5	6
1	β_j	1					
2	$2\beta_j$	3	-1				
3	$12\beta_j$	23	-16	5			
4	$24\beta_j$	55	-59	37	-9		
5	$720\beta_j$	1901	-2774	2616	-1274	251	
6	$1440\beta_j$	4277	-7923	9982	-7298	2877	-475

吉尔（Gear）为了变阶和变步长的方便，采用诺特西克(Nordsieck)的表示形式，将式(10-3)、(10-4)进行改写，组合成一个预估-校正迭代格式。式（10-3）为预估公式，式（10-4)作为校正公式。令

表10-2　亚当斯-莫尔顿公式的系数

K \ j		0	1	2	3	4	5
1	β_j^*	1					
2	$2\beta_j^*$	1	1				
3	$12\beta_j^*$	5	8	−1			
4	$24\beta_j^*$	9	19	−5	1		
5	$720\beta_j^*$	251	646	−264	106	−19	
6	$1440\beta_j^*$	475	1427	−798	482	−173	27

$$z_i=\left(y_i,\ \frac{h}{1!}y_i',\ \cdots,\ \frac{h^K}{K!}y_i^{(K)}\right)^T$$

为计算中得到的在t_i处问题（10-1）的解$y(t)$及其各阶导数的近似值，于是由式（10-3）可得到z_{i+1}的预估值$z_{i+1(0)}$

$$z_{i+1(0)}=Az_i \tag{10-5}$$

其中A是巴斯卡尔（Pascal）上三角形矩阵

$$A=\begin{pmatrix} a_{11}I & a_{12}I\cdots & a_{1,\ K+1}I \\ & a_{22}I & a_{2,K+1}I \\ & 0 & \ddots \\ & & a_{K+1,K+1}I \end{pmatrix},\quad I=\begin{pmatrix} 1 & & \\ & 1 & 0 \\ 0 & & \ddots \\ & & 1 \end{pmatrix}$$

其中a_{ij}为

$$a_{ij}=\begin{cases} \binom{j-1}{i-1} & j\geqslant i \\ 0 & j<i \end{cases}$$

若记z_i的列向量分别为z^0，z^1，…，z^K，则$z_{i+1(0)}$的列向量可通过下面一系列的向量加法来实现

$$z^k + z^{k-1} \longrightarrow z^{k-1}$$
$$z^{k-1} + z^{k-2} \longrightarrow z^{k-2}$$
$$\cdots$$
$$z^1 + z^0 \longrightarrow z^0$$
$$z^k + z^{k-1} \longrightarrow z^{k-1}$$
$$\cdots$$
$$z^2 + z^1 \longrightarrow z^1$$
$$\cdots$$
$$z^k + z^{k-1} \longrightarrow z^{k-1}$$

以$z_{i+1(0)}$为初值，由公式（10-4）可得到校正迭代

$$z_{i+1(m+1)} = z_{i+1(m)} + lF(z_{i+1(m)}) \quad m = 0, 1, \cdots \qquad (10-6)$$

其中

$$l = \begin{pmatrix} l^0 I \\ l^1 I \\ \vdots \\ l^k I \end{pmatrix} \qquad F(z) = -z' + hf(t, z^0)$$

当按式（10-6）迭代到满足精度时，令$z_{i+1} = z_{i+1(m)}$。由这些公式可以看出，若不考虑右函数$f(t, y)$的计算，算法是已向量化的。对于应用向后微分公式的刚性方法可作类似的处理。表10-3中列出相应于表10-2的l^0，l^1，…，l^k的值。

表10-3　亚当斯—莫尔顿公式的系数l^0，l^1，…，l^k

K	l^0	l^1	l^2	l^3	l^4	l^5	l^6
1	1	1					
2	$\frac{1}{2}$	1	$\frac{1}{2}$				
3	$\frac{5}{12}$	1	$\frac{3}{4}$	$\frac{1}{6}$			
4	$\frac{3}{8}$	1	$\frac{11}{12}$	$\frac{1}{3}$	$\frac{1}{24}$		

（续）

K	l^0	l^1	l^2	l^3	l^4	l^5	l^6
5	$\frac{251}{720}$	1	$\frac{25}{24}$	$\frac{35}{72}$	$\frac{5}{48}$	$\frac{1}{120}$	
6	$\frac{95}{288}$	1	$\frac{137}{120}$	$\frac{5}{8}$	$\frac{17}{96}$	$\frac{1}{40}$	$\frac{1}{720}$

10.2 预估-校正并行计算格式

在10.1节中描述的数值积分方法都是串行方法，象亚当斯公式类型的线性多步公式或者龙格-库塔公式，每一次调用只能得到微分方程的解在一个格点上的值，每一个新的格点上的值依赖于前面一些格点上的值，可以看成是这些值的函数。在10.1节中，我们描述了这些传统算法的向量化处理，其中我们对右函数的计算没有考虑，但是对一般的常微分方程初值问题的数值求解，主要的计算量集中在计算式（10-1）的右函数 $f(t, y)$ 上。在10.1节中描述的传统方法只能一个点接着一个点计算右函数的值，所以在并行处理机上来实现是低效的。这一节描述的算法希望能同时对若干个点上的右函数值进行计算。

仍考虑10.1节中的问题(10-1)和用步长 h 的离散化，用 y_i^p 和 y_i^c 分别表示 y_i 的预估值和校正值，记 $f_i^c = f(t_i, y_i^c)$，$f_i^p = f(t_i, y_i^p)$，考虑串行的预估校正公式

$$y_{i+1}^p = y_i^c + \frac{h}{2}[3f_i^c - f_{i-1}^c]$$

$$y_{i+1}^c = y_i^c + \frac{h}{2}[f_{i+1}^p + f_i^c] \tag{10-7}$$

它的计算次序是这样的

$$\cdots \longrightarrow y_{i+1}^p \longrightarrow f_{i+1}^p \longrightarrow y_{i+1}^c \longrightarrow f_{i+1}^c \longrightarrow \cdots$$

在图10-1上这样表示：上面的线表示在格点上对 y_{i+1}^p, f_{i+1}^p 的计算进程，下面的线表示 y_{i+1}^c，f_{i+1}^c 的计算进程，虚线表示计算的前

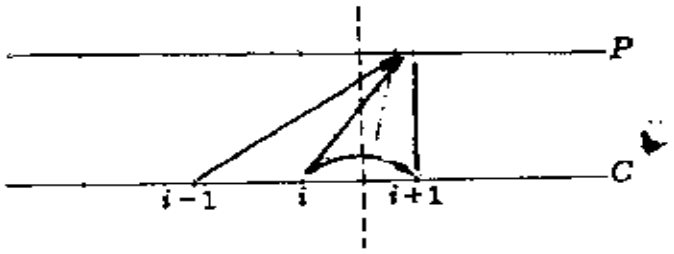


图10-1　进程执行示意图

锋面，用箭头表示对第 $i+1$ 个点上的计算的影响。从图上看出，计算前锋面前面的点上的计算依赖于这个面二边的信息，这是串行计算的一个特征。作为说明的例子，我们再考虑一个并行的预估-校正格式

$$y_{i+1}^{p}=y_{i-1}^{c}+2hf_{i}^{p} \tag{10-8}$$

$$y_{i}^{c}=y_{i-1}^{c}+\frac{h}{2}(f_{i}^{p}+f_{i-1}^{c})$$

它的计算序列由图10-2中的箭头表示。

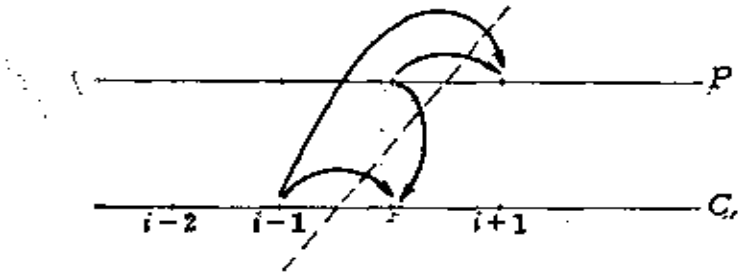


图10-2　计算顺序示意图

从图中看出，计算前锋面前面的点上的计算仅依赖于计算前锋面后面的点上的信息，这是并行计算的特征，整个计算的序列可以分成二个子序列

$$\longrightarrow y_{i+1}^{p}\longrightarrow f_{i+1}^{p}\longrightarrow$$

$$\longrightarrow y_{i}^{c}\longrightarrow f_{i}^{c}\longrightarrow$$

它们的计算可并行进行，特别f_{i+1}^{p}和f_{i}^{c}的值可同时计算。

下面描述 $2s$ 个右函数值可同时计算的一般的预估-校正公

式类，$s=1$，2，…。为简单起见，假定式(10-1)是单个方程，y是标量，下面的叙述和公式容易推广到y为向量的情形。

令Y_ν是s维向量

$$Y_\nu=(y_{\nu s-1},\ y_{\nu s-2},\ \cdots,\ y_{(\nu-1)s})^T \qquad \nu=1,\ 2,\ \cdots \tag{10-9}$$

其中y_i($i=0,\ 1,\ 2,\ \cdots$)是格点t_i处的量，令F_ν是s维向量

$$F_\nu=(f_{\nu s-1},\ f_{\nu s-2},\ \cdots,\ f_{(\nu-1)s})^T \tag{10-10}$$

其中$f_i=f(t_i,\ y_i)$($i=0,\ 1,\ 2,\ \cdots$)所有的量y_i，f_i($i=0,\ 1,\ \cdots$)和Y_ν，F_ν均可以带上标p，例如$f_i^p=f(t_i,\ y_i^p)$，$Y_\nu^p=(y_{\nu s-1}^p,\ \cdots,\ y_{(\nu-1)s}^p)^T$，$F_\nu^p=(f_{\nu s-1}^p,\ \cdots,\ f_{(\nu-1)s}^p)^T$。

用$|M|$表示矩阵M的行列式，令

$$A_0^*,\ A_\mu,\ A_\mu^p,\ B_\mu,\ B_\mu^p,\ \mu=0,\ 1,\ \cdots,K$$

表示$s\times s$矩阵，并且有$|A_0|\neq 0$，$|A_0^*|\neq 0$，考虑对$Y_{\nu+1}^p$和Y_ν的显式差分方程组

$$A_0^*Y_{\nu+1}^p+A_1^pY_\nu^p+\sum_{\mu=2}^{K}A_\mu^pY_{\nu-\mu+1}+h\left(B_1^pF_\nu^p+\sum_{\mu=2}^{K}B_\mu^pF_{\nu-\mu+1}\right)=0 \tag{10-11}$$

$$A_0Y_\nu+A_0^*Y_\nu^p+\sum_{\mu=1}^{K}A_\mu Y_{\nu-\mu}+h\left(B_0F_\nu^p+\sum_{\mu=1}^{K}B_\mu F_{\nu-\mu}\right)=0 \tag{10-12}$$

$$\nu=1,\ 2,\ \cdots$$

用W_ν表示s维向量，它的分量是问题(10-1)的精确解$y(t)$在格点t_i($i=\nu s-1,\ \nu s-2,\ \cdots,(\nu-1)s$)上的值。用$W_\mu$代替式(10-11)(10-12)中除$Y_{\nu+1}^p$和$Y_\nu$的所有$Y_\mu$和$Y_\mu^p$项，然后从中将$Y_{\nu+1}^p$和$Y_\nu$解出，将解出的值分别记成$\bar{Y}_{\nu+1}^p$和$\bar{Y}_\nu$。

记

$$Y_\nu=(Y_{\nu+1}^p,\ Y_\nu)^T,\quad W_\nu=(W_{\nu+1},\ W_\nu)^T,\quad F_\nu=(F_{\nu+1}^p,\ F_\nu)^T \tag{10-13}$$

它们都是$2s$维向量，令

$$\boldsymbol{A}_0=\begin{pmatrix} A_0^p & 0 \\ 0 & A_0 \end{pmatrix} \quad \boldsymbol{A}_1=\begin{pmatrix} A_1^p & A_2^p \\ A_0^* & A_1 \end{pmatrix} \quad \boldsymbol{A}_\mu=\begin{pmatrix} 0 & A_{\mu+1}^p \\ 0 & A_\mu \end{pmatrix}$$

$$\boldsymbol{B}_1=\begin{pmatrix} B_1^p & B_2^p \\ B_0 & B_1 \end{pmatrix} \quad \boldsymbol{B}_\mu=\begin{pmatrix} 0 & B_{\mu+1}^p \\ 0 & B_\mu \end{pmatrix} \quad \mu=2,\cdots,K \tag{10-14}$$

它们都是$2s\times 2s$矩阵，其中取$A_{k+1}^p=B_{k+1}^p=0$。

利用这些记号，差分方程（10-11）（10-12）可以写成

$$\sum_{\mu=0}^{K} \boldsymbol{A}_\mu \boldsymbol{Y}_{\nu-\mu}+h\sum_{\mu=1}^{K}\boldsymbol{B}_\mu \boldsymbol{F}_{\nu-\mu}=0 \quad \nu=k+1,k+2,\cdots \tag{10-15}$$

其中$|\boldsymbol{A}_0|\neq 0$，对于任意的$2s\times 2s$矩阵$\boldsymbol{C}$，用$(\overline{\boldsymbol{C}})_{qr}$，$q,r=1,2,\cdots,2s$表示它的元素，而用$(V)_r$，$r=1,2,\cdots,2s$表示$2s$维向量$\boldsymbol{V}$的分量，用$(V)_r$（$r=1,2,\cdots,s$）表示$s$维向量$\boldsymbol{V}$的分量。于是式（10-15）中的项$\boldsymbol{B}_\mu \boldsymbol{F}_{\nu-\mu}$的元素有

$$(\boldsymbol{B}_\mu \boldsymbol{F}_{\nu-\mu})_q=\sum_{r=1}^{s}(\boldsymbol{B}_\mu)_{qr}(F_{\nu-\mu}^p)_r+\sum_{r=s+1}^{2s}(\boldsymbol{B}_\mu)_{qr}(F_{\nu-\mu})_r$$

$$\mu=1,\cdots,K,\quad q=1,\cdots,2s \tag{10-16}$$

由于$|\boldsymbol{A}|\neq 0$，在式（10-15）的两边左乘$\boldsymbol{A}_0$的逆$\boldsymbol{A}_0^{-1}$，得到

$$\boldsymbol{Y}_\nu=-\sum_{\mu=1}^{K}(\boldsymbol{A}_0^{-1}\boldsymbol{A}_\mu \boldsymbol{Y}_{\nu-\mu}+h\boldsymbol{A}_0^{-1}\boldsymbol{B}_\mu \boldsymbol{F}_{\nu-\mu})$$

$$\nu=K+1,K+2,\cdots, \tag{10-17}$$

实际计算时可按下述计算步骤应用式（10-17）。

步0　给出$\boldsymbol{Y}_1,\boldsymbol{Y}_2,\cdots,\boldsymbol{Y}_k$和$\boldsymbol{F}_1,\boldsymbol{F}_2,\cdots,\boldsymbol{F}_k$，令$\nu=k+1$。

步1　按公式（10-17）计算$\boldsymbol{Y}_\nu=(Y_{\nu+1}^p,Y_\nu)^T$

步2　并行计算右函数

$f_{(\nu+1)s-1}^p=f(t_{(\nu+1)s-1},y_{(\nu+1)s-1}^p),\cdots,f_{\nu s}^p=f(t_{\nu s},y_{\nu s}^p)$

$f_{\nu s-1}=f(t_{\nu s-1},y_{\nu s-1}),\cdots,f_{(\nu-1)s}=f(t_{(\nu-1)s},y_{(\nu-1)s})$

组成向量F.

步 3　令 $\nu = \nu + 1$，转步 1。

由步 1 和式（10-17）可知，计算Y_ν主要是向量运算，而由步 2 可知，在计算$2s$个右函数时，右函数的自变量是已知的，这$2s$个右函数值可并行计算，因而这算法适合在多处理机，特别是在向量机上进行计算。

现在简要地叙述由式（10-16）或式（10-17）给出的显式并行算法的一些理论结果。令$A(z)$和$B(z)$是由

$$\begin{aligned} A(z) &= \sum_{\mu=0}^{K} A_\mu z^{k-\mu} \\ B(z) &= \sum_{\mu=1}^{K} B_\mu z^{k-\mu} \end{aligned} \tag{10-18}$$

给出的复变量 z 的矩阵系数多项式。

定义10-1　如果有

$$\begin{aligned} A(1) &= 0 \\ B(1) + sA'(1) &= 0 \end{aligned} \tag{10-19}$$

则差分方程（10-15）称为相容的，其中“′”表示d/dz。差分方程（10-15）的第 q 行为

$$\sum_{r=1}^{2s} \sum_{\mu=0}^{K} (A_\mu)_{q,r}(Y_{\nu-\mu})_r + h \sum_{r=1}^{2s} \sum_{\mu=1}^{K} (B_\mu)_{q,r}(F_{\nu-\mu})_r = 0$$

$$q = 1,\ 2,\ \cdots,\ 2s \tag{10-20}$$

其中上标 p 均省略掉，这是一个步数为 $(k+1)s - \varphi(q) - 1$ 的y_i的线性差分方程，

$$\varphi(q) = \min\{r \mid (A_0)_{q,r} \neq 0,\ r = 1,\ 2,\ \cdots 2s\}$$

定义10-2　如果对每个 $q = 1,\ 2,\ \cdots,\ 2s$，差分方程(10-20）都是相容的，称差分方程（10-15）为按行相容的。

定义10-3　如果多项式 $|A(z)|$ 的根均在闭单位圆上，并且模为 1 的根为单根，则称多项式$|A(z)|$是满足根条件的。

在式（10-17）中取 $Y_{\nu-\mu} = W_{\nu-\mu}$，$F_{\nu-\mu} = W'_{\nu-\mu}$，记其左边量

为

$$\widetilde{\boldsymbol{Y}}_n=(\widetilde{Y}_{n+1}^{p},\ \widetilde{Y}_n)^T$$

则称量

$$\widetilde{\boldsymbol{e}}_n=\widetilde{\boldsymbol{Y}}_n-\boldsymbol{W}_n$$

为差分方程（10-15）的局部截断误差。

下面我们假定 $f(t,\ y)$ 对 t 是连续的，对 y 是李普希茨连续的，其李普希茨常数为 l。

我们在固定区间〔0，1〕中讨论，引进 $y'(t)$ 的连续模 $\xi(\delta)$

$$\xi(\delta)=\max\{|y'(t^*)-y'(t)|\ \|\ t-t^*|\leqslant\delta,\ \ t,\ t^*\in〔0,\ 1〕\} \tag{10-21}$$

再引进两组数 θ_j 和 θ'_j，$j=-s,\ -s+1,\ \cdots,\ 0,\ 1,\ \cdots$ 使下列关系式成立

$$y'(t_{i-j})=y'(t_i)+\theta_j\xi(jh)$$

$$y(t_{i-j})=y(t_i)+jh〔y'(t_i)+\theta'_j\xi(jh)〕 \tag{10-22}$$

对局部截断误差的量级有下面的引理

引理10-1 如果差分方程是相容的，或按行相容的，则当 $h\to 0$ 时，有

$$\widetilde{\boldsymbol{e}}_n=0(h)$$

证明 记 u 为 $2s$ 维向量 $(1,\ 1,\ \cdots,\ 1)^T$，于是

$$\begin{aligned}-\overline{\boldsymbol{A}}_0\widetilde{\boldsymbol{e}}_n&=\sum_{\mu=0}^{K}\boldsymbol{A}_\mu\boldsymbol{W}_{n-\mu}+h\sum_{\mu=1}^{K}\boldsymbol{B}_\mu\boldsymbol{W}'_{n-\mu}\\&=\sum_{\mu=0}^{K}\boldsymbol{A}_\mu\boldsymbol{W}_{n-k}+sh\sum_{\mu=0}^{K}(k-\mu)\boldsymbol{A}_\mu〔\boldsymbol{W}'_{n-k}\\&\quad+u\theta'_{(k-u)s}\xi((k-\mu)sh)〕\\&\quad+h\sum_{\mu=1}^{K}\boldsymbol{B}_\mu〔\boldsymbol{W}'_{n-k}+u\theta_{(k-u)s}\xi((k-\mu)sh)〕\end{aligned} \tag{10-23}$$

由于 $\boldsymbol{B}_0=0$

$$-A_0\tilde{e}_n=\sum_{\mu=0}^{K}A_\mu W_{n-k}+h\sum_{\mu=0}^{K}(B_\mu+(k-\mu)sA_\mu)W'_{n-k}+0(h) \tag{10-24}$$

由相容性和$|A_0|\neq 0$，推得

$$\tilde{e}_n=0(h)$$

为了从按行相容性推得同样的结论，可以这样进行。记$(W_{n-k})_q$为W_{n-k}的第q个分量，由泰勒（Taylor）定理得

$$W_{n-k}=(W_{n-k})_s u+h(W'_{n-k})_s\begin{pmatrix}s-1\\ \vdots\\ -s\end{pmatrix}+h\begin{pmatrix}(s-1)\theta'_{s-1}\xi((s-1)h)\\ \vdots\\ (-s)\theta'_s\xi(sh)\end{pmatrix} \tag{10-25}$$

$$W'_{n-k}=(W'_{n-k})_s u+\begin{pmatrix}\theta_{s-1}\xi((s-1)h)\\ \vdots\\ \theta_s\xi(sh)\end{pmatrix} \tag{10-26}$$

$$W_{n-k}=(W_{n-k})_{2s}u+h(W'_{n-k})_{2s}\begin{pmatrix}2s-1\\ \vdots\\ 0\end{pmatrix}+h\begin{pmatrix}(2s-1)\theta'_{2s-1}\xi((2s-1)h\\ \vdots\\ 0\end{pmatrix} \tag{10-27}$$

$$W'_{n-k}=(W'_{n-k})_{2s}u+\begin{pmatrix}\theta_{2s-1}\xi((2s-1)h)\\ \vdots\\ 0\end{pmatrix} \tag{10-28}$$

令γ是取值为1或2的变量，将式(10-25)(10-26)代入式(10-24)，或将式(10-27)、(10-28)代入式(10-24)都得到下面的公式(10-29)，但其中γ分别取1和2

$$-A_0\tilde{e}_n=\sum_{\mu=0}^{K}A_\mu(W_{n-k})_{\gamma s}u+h\sum_{\mu=0}^{K}A_0(W'_{n-k})_{\gamma s}\begin{pmatrix}\gamma s-1\\ \vdots\\ (\gamma-2)s\end{pmatrix}$$

$$+h\sum_{\mu=0}^{K}[\overline{\boldsymbol{B}}_{\mu}+(k-\mu)s\boldsymbol{A}_{\mu}](\boldsymbol{W}'_{\nu-k})_{\nu s}\boldsymbol{u}$$
$$+0(h) \tag{10-29}$$

于是

$$(-\boldsymbol{A}_0\tilde{\boldsymbol{e}}_{\nu})_q=(\boldsymbol{W}_{\nu-k})_{\nu s}\sum_{r=1}^{2s}\sum_{\mu=0}^{K}(\boldsymbol{A}_{\mu})_{q,r}$$
$$+h(\boldsymbol{W}'_{\nu-k})_{\nu s}\sum_{r=1}^{2s}\sum_{\mu=0}^{K}(\boldsymbol{B}_{\mu})_{q,r}$$
$$+h(\boldsymbol{W}'_{\nu-k})_{\nu s}\sum_{r=1}^{2s}\sum_{\mu=0}^{K}[(k-\mu)s$$
$$+\nu s-r](\boldsymbol{A}_{\mu})_{q,r}+0(h) \tag{10-30}$$

令

$$\nu=\begin{cases}1 & 1\leqslant q\leqslant s\\ 2 & s+1\leqslant q\leqslant 2s\end{cases} \tag{10-31}$$

通过考察式（10-15）可看出按行相容性给出

$$(-\boldsymbol{A}_0\tilde{\boldsymbol{e}}_{\nu})_q=0(h) \tag{10-32}$$

这式和$|\boldsymbol{A}_0|\neq 0$推出引理。

下面的引理给出满足根条件的多项式的一个性质。

引理10-2 设多项式$|M(z)|=\left|\sum_{\mu=0}^{K}M_{\mu}z^{K-\mu}\right|$满足根条件，其中$|M_0|\neq 0$，$M_{\mu}$是矩阵，则矩阵$[z^KM(1/z)]^{-1}$在$z=0$的邻域中是解析的，如果$M_{\mu}$，$\mu=0$，1，…是由

$$\left[z^KM\left(\frac{1}{z}\right)\right]^{-1}=\sum_{\mu=0}^{\infty}M_{\mu}z^{\mu} \tag{10-33}$$

给出的矩阵，则有

$$\max_{\mu}\|M_{\mu}\|=M<\infty \tag{10-34}$$

证明 由于 $z^kM(1/z)=\sum_{\mu=0}^{K}M_\mu z^\mu$ 和 $|M_0|\neq 0$，显然 $[z^kM(1/z)]^{-1}$ 在原点的邻域中是解析的，由于

$$|z^kM(1/z)|=z^{2rk}|M(1/z)|$$

由根条件推出多项式$|z^kM(1/z)|$的根在开单位圆的外面，并且模为 1 的根是单根，由于

$$[z^kM(1/z)]^{-1}=[\text{多项式矩阵}]/|z^kM(1/z)|$$

只要证明多项式的逆$|z^kM(1/z)|^{-1}$的幂级数具有有界的系数，进行直接计算可以证明这个结论。

称量

$$e_n=Y_n-W_n$$

为累积截断误差。利用上面两个引理，按照亨利西（Henrici)的书中证明的思路，得到下面的定理。

定理10-1 设差分方程（10-15）是相容的，或按行相容的，并且求解时的起始值 Y_ν，$\nu=1,2,\cdots,k$ 的误差 $e_\nu=Y_\nu-W_\nu$，当 $h\to 0$ 时有

$$e_n=0(1)$$

如果$|A(z)|$满足根条件，则有

$$e_n=Y_n-W_n=0(1)\quad \nu=1,2,\cdots,\lfloor 1/sh\rfloor$$

其中$\lfloor\ \rfloor$表示 $1/sh$的整数部分。

这定理说明相容性和根条件是方法收敛性的充分性条件。

下面给出一些具体的并行积分公式：

1. 二个处理器的情形（$s=1$）。考虑单个预估公式和单个校正公式的组合，构成广义的亚当斯型公式组，其中

$$A(z)=A_0z^k+A_1z^{k-1}A_j=0,\quad 2\leqslant j\leqslant k \qquad (10\text{-}35)$$

由于 $s=1$，A_j，$j=0,1,\cdots$均是 2×2 矩阵，A_j和 B_j 均是数，将它们分别表成a_j，b_j。由式（10-35）推得$a_j=0(2\leqslant j\leqslant k)$ 和$a_j^\rho=0(3\leqslant j\leqslant k)$，于是两个积分公式具有形式

$$a_0^\rho y_{i+1}^\rho+a_1^\rho y_i^\rho+a_2^\rho y_{i-1}+h(b_1^\rho f_i^\rho+b_2^\rho f_{i-1}+\cdots)=0 \qquad (10\text{-}36)$$

$$a_0y_i+a_0^*y_i^p+a_1y_{i-1}+h(b_0f_i^p+b_1f_{i-1}+\cdots)=0$$

对于这种型式的公式，

$$|A(z)|=z^{2(k-1)}|A_0z+A_1|$$

因此，$z=0$ 至少是 $2(k-1)$ 重根，$|A(z)|=0$ 的其余两个根是二次方程 $|A_0z+A_1|=0$，即

$$a_0a_0^pz^2+(a_0^pa_1+a_0a_1^p)z+(a_1a_1^p-a_0^*a_1^p)=0 \qquad (10\text{-}37)$$

的根，按行相容性要求

$$a_1^p=-(a_0^p+a_1^p) \quad a_0^*=-(a_0+a_1) \qquad (10\text{-}38)$$

将式（10-38）代入式（10-37），并解出 z，得到两个根分别为 $z_1=1$ 和 $z_2=-[1+(a_1/a_0)+(a_1^p/a_0^p)]$。不损失一般性，可以进行正则化，使 $a_0^p=a_0=-1$，于是 $z_2=-1+(a_1+a_1^p)$，$a_2^p=1-a_1^p$，$a_0^*=1-a_1$。如果取

$$a_1=1-a_1^p=a_2^p \qquad (10\text{-}39)$$

若取 $a_0^*=a_1^p$，则有 $z_2=0$，这时根条件是成立的。更一般的，若取 a_1^p，a_1，满足

$$-a_1^p\leqslant a_1<2-a_1^p \qquad (10\text{-}40)$$

则有 $-1\leqslant z_2<1$，为避免 $z=1$ 为二重根，$z_2=1$ 是不能取的，在式（10-40）给出的范围中 a_1 可自由选取，通常选取 a_1 有两个目的，一个目的是使稳定区域大，另一个目的是提高公式的精度。

为简单起见，取 $a_1^p=0$，这时若应用式（10-39），得 $a_1=a_2^p=1$，$a_0^*=0$，若应用式（10-40）得 $0\leqslant a_1<2$。现应用式(10-39)，由式（10-38）得到

$$-y_{i+1}^p+y_{i-1}+h\left(b_1^pf_i^p+\sum_{j=2}^{r+1}b_j^pf_{i-j+1}\right)=0 \qquad (10\text{-}41)$$

$$-y_i+y_{i-1}+h\left(b_0^pf_i^p+\sum_{j=1}^{r}b_jf_{i-j}\right)=0$$

式（10-41）中的 b^p 和 b 可以选成按行的精度阶 $\geqslant 1$，于是按行相容性成立，下面是三组公式，它们的阶分别为 2，3，4。

$$\begin{aligned} y_{i+1}^{p} &= y_{i-1}+2hf_i^{p} \\ y_i &= y_{i-1}+\frac{h}{2}(f_i^{p}+f_{i-1}) \end{aligned} \tag{10-42}$$

$$\begin{aligned} y_{i+1}^{p} &= y_{i-1}+\frac{h}{3}(7f_i^{p}-2f_{i-1}+f_{i-2}) \\ y_i &= y_{i-1}+\frac{h}{12}(5f_i^{p}+8f_{i-1}-f_{i-2}) \end{aligned} \tag{10-43}$$

$$\begin{aligned} y_{i+1}^{p} &= y_{i-1}+\frac{h}{3}(8f_i^{p}-5f_{i-1}+4f_{i-2}-f_{i-3}) \\ y_i &= y_{i-1}+\frac{h}{24}(9f_i^{p}+19f_{i-1}-5f_{i-2}+f_{i-3}) \end{aligned} \tag{10-44}$$

2. 处理器的个数$\geqslant 4$的情形（$s\geqslant 2$）。对于$s\geqslant 2$的情形，预估-校正公式(10-11)(10-12)中的系数均是$s\times s$矩阵，再取$\boldsymbol{A}_\mu=0$，$2\leqslant\mu\leqslant k$或取$\boldsymbol{A}_\mu=0$，$2\leqslant\mu\leqslant k$和$A_0^{p}=0$，$3\leqslant\mu\leqslant k$，则$z=0$是多项式$|\boldsymbol{A}(z)|=z^{2s(k-1)}|\boldsymbol{A}_0z+\boldsymbol{A}_1|$的重数至少为$2s(k-1)$的根，其余的$2s$个根为多项式$|\boldsymbol{A}_0z+\boldsymbol{A}_1|$的零点，如果取$\boldsymbol{A}_0=-I_{2s\times 2s}$使有$A_0^{p}=A_0=-I_{s\times s}$（其中$I_{q\times q}$表示$q\times q$单位矩阵）则有

$$\boldsymbol{A}_0z+\boldsymbol{A}_1=-\begin{pmatrix} zI_{s\times s}-A_1^{p} & -A_2^{p} \\ -A_0^{*} & zI_{s\times s}-A_1 \end{pmatrix} \tag{10-45}$$

取$A_0^{*}=0$，并取A_1^{p}和A_1具有构造

$$A_1^{p}=\begin{pmatrix} 0 & \times & \times & \cdots & \times \\ 0 & 0 & \times & \cdots & \\ 0 & & 0 & \times & \times \\ & \ddots & & 0 & \times \\ 0 & \cdots\cdots & & & 0 \end{pmatrix} \quad A_1=\begin{pmatrix} 0 & \times & \times & \cdots & & \times \\ & 0 & \times & \cdots & & \vdots \\ & & 0 & \times & \cdots & \vdots \\ & & & 0 & \times & \times \\ & 0 & & & 0 & \times \\ & & & & & \times \end{pmatrix} \tag{10-46}$$

其中“×”的位置的元素不要求为零，对于$\boldsymbol{A}_0=-I_{2s\times 2s}$，由式（10-

46）和对校正公式的最后一行的相容性要求推出A_1的右下角处的元素为1，由A_1^p和A_1的结构给出$|A_0z+A_1|=-z^{2s-1}(z-1)$，因此$z=0$是$|A(z)|$的重数为$(2sk-1)$的根，$z=1$是它的单根（主根）。这表示$|A(z)|$满足根条件，如果对$y^p$和$y$的单个行所表示的公式中的系数选成使精度阶$\geqslant 1$，则按行的相容性也是有保证的，满足上述条件的公式的一般形式为

$$-y^p_{(\nu+1)s-q+1}+\sum_{r=q+1}^{s}(A_1^p)_{qr}y^p_{\nu s-r+1}+\sum_{r=1}^{s}(A_2^p)_{qr}y_{(\nu-1)s-r+1}+h(\cdots)=0$$
$$q=1,\ 2,\ \cdots,\ s \tag{10-47}$$

$$-y_{\nu s-q+1}+\sum_{r=q+1}^{s}(A_1)_{qr}y_{(\nu-1)s-r+1}+h(\cdots)=0$$
$$q=1,\ 2,\ \cdots,\ s-1 \tag{10-48}$$
$$-y_{(\nu-1)s+1}+y_{(\nu-2)s+1}+h(\cdots)=0$$

其中$h(\cdots)$表示对f的求和项，如果取$A_1^p=0$，上面给出的公式将变得特别简单。下面列出二组具体的公式，都是$s=2$的，按行的精度分别为$O(h)$和$O(h^2)$

$$\left.\begin{aligned}y^p_{2i+2}&=y_{2i-2}+4hf^p_{2i}\\y^p_{2i+1}&=y_{2i-2}+3hf^p_{2i}\\y_{2i}&=y_{2i-3}+3hf^p_{2i}\\y_{2i-1}&=y_{2i-3}+2hf^p_{2i-1}\end{aligned}\right\} \tag{10-49}$$

$$\left.\begin{aligned}y^p_{2i+2}&=y_{2i-2}+4hf^p_{2i}\\y^p_{2i+1}&=y_{2i-2}+\frac{3h}{2}(f^p_{2i}+f^p_{2i-1})\\y_{2i}&=y_{2i-3}+\frac{h}{2}(3f^p_{2i}+9f^p_{2i-1})\\y_{2i-1}&=y_{2i-3}+2hf_{2i-2}\end{aligned}\right\} \tag{10-50}$$

10.3 应用理查森外插的梯形公式的并行计算

线性多步公式的传统应用方式是一个点接着一个点来确定y_i，是串行的方式。虽然按照10.1节的处理，方法本身的计算可向量化，从而可进行并行处理，但右函数的求值仍无法并行进行，因而，需要对传统数值积分方法的应用方式进行改造，以适应并行处理的需要。

目前，由于对数值积分的精度和稳定性要求愈来愈高，特别是数值积分刚性问题的需要，采用隐式公式进行数值积分已很普遍。对于隐式公式，需要进行迭代。既然求一个点的值需要进行迭代，我们可以同时对若干个点进行迭代，从而使计算并行化。基于这种思想，文献〔22〕〔23〕改变了传统积分公式的使用方式，使在每一步计算中可以确定N个点t_i上的值y_i，并且使得N个点上的右函数可同时计算。下面我们以梯形公式为例来说明算法的构造。

达奎斯脱（Dahlquist）证明A-稳定的线性多步公式一定是隐式的，其中梯形公式达到最高精度阶，并且具有最小的误差系数，因而梯形公式经常作为构造求解刚性方程数值方法的基础。由于梯形公式的精度阶是2，并且需要求解隐式方程，因而与其它算法相比，在达到同样精度的条件下，它所花费的工作量较大。为了克服这个缺点，将下面推导的并行算法结合理查森(Richardson)外插和林德伯格（Lindberg）内插的技巧、构造了在向量机或多处理机上能高效地实现的并行算法。这算法也可在串行机上实现。每算一个点仅需计算$\frac{15}{8}$个点上的右函数，而对同样精度的隐式龙格-库塔方法，需4个点上的右函数。

10.3.1 梯形公式的并行计算方法

考虑常微分方程组初值问题

$$y' = f(t, y), \quad t \in [t_a, t_b] \tag{10-51}$$

$$y(t_a) = y_a \tag{10-52}$$

其中y为n维列向量，$f(\cdot,\ \cdot)$为t，y的n维向量值函数，t为独立变量。记式（10-51）、（10-52）的精确解为$y(t)$，假定$f(\cdot,\ \cdot)$对其变量t，y充分光滑，具有下文所需要的各阶导数。

设已知$y(t)$及其导数$y'(t)$在点$t_0\in[t_a,\ t_b]$处的近似值y_0和$y_0'=f(t_0,\ y_0)$，要求确定$y(t)$在格点$t_i=t_0+iH$上的值$y(t_i)$的近似值y_i，其中$i=1,\ 2,\ \cdots,\ N$，H为步长，计算y_i的梯形公式为

$$y_{i+1}=y_i+\frac{H}{2}[f(t_i,\ y_i)+f(t_{i+1},\ y_{i+1})] \tag{10-53}$$

引进下面的记号

$$\mathbf{Y}=\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix} \qquad \mathbf{Y}'=\begin{bmatrix} f(t_1,\ y_1) \\ f(t_2,\ y_2) \\ \vdots \\ f(t_N,\ y_N) \end{bmatrix}$$

$$\mathbf{Y}_l=\begin{bmatrix} y_1(l) \\ y_2(l) \\ \vdots \\ y_N(l) \end{bmatrix} \quad \mathbf{Y}'_l=\begin{bmatrix} y_1(l)' \\ y_2(l)' \\ \vdots \\ y_N(l)' \end{bmatrix}$$

其中$\mathbf{Y}_l$，$\mathbf{Y}'_l$分别为$\mathbf{Y}$和$\mathbf{Y}'$的近似，按下述方式逐次计算$\mathbf{Y}_l$，$\mathbf{Y}'_l$，设已给出$y_1^{(0)},\ y_2^{(0)},\ \cdots,\ y_N^{(0)}$，并行计算$N$个右函数值$y_i^{(0)\prime}=f(t_i,\ y_i^{(0)})(i=1,\ \cdots,\ N)$即得到$\mathbf{Y}_0$，$\mathbf{Y}'_0$。若有

$$y_{i+1}^{(0)}=y_i^{(0)}+\frac{H}{2}(y_i^{(0)\prime}+y_{i+1}^{(0)\prime})$$

$$i=0,\ 1,\ 2,\ \cdots N-1 \tag{10-54}$$

则$\mathbf{Y}_0$，$\mathbf{Y}'_0$即为所求的$\mathbf{Y}$和$\mathbf{Y}'$。在式（10-54）中约定$y_0^{(0)}=y_0$，$y_0^{(0)\prime}=y_0'$，假设式（10-54）不成立，记$\xi_0=0$，$\xi_i=y_i-y_0^{(0)}(i=1,\ 2,\ \cdots,\ N)$，于是$\xi_i$满足方程

$$y_{i+1}^{(0)}+\xi_{i+1}=y_i^{(0)}+\xi_i+\frac{H}{2}[f(t_i,\ y_i^{(0)}+\xi_i)+f(t_{i+1},\ y_{i+1}^{(0)}+\xi_{i+1})],$$

$$i=0,\ 1,\ \cdots,\ N-1 \tag{10-55}$$

将上式右边进行展开，并令

$$d_{i+1}=y_i^{(0)}+\frac{H}{2}[y^{(0)\prime}+y_{i+1}^{(0)\prime}]-y_{i+1}^{(0)} \tag{10-56}$$

得

$$\xi_{i+1}=\xi_i+\frac{H}{2}\left[\frac{\partial f(t_i,\ y_i^{(0)})}{\partial y}\xi_i+\frac{\partial \mathbf{Y}(t_{i+1},\ y_{i+1}^{(0)})}{\partial y}\xi_{i+1}\right]$$
$$+d_{i+1}+O(H\|\xi_i\|^2)+O(H\|\xi_{i+1}\|^2) \tag{10-57}$$

略去$O(H\|\xi_i\|^2)$和$O(H\|\xi_{i+1}\|^2)$项，并将式（10-57）中的$n\times n$矩阵$\partial f(t_i,\ y_i^{(0)})/\partial y$和$\partial f(t_{i+1},\ y_{i+1}^{(0)})/\partial y$换成固定的矩阵$A$，得到$\xi_i$的近似值$\pi_i$的递推式

$$\pi_0=0,\quad \pi_{i+1}=\left[I-\frac{H}{2}A\right]^{-1}\left[\left(I+\frac{H}{2}A\right)\pi_i+d_{i+1}\right],$$
$$i=0,\ 1,\ \cdots,\ N-1 \tag{10-58}$$

由式（10-58）解得π_i，$i=1$，$2\cdots$，N后，令

$$y_i^{(1)}=y_i^{(0)}+\pi_i \quad i=1,\ 2,\ \cdots,\ N \tag{10-59}$$

得$\mathbf{Y}_1$，以$\mathbf{Y}_1$代替上面推导中的$\mathbf{Y}_0$，进行迭代，逐次得到$\mathbf{Y}_2$，$\mathbf{Y}_3$，…。当$\mathbf{Y}_0$和H满足适当的条件，这迭代过程将收敛，对于给定的容许误差ε，可找到l，使$\|d_{i+1}\|\leqslant\varepsilon$对所有$i=0$，$1$，…，$N-1$均成立，这时令$\mathbf{Y}=\mathbf{Y}_l$，上述的迭代收敛性证明可按文献〔23〕类似地进行。由于上述的迭代过程实际上是加进扰动后的牛顿方法，若迭代收敛，收敛速度是很快的，通常迭代2至3次就能满足精度。

在上述迭代过程中，每次可并行计算N个右函数的值。

若容许误差ε取得适当小，例如小于下面式（10-60）、（10-61）中的$O(H^9)$项，并且y_0具有渐近展开式

$$y_0=y(t_0)+a_2(t_0)H^2+a_4(t_0)H^4+a_6(t_0)H^6+a_8(t_0)^8+O(H^9) \tag{10-60}$$

则求得的$y_i=y_i^{(1)}$有渐近展开式：记$y(t_i,\ H)=y_i$，

$$y(t_i,\ H)=y(t_i)+a_2(t_i)H^2+a_4(t_i)H^4+a_6(t_i)H^6$$
$$+a_8(t_i)H^8+O(H^9) \tag{10-61}$$

式（10-60）、（10-61）中的$a_2(\cdot)$，$a_4(\cdot)$，$a_6(\cdot)$，$a_8(\cdot)$均是t的

充分光滑的函数，与步长H无关。

10.3.2 理查森外插过程

由式（10-61）可知，用梯形公式（10-53）求解得到的近似解y_i的误差为

$$y_i - y(t_i) = a_2(t_i)H^2 + a_4(t_i)H^4 + a_6(t_i)H^6 + a_8(t_i)H^8 + O(H^9) \tag{10-62}$$

为了提高精度，需要将式（10-62）右边的各项消去，这就是理查森外插过程。为了叙述这个过程，选定基本步长h、建立相应的基本格点$t_i = t_0 + ih$（$i = 1, 2, \cdots, N$），其中$N = 2^3L$，L为自然数，以$H = 2^j h$为步长，按10.3.1中的方式计算，得到格点t_{2^jk}上的近似值$Y(t_{2^jk}, 2^j h)$，$K = 1, 2, \cdots N/2^j$，上述的j取0，1，2，3。

为了消去式（10-62）中的$a_2(t)H^2$项，作理查森一次外插

$$y_1(t_{2^jk}, 2^{j-1}h) = -\frac{4}{3} y(t_{2^jk}, 2^{j-1}h) - \frac{1}{3} y(t_{2^jk}, 2^j h)$$

$$j = 1, 2, 3, \quad K = 0, 1, \cdots, N/2^j \tag{10-63}$$

由式（10-61）、（10-63）确定的$y_1(t_i, H)$具有展开式

$$y_1(t_i, H) = y(t_i) + b_4(t_i)H^4 + b_6(t_i)H^6 + b_8(t_i)H^8 + O(H^9) \tag{10-64}$$

其中$b_4(\cdot)$，$b_6(\cdot)$，$b_8(\cdot)$由下式确定

$$b_4(t) = -4a_4(t)$$

$$b_6(t) = -20a_6(t)$$

$$b_8(t) = -84a_8(t)$$

再作理查森二次外插，消去式（10-64）中的H^4项，得

$$y_2(t_{2^jk}, 2^{j-2}h) = \frac{16}{15} y_1(t_{2^jk}, 2^{j-2}h) - \frac{1}{15} y_1(t_{2^jk}, 2^{j-1}h)$$

$$j = 2, 3, \quad k = 0, 1, \cdots, N/2^j \tag{10-65}$$

由式（10-64）、（10-65）式确定的$y_2(t_i,\ H)$具有展开式

$$y_2(t_i,\ H)=y(t_i)+C_6(t_i)H^6+C_8(t_i)H^8+O(H^9) \tag{10-66}$$

其中

$$C_6(t)=64a_6(t)$$
$$C_8(t)=1344a_8(t)$$

为了消去式（10-66）中的H^6项，作理查森三次外插，得

$$y_3(t_{8k},\ h)=\frac{64}{63}y_2(t_{8k},\ h)-\frac{1}{63}y_2(t_{8k},\ 2h)$$
$$k=0,1,\cdots,N/8 \tag{10-67}$$

$y_3(t_i,\ h)$有展开式

$$y_3(t_i,\ h)=y(t_i)+d_8(t_i)H^8+O(H^9) \tag{10-68}$$

其中

$$d_8(t)=4096a_8(t)$$

10.3.3 林德伯格内插法求中间点上的精确值

由10.3.2中的结果可知，若在t_{8k}（$k=0,1,\cdots,N/8$）上取$y_3(t_{8k},\ h)$作为$y(t_{8k})$的近似值，误差为$O(H^8)$的量级，精度比较高，但这些格点比较稀；为了得到所有格点t_i（$i=0,1,\cdots,N$）上的高精度的解，按常规需要将h缩小，从而加大计算量，林德伯格推导了一个算法，直接利用10.3.2中理查森外插得到的信息进行内插，得到全部格点上的精度高的近似值。我们不考虑一般情形，只根据构造算法的实际需要对理查森三次外插的情形进行讨论。为简单起见，只讨论格点t_0，t_1，…，t_8，其它格点的公式相似。

令

$$y_0=y_3(t_0,\ h)$$
$$y_8=y_3(t_8,\ h) \tag{10-69}$$
$$z_2(t_0)=y_0-y_2(t_0,\ h)$$
$$z_2(t_8)=y_8-y_2(t_8,\ h) \tag{10-70}$$

由式（10-66）、（10-68）

$$z_2(t_i)=-C_6(t_i)h^6+O(h^8)\quad i=0,\ 8 \tag{10-71}$$

由线性内插确定 $z_2(t_4)$为

$$z_2(t_4)=\frac{1}{2}z_2(t_0)+\frac{1}{2}z_2(t_8)$$

由函数$C^6(t)$ 的可微性和线性内插的误差估计，

$$z_2(t_4)=-C_6(t_4)h^6+O(h^7) \tag{10-72}$$

令

$$y_4=y_2(t_4,\ h)+z_2(t_4) \tag{10-73}$$

由式(10-72)、(10-66)

$$y_4=y(t_4)+O(h^7) \tag{10-74}$$

对t_i，$i=0,4,8$，令

$$z_1(t_i)=y_i-y_1(t_i,\ h) \tag{10-75}$$

由式 (10-64)、(10-68)、(10-74)，有估计式

$$z_1(t_i)=-b_4(t_i)h^4-b_6(t_i)h^6+O(h^8) \tag{10-76}$$

其中

$$P=\begin{cases}8 & i=0,\ 8\\ 7 & \text{其它}\end{cases} \tag{10-77}$$

以t_0，t_4，t_8为格点作拉格朗日插值多项式

$$z_1(t)=l_{10}(t)z_1(t_0)+l_{14}(t)z_1(t_4)+l_{18}(t)z_1(t_8) \tag{10-78}$$

其中$l_{10}(t)$，$l_{14}(t)$，$l_{18}(t)$ 分别为点t_0，t_4，t_8处的插值基函数，将t_2，t_6代入式 (10-78) 得

$$z_1(t_2)=\frac{3}{8}z_1(t_0)+\frac{3}{4}z_1(t_4)-\frac{1}{8}z_1(t_8)$$

$$z_1(t_6)=-\frac{1}{8}z_1(t_0)+\frac{3}{4}z_1(t_4)+\frac{3}{8}z_1(t_8)$$

由式 (10-76)，$b_4(t)$，$b_6(t)$ 的可微性，以及插值的误差估计，对$t_i=t_2$，t_6，式 (10-76) 式仍成立，令

$$\begin{aligned}y_2&=y_1(t_2,\ h)+z_1(t_2)\\ y_6&=y_1(t_6,\ h)+z_1(t_6)\end{aligned} \tag{10-79}$$

由式(10-78)、(10-64)得

$$y_2 = y(t_2) + O(h^7)$$
$$y_6 = y(t_6) + O(h^7) \tag{10-80}$$

对t_i，$i = 0, 2, 4, 6, 8$令

$$z_0(t_i) = y_i - y(t_i, h) \tag{10-81}$$

由式(10-61)、(10-68)、(10-73)、(10-80)，有

$$z_0(t_i) = -a_2(t_i)h^2 - a_4(t_i)h^4 - a_6(t_i)h^6 + O(h^p) \tag{10-82}$$

其中p仍由式(10-77)确定。以t_i($i = 0, 2, 4, 6, 8$)为插值格点作拉格朗日插值多项式

$$z_0(t) = l_{00}(t)z_0(t_0) + l_{02}(t)z_0(t_2) + l_{04}(t)z_0(4) + l_{06}(t)z_0(t_6) + l_{08}(t)z_0(t_8) \tag{10-83}$$

其中$l_{0j}(t)$($j = 0, 2, 4, 6, 8$)为对应格点处的插值基函数，将$t = t_i$($i = 1, 3, 5, 7$)代入式(10-83)，得

$$\begin{pmatrix} z_0(t_1) \\ z_0(t_3) \\ z_0(t_5) \\ z_0(t_7) \end{pmatrix} = z_0(t_0)\begin{pmatrix} \frac{35}{128} \\ -\frac{5}{128} \\ \frac{3}{128} \\ -\frac{5}{128} \end{pmatrix} + z_0(t_2)\begin{pmatrix} \frac{35}{32} \\ \frac{15}{32} \\ -\frac{5}{32} \\ \frac{7}{32} \end{pmatrix}$$

$$+ z_0(t_4)\begin{pmatrix} -\frac{35}{64} \\ \frac{45}{64} \\ \frac{45}{64} \\ -\frac{35}{64} \end{pmatrix} + z_0(t_6)\begin{pmatrix} \frac{7}{32} \\ -\frac{5}{32} \\ \frac{15}{32} \\ \frac{35}{32} \end{pmatrix} + z_0(t_8)\begin{pmatrix} -\frac{5}{128} \\ \frac{3}{128} \\ -\frac{5}{128} \\ \frac{35}{128} \end{pmatrix}$$

由式(10-82)和拉格朗日插值的余项估计推得式(10-82)对$i = 0, 1, \cdots, 8$均成立，令

$$y_i = y(t_i, h) + z_0(t_i) \quad i = 1, 3, 5, 7 \tag{10-84}$$

由式(10-82)、(10-61)推得

$$y_i = y(t_i) + O(h^7) \quad i = 1, 3, 5, 7 \qquad (10\text{-}85)$$

由式(10-68)、(10-74)、(10-80)、(10-85)推得由内插过程得到的y_i($i = 0, 1, \cdots, 8$)的误差为$O(h^7)$，对$i = 0, 8$为$O(h^8)$。

类似地，若用理查森二次外插，再通过内插求得的$y(t_i)$的近似值y_i的误差为$O(h^6)$，而t_0，t_4上的误差为$O(h^8)$。若用一次理查森外插再进行内插求得的y_i的误差为$O(h^3)$，在t_0，t_2上的误差为$O(h^4)$。

10.3.4 算　　法

将10.3.1-10.3.3中的推导结果总结成下面的算法。算法由三部分组成：(1)梯形公式的并行计算；(2)理查森外插计算；(3)内插计算。对给定的步长h和t_0，令$t_i = t_0 + ih$($i = 0, 1, \cdots, N$)，其中$N = 2^3L$，为简化记号，用下标k_j表示用步长$2^j h$计算得到的格点$t_{2^j k}$上的值，相应的下标$k+1$，j表示在格点$t_{2^j(k+1)}$处的值。

算法10-1（梯形公式三次外插算法）

(1) 梯形公式的并行计算

步骤0　给出t_0处的解$y(t_0)$的近似值y_{00}，y_{01}，y_{02}，y_{03}，给出迭代初值$y_{kj}^{(0)}$($j = 0, 1, 2, 3$)，($k = 1, 2, \cdots, N/2^j$)，计算矩阵A，$(I + 2^{j-1}hA)$，$(I - 2^{j-1}hA)^{-1}$($j = 0, 1, 2, 3$)给出容许误差ε。令$l = 0$。

步1　并行计算右函数

$$y_{kj}^{(l)\prime} = f(t_{2^j k}, y_{kj}^{(l)}) \quad j = 0, 1, 2, 3$$
$$k = 1, 2, \cdots, N/2^j$$

步2　并行计算

$$d_{k+1,j} = y_{kj}^{(l)} + 2^{j-1}h[y_{kj}^{(l)\prime} + y_{k+1,j}^{(l)\prime}] - y_{k+1,j}^{(l)}$$
$$j = 0, 1, 2, 3 \quad k = 0, 1, \cdots, N/2^j - 1$$

其中约定

$$y_0^{(j)\prime}=y'_{0j},\quad y_0^{(j)}=y_{0j}$$

步 3　对 $j=0,1,2,3$　$k=0,1,\cdots,N/2^j-1$ 判别估计式

$$\|d_{k+1,j}\|\leqslant\varepsilon$$

是否均成立。若均成立，转步 7；否则转步 4。

步 4　取 $\pi_{0j}=0$，由递推式

$$\pi_{k+1,j}=[I-2^{j-1}hA]^{-1}[(I+2^{j-1}hA)\pi_{kj}+d_{k+1,j}]$$

逐次计算 π_{kj}（$j=0,1,2,3$），（$k=1,2,\cdots,N/2^j$）

步 5　令

$$y_{kj}^{(l+1)}=y_{kj}^{(l)}+\pi_{kj}\quad j=0,1,2,3$$
$$k=1,2,\cdots,N/2^j$$

步 6　令 $l=l+1$，转步1。

步 7　记

$$y(t_{2^jk},\ 2^jh)=y_{kj}^{(l)},\quad j=0,1,2,3$$

$k=0,1,\cdots,N/2^j$，转步 8

（2）理查森外插计算

步 8　计算

$$y_1(t_{2^jk},\ 2^{j-1}h)=\frac{4}{3}y\ (t_{2^jk},\ 2^{j-1}h)\ -\frac{1}{3}y\ (t_{2^jk},\ 2^jh)$$

$j=1,2,3$；　$k=0,1,\cdots,N/2^j$

步 9　计算

$$y_2(t_{2^jk},\ 2^{j-2}h)=\frac{16}{15}y_1\ (t_{2^jk},\ 2^{j-1}h)\ -\frac{1}{15}y\ (t_{2^jk},\ 2^jh)$$
$$j=2,3;\quad k=0,1,\cdots,N/2^j$$

步10　对 $k=0,1,\cdots,N/8$，计算

$$y_3(t_{8k},\ h)=\frac{64}{63}y_2(t_{8k},\ h)\ -\frac{1}{63}y_2(t_{8k},\ 2h)$$

（3）内插计算

步11　对 $k=0,1,\cdots,N/8$ 令

$$y_{8k}=y_3(t_{8k},\ h)$$

$$z_2(t_{8k})=y_{8k}-y_2(t_{8k},\ h)$$

步12　对 $k=0,1,\cdots,N/8-1$ 令

$$y_{8k+4}=y_2(t_{8k+4},\ h)+\frac{1}{2}z_2(t_{8k})+\frac{1}{2}z_2(t_{8(k+1)})$$

步13　对 $k=0,1,\cdots,N/4$ 计算

$$z_1(t_{4k})=y_{4k}-y_1(t_{4k},\ h)$$

步14　对 $k=0,1,\cdots,N/8-1$ 计算

$$y_{8k+2}=y_1(t_{8k+2},\ h)+\frac{3}{8}z_1(t_{8k})+\frac{3}{4}z_1(t_{8k+4})-\frac{1}{8}z_1(t_{8(k+1)})$$

$$y_{8k+6}=y_1(t_{8k+6},\ h)-\frac{1}{8}z_1(t_{8k})+\frac{3}{4}z_1(t_{8k+4})+\frac{3}{8}z_1(t_{8(k+1)})$$

步15　对 $k=0,1,\cdots,N/2$ 计算

$$z_0(t_{2k})=y_{2k}-y(t_{2k},\ h)$$

步16　对 $k=0,1,\cdots,N/8-1$ 计算

$$y_{8k+1}=y(t_{8k+1},\ h)+\frac{35}{128}z_0(t_{8k})+\frac{35}{32}z_0(t_{8k+2})-\frac{35}{64}z_0(t_{8k+4})+\frac{7}{32}z_0(t_{8k+6})-\frac{5}{128}z_0(t_{8(k+1)})$$

$$y_{8k+3}=y(t_{8k+3},\ h)-\frac{5}{128}z_0(t_{8k})+\frac{15}{32}z_0(t_{8k+2})+\frac{45}{64}z_0(t_{8k+4})-\frac{5}{32}z_0(t_{8k+6})+\frac{3}{128}z_0(t_{8(k+1)})$$

$$y_{8k+5}=y(t_{8k+5},\ h)+\frac{3}{128}z_0(t_{8k})-\frac{5}{32}z_0(t_{8k+2})+\frac{45}{64}z_0(t_{8k+4})+\frac{15}{32}z_0(t_{8k+6})-\frac{5}{128}z_0(t_{8(k+1)})$$

$$y_{8k+7}=y(t_{8k+7},\ h)-\frac{5}{128}z_0(t_{8k})+\frac{7}{32}z_0(t_{8k+2})$$
$$-\frac{35}{64}z_0(t_{8k+4})+\frac{35}{32}z_0(t_{8k+6})+\frac{35}{128}z_0(t_{8(k+1)})$$

步17　在区间段$[t_0,\ t_N]$中的计算停止。

上述算法是求$y(t)$在整个区间$[t_a,\ t_b]$上的近似的一个子区间段中的算法。开始时取$t_0=t_a$，$t_N=t_a+Nh$，$y_{00}=y_{01}=y_{02}=y_{03}=y_a$，由算法10-1得到$y(t)$在区间$[t_a,\ t_a+Nh]$中的格点上的近似$y_i$，接着取$t_0=t_a+Nh$，$t_N=t_a+2Nh$，重新调用算法10-1，计算子区间$[t_a+Nh,\ t_a+2Nh]$的格点上的近似值。$y_{0j}$的取值有二种处理方法，一种处理方法是取

$$y_{0j}=y(t_{N0},\ 2^jh)\quad j=0,1,2,3$$

其中t_{N0}为前一次调用算法10-1的t_N的值，这种处理方法称作整体方法；另一种处理方法称为局部方法，取

$$y_{00}=y_{01}=y_{02}=y_{03}=y_{N0}$$

其中y_{N0}指上一次调用算法10-1得到的y_N值。按这两种处理方法逐段计算，可得到$y(t)$在整个$[t_a,\ t_b]$中格点上的近似值。

由10.3.3中的结果，算法10-1得到的y_i的误差为$O(h^7)$，对$i=8k$，误差为$O(h^8)$，局部方法和整体方法达到的误差阶是相同的，但由10.3.3中的推导，实际误差主要依赖于$a_2(t)$，$a_4(t)$，$a_6(t)$，$a_8(t)$以及它们对t的导数的量值。在第一个$[t_0,\ t_N]$中它们的量级为Nh的值，若用整体方法，在第二个$[t_0,\ t_N]$中，它们的量级为$2Nh$，而对于局部方法，在第二个$[t_0,\ t_N]$中，它们的量级仍为Nh（将$O(h^p)$写成$O(h^{p-2})h^2$，于是y_0有展开式$y_0=y(t_0)+a_2(t_0)h^2+a_4(t_0)h^4$，其中$a_2(t_0)=O(h^{p-2})$，$a_4(t_0)=a_6(t_0)=a_8(t_0)=0$，将$a_2(t_0)$，$a_4(t_0)$，$a_6(t_0)$，$a_8(t_0)$作为确定它们的微分方程的初值，即可得到上面的结论）。由这种分析，可以看出局部方法的精度要好一些。

数值求解常微分方程初值问题的计算量，通常主要集中在右函数的计算中，具体到算法10-1主要集中在步1，这里共要计算

$N+\frac{N}{2}+\frac{N}{4}+\frac{N}{8}=\frac{15}{8}N$个点上的右函数值。在计算时$\frac{15}{8}N$个变量$(t_2{}^i{}_k,\ y_k^{(i)})$是已知的，因而可并行计算，其它步中的计算主要是向量运算，特别当y的维数n很大时，向量运算的长度较大，适宜在向量机上实现。

算法10-1是三次外插算法，将其中各步作适当的改变后，还可以构造一次外插，二次外插算法，当然也可以构造四次五次外插算法。但由于高次内插的精度不好，超过三次的外插算法不一定适宜。

步0中的迭代初值$y_k^{(0)}$可以用某种预估过程给出，这里不讨论。

10.3.5 算法的稳定性

对于整体方法，10.3.4中的算法将能保持梯形公式的A-稳定性，对局部方法，算法将失去A-稳定的性质。对$N=2^j$的j次外插（$j\leqslant 5$）进行了讨论，并画出了稳定区域、结果表明，当j是奇数时，算法的稳定区域是有限的，在无穷远处是不稳定的。当j为偶数时，算法的稳定区域仍是无限的，对于大多数实际问题这些稳定区域已足够大，对于$N=2^jL$的j次外插的局部算法，当L为偶数时，j次外插都具有无限的稳定区域。下面举例来说明这一点。取$N=2^1\cdot 2$的一次外插的局部算法，考虑试验方程

$$y'=\lambda y,\quad y(t_0)=y_0\quad R_e\lambda<0$$

于是可推得

$$y_{4k}=\mu(\lambda h)^k y_0\quad k=1,\ 2,\ \cdots$$

其中

$$\mu(\lambda h)=\frac{4}{3}\left(\frac{1+\frac{\lambda h}{2}}{1-\frac{\lambda h}{2}}\right)^4-\frac{1}{3}\left(\frac{1+\lambda h}{1-\lambda h}\right)^2$$

当$\lambda h\to\infty$时，$\mu(\lambda h)\to 1$，并且当λh为负实数，$|\lambda h|$充分大时$\left(|\lambda h|>\sqrt{-\frac{8}{6}}\right)$，有

$$\left(\frac{1+\frac{\lambda h}{2}}{1-\frac{\lambda h}{2}}\right)^4<\left(\frac{1+\lambda h}{1-\lambda h}\right)^2$$

因而，对充分大的这样的λh，有

$$0<\frac{4}{3}\left(\frac{1+\frac{\lambda h}{2}}{1-\frac{\lambda h}{2}}\right)^4-\frac{1}{3}\left(\frac{1+\lambda h}{1-\lambda h}\right)^2<\left(\frac{1+\frac{\lambda h}{2}}{1-\frac{\lambda h}{2}}\right)^4<1$$

所以有无限大的稳定区域。

当L为奇数时，一次、三次外插的稳定区域仍是有限的，估计会比文献〔32〕中的大，原因是经过梯形公式较多次的求解，误差要衰减，当然，L取偶数会好一些。

10.4 线性多步公式的平均算法

在反应器设计、电流分析、飞行动力学及控制系统工程等学科中，遇到的常微分方程初值问题（10-1)常常具有这样的性质，它的右函数$f(t、y)$的雅可比矩阵$\frac{\partial f(t,y)}{\partial y}$的特征值$\lambda$的实部$R_e\lambda$是负的，并且$|R_e\lambda|$的最大值或者最大值与最小值的比值相当大。这种类型的常微分方程称作刚性方程（stiff方程)。由于问题的这种特性，使得传统的数值积分方法遇到了很大的困难。对产生困难的原因可以作下面的描述，对于传统的数值积分方法，用固定步长$h>0$数值积分组（10-1）的结果通常依赖于积分方法的稳定性。为达到这种稳定性，要求选取步长h使所有的量λh均属于复平面上的一个区域中，这个区域称作稳定区域。对于大多数有效的传统数值积分方法，所对应的稳定区域是复平

面上的一个含原点的有限区域。因此，对刚性方程，为保证数值积分的稳定性，步长 h 应选得足够小。当组（10-1）的解达到稳态时，解的变化很慢，积分步长似乎可以放大，但由于计算稳定性的限制，仍应用很小的步长进行积分。这就是使用具有有限稳定区域的传统数值积分方法的步长障碍，它使得这些方法对刚性方程的效率非常低。

为了克服上述困难，希望得到具有无限稳定区域的算法。为此，达奎斯脱引进了 A-稳定性的概念，并且给出了一个限制性的结果：显式的线性多步方法（包括显式的龙格-库塔方法）不可能是 A-稳定的；A-稳定的隐式线性多步方法的阶不能超过 2；并且在所有 A-稳定的二阶方法中梯形公式具有最小的局部截断误差常数。为了突破达奎斯脱的这个结果的限制，广泛开展了刚性方程数值积分方法的研究。关于这方面的详细情形及这一节所用到的一些术语的意义见文献[24]。

为了提高适合于刚性方程的计算公式的阶，通常可通过下面几个途径进行研究：

（1）减弱 A-稳定性的定义；

（2）允许计算公式中出现二阶或更高阶的导数；

（3）利用 A-稳定的二阶公式，采用一些技巧，构造方程的解的高阶近似。

对于途径（3）的一个例子是在10.3节中讨论的梯形公式的外插方法。以不同的步长用梯形公式进行积分，然后用理查森外插得到高精度的解。按照10.3节中的处理是有效的。在10.3节所述的计算过程中，积分方法是固定的，步长作为一个可变参数。而理查森外插过程可以看成对不同参数（步长）的计算结果的某种平均过程，从而得到高阶的计算结果。在这一节中，我们将积分步长固定，而在保持 A-稳定的条件下设法在积分公式中引进变参数，然后以固定的步长而用不同的参数的公式进行积分，最后将积分结果进行平均来得到高阶的计算结果。这就是下面要构造的平均算法的设计思想。计算时，由于各公式仅参数不同，右

函数和方法本身均可并行计算。

设组（10-1）的精确解$y(t)$是区间$[t_0, T]$上的无限连续的可微向量值函数，$t_0=0$，格点集$\{t_m\}$是t的一组离散值，有$t_m=t_0+mh$，其中$h>0$称作步长。考虑亚当斯型线性多步公式

$$y_{m+1}=y_m+h\left[c\dot{y}_{m+1}+(1-c)\dot{y}_m+\sum_{j=1}^{K-1}(b_j^*-c)\nabla^j\dot{y}_m\right] \tag{10-86}$$

其中∇^j表示第j阶向后差分，$\nabla y_j=y_j-y_{j-1}$，b_j^*和c是未定常数（称c为"第二参数"），"$\cdot$"表示对t的导数。

对式（10-86），我们引进一个算子

$$\Omega^*=\Delta-D\left[c\Delta+I+\sum_{j=1}^{K-1}(b_j^*-c)\nabla^j\right] \tag{10-87}$$

其中Δ是向前差分算子，$D=h(d/dt)$，形式为式(10-87)的算子称作是亚当斯型算子。将Ω^*中的差分展开，可以写成更熟悉的形式，适于进行稳定性分析。

用下面的方式来定义Ω^*的精度：将$y(t)$及其差分在$t=t_m$处展开，并用记号$\simeq$表示具有误差$O(h^k)$的相等，其中k为Ω^*中的步数，于是有

$$\Delta\simeq\sum_{i=1}^{K}(i!)^{-1}D^i \tag{10-88}$$

$$\nabla^j\simeq\sum_{i=j}^{K}\gamma_{ij}D^i\qquad j=1,\cdots,k-1 \tag{10-89}$$

其中γ_{ij}是形式幂级数的形式乘法

$$\left(D-\frac{1}{2!}D^2+\frac{1}{3!}D^3\cdots\right)^j=D^j+\cdots \tag{10-90}$$

得到的右端项的系数，将式（10-88）、（10-89）代入式（10-87）得到

$$\Omega^*\simeq\sum_{i=2}^{K}\nu_iD^i \tag{10-91}$$

其中

$$\nu_i=(i!)^{-1}-\sum_{j=1}^{i-1}\gamma_{i-1,j}b_j^* \quad 2\leqslant i\leqslant k \tag{10-92}$$

注意：如果第二参数 c 如式（10-87）引进，则ν_i与b_j^*之间的关系式（10-92）与 c 无关。由式（10-90），对所有 i 有 $\gamma_{ii}=1$。另外，由于由式（10-92）确定的系数矩阵是三角形的，对于任意给定的ν_i的组，可以唯一地解出b_j^*，若在式（10-91）中 $\nu_2=\nu_3=\cdots=\nu_p=0$，则称由式（10-91）定义的算子 Ω^*的误差为$O(h^{p+1})$，并称它有 p 阶精度。对任意的 p，$2\leqslant p\leqslant k$，我们总可以选取常数b_j^*，使有$\nu_i=O(i=2,\cdots,p)$，这时Ω^* 有 p 阶精度。

令b_j为使式（10-92）给出的所有ν_i均为O的b_j^*的解，这时Ω^*的阶 $p=k$。但是当 $p>2$ 时，这个算子对应的方法不可能是 A-稳定的，用它来解刚性方程是不合适的。现在我们考虑另外一种应用算子Ω^*的方式，同样能得到$p=k$阶的方法的效果。

将式（10-86）中的常数b_j^*取成

$$b_j^*=b_j \quad j=1,\cdots,\ p-1 \tag{10-93}$$

$$b_j^*=b_j+u_{j-p+1} \quad j=p,\ p+1,\ \cdots,\ K-1$$

其中 u_1，u_2，$\cdots$，u_d 是任意常数 $d=K-p$。容易看出，这时Ω^*有

$$\Omega^*\simeq D^p\sum_{i=1}^{d}\mu_iD^i \tag{10-94}$$

其中

$$\mu_i=-\sum_{j=1}^{i}\gamma_{p-1+i,p-1+j}u_j \quad i=1,\ 2,\ \cdots,\ d \tag{10-95}$$

因此对u_1，u_2，$\cdots$，u_d的任何组，Ω^*的阶$\geqslant p$，每一组参数组成一个 d 维向量 $u=(u_1,\ u_2,\ \cdots,\ u_d)^T$。考虑 $d+1$ 个向量 $u_\rho=(u_{1\rho},\ u_{2\rho},\ \cdots,\ u_{d\rho})^T$，对于每个 ρ，取式（10-93）中的参数$u_j=u_{j\rho}$，可得到相应于u_ρ 的算子Ω^* 及方法（10-86）。将这样得到的算子记成Ω_ρ，对于给定的 $d+1$ 个向量 u_ρ 及任意 $d+1$ 个数 ν_ρ

（$\rho=1,\ 2,\ \cdots,\ d+1$）作算子

$$\Omega=\sum_{\rho=1}^{d+1} v_\rho \Omega_\rho \tag{10-96}$$

Ω可以看成是算子Ω_ρ（$\rho=1,\ \cdots,\ d+1$）以权v_ρ的平均，下面我们来说明如何选取权v_ρ，使算子Ω的阶为$P=K$，由式(10-94)，我们得到

$$\Omega\simeq D^p \sum_{i=1}^{d} \pi_i D^i \tag{10-97}$$

其中

$$\pi_i=\sum_{\rho=1}^{d+1} v_\rho u_{i,\rho} \tag{10-98}$$

由此可以看到，为了Ω具有阶K，必须有

$$\pi_i=0 \quad i=1,\ 2,\ \cdots,\ d \tag{10-99}$$

通过简单的计算，容易证明式（10-99)与

$$\sum_{\rho=1}^{d+1} u_{i\rho}=0 \quad i=1,\ \cdots,\ d \tag{10-100}$$

是等价的，对v_ρ再加上正则化条件

$$\sum_{\rho=1}^{d+1} v_\rho=1 \tag{10-101}$$

这样，如果v_ρ满足式(10-100)、(10-101)，则与算子Ω相应的算法（10-86)将具有阶K。

容易看出，式（10-100)、(10-101）对v_ρ（$\rho=1,\ \cdots,d+1$）的唯一可解性等价于d个向量$u_\rho-u_{d+1}$（$\rho=1,\ 2,\cdots,d$）可展成d维线性空间，对于$d=2$的特殊情形，唯一可解性条件是以二维向量u_1，u_2，u_3为顶点组成一个非退化三角形。

上述的算法可以按下面的方式在并行处理计算机上实现。按前面叙述的条件选取参数向量u_ρ和权系数v_ρ（$\rho=1,\cdots,d+1$)。对于每一个ρ，利用参数向量u_ρ，得到Ω_ρ及相应的公式(10-86)。

由公式（10-86）解出$y_{m+1,\rho}$（$\rho = 1, 2, \cdots, d+1$）。令

$$y_{m+1} = \sum_{\rho=1}^{d+1} v_\rho y_{m+1,\rho} \tag{10-102}$$

这种计算可由图10-3给出的示意框图来说明。

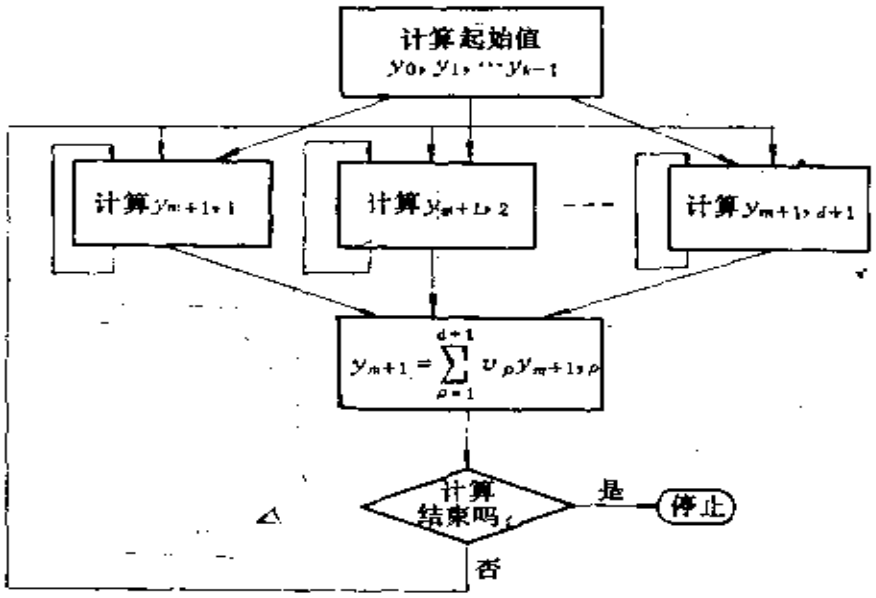


图10-3　平均算法示意框图

由方法（10-86）的形状可以看出，只要它的阶$p \geqslant 1$，则一定有零稳定性，下面我们主要考虑方法的A-稳定性。另外，在一般情形下，方法（10-86）的A-稳定性问题还未彻底解决，所以我们只对几个具体方法来详细讨论这个问题。

林尼格（Liniger）在文献[25]中对线性多步方法给出了一个A-稳定性的充分条件，我们将其作一些明显的改变后，可得到方法（10-86）为A-稳定的充分条件。

将方法（10-86）应用到试验方程

$$\frac{dy}{dt} = \lambda y \quad R_e\lambda < 0 \tag{10-103}$$

我们得到特征方程

$$\rho(\zeta) - \lambda h\sigma(\zeta) = 0 \tag{10-104}$$

其中多项式$\rho(\xi)$和$\sigma(\xi)$可表成

$$\rho(\xi)=\sum_{j=0}^{K} d_j\xi^j \tag{10-105}$$

$$\sigma(\xi)=\sum_{j=0}^{K} \beta_j\xi^j \tag{10-106}$$

对方法 (10-86)有

$$\rho(\xi)=\xi^k-\xi^{k-1} \tag{10-107}$$

令$\sigma_i(i=1,2,\cdots,k)$是多项式$\sigma(\xi)$的零点，构造多项式

$$p_k(\xi)=\sum_{j=0}^{K} \gamma_j T_j(\xi) \tag{10-108}$$

其中$T_j(\xi)$是契比雪夫多项式

$$T_0(\xi)=1$$
$$T_1(\xi)=\xi$$
$$T_2(\xi)=2\xi^2-1$$
$$\cdots$$

而系数γ_j定义成

$$\gamma_0=\sum_{j=0}^{K} \alpha_j\beta_j$$

$$\gamma_j=\sum_{l=0}^{K-j} (\alpha_{l+j}\beta_l+\alpha_l\beta_{j+l}) \quad 1\leqslant j\leqslant k$$

则有下面的定理。

定理10-2 如果条件

$$|\sigma_i|<1 \quad i=1,\cdots,k \tag{10-109}$$

和

$$p_k(\xi)\geqslant 0 \quad -1\leqslant\xi\leqslant 1 \tag{10-110}$$

成立，则方法 (10-86) 是A-稳定的。

〔例10-1〕 方法A_1

考虑由下面的算子

$$\Omega_1^*=\Delta-D\left[c\Delta+I+\left(\frac{1}{2}-c+r\right)\nabla\right]$$

所对应的方法，应用到标量试验方程，得到的特征方程(10-104)中的$\rho(\zeta)$和$\sigma(\zeta)$为

$$\rho(\zeta)=\zeta^2-\zeta$$

$$\sigma(\zeta)=c\zeta^2+\left(\frac{3}{2}-2c+r\right)\zeta-\left(\frac{1}{2}-c+r\right)$$

(10-111)

作变换

$$\zeta=\frac{z+1}{z-1}$$

并令

$$S(z)=\left(\frac{z-1}{2}\right)^2\sigma(\zeta)$$

$$=\frac{1}{4}[z^2+(1+2r)z+(-2+4c-2r)]$$

为满足条件（10-109），即$\sigma(\zeta)$的零点均在单位圆内部，等价于要求$S(z)$的零点均在左半平面，即其实部均是负的。由路斯(Routh)准则，这又等价于

$$1+2r>0$$

(10-112)

$$1-2c+r<0$$

由式(10-111)，式（10-105)、(10-106）中的系数α_j，β_j为

$$\alpha_0=0 \qquad \beta_0=-\left(\frac{1}{2}-c+r\right)$$

$$\alpha_1=-1 \qquad \beta_1=\left(\frac{3}{2}-2c+r\right)$$

$$\alpha_2=1 \qquad \beta_2=c$$

于是$p_2(\xi)$中的γ_j为

$$\gamma_0=-\frac{3}{2}+3c-r$$

$$\gamma_1=2-4c+2r$$

$$\gamma_2 = -\left(-\frac{1}{2} - c + r\right)$$

对应于方法A_1的多项式$p_2(\xi)$为

$$p_2(\xi) = (\xi - 1)[(-1 + 2c - 2r)\xi + (1 - 2c)] \tag{10-113}$$

因此，要求条件（10-110）成立等价于要求成立

$$(-1 + 2c - 2r)\xi + (1 - 2c) \leqslant 0 \quad -1 \leqslant \xi \leqslant 1 \tag{10-114}$$

令$\xi = 1$和$\xi = -1$，立即推得r、c应满足的条件

$$-1 + 2c > r \geqslant 0 \tag{10-115}$$

满足条件（10-115）的点（r，c）组成一个楔形G_1(见图10-4)。

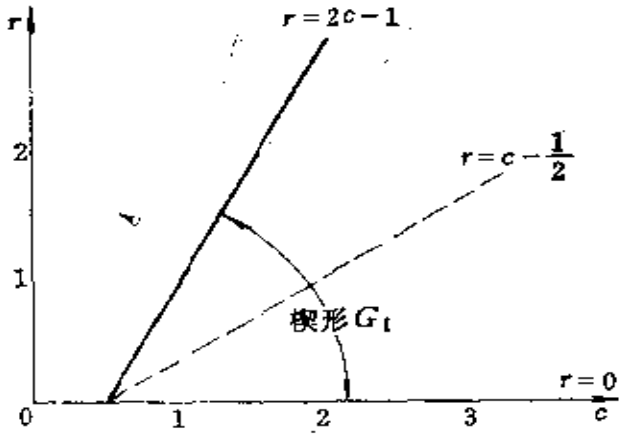


图10-4　方法A_1满足条件（10-115）的（r，c）的区域

容易验证，除边界$r = 2c - 1$，$r = 0$及$r = c - \frac{1}{2}$的点外，楔形G_1中的点还满足条件（10-112)。因此上述区域中的所有点（r，c）所对应的方法均是A-稳定的。

方法A_1的阶$p = 1$，$h = 2$，通过选取（r，c）的值进行并行计算和平均，可提高计算结果的精度阶。

〔例10-2〕　方法A_2

考虑由算子

$$\Omega_2^*=\Delta-D\left[c\Delta+I+\left(-\frac{1}{2}-c\right)\nabla+\left(\frac{5}{12}-c+r\right)\nabla^2\right]$$

(10-116)

确定的方法 (10-86)，这个方法的阶为 $p=2$，$k=3$，它所对应的多项式 $\rho(\zeta)$ 和 $\sigma(\zeta)$ 为

$$\rho(\zeta)=\zeta^3-\zeta^2$$

(10-117)

$$\sigma(\zeta)=c\zeta^3+\left(\frac{23}{12}-3c+r\right)\zeta^2-\left(\frac{8}{6}-3c+2r\right)\zeta+\left(\frac{5}{12}-c+r\right)$$

作变换

$$\zeta=\frac{z+1}{z-1}$$

并令

$$S(z)=\left(\frac{z-1}{2}\right)^3\quad\sigma(\zeta)=-\frac{1}{24}\left[3z^3+6z^2+(2+12r)z+(-11+24c-12r)\right]$$

(10-118)

应用路斯准则，要求 $S(z)$ 的根均在左半平面等价于要求 r 满足不等式

$$-\frac{5}{12}+\frac{2}{3}c<r<-\frac{11}{12}+2c \qquad (10\text{-}119)$$

对应于方法 A_2 的 $p_3(\xi)$ 为

$$p_3(\xi)=(\xi-1)^2\left[\left(\frac{5}{3}-4c+4r\right)\xi-\left(\frac{1}{6}-2r\right)\right]$$

(10-120)

因此要求对 $-1\leqslant\xi\leqslant1$ 有 $p_3(\xi)\geqslant0$ 等价于要求有

$$\left(\frac{5}{3}-4c+4r\right)\xi-\left(\frac{1}{6}-2r\right)\geqslant0\quad-1\leqslant\xi\leqslant1$$

取 $\xi=1$，$\xi=-1$ 立即推得不等式

$$2c-\frac{11}{12}\geqslant r\geqslant\frac{2}{3}c-\frac{1}{4} \tag{10-121}$$

满足式（10-121）的点（r，c）组成图10-3中的楔形G_2，与式(10-119)相比较，可知对于除去上界$r=2c-\frac{11}{12}$后的楔形G_2中的所有点（r，c）(见图10-5)，方法A_2都是A-稳定的。

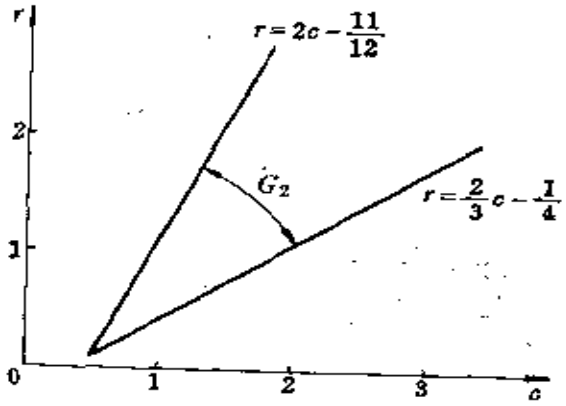


图10-5　方法A_2满足条件（10-109）（10-110）的（r，c）的区域

〔例10-3〕　方法A_3

考虑由算子

$$\Omega_3^*=\Delta-D\left[c\Delta+I+\left(-\frac{1}{2}-c\right)\nabla+\left(\frac{5}{12}-c+r\right)\nabla^2+\left(\frac{3}{8}-c+s\right)\nabla^3\right] \tag{10-122}$$

确定的方法（10-86），它的阶$p=2$，$k=4$，它对应的多项式$\rho(\zeta)$和$\sigma(\zeta)$为

$$\rho(\zeta)=\zeta^4-\zeta^3$$

$$\sigma(\zeta)=c\zeta^4+\left(\frac{55}{24}-4c+r+s\right)\zeta^3-\left(\frac{59}{24}-6c+2r+3s\right)\zeta^2+\left(\frac{37}{24}-4c+r+3s\right)\zeta-\left(\frac{3}{8}-c+s\right) \tag{10-123}$$

作变换

$$s=\frac{z+1}{z-1}$$

并令

$$s(z)=\left(\frac{z-1}{2}\right)^4 \quad \sigma(\xi)=\frac{1}{48}[3z^4+9z^3+(8+12r)z^2$$
$$+24sz+(-20+48c-12r-24s)] \qquad (10\text{-}124)$$

应用路斯准则，要求$S(z)$的根均在左半平面内等价于要求成立不等式组

$$c_0=2+3r-2s>0$$
$$d_0=\frac{(15-36c+9r+34s+24rs-16s^2)}{2+3r-2s}>0$$
$$e_0=-5+12c-3r-6s>0 \qquad (10\text{-}125)$$

对于任意的$c\neq 0$和$c_0\neq 0$，$d_0=0$表示一非退化的双曲线，其中心在$\left(-\frac{23}{12},\ -\frac{3}{8}\right)$处，渐近线为$As_1$：$s=-\frac{3}{8}$和$As_2$：$s=\frac{3}{2}r+\frac{5}{2}$。$As_2$与$c_0=0$是平行的，这双曲线与$c_0=0$和$e_0=0$交在点$p_2=\left(-\frac{11}{12}+c,\ -\frac{3}{8}+\frac{3}{2}c\right)$，和$e_0=0$的另一个交点为$p_1=\left(-\frac{5}{3}+4c,\ 0\right)$，对于$c=\frac{1}{4}$，这两个交点将重合。如果$c>0$，双曲线$d_0=0$的另一支（位于$s<-\frac{3}{8}$中）可以不考虑，因为它所构成的区域与$c_0>0$不相交。下面还将证明，如果$c<\frac{1}{2}$，则条件(10-110)不满足（见图10-6），因此我们仅限于讨论$c\geqslant\frac{1}{2}$的情形。

对应于方法A_3的多项式$p_4(\xi)$为

$$p_4(\xi)=(\xi-1)^2Q(\xi) \qquad (10\text{-}126)$$

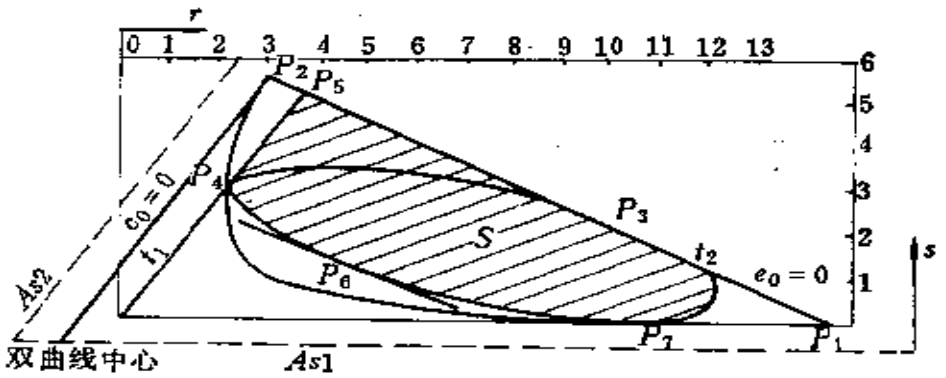


图10-6　方法A_3满足条件（10-109)、(10-110）的区域

其中

$$Q(\xi)=(-9+24c-24s)\xi^2+(5-12c+12r)\xi$$
$$+(4-12c+6r+12s) \qquad (10-127)$$

因此要求当$-1\leqslant\xi\leqslant1$时成立$p_4(\xi)\geqslant1$等价于要求当$-1\leqslant\xi\leqslant1$时有

$$Q(\xi)\geqslant0$$

对于固定的$c>0$和固定的ξ，$-1\leqslant\xi\leqslant1$，$Q(\xi)=0$为参数r和s平面上的一条直线，当ξ在$-1\leqslant\xi\leqslant1$中变化时，这些直线的包络为椭圆。

$$6r^2+24rs+48s^2+(14-36c)r+(34-96c)s$$
$$+\frac{1}{24}(13-36c)^2=0 \qquad (10-128)$$

位于点p_4和p_3之间的部分（见图10-6）$p_4=\left(-\frac{13}{60}+\frac{3}{5}c,\ -\frac{13}{40}+\frac{9}{10}c\right)$，它对应于$\xi=+1$的直线$Q(1)=0$；$s=3r/2$与椭圆(10-128)的切点，该直线记成$t_1$，$p_3=\left(-\frac{17}{12}+3c,\ -\frac{1}{8}+\frac{1}{2}c\right)$是对应于$\xi=-1$的直线$Q(-1)=0$与椭圆(10-128）的切点，这直线记作$t_2$

由式（10-127）和

$$Q(0)\geqslant 0$$

解得

$$s\geqslant -\frac{1}{2}r+c-\frac{1}{3} \tag{10-129}$$

而由

$$Q(-1)\geqslant 0$$

解得

$$s\leqslant -\frac{5}{6}+2c-\frac{1}{2}r \tag{10-130}$$

为保证式（10-129）和式（10-130）是相容的，必须有

$$c-\frac{1}{3}\leqslant -\frac{5}{6}+2c$$

因此，仅当 $c\geqslant -\frac{1}{2}$ 时式（10-129）和（10-130）是相容的。

由上面的分析可知（参看图10-6），在以 p_3 和 p_4 间的椭圆的下面部分，直线 t_1 和直线 t_2 所围成的闭集 $\overline{S}$ 上，$Q(\xi)\geqslant 0$，因而条件（10-110）满足。另外，这里指出，当 c 变化时，椭球心 $\left(-\frac{11}{12}+2c,-\frac{1}{8}+\frac{1}{2}c\right)$ 在过双曲线 $d_0=0$ 的心的直线 $s=-\frac{1}{4}r+\frac{5}{48}$ 上，与 c 无关。正 r 轴与椭圆的主轴的夹角是 $-14^\circ 50'$。

现在我们证明在将 $\overline{S}$ 的直线 t_2 上的闭边界部分〔p_3，p_6〕去掉后的开闭集合 S 上，方法 A_3 是 A-稳定的，这只要证明在 S 上不等式（10-129）均成立。首先，不等式 $Q(-1)>0$ 与 $e_0>0$ 定义同样的开半平面（在切线 t_2 的下半部分），包含所有的 S。其次，S 的所有点在以 t_1 为界的半平面 $S\geqslant -\frac{3}{2}r$ 中，因此更满足较弱的不等式 $c_0>0$。第三，为了看出 S 的所有点在不等式 $d_0>0$ 所确定的区域中，只要证明如果 p_3 在 $d_0>0$ 中，S 的边界与双曲线

不相交。由于$p_5\left(-\frac{5}{12}+c,\ -\frac{5}{8}+\frac{3}{2}c\right)$是$l_1$和$l_2$的交点，容易证明区间$[p_3,\ p_5]$严格地位于$l_2$与双曲线二个交点$p_1$与$p_2$之间。因此$[p_3,\ p_5]$全部在区域$d_0>0$中，类似地$p_4$和$p_5$均在$l_1$与双曲线的单个交点的（$d_0>0$）一边，因而整个区间$[p_4,\ p_5]$也在区域$d_0>0$中。最后，为了说明$S$的边界的椭圆部分与双曲线不交，我们证明这对整个椭圆均成立。由式(10-128)和$d_0=0$消去r，并令$s=3\sigma/4$，则椭圆和双曲线的交点一定对应于σ的方程

$$(72\sigma^2-24\sigma+2)c^2+(-120\sigma^3-28\sigma^2+14\sigma-1)c$$
$$+\left(50\sigma^4+40\sigma^3+3\sigma^2-2\sigma+\frac{1}{8}\right)=0 \qquad (10\text{-}131)$$

的实根，但是方程（10-131）对c的判别式等于$-216\sigma^2$。因此，除非$\sigma=0$，对任何实值σ，任何c的实值均不满足方程(10-131)。这等价于对任何实值c，方程（10-131）对σ无实解，除非$\sigma=0$，但$\sigma=0$对应于$c=-\frac{1}{4}$，它是不考虑的值，这就是我们要证明的。

下面我们来描述由算法A_3得到的一个计算过程。在方法A中取$c=4$，$(r,\ s)$取三组值$(r_1,s_1)=(7,\ 2)$，$(r_2,\ s_2)=(5,\ 2)$，$(r_3,\ s_3)=(7,\ 1)$这三组值均在图10-6的集合S内，然后选$v_1,\ v_2,\ v_3$满足

$$v_1+v_2+v_3=1$$

并且使算子

$$\Omega_{3,}=\Delta-D\Big[4\Delta+I-\frac{7}{2}\nabla+\Big(-\frac{43}{12}+7v_1+5v_2$$
$$+7v_3\Big)\nabla^2+\Big(-\frac{29}{8}+2v_1+2v_2+v_3\Big)\nabla^3\Big]$$

所对应的方法的局部截断误差为$O(h^5)$，由这二个条件选得的v_1，v_2，v_3的值为

$$v_1=-4.5,\ v_2=3.5,\ v_3=2$$

将这三组参数$(r_i,\ s_i)$相应的算法A_3写成通常的形式，得到

$$y_{i+1}=y_i+h\beta_1 f_{i+1}+h(\beta_0 f_i+\beta_{-1}f_{i-1}+\beta_{-2}f_{i-2}+\beta_{-3}f_{i-3})$$

其中系数为

$$\beta_1=4$$

$$\beta_{01}=-\frac{113}{24} \qquad \beta_{02}=-\frac{161}{24} \qquad \beta_{03}=-\frac{137}{24}$$

$$\beta_{-11}=\frac{37}{24} \qquad \beta_{-12}=\frac{133}{24} \qquad \beta_{-13}=\frac{109}{24}$$

$$\beta_{-21}=-\frac{35}{24} \qquad \beta_{-22}=-\frac{83}{24} \qquad \beta_{-23}=-\frac{107}{24}$$

$$\beta_{-31}=\frac{39}{24} \qquad \beta_{-32}=-\frac{39}{24} \qquad \beta_{-33}=\frac{63}{24}$$

现在来叙述用算法A_3的平均算法的主要计算步骤。

设已得到组(10-1)在格点$t_i=t_0+ih$上的解$y(t_i)$的三组近似值$y_{i,1}$，$y_{i,2}$，$y_{i,3}$，并且假定已得到组(10-1)在格点t_i，t_{i-1}，t_{i-2}，t_{i-3}上的相应的三组右函数的近似值。

$$\begin{matrix} f_{01} & f_{-11} & f_{-21} & f_{-31} \\ f_{02} & f_{-12} & f_{-22} & f_{-32} \\ f_{03} & f_{-13} & f_{-23} & f_{-33} \end{matrix}$$

作向量

$$Y_i=\begin{bmatrix} y_{i,1} \\ y_{i,2} \\ y_{i,3} \end{bmatrix} \qquad F_0=\begin{bmatrix} f_{01} \\ f_{02} \\ f_{03} \end{bmatrix} \qquad F_{-1}=\begin{bmatrix} f_{-11} \\ f_{-12} \\ f_{-13} \end{bmatrix}$$

$$F_{-2}=\begin{bmatrix} f_{-21} \\ f_{-22} \\ f_{-23} \end{bmatrix} \qquad F_{-3}=\begin{bmatrix} f_{-31} \\ f_{-32} \\ f_{-33} \end{bmatrix}$$

应用下述计算得到组(10-1)在格点$t_{i+1}=t_0+(i+1)h$上的解$y(t_{i+1})$及其右函数的三组近似值

$$Y_{i+1}=\begin{bmatrix} y_{i+1,1} \\ y_{i+1,2} \\ y_{i+1,3} \end{bmatrix} \qquad F_1=\begin{bmatrix} f_{11} \\ f_{12} \\ f_{13} \end{bmatrix}$$

步 1　计算

$$\tilde{Y}_{i+1}=Y_i + h\left[\begin{bmatrix}\beta_{01}f_{01}\\ \beta_{02}f_{02}\\ \beta_{03}f_{03}\end{bmatrix}+\begin{bmatrix}\beta_{-11}f_{-11}\\ \beta_{-12}f_{-12}\\ \beta_{-13}f_{-13}\end{bmatrix}+\begin{bmatrix}\beta_{-21}f_{-21}\\ \beta_{-22}f_{-22}\\ \beta_{-23}f_{-23}\end{bmatrix}+\begin{bmatrix}\beta_{-31}f_{-31}\\ \beta_{-32}f_{-32}\\ \beta_{-33}f_{-33}\end{bmatrix}\right]$$

步 2　计算Y_{i+1} 的迭代初值

$$Y_{i+1}^{(k)}=\tilde{Y}_{i+1}+h\beta_1(3F_0-3F_{-1}+F_{-2})$$

并令 $k=0$

步 3　并行计算右函数

$$f(t_{i+1},\ y_{i+1,1}^{(k)}) \qquad f(t_{i+1},\ y_{i+1,2}^{(k)}) \qquad f(t_{i+1},\ y_{i+1,3}^{(k)})$$

并组成向量

$$F_1^{(k)}=\begin{bmatrix} f(t_{i+1},\ y_{i+1,1}^{(k)})\\ f(t_{i+1},\ y_{i+1,2}^{(k)})\\ f(t_{i+1},\ y_{i+1,3}^{(k)})\end{bmatrix}$$

步 4　计算偏差量

$$\Delta=Y_{i+1}^{(k)}-\tilde{Y}_{i+1}-h\beta_1F_1^{(k)}$$

步 5　按分量检验偏差量Δ是否满足精度要求，若满足，转步 8；否则转步 6。

步 6　计算

$$Y_{i+1}^{(k+1)}=Y_{i+1}^{(k)}-A_{i+1}^{(k)}\Delta$$

其中

$$A_{i+1}^{(k)}=\begin{bmatrix} A & 0 & 0\\ 0 & B & 0\\ 0 & 0 & C\end{bmatrix}$$

$$A=\left[\ I-h\beta_1\frac{\partial f(t_{i+1},\ y_{i+1,1}^{(k)})}{\partial y}\right]^{-1}$$

$$B=\left[\ I-h\beta_1\frac{\partial f(t_{i+1},\ y_{i+1,2}^{(k)})}{\partial y}\right]^{-1}$$

$$C=\left[\ I-h\beta_1\frac{\partial f(t_{i+1},\ y_{i+1,3}^{(k)})}{\partial y}\right]^{-1}$$

I 是单位矩阵，为了节省计算量，$A_i^{(k)}$ 中的矩阵$\frac{\partial f}{\partial y}$不必每步重新计算，并且在$A_i^{(k)}$中的三个$\frac{\partial f}{\partial y}$可采用同一个矩阵。

步 7　令 $k = k + 1$，转步 3

步 8　计算

$$Y_{i+1} = Y_i^{(k)}, \quad F_1 = F_1^{(k)}$$

其分量仍分成三组，分别记为$y_{i+1,1}$，$y_{i+1,2}$，$y_{i+1,3}$和f_{11}，f_{12}，f_{13}。

得到Y_{i+1}和F_1后作平均得到$y(t_{i+1})$及其导数的近似值 y_{i+1}和Dy_{i+1}。

$$y_{i+1} = v_1 y_{i+1,1} + v_2 y_{i+1,2} + v_3 y_{i+1,3}$$

$$Dy_{i+1} = v_1 f_{11} + v_2 f_{12} + v_3 f_{13}$$

然后作传送

$$F_{-2} \Rightarrow F_{-3} \quad F_{-1} \Rightarrow F_{-2} \quad F_0 \Rightarrow F_{-1} \quad F_1 \Rightarrow F_0$$

令 $i = i + 1$，继续进行计算。

由上面的计算过程可以看到，每积分一步必须进行迭代，并且在迭代过程中必须计算右函数 $f(t, y)$的雅可比矩阵及其有关的逆矩阵。由此可以想到，能否将这雅可比矩阵直接加到数值积分公式中去，构造更适合于求解的公式。这样，在文献[24]中我们构造了线性隐式的变系数多步方法

$$y_{i+1} = y_i + chJ_i\left[Ay_i - \sum_{j=1}^{k-1} \nabla^j y_i \right] - h\left[f_i + \sum_{j=1}^{k-1} b_j^* \nabla^j f_i \right] \tag{10-132}$$

其中J_i是雅可比矩阵$\partial f(t, y)/\partial y$在$(t_i, y_i)$处的近似值。$c$ 和b_j^*可类似于式(10-87)进行选取。由式(10-132)可得

$$y_{i+1} = (I - chJ_i)^{-1}\left[y_i - cI_i \sum_{i=1}^{k-1} \nabla^j y_i + h\left(f_i + \sum_{j=1}^{k-1} b_j^* \nabla^j f_i \right) \right] \tag{10-133}$$

因而不需要迭代。对方法(10-132) 的分析可象对方法(10-87)的分析一样进行。

10.5 插值分解算法

许多大系统的运动可以用大型的常微分方程组的初值问题(10-1) 来描述。这时组 (10-1) 的阶可以非常高，而且y的各个分量所描述的量的变化特性可以是非常不同的，从而使组(10-1) 常常表现为大型的刚性方程组，对数值求解的方法和步长的选取提出非常苛刻的要求。

文献〔25〕提出将两种方法联合应用求解一个常微分方程组的思想。这种思想进一步拓展，将组 (10-1) 分解成一定程度上可以独立求解的子组，分别进行求解。在求解时用到另外一个子组中的量时，可以通过插值得到。各个子组所用的数值积分方法和积分步长可以不同，这为方法和步长的选取提供了一定的灵活性，应用这种思想，还可以将复杂的右函数进行分块计算。根据这种思想构造的算法将特别适合在 MIMD 计算机系统上运行。

本节的算法针对多处理机系统进行构造，假设该系统由p台处理机和一个共享存储器组成，该系统的每台处理机具有自己的存储器 (局部存储器)，可根据处理机中的指令流运行，并通过共享存储器与其它处理机交换信息。对于这种并行计算机系统，允许在属于它的各个处理机上独立地执行多个程序。这些程序可以是不同的，在这种并行计算机系统上，这一节构造的算法将能使问题 (10-1) 的数值求解达到较高的并行性。

在 10.5.1 节中，我们对简化的插值分解算法的模型给出一些理论结果。在该简化模型中，组 (10-1) 只分解成两个子组，对它们的数值积分公式均是显式单步公式，而数值积分步长都是常数，而且其中一个步长是另外一个步长的整数倍。10.5.1 节中的结果对于较一般的情形 (如 10.5.2 节和 10.5.3 节中给出的算法) 也成立，但需要更复杂的证明[28]。在 10.5.2 节和 10.5.3 节中，我们将分别给出异步并行迭代算法和并行向前算法，同时

给出相应的数值例子。

10.5.1 插值分解方法的一些理论结果

这一节我们将给出插值分解方法的收敛性和收敛阶。为简单起见，只考虑将组（10-1）分解成两个子组的情形，类似地可以处理分解成更多个子组，设组（10-1）可以分解成下面的两个组

$$\frac{dz(t)}{dt}=g(t,\ z(t),\ w(t))\quad z(0)=z_0 \tag{10-134a}$$

$$\frac{dW(t)}{dt}=h(t,\ z(t),\ W(t))\quad W(0)=W_0 \tag{10-134b}$$

即将向量 y 的所有分量进行适当归并分成两个子向量 z 和 W，相应地将式（10.1）的右函数 $f(t,y)$ 也进行归并，得到组（10-134），设 z 和 W 的维数分别是 n_1 和 n_2，$n=n_1+n_2$。这一节应用的向量模为最大模。

对式（10-134）作下面的假定。

假定10-1 函数 $g(t,\ z,\ w)$ 和 $h(t,\ z,\ w)$ 在所考虑的区间上对 t 连续，而对 z 和 w 满足李普希兹条件，即存在正常数 L_{g_1}，L_{g_2}，L_{h_1}，L_{h_2} 对任意的 $z^{(1)}$，$z^{(2)}\in E^{n_1}$ 和 $w^{(1)}$，$w^{(2)}\in E^{n_2}$，有

$$\|g(t,\ z^{(1)},\ w^{(1)})-g(t,\ z^{(2)},w^{(1)})\|\leqslant L_{g_1}\|z^{(1)}-z^{(2)}\|$$

$$\|g(t,\ z^{(1)},\ w^{(1)})-g(t,\ z^{(1)},\ w^{(2)})\|\leqslant L_{g_2}\|w^{(1)}-w^{(2)}\|$$

$$\|h(t,\ z^{(1)},\ w^{(1)})-h(t,\ z^{(2)},\ w^{(1)})\|\leqslant L_{h_1}\|z^{(1)}-z^{(2)}\|$$

$$\|h(t,\ z^{(1)},\ w^{(1)})-h(t,\ z^{(1)},\ w^{(2)})\|\leqslant L_{h_2}\|w^{(1)}-w^{(2)}\|$$

在这些假定下，组（10-134）在所考虑的区间上有唯一的连

续可微解，记其解为 $\tilde{z}(t)$，$\tilde{W}(t)$。

现在来描述数值求解组（10-134）的一个插值分解算法的模型。用点 $t_i=ih_1(i=0,1,\cdots)$ 将积分区间离散化。用点 $t_{ij}=t_{i-1}+jh_2(j=0,1,\cdots,M)$ 将区间$[t_{i-1},t_i]$离散化。$t_{iM}=t_i$，这要求步长 h_1是步长 h_2 的整数倍。点 t_i 和 t_{ij} 称作离散化格点。为简单起见，只考虑定步长，设已得到序列 z_k，$dz_k(k=0,1,\cdots,i)$ 和 w_{ij}，$dw_{ij}(l=1,2,\cdots,i)$，$(j=0,1,\cdots,M)$ 并记 $w_i=w_{iM}$，其中 z_k、dz_k 分别是 $z(t_k)$、$z'(t_k)$ 的近似值，w_{ij}、dw_{ij} 分别是 $w(t_{ij})$、$w'(t_{ij})$ 的近似，这里我们取$dz_k=g(t_k,z_k,w_k)$，$dw_{ij}=h(t_{ij},\tilde{z}_i(t_{ij}),w_{ij})$。其中量 $\tilde{z}(t)$ 在下面定义。记

$$
\begin{aligned}
&T_{zi}=(t_0,t_1,\cdots,t_i)^T,\\
&T_{wi}=(t_{10},t_{11},\cdots,t_{1M},t_{21},\cdots,t_{iM})^T\\
&Z_i=(z_0,z_1,\cdots,z_i)^T,\\
&DZ_i=(dz_0,dz_1,\cdots,dz_i)^T\\
&W_i=(W_{10},W_{11},\cdots,W_{iM})^T,\\
&DW_i=(dW_{10},dW_{11},\cdots,dW_{iM})^T
\end{aligned}
$$

并记它们所属的向量空间为$E_{T_{zi}}$，E_{TW_i}，E_{z_i}，E_{W_i}，即有$T_{zi}\in E_{T_{zi}}$，$T_{wi}\in E_{TW_i}$，Z_i，$DZ_i\in E_{zi}$，W_i，$DW_i\in E_{wi}$，构造分别由 (Z_i,DZ_i,T_{zi}) 和 (W_i,DW_i,T_{wi}) 确定的插值函数 $I_{zi}(Z_i,DZ_i,T_{zi})(t)$ 和 $I_{wi}(W_i,DW_i,T_{wi})(t)$，它们分别是由求解区间到$E^{n_1}$和$E^{n_2}$中的函数。记

$$\tilde{z}_i(t)=I_{zi}(Z_i,DZ_i,T_{zi})(t) \qquad (10-135)$$

$$\tilde{W}_i(t)=I_{wi}(W_i,DW_i,TW_i)(t) \qquad (10-136)$$

假定在构造 $\tilde{z}_i(t)$ 时，插值函数$I_{zi}(Z_i,DZ_i,T_{zi})(t)$ 中未用到dz_i，并且对于 $l\leqslant i$ 时有 $\tilde{z}_l(t_{ll})=z_l$、$\tilde{W}_l(t_{ll})=W_{ll}$。用 $\tilde{W}_i(t)$ 代替式（10-134 a）中的$W(t)$，得到初值问题

$$\frac{dz(t)}{dt}=g(t,z(t),\tilde{W}_i(t)) \quad z(t_i)=z_i \qquad (10-137)$$

用单步公式以步长 h_1 数值积分一步，得

$$z_{i+1}=z_i+h_1\psi_g(t_i,\ z_i,\ \tilde{w}_i,\ h_1) \qquad (10\text{-}138)$$

其中 ψ_g 是增量函数，它由所用的单步公式和函数 g 确定，并且仅是 t_i，h_1，z_i 和 $\tilde{w}_i(s)$ 的函数。由 Z_{i+1}、DZ_{i+1}、$T_{z_{i+1}}$ 按式（10-135）构造 $\tilde{z}_{i+1}(t)=I_{z_{i+1}}(Z_{i+1},\ DZ_{i+1},\ T_{z_{i+1}})(t)$，并代替式（10-134 b）中的 $z(t)$，得到初值问题

$$\frac{dW(t)}{dt}=h(t,\ \tilde{z}_{i+1}(t),\ W(t))\quad W(t_{i+1,0})=W_{iM}=W_i \qquad (10\text{-}139)$$

用单步公式以步长 h_2 数值积分，得到递推式

$$W_{i+1,j+1}=W_{i+1,j}+h_2\psi_h(t_{i+1,j},\tilde{z}_{i+1},W_{i+1,j},h_2)$$
$$j=0,\ 1,\ \cdots,\ M-1\quad W_{i+1,0}=W_{iM} \qquad (10\text{-}140)$$

ψ_h 是增量函数，它由所用的单步公式和函数 h 确定，仅是 $t_{i+1,j}$、h_2、$\tilde{z}_{i+1}(s)$、$W_{i+1,j}$ 的函数。

这样，我们描述了一个从 t_i 推进到 t_{i+1} 的完整过程。将公式中的 i 用 $i+1$ 代替，不断重复这个过程，直到 t_i 等于要求的终端时刻为止。将上面的描述叙述成下面的分解算法。

算法10-2

步 0　给出步长 h_1 和 h_2，构造函数 $\tilde{w}_0(t)$，令 $i=0$

步 1　按公式（10-138）计算 z_{i+1}

步 2　构造函数 $\tilde{z}_{i+1}(t)$，令 $j=0$

步 3　按公式（10-140）计算 $W_{i+1,j+1}$

步 4　判断 $j+1=M$ 是否成立，若成立，转步 5；否则令 $j=j+1$ 转步 3。

步 5　判断 t_{i+1} 是否为计算的终端时刻，若是，停止计算；否则构造函数 $\tilde{w}_{i+1}(t)$，令 $i=i+1$，转步 1。

现在对算法10-2作一些说明

（1）式（10-138）和（10-140）中的单步公式可以换成多步公式，相应的增量函数中的变量要进行改变。这一节后面的处理方法作一些修改后也可以用来处理多步公式的过程。

（2）若在构造 $\tilde{w}_i(t)$ 时，让式（10-136）不使用 t_i 邻近的一些点 $t_{i,i+1}, t_{i,i+2}, \cdots, t_{iM}$ 上 $\bar{w}(t)$ 和 $\bar{w}'(t)$ 的近似值，于是公式（10-138）和公式

$$W_{i,j+1}=W_{ij}+h_2\psi_A(t_{ij},\ \tilde{z}_i,\ W_{ij},\ h_2)$$
$$j=l,\ l+1,\cdots,\ M-1 \qquad (10\text{-}141)$$

的计算可以并行进行。另外一种改变是将式（10-139）、(10-140) 中的 $\tilde{z}_{i+1}(t)$ 改成 $\tilde{z}_i(t)$，则公式（10-138）、(10-140) 可并行计算。

在算法 10-2 中要求步长 h_1 是步长 h_2 的整倍数，现在我们来考虑 h_1 和 h_2 可以是任意正数的情形，记 $t_i=ih_1$，$s_j=jh_2$，i，$j=0$，1，2，…，记

$T_{wj}=(s_0, s_1, \cdots, s_j)^T$，$W_j=(W_0, W_1, \cdots, W_j)^T$，$DW_j=(dW_0, dW_1, \cdots, dW_j)^T$ 其中 W_j、dW_j 分别是 $\bar{w}(t)$ 和 $\bar{w}'(t)$ 在格点 s_j 上的近似值。$\tilde{z}_i(t)$，$\tilde{w}_j(t)$ 仍由式（10-135）、(10-136) 定义。只要将式（10-136）中的 i 改写成 j。

算法10-3

步 0　给出步长 h_1 和 h_2，构造函数 $\tilde{z}_0(t)$ 和 $\tilde{w}_0(t)$，令 $i=0$，$j=0$。

步 1　判断计算是否结束，若是，停止计算；否则若有 $t_i\leqslant s_j$，转步 2，若有 $t_i>s_j$，转步 5。

步 2　计算

$$z_{i+1}=z_i+h_1\psi_g(t_i,\ z_i,\ \tilde{w}_j,\ h_1) \qquad (10\text{-}142)$$

步 3　构造函数 $\tilde{z}_{i+1}(t)$

步 4　令 $i=i-1$，转步 1。

步 5　计算

$$W_{j+1}=W_j+h_2\psi_A(s_j,\ \tilde{z}_i,\ W_j,\ h_2) \qquad (10\text{-}143)$$

步 6　构造函数 $\tilde{w}_{j+1}(t)$

步 7　令 $j=j+1$，转步 1。

算法 10-3 的这种叙述是串行计算，其实步 2 到步 4 和步 5 到步 7 中的计算可在两个处理机中并行进行。只要取公式（10-

142）和（10-143）中用到的函数 $\tilde{w}_j(t)$ 和 $\tilde{z}_i(t)$ 为当时已构造好的函数，而不管 t_i 和 s_j 的大小，其计算框图在图 10-7 中给出。

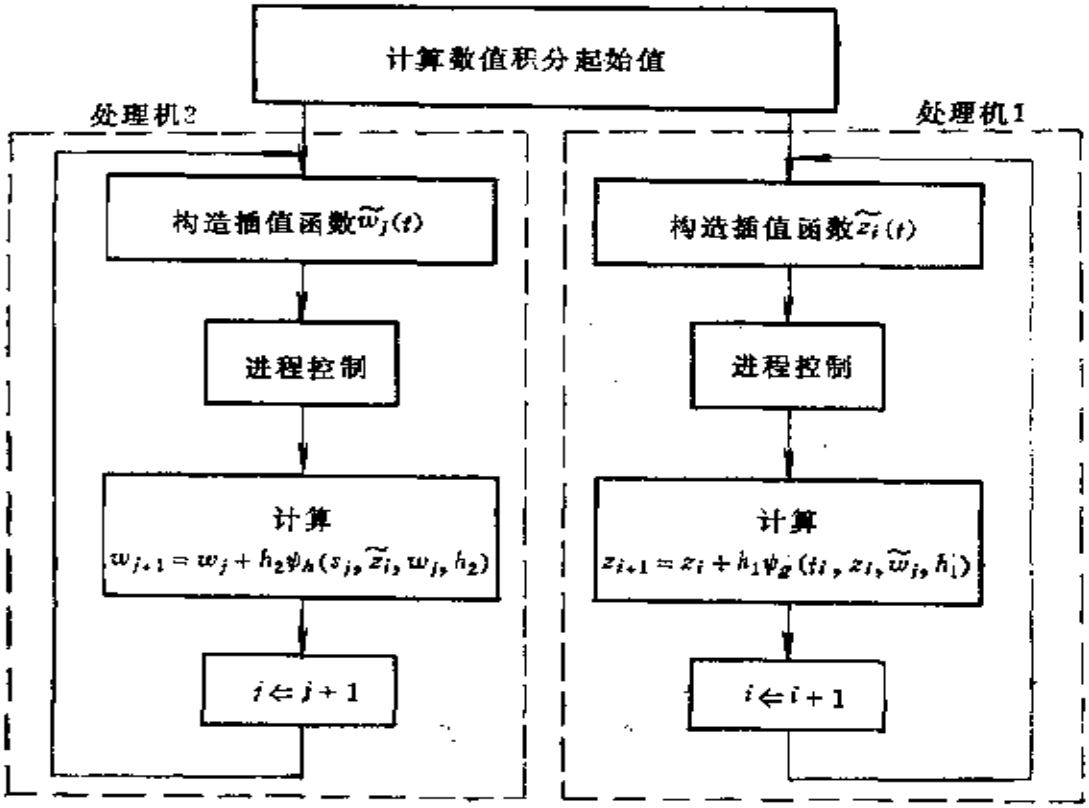


图10-7　算法10-3并行计算框图

在求解实际问题时，大量的计算集中在计算组(10-134)的右函数 g，h 上。$\tilde{z}_i(t)$ 和 $\tilde{w}_j(t)$ 的插值计算相对来说是微不足道的。通过算法 1 和算法 2 的方式将完整的组（10-134）分解成两个独立的方程组求解，而用简单的插值法将它们联系起来，从而可以用不同的方法，不同的步长独立地求解这两个方程组。特别，由于计算是独立的。如图 10-7 所示，这两个组的计算可并行地由两台处理机完成。

对于组（10-1）的分解，要根据实际需要进行，可以根据物理模型分块。若是为了实时数字仿真，可以将系统中需要取得实

时信息的量并成一组，而将不需要取得实时信息的量合成一个组；若是为了克服刚性方程组给计算带来的困难，可以将系统中变化速度快的重合成一个组，而将变化速度慢的分量合成另外一个组。

若按图 10-7 来实现算法 10-3 时，最好将组（10-134 a）和（10-134 b）划分得使计算过程中的 t_i 和 s_i 比较接近。这样可保证计算精度和数值稳定性。

在实际实现时，算法 10-2 和算法 10-3 可以作各种变动。特别是插值方法和数值积分方法，可以根据组（10-1）中量的变化特性和实际需要进行选取和灵活变动。

为了证明简单起见，我们只给出关于算法 10-2 的一些理论结果，但这些结果可以推广到更一般和复杂的算法。

为给出算法 10-2 的收敛性定理，我们先叙述一些假定和简单的引理。

设量 e_i 满足式

$$\|e_{i+1}\|\leqslant(1+hL)\|e_i\|+D \tag{10-144}$$

文献〔22〕给出下面的引理。

引理10-3 如果$\|e_i\|$满足（10-144），并且 $0\leqslant ih\leqslant B$，则

$$\|e_i\|\leqslant\frac{e^{LB}-1}{hL}D+e^{iLB}\|e_0\| \tag{10-145}$$

对组（10-134）的解 $\bar{z}(t)$，$\bar{w}(t)$，记

$$\begin{aligned}\overline{Z}_i&=(\bar{z}(t_0),\bar{z}(t_1),\cdots,\bar{z}(t_i))^T\\ D\overline{Z}_i&=(\bar{z}'(t_0),\ \bar{z}'(t_1),\cdots,\bar{z}'(t_i))^T\\ \overline{W}_i&=(\bar{w}(t_0),\ \bar{w}(t_1),\ \cdots,\bar{w}(t_i))^T\\ D\overline{W}_i&=(\bar{w}'(t_0),\bar{w}'(t_1),\ \cdots,\ \bar{w}'(t_i))^T\end{aligned} \tag{10-146}$$

记

$$e_{insz}=\max_i\Big(\max_{0\leqslant t\leqslant t_i}\|\bar{z}(t)-I_{z_i}(\overline{Z}_i,\ D\overline{Z}_i,\ T_{z_i})(t)\|\Big) \tag{10-147}$$

$$e_{insw}=\max_i\Big(\max_{0\leqslant t\leqslant t_i+h_1}\|\bar{w}(t)-I_{w_i}(\overline{W}_i,\ D\overline{W}_i,\ T_{wi})(t)\|\Big) \tag{10-148}$$

假定10-2 映象 $I_{z_i}(\cdot,\ \cdot,\ \cdot)(t)$ 和 $I_W(\cdot,\ \cdot,\ \cdot)(t)$ 满足下面的条件。

(1) $\lim\limits_{h_1\to 0} e_{intz}=0$，$\lim\limits_{\substack{h_1\to 0\\ h_2\to 0}} e_{intw}=0$

(2) 存在 L_{z_1}，L_{z_2}和 L_{w_1}，L_{w_2}，使对任何 $(Z_i^{(1)}, DZ_i^{(1)})\in E_{z_i}\times E_{z_i}$，$(Z_i^{(2)},\ DZ_i^{(2)})\in E_{z_i}\times E_{z_i}$，和$(W_i^{(1)}, DW_i^{(1)})\in E_{wi}\times E_{wi}$，$(W_i^{(2)}, DW_i^{(2)})\in E_{wi}\times E_{wi}$，成立估计式

$$\|I_{z_i}(Z_i^{(1)},\ DZ_i^{(1)},\ T_{z_i})(t)-I_{z_i}(Z_i^{(2)},\ DZ_i^{(2)},\ T_{z_i})(t)\|$$
$$\leqslant L_{z_1}\max_{j\leqslant i}\|z_j^{(1)}-z_j^{(2)}\|+L_{z_2}\max_{j\leqslant i-1}\|dz_j^{(1)}-dz_j^{(2)}\|\qquad t\in[0,\ t_i]$$

$$\|I_{w_i}(W_i^{(1)}, DW_i^{(1)},\ T_{w_i})(t)-I_{w_i}(W_i^{(2)},\ DW_i^{(2)},\ T_{wi})(t)\|$$
$$\leqslant L_{w_1}\max_{\substack{l\leqslant i\\ j\leqslant M}}\|W_{lj}^{(1)}-W_{lj}^{(2)}\|+L_{w_2}\max_{\substack{l\leqslant i\\ j\leqslant M}}\|dW_{lj}^{(1)}-dW_{lj}^{(2)}\|$$
$$t\in[0,\ t_i+h_1]$$

(3) 若 Z_{i-1}，DZ_{i-1} 分别是由 Z_i 和 DZ_i 去掉 z_i 和 dz_i 得到的向量，则当 $t\leqslant t_{i-1}$时，有

$$I_{z_i}(Z_i,\ DZ_i,\ T_{zi})(t)$$
$$=I_{z_{i-1}}(Z_{i-1},\ DZ_{i-1},\ T_{z_{i-1}})(t)$$

由这假定的(3)可以看出，分别将 $\tilde{z}_{i-1}(t)=I_{z_{i-1}}(Z_{i-1}, DZ_{i-1},\ I_{zi-1})(t)$ 和 $\tilde{z}_i(t)=I_{z_i}(Z_i,\ DZ_i,\ T_{z_i})(t)$ 代入组(10-134 b)，得到的两个微分方程组的初值问题的解，当 $t\leqslant t_{i-1}$ 时，是重合的。因而，对常微分方程初值问题

$$\frac{dW(t)}{dt}=h(t,\ \tilde{z}_{i+1}(t),\ W(t)),W(0)=W_0 \tag{10-149}$$

用步长 h_2 的递推公式

$$W_{l,j+1}=W_{lj}+h_2\psi_h(t_{lj},\ \tilde{z}_{i+1},\ W_{lj},\ h_2),\ W_{l0}=W_0$$
$$w_{l+1,0}=w_{lM}$$

数值积分得到的 $W_{lj}(l\leqslant i+1,\ j=0,\ 1,\ \cdots,\ M)$ 将与算法 10-2 得到的 W_{lj}是重合的。

用记号 $\Delta g(t, \bar{z}, \bar{w}, h)$ 表示组 (10-134 a) 的精确增量函数，即有

$$\bar{z}(t_{i+1})=\bar{z}(t_i)+h_i\Delta g(t_i, \bar{z}(t_i), \bar{w}, h_1)$$

其中 $\bar{z}(t)$，$\bar{w}(t)$ 是指组 (10-134) 的精确解。记

$$e_{\psi g}(t, h)=\Delta g(t, \bar{z}(t), \bar{w}, h)-\psi_g(t, z, \bar{w}, h)$$

$$\bar{e}_{\psi g}(h)=\max_{0\leqslant t\leqslant T}\|e_{\psi g}(t, h)\|$$

其中 T 是任意给定的积分终点。

对增量函数 $\psi_g(t, z, w, h)$，作下面的假定。

假定10-3 增量函数 $\psi_g(t, z, w, h)$ 满足下面的条件：

(1) 在所考虑的区域中，$\psi_g(t, z, w, h)$ 对于 t、h 连续，对 z、w 满足李普希兹条件，即存在正常数 $L_{\psi g_1}$、$L_{\psi g_2}$ 和 h_{10}，使对任意的 $z^{(1)}$、$z^{(2)}$ 和连续函数 $W^{(1)}(s)$，$W^{(2)}(s)$，当 $h\leqslant h_{10}$ 时成立

$$\|\psi_g(t, z^{(1)}, w^{(1)}, h)-\psi_g(t, z^{(2)}, w^{(1)}, h)\|\leqslant L_{\psi g_1}\|z^{(1)}-z^{(2)}\|$$

$$\|\psi_g(t, z^{(1)}, w^{(1)}, h)-\psi_g(t, z^{(1)}, w^{(2)}, h)\|\leqslant L_{\psi g_2}\max_{t\leqslant s\leqslant t+h}\|w^{(1)}(s)-w^{(2)}(s)\|$$

(2) $\lim\limits_{h\to 0}\bar{e}_{\psi g}(h)=0$

类似地，对函数 $\bar{z}(t)$，$\bar{w}(t)$ 和增量函数 $\psi_h(t, z, w, h)$ 定义 $\Delta_h(t, \bar{z}, \bar{w}, h)$ 和 $e_{\psi h}(t, h)$、$\bar{e}_{\psi h}(h)$，并作下面的假定。

假定10-4 增量函数 $\psi_h(t, z, w, h)$ 满足下面的条件：

(1) 在所考虑的区域中 $\psi_h(t, z, w, h)$ 对于 t，h 连续，对 z，w 满足李普希兹条件，即存在正常数 $L_{\psi h1}$，$L_{\psi h2}$ 和 h_{20}，使对任意的 $w^{(1)}$，$w^{(2)}$ 和连续函数 $z^{(1)}(s)$、$z^{(2)}(s)$，当 $h\leqslant h_{20}$ 时，成立

$$\|\psi_h(t, z^{(1)}, w^{(1)}, h)-\psi_h(t, z^{(2)}, w^{(1)}, h)\| \leqslant L_{\psi h 2} \max_{t \leqslant s \leqslant t+h} \|z^{(1)}(s)-z^{(2)}(s)\|$$

$$\|\psi_h(t, z^{(1)}, w^{(1)}, h)-\psi_h(t, z^{(1)}, w^{(2)}, h)\| \leqslant L_{\psi h 1}\|w^{(1)}-w^{(2)}\|$$

(2) $\lim_{h \to 0} \bar{e}_{wh}(h)=0$

下面二个定理分别给出分解算法 10-2 的收敛性和收敛的阶。

定理10-3 设假定10-1～假定10-4 成立，则当$h_1 \to 0$ 时，由算法10-2构造的序列$\{z_i\}$、$\{w_i\}$将收敛到组 (10-134) 的 解$\bar{z}(t)$、$\bar{w}(t)$。即当 $h_1 \to 0$ 时，若有 $t_i \to t$，则 $z_i \to \bar{z}(t)$，$w_i \to \bar{w}(t)$。

证明 记 $u_{z_i}=\bar{z}(t_i)-z_i$，$u_{w_{ij}}=\bar{w}(t_{ij})-w_{ij}$，$u_{w_i}=w(t_i)-w_i$，由 $\tilde{w}_i(s)$ 的定义式 (10-136) 和假定10-2，有

$$\begin{aligned}\|\bar{w}(s)-\tilde{w}_i(s)\| &\leqslant \|\bar{w}(s)-I_{w_i}(\bar{w}_i, D\bar{w}_i, T_{w_i})(s)\| \\ &+\|I_{wi}(\bar{w}_i, D\bar{w}_i, T_{w_i})(s)-I_{w_i}(w_i, Dw_i, T_{w_i})(s)\| \\ &\leqslant e_{inw}+L_{w_1} \max_{\substack{l \leqslant i \\ j \leqslant M}} \|\bar{w}(t_{lj})-w_{lj}\|+L_{w_2} \max_{\substack{l \leqslant i \\ j \leqslant M}} \|\bar{w}'(t_{lj}) \\ &- dw_{lj}\| \qquad (10\text{-}150)\end{aligned}$$

同样，对 $\tilde{z}_{i+1}(s)$，有

$$\begin{aligned}\|\bar{z}(s)-\tilde{z}_{i+1}(s)\| &\leqslant \|\bar{z}(s)-I_{z_{i+1}}(\bar{Z}_{i+1}, D\bar{Z}_{i+1}, T_{zi+1})(s)\| \\ &+\|I_{z_{i+1}}(\bar{Z}_{i+1}, D\bar{Z}_{i+1}, T_{z_{i+1}})(s)-I_{z_{i+1}}(Z_{i+1}, DZ_{i+1}, \\ &T_{z_{i+1}})(s)\| \leqslant e_{inzz}+L_{z_1} \max_{j \leqslant i+1} \|\bar{z}(t_j)-z_j\| \\ &+L_{z_2} \max_{j \leqslant i} \|\bar{z}'(t_j)-dz_j\| \qquad (10\text{-}151)\end{aligned}$$

由 dz_j 的计算方法

$$\begin{aligned}\|\bar{z}'(t_j)-dz_j\| &=\|g(t_j, \bar{z}(t_j), \bar{w}(t_j))-g(t_j, z_j, w_j)\| \\ &\leqslant L_{g_1}\|\bar{z}(t_j)-z_j\|+L_{g_2}\|\bar{w}(t_j)-w_j\| \\ &\qquad (10\text{-}152)\end{aligned}$$

将其代入式 (10-151) 得

$$\|\bar{z}(s)-\tilde{z}_{i+1}(s)\|\leqslant e_{intz}+(L_{z_1}+L_{z_2}L_{g_1})\max_{j\leqslant i+1}\|u_{z_j}\|$$
$$+L_{z_2}L_{g_2}\max_{j\leqslant i}\|u_{w_j}\| \tag{10-153}$$

由于当 $l\leqslant i$ 时

$$\|\bar{w}'(t_{lj})-dw_{lj}\|=\|h(t_{lj},\ \bar{z}(t_{lj}),\ \bar{w}(t_{lj}))-h(t_{lj},$$
$$\bar{z}_i(t_{lj}),\ w_{lj})\|\leqslant L_{h_1}\|\bar{z}(t_{lj})-\tilde{z}_i(t_{lj})\|$$
$$+L_{h_2}\|\bar{w}(t_{lj})-w_{lj}\| \tag{10-154}$$

将式 (10-153) 中的 $i+1$ 改成 i，并代入上式，当 $l\leqslant i$ 时得

$$\|\bar{w}'(t_{lj})-dw_{lj}\|\leqslant L_{h_1}e_{intz}+L_{h_1}(L_{z_1}+L_{z_2}L_{g_1})\max_{j\leqslant i}\|u_{z_j}\|$$
$$+L_{h_1}L_{z_2}L_{g_2}\max_{j\leqslant i-1}\|u_{w_j}\|+L_{h_2}\|u_{w_{lj}}\| \tag{10-155}$$

将式 (10-155) 代入 (10-150)，得

$$\|\bar{w}(s)-\tilde{w}_l(s)\|\leqslant e_{intw}+L_{w_1}\max_{\substack{l\leqslant i\\ j\leqslant M}}\|u_{w_{lj}}\|+L_{w_2}L_{h_1}e_{intz}$$
$$+L_{w_2}L_{h_1}(L_{z_1}+L_{z_2}L_{g_1})\cdot\max_{j\leqslant i}\|u_{z_j}\|$$
$$+L_{w_2}L_{h_1}L_{z_2}L_{g_2}\max_{j\leqslant i-1}\|u_{w_j}\|+L_{w_2}L_{h_2}\max_{\substack{l\leqslant j\\ j\leqslant M}}\|u_{w_{lj}}\|$$
$$\leqslant e_{intw}+L_{w_2}L_{h_1}e_{intz}+L_{w_2}L_{h_1}(L_{z_1}+L_{z_2}L_{g_1})\max_{j\leqslant i}\|u_{z_j}\|$$
$$+(L_{w_1}+L_{w_2}L_{w_2}+L_{w_2}L_{h_1}L_{z_2}L_{g_2})\max_{\substack{l\leqslant i\\ j\leqslant M}}\|u_{w_{lj}}\| \tag{10-156}$$

由假定10-2 后面的说明得到，当 $l\leqslant i+1$ 时，z_{lj} 满足递推式

$$w_{l,j+1}=w_{lj}+h_2\psi_h(t_{lj},\ \tilde{z}_{i+1},\ w_{lj},\ h_2),\quad w_{l0}=w_{l-1,M}$$
$$=w_{l-1},\quad l\leqslant i+1,\quad j\leqslant M-1$$

于是

$$u_{w_{l,j+1}}=\bar{w}(t_{l,j+1})+W_{l,j+1}$$
$$=\bar{w}(t_{lj})+h_2\Delta_h(t_{lj},\ \bar{z},\ \bar{w}(t_{lj}),h_2)$$
$$-w_{lj}-h_2\psi_h(t_{lj},\ \tilde{z}_{i+1},w_{lj},h_2)$$
$$=u_{w_{lj}}+h_2[\Delta_h(t_{lj},\ \bar{z},\ \bar{w}(t_{lj}),h_2)$$

$$
\begin{aligned}
&-\psi_h(t_{ij},\ \bar{z},\ \bar{w}(t_{ij}),\ h_2)]\\
&+h_2[\psi_h(t_{ij},\bar{z},\bar{w}(t_{ij}),h_2)\\
&-\psi_h(t_{ij},\ \bar{z},\ w_{ij},\ h_2)]\\
&+h_2[\psi_h(t_{ij},\ \bar{z},\ w_{ij},\ h_2)\\
&-\psi_h(t_{ij},\ \bar{z}_{i+1},\ w_{ij},\ h_2)]
\end{aligned}
\tag{10-157}
$$

由定理的假定，得到估计式

$$
\|u_{w_{i+1,j+1}}\|\leqslant\|u_{w_{ij}}\|+h_2\bar{e}_{\psi h}(h_2)+h_2L_{\psi h_1}\|u_{w_{ij}}\|
$$

$$
+h_2L_{\psi h_2}\max_{0\leqslant s\leqslant t_{i+1}}\|\bar{z}(s)-\bar{z}_{i+1}(s)\| \tag{10-158}
$$

将式（10-153）代入，得

$$
\begin{aligned}
\|u_{w_{i+1,j+1}}\|\leqslant&(1+h_2L_{\psi h_1})\|u_{w_{ij}}\|+h_2\bar{e}_{\psi h}(h_2)+h_2L_{\psi h_2}e_{intw}\\
&+h_2L_{\psi h_2}(L_{x_1}+L_{x_2}L_{g_1})\max_{j\leqslant i+1}\|u_{x_j}\|\\
&+h_2L_{\psi h_2}L_{x_2}L_{g_2}\max_{j\leqslant i}\|u_{w_j}\|
\end{aligned}
\tag{10-159}
$$

令 $l=i+1$，对 $j=0$，1，…，M，由引理10-3，我们得到

$$
\begin{aligned}
\|u_{w_{i+1,j}}\|\leqslant&\frac{e^{L_{\psi h_1}h_1}-1}{L_{\psi h_1}}[\bar{e}_{\psi h}(h_2)+L_{\psi h_2}e_{intx}+L_{\psi h_2}(L_{x_1}\\
&+L_{x_2}L_{g_1})\max_{j\leqslant i+1}\|u_{x_j}\|+L_{\psi h_2}L_{x_2}L_{g_2}\max_{j\leqslant i}\|u_{w_j}\|\\
&+e^{L_{\psi h_1}h_1}\|u_{w_{ij}}\|=\frac{e^{L_{\psi h_1}h_1}-1}{L_{\psi h_1}}L_{\psi h_2}L_{x_2}L_{g_2}\max_{j\leqslant i}\|u_{w_j}\|\\
&+e^{L_{\psi h_1}h_1}\|u_{w_{ij}}\|+\frac{e^{L_{\psi h_1}h_1}-1}{L_{\psi h_1}}L_{\psi h_2}(L_{x_1}+L_{x_2}L_{g_1})\max_{j\leqslant i+1}\|u_{x_j}\|\\
&+\frac{e^{L_{\psi h_1}h_1}-1}{L_{\psi h_1}}[\bar{e}_{\psi h}(h_2)+L_{\psi h_2}e_{intx}]
\end{aligned}
\tag{10-160}
$$

特别，当 $j=M$ 时，这式也成立，得

$$\|u_{w_{i+1}}\| \leqslant \frac{e^{L_{\psi h 1}h_1}-1}{L_{\psi h_1}} L_{\psi h_2} L_{z_2} L_{g_2} \max_{j \leqslant i} \|u_{w_j}\| + e^{L_{\psi h 1}h_1}\|u_{w_i}\|$$
$$+ \frac{e^{L_{\psi h 1}h_1}-1}{L_{\psi h_1}} L_{\psi h_2}(L_{z_1}+L_{z_2}L_{g_1}) \max_{j \leqslant i+1} \|u_{z_j}\|$$
$$+ \frac{e^{L_{\psi h 1}h_1}-1}{L_{\psi h_1}} [\bar{e}_{\psi h}(h_2) + L_{\psi h_2} e_{intz}] \qquad (10\text{-}161)$$

由于

$$\begin{aligned} u_{z_{i+1}} &= \bar{z}(t_{i+1}) - z_{i+1} \\ &= \bar{z}(t_i) + h_1 \Delta_g(t_i,\ \bar{z}(t_i), \bar{w},\ h_1) \\ &\quad - [z_i + h_1 \psi_g(t_i,\ z_i,\ \tilde{w}_i,\ h_1)] \\ &= u_{z_i} + h_1[\Delta_g(t_i,\ \bar{z}(t_i),\ \bar{w},\ h_1) \\ &\quad - \psi_g(t_i,\ \bar{z}(t_i),\ \bar{w},\ h_1)] \\ &\quad + h_1[\psi_g(t_i,\ \bar{z}(t_i), \bar{w},\ h_1) - \psi_g(t_i, \\ &\quad z_i,\ \bar{w},\ h_1)] + h_1[\psi_g(t_i,\ z_i, \\ &\quad \bar{w},\ h_1) - \psi_g(t_i,\ z_i,\ \tilde{w}_i,\ h_1)] \end{aligned} \qquad (10\text{-}162)$$

应用定理的假定，得

$$\|u_{z_{i+1}}\| \leqslant \|u_{z_i}\| + h_1 \bar{e}_{\psi g}(h_1) + h_1 L_{\psi h_1}\|u_{z_i}\|$$
$$+ h_1 L_{\psi g_2} \max_{t_i \leqslant s \leqslant t_{i+1}} \|\bar{w}(s) - \tilde{w}_i(s)\| \qquad (10\text{-}163)$$

将式（10-156）代入，得

$$\|u_{z_{i+1}}\| \leqslant (1 + h_1 L_{\psi g_1})\|u_{z_i}\| + h_1 \bar{e}_{\psi g}(h_1) + h_1 L_{\psi g_2} e_{intw}$$
$$+ h_1 L_{\psi g_2} L_{w_2} L_{h_1} e_{intz} + h_1 L_{\psi g_2} L_{w_2} L_{h_1}(L_{z_1}$$
$$+ L_{z_2} L_{g_1}) \max_{j \leqslant i} \|u_{z_j}\| + h_1 L_{\psi g_2}(L_{w_1} + L_{w_2} L_{h_2}$$
$$+ L_{w_2} L_{h_1} L_{z_2} L_{h_2}) \max_{\substack{l \leqslant i \\ j \leqslant M}} \|u_{w_{lj}}\| = (1 + h_1 L_{\psi g_1})\|u_{z_i}\|$$
$$+ h_1 L_{\psi g_2} L_{w_2} L_{h_1}(L_{z_1} + L_{z_2} L_{g_1}) \max_{j \leqslant i} \|u_{z_j}\|$$
$$+ h_1 L_{\psi g_2}(L_{w_1} + L_{w_2} L_{h_2} + L_{w_2} L_{h_1} L_{z_2} L_{g_2}) \max_{\substack{l \leqslant i \\ j \leqslant M}} \|u_{w_{lj}}\|$$
$$+ h_1[\bar{e}_{\psi g}(h_1) + L_{\psi g_2} e_{intw} + L_{\psi g_2} L_{w_2} L_{h_1} e_{intz}] \qquad (10\text{-}164)$$

令量V_{x_j}，V_{w_j}满足递推式

$$V_{x_{i+1}}=A_{11}V_{x_i}+A_{12}V_{w_i}+B_1 \quad V_{x_0}=0$$
$$V_{w_{j+1}}=A_{21}V_{x_i}+A_{22}V_{w_i}+B_2 \quad V_{w_0}=0$$

其中

$$A_{11}=1+h_1[L_{\psi g_1}+L_{\psi g_2}L_{w_2}L_{h_1}(L_{x_1}+L_{x_2}L_{g_1})]$$
$$A_{12}=h_1L_{\psi g_2}(L_{w_1}+L_{w_2}L_{h_2}+L_{w_2}L_{h_1}L_{x_2}L_{g_2})$$
$$A_{21}=\frac{e^{L_{\psi h1}h_1}-1}{L_{\psi h_1}}L_{\psi h_2}(L_{x_1}+L_{x_2}L_{g_1})A_{11}$$
$$A_{22}=e^{L_{\psi h1}h_1}+\frac{e^{L_{\psi h1}h_1}-1}{L^{\psi h_1}}[L_{\psi h_2}L_{x_2}L_{g_2}+L_{\psi h_2}(L_{x_1}+L_{x_2}L_{g_1})A_{12}] \tag{10-165}$$
$$B_1=h_1(\bar{e}_{\psi g}(h_1)+L_{\psi g_2}e_{inw}+L_{\psi g_2}L_{w_2}L_{h_1}e_{inx}]$$
$$B_2\frac{e^{L_{\psi h1}h_1}-1}{L_{\psi h_1}}[\bar{e}_{\psi h}(h_2)+L_{\psi h_2}e_{inx}+L_{\psi h_2}(L_{x_1}+L_{x_2}L_{g1})B_1]$$

由式(10-161)、(10-164)、并将式（10-164）代入式（10-161）中的$\|u_{\tau_{i+1}}\|$，容易看出有

$$\|u_{x_j}\|\leqslant V_{x_i}$$
$$\|u_{w_ij}\|\leqslant V_{w_i} \quad i=0,\ 1,\ \cdots \quad j=0,\ 1,\ \cdots,\ M \tag{10-166}$$

由A_{11}，A_{12}，A_{21}，A_{22}的表达式可知它们可表示成

$$A_{11}=1+h_1a_{11}$$
$$A_{12}=h_1a_{12}$$
$$A_{21}=h_1a_{21}$$
$$A_{22}=1+h_1a_{22}$$

其中a_{11}、a_{12}、a_{21}、a_{22}均是h_1的正函数，当$h_1\to 0$时都以正数为极限，可找到正数$\bar{h}_1$和正常数$\bar{a}_{11}$，$\bar{a}_{12}$，$\bar{a}_{21}$，$\bar{a}_{22}$。使当$h_1\leqslant\bar{h}_1$时，有$a_{11}\leqslant\bar{a}_{11}$，$a_{12}\leqslant\bar{a}_{12}$，$a_{21}\leqslant\bar{a}_{21}$，$a_{22}\leqslant\bar{a}_{22}$，设量$V_i$满足递推式

$$V_{i+1}=(1+h_1\bar{a})V_i+B,\ V_0=0 \tag{10-167}$$

其中

$$\bar{a}=\max\{\bar{a}_{11}+\bar{a}_{12},\ \bar{a}_{21}+\bar{a}_{22}\}$$

$$B=\max\{B_1,\ B_2\}$$

则易证明有 $\max\{V_{zi},V_{w_i}\}\leqslant V_i$，由引理 10-3 知 V_i 有估计式

$$V_i\leqslant\frac{e^{\bar{a}(t_i-t_0)}-1}{\bar{a}}\cdot\frac{B}{h_1}\tag{10-168}$$

由 B_1，B_{2+J} 表示式及定理的假定易知当 $h_0\to 0$ 时，有 $\frac{1}{h_1}B\to 0$，由式（10-168）知，当 t_i 小于某个有限数 $\bar{t}$ 时，一致地有 $\|\bar{z}(t_i)-z_i\|\to 0$，$\|\bar{w}(t_i)-w_i\|\to 0$。于是若有 $h_1\to 0$ 和 $t_i\to t$，由于

$$\|\bar{z}(t)-z_i\|\leqslant\|\bar{z}(t)-\bar{z}(t_i)\|+\|\bar{z}(t_i)-z_i\|$$

和 $\bar{z}(t)$ 的连续性，将有 $\|\bar{z}(t)-z_i\|\to 0$，即有 $z_i\to\bar{z}(t)$。同理可证 $w_i\to\bar{w}(t)$，定理证毕。

定理10-3 给出了分解算法10-2收敛的条件，在对 $\bar{e}_{\varphi g}(h_1)$，$\bar{e}_{\varphi h}(h_2)$，e_{intz}，e_{intw} 的更精细的假定下，下面的定理给出了当 $h_1\to 0$ 时，$z_i\to\bar{z}(t)$、$w_i\to\bar{w}(t)$ 的收敛速度（收敛阶）。

定理10-4 设定理 10-3 的条件成立，并且存在 h_0 和正常数 A_z，A_w，A_{z_1}，A_{w_1}，使当 $h_1\leqslant h_0$、$h_2\leqslant h_0$时成立估计

$$\bar{e}_{\varphi g}(h_1)\leqslant A_z h_1^{p_z},\ e_{intz}\leqslant A_{z1}h_1^{q_z}\tag{10-169}$$

$$\bar{e}_{\varphi h}(h_2)\leqslant A_w h_2^{p_w},\ e_{intw}\leqslant A_{w1}h_2^{q_w}$$

其中 p_z，p_w，q_z，q_w 是自然数，则存在正常数 A 和 $\bar{h}_0$，使当 $h_1\leqslant\bar{h}_0$时，有估计

$$\|\bar{z}(t_i)-z_i\|\leqslant Ah_1^p$$

$$\|\bar{w}(t_{ij})-w_{ij}\|\leqslant Ah_1^p$$

其中 $p=\min\{p_z,\ p_w,\ q_z,\ q_w\}$

证明 由估计式（10-168）和 B 的定义，将式（10-169）代入，就能得到所要求的结果。

算法 10-2 和算法 10-3 中所用的积分方法可取通常的四阶龙格-库塔方法，而插值方法可取分段的拉格朗日公式或爱尔米特公式插值，但点数不宜过多，这时定理 10-3 和定理 10-4 中的条

件都将成立。

10.5.2 插值分解异步并行迭代算法

算法 10-2 是一个同步算法，算法的同步化是降低多处理机使用效率的重要因素。现在来构造一个数值求解问题（10-1）的插值分解异步并行迭代算法，我们首先将向量 y 的分量归并成 I 个子向量 $y_1, y_2, \cdots, y_I$，相应地将问题(10-1)分割成 I 个子问题。

$$\frac{dy_i}{dt}=f_i(t, y_1, y_2, \cdots, y_I),\ y_i(t_{i0})=y_{i0}$$

$$i=1, 2, \cdots, I \qquad (10\text{-}170)$$

其中 y_i 是 n_i 维向量 $f_i(t, y_1, y_2, \cdots, y_I)$ 是 n_i 维向量值函数，有

$$n=n_1+n_2+\cdots+n_I$$

式（10-170）中的函数 f_i 的计算可能非常复杂，计算量很大，并且可能有些量的计算是若干个子组所共有的，为了使各个右函数 f_i 的计算量比较均衡，并避免重复计算，我们将右函数 f_i 的计算进行分割，将其表示成

$$f_i(t, y_1, y_2, \cdots, y_I)=F_i(t, g_1, g_2, \cdots, g_J, y_1, y_2, \cdots, y_I) \qquad (10\text{-}171)$$

其中 F_i 为其变量 $g_1, \cdots, g_J, y_1, \cdots y_I, t$ 的比较简单的函数，g_j 是 $y_1, y_2, \cdots, y_I, t$ 的函数

$$g_j=g_j(y_1, y_2, \cdots, y_I, t) \quad j=1, 2, \cdots, J \qquad (10\text{-}172)$$

它是右函数 f_i 计算过程中的子块。

为了各个处理机之间进行通讯，在共享存储器中设置变量 Gy_i 和 Gg_j，$i=1, 2, \cdots, I$，$j=1, 2, \cdots, J$，分别存放变量 y_i 和变量 g_j 在一些时间点上的值。这些变量可以为各个处理机执行读或写，称作整体变量，它们的构造分别由表10-4和表 10-5 表示。整体变量 Gy_i 和 Gg_j 中的时间集 $T_{y_i}=\{t_{y_{il}} | l=0, 1, \cdots, L_i\}$ 和 $T_{g_j}=\{t_{g_{jm}} | m=0, 1, \cdots, M_j\}$ 分别称作计算

表10-4　整体变量Gy_i的结构

$t_{y_{i0}}$	$t_{y_{i1}}$	$t_{y_{i2}}$	…	$t_{y_{iL_i}}$
y_{i0}	y_{i1}	y_{i2}	…	y_{iL_i}

表10-5　整体变量Gg_j的结构

$t_{g_{j0}}$	$t_{g_{j1}}$	$t_{g_{j2}}$	…	$t_{g_{jM_j}}$
g_{j0}	g_{j1}	g_{j2}	…	g_{jM_j}

量 y_i 和 g_j 的格点集。通过插值过程，由变量 G_{y_i} 可得到组（10-170）的解 $y_i(t)$ 的近似 $\tilde{y}_i(t)$，而由变量 G_{g_j} 可得到时间 t 的函数

$$g_j(t)=g_j(y_1(t),\ y_2(t),\ \cdots,\ y_I(t)\ t)$$

的近似 $\tilde{g}_j(t)$。

将由整体变量 $G_{y_l}(l=1,\ 2,\ \cdots,\ i-1,\ i+1,\ \cdots,\ I)$ 和 $G_{g_j}(j=1,\ 2,\ \cdots,\ J)$ 构造的函数 $\tilde{y}_l(t)$ 和 $\tilde{g}_j(t)$ 代入式（10-170）的第 i 个子问题，我们得到初值问题

$$\frac{dy_i}{dt}=F_i(t,\ \tilde{g}_1(t),\ \tilde{g}_2(t),\ \cdots,\ \tilde{g}_J(t),\ \tilde{y}_1(t),\cdots,\ \tilde{y}_{i-1}(t),\ y_i,\ \tilde{y}_{i+1}(t),\ \cdots,\ \tilde{y}_I(t)),\ y_i(t_{i0})=y_{i0} \tag{10-173}$$

应用数值积分方法 ME_i 对问题（10-173）进行数值积分，得到初值问题（10-173）的解在格点集 T_{y_i} 上的近似值 y_{i_l}，我们用这些值更新整体变量 Gy_i。

将由整体变量 Gy_i 构造的函数 $\tilde{y}_i(t)(i=1,\ 2,\ \cdots,\ I)$ 代入 g_j 的表达式（10-172），计算格点集 T_{g_j} 上的值

$$g_{jm}=g_j(\tilde{y}_1(t_{jm}),\ \tilde{y}_2(t_{jm}),\ \cdots,\ \tilde{y}_I(t_{jm}),\ t_{jm}),$$
$$m=1,\ 2,\ \cdots,\ M_j \tag{10-174}$$

并用这些值更新整体变量 Gg_j。

我们把整体变量 Gy_i，Gg_j 的每一次更新计算称作一个过程

段。由上面的构造，共有 $I+J$ 个过程段，每个过程段的计算只依赖于整体变量的值，与其它过程段的计算无关。将这 $I+J$ 个过程段分配给 p_r 台处理机进行并行计算，分配的方法称作进程控制。我们把上面的构造总结成下面的算法。

算法10-4 （整体变量 Gy_i 的更新计算）

步 1　读整体变量 Gy_l，$l \neq i$，$l=1, 2, \cdots, I$，Gg_j，$j=1, 2, \cdots J$

步 2　构造插值函数 $\tilde{y}_l(t)$，$\tilde{g}_j(t)$

步 3　应用方法 ME_i 数值积分初值问题（10-173）

步 4　更新整体变量 Gy_i，并给出 Gy_i 是否满足精度的标志

算法10-5 （整体变量 Gg_j 的更新计算）

步 1　读整体变量 Gy_i，$i=1, 2, \cdots, I$

步 2　构造插值函数 $\tilde{y}_i(t)$，$i=1, 2, \cdots, I$

步 3　按式（8-174）计算 g_{jm}，$m=0, 1, 2, \cdots, M_j$

步 4　更新整体变量 G_{gi}

算法10-6 （插值分解异步并行迭代算法）

步 1　给出整体变量 G_{y_i}，$i=1, 2, \cdots, I$ 的初始值

步 2　计算整体变量 G_{gj}，$j=1, 2, \cdots J$

步 3　按照进程控制在 p_r 台处理机上按算法 10-4 或算法 10-5对 $I+J$ 个过程段进行并行计算，并由步 4 控制迭代计算是否结束。

步 4　若 I 个更新 Gy_i 的过程段均给出满足精度的标志，则所有 Gy_i 的格点集 $Ty_i(i=1, 2, \cdots, I)$ 上的迭代结束，否则继续迭代。

若整体变量 Gy_i 中的时间值 $t_{y_i l_i}$ 都已达到要求解的时间区间的右端点，则停止计算，否则重新形成格点集 T_{y_i} 和 T_{g_j}，调用算法10-6，重复进行计算，直到达到时间区间的右端点为止。

算法 10-6 的步 3 是一个异步并行迭代过程，即 $I+J$ 个过程段中的任何一个的运行并不取决其它过程段的运行是否完成，

对于给定的格点集 T_{y_i} 和 T_{g_j}（$i=1,\ 2,\ \cdots,\ I$），（$j=1,\ 2,\ \cdots,\ J$），在计算过程中只要整体变量 G_{y_i}（$i=1,\ 2,\ \cdots,\ I$）中有一个得到更新，我们就称算法 10-6 完成一次迭代。我们将第 l 次迭代中得到更新的整体变量 G_{y_i} 的足标 i 的集合记为 $I(l)$。而将第 l 次迭代得到的整体变量 G_{y_i} 的值记为 $G^l_{y_i}$。

当 $i\notin I(l)$ 时，G_{yi} 保持前一次迭代的值，有

$$G^l_{yi}=G^{l-1}_{yi}\quad i\notin I(l) \tag{10-175}$$

当 $i\in I(l)$ 时，G^l_{yi} 的值由算法 8-4 得到，其计算过程中用到整体变量 Gy_k（$k=1,\ 2,\ \cdots,\ I$），$k\neq i$ 和 Gg_j（$j=1,\ 2,\ \cdots,\ J$）的值。设 Gy_k 的值是在第 $s_k(l,\ i,\ 0)$ 次迭代时的更新值，Gg_j 的值是由算法 10-5 计算得到的，在其计算过程中又用到 Gy_k（$k=1,\ 2,\ \cdots,\ I$）的值，设其中 Gy_k 的值是第 $s_k(l,\ i,\ j)$ 次迭代时的更新值。这样，当 $i\in I(l)$ 时，Gy_i 的值 G^l_{yi} 是值 $G^{s_k(l,i,j)}_{yk}$（$k=1,\ 2,\ \cdots,\ I$），（$j=1,\ 2,\ \cdots,\ J$）的函数记

$$z_j(l,\ i)=(G^{s_1(l,i,j)}_{y1},\ G^{s_2(l,i,j)}_{y2},\ \cdots,\ G^{s_I(l,i,j)}_{yI})$$

则 G^l_{yi} 可以表示成函数

$$G^l_{yi}=P_i(z_0(l,\ i),\ z_1(l,\ i),\ \cdots,\ z_J(l,\ i))$$
$$i\in I(l) \tag{10-176}$$

现在我们来证明由式（10-175），（10-176）给出的异步迭代的收敛性。记 Gy_i 的取值空间为 V^{ni}，对于任何的 $Gy_{ij}\in V^{ni}$（$j=0,\ 1,\ 2,\ \cdots,\ J$），（$i=1,\ 2,\ \cdots,\ I$），记

$$z_j=(Gy_{1j},\ Gy_{2j},\ \cdots,\ Gy_{Ij})$$

$$Z=(z_0,\ z_1,\ \cdots,\ z_J)$$

$$\|Z\|=\max_j\ \max_k\|Gy_{kj}\|$$

令

$$h_{y_i}=\max_{1\leqslant k\leqslant L_i}(t_{y_{ik}}-t_{y_{ik-1}})$$

$$h_{g_j}=\max_{1\leqslant m\leqslant M_j}(t_{g_{jm}}-t_{g_{jm-1}})$$

$$h_i=\max\{t_{y_{i0}},\ t_{y_{i20}},\ \cdots,\ t_{y_{iI}0},\ t_{g_{i0}},\ t_{g_{i20}},\ \cdots,\ t_{g_{iJ}0}\}$$
$$-\min\{t_{y_{i0}},\ t_{y_{i20}},\ \cdots,\ t_{y_{iI}0},\ t_{g_{i0}},\ t_{g_{i20}},\ \cdots,\ t_{g_{iJ}0}\}$$
$$h_K=\max\{t_{y_{i1L_1}},\ t_{y_{i2L_2}},\ \cdots,\ t_{y_{iI_iL_{I_i}}},\ t_{g_{i1M_1}},\ t_{g_{i2M_2}},\ \cdots,t_{g_{iJ_iM_{J_i}}}\}$$
$$-\min\{t_{y_{i1L_1}},\ t_{y_{i2L_2}},\ \cdots,\ t_{y_{iI_iL_{I_i}}},\ t_{g_{i1M_1}},\ t_{g_{i2M_2}},$$
$$\cdots,t_{g_{iJ_iM_{J_i}}}\}$$
$$h=\max\{h_{y_1},\ h_{y_2},\ \cdots,\ h_{y_I},\ h_{g_1},\ h_{g_2},\ \cdots,\ h_{g_J}\}$$

作下面的假定。

假定10-5 问题（10-170）中的右端函数 $f_i(t,y_1,y_2,\cdots,y_r)$，数值积分方法 ME_i 和算法 10-4，算法 10-5 中的插值方法使函数 $P_i(\cdots)$ 对任意的 $Z^{(1)}=(z_0^{(1)},\ z_1^{(1)},\ \cdots,\ z_I^{(1)})$ 和 $z^{(2)}=(z_0^{(2)},z_1^{(2)},\cdots z_I^{(2)})$有估计式

$$\|P_i(Z^{(1)})-P_i(Z^{(2)})\|\leqslant h_{g_i}Q_i\|Z^{(1)}-Z^{(2)}\| \quad (10\text{-}177)$$

其中 Q_i 为固定的常数。

假定10-6 算法 10-6 的步 3 中的进程控制使对每个 $l=1,2,\cdots,i=1,2,\cdots,I,k=1,2,\cdots,J,j=0,1,2,\cdots,J$ 成立

1. $S_k(l,i,j)<l$
2. 每个 $i=1,2,\cdots,I$ 都在无限多个 $I(l)$ 中出现
3. 记 $S(l)=\min\limits_{\substack{i=1,2,\cdots,I\\K=1,2,\cdots,J\\j=0,1,2,\cdots,J}} S_k(l,i,j)$，则有 $\lim\limits_{l\to\infty}S(l)=\infty$。

我们有下面的定理。

定理10-5 设假定 10-5 和假定 10-6 成立，则当

$$\alpha=\max_i h_iQ_i<1$$

时，算法 10-6 步 3 中的异步并行迭代过程收敛。

证明 为了证明异步并行迭代过程的收敛性，我们首先证明下面的同步迭代算法的收敛性。取初值 $G_{yi}^{(0)}\in V^{m_i}(i=1,2,\cdots,I)$ 构造 $z_j^{(0)}=(G_{y1}^{(0)},\ G_{y2}^{(0)},\ \cdots,\ G_{yI}^{(0)})(j=0,\ 1,\ \cdots,J)$ 和 $Z^{(0)}=(z_0^{(0)},\ z_1^{(0)},\ \cdots,\ z_J^{(0)})$，计算

$$Gy_i^{(1)}=P_i(Z^{(0)}) \quad i=1,\ 2,\ \cdots,\ I$$

类似地组成 $z_i^{(1)}=(G^{(1)}{}_{y_1}, G^{(1)}{}_{y_2}, \cdots, G^{(1)}{}_{y_I})$ 和 $Z^{(1)}=(z_0^{(1)}, z_1^{(1)}, \cdots, z_J^{(1)})$，并计算 $G_{yi}^{(2)}$，一般若已知 $z_i^{(d)}=(Gy_1^{(d)},\ Gy_2^{(d)}, \cdots, Gy_I^{(d)})$ 构造 $Z^{(d)}=(z_0^{(d)},\ z_1^{(d)},\ \cdots,\ z_J^{(d)})$，并计算

$$Gy_i^{(d+1)}=P_i(Z^{(d)}),\quad i=1,\ 2,\ \cdots,\ I \qquad (10\text{-}178)$$

这是同步迭代，由假定10-5

$$\|Z^{(d+1)}-Z^{(d)}\|=\max_i \|Gy_i^{(d+1)}-Gy_i^{(d)}\|\leqslant \max_i(h_{y_i}Q_i)\|Z^{(d)}-Z^{(d-1)}\|=\alpha\|Z^{(d)}-Z^{(d-1)}\|\leqslant\cdots\leqslant\alpha^d\|Z^{(1)}-Z^{(0)}\|$$

因此对任意自然数 m 有

$$\begin{aligned}\|Gy_i^{(d+m)}-Gy_i^{(d)}\|&\leqslant\|Gy_i^{(d+m)}-Gy_i^{(d+m-1)}\|+\|Gy_i^{(d+m-1)}\\&\quad -Gy_i^{(d+m-2)}\|+\cdots+\|Gy_i^{(d+1)}-Gy_i^{(d)}\|\\&\leqslant\|Z^{(d+m)}-Z^{(d+m-1)}\|+\|Z^{(d+m-1)}-Z^{(d+m-2)}\|+\cdots\\&\quad+\|Z^{(d+1)}-Z^{(d)}\|\leqslant(\alpha^{d+m-1}+\alpha^{d+m-2}+\cdots+\alpha^d)\|Z^{(1)}\\&\quad -Z^{(0)}\|<\frac{\alpha^d}{1-\alpha}\|Z^{(1)}-Z^{(0)}\|\end{aligned}$$

因此序列 $\{Gy_i^{(d)}\}$ 是哥西序列，它收敛到一个极限值，记其极限值为 Gy_i^*。

构造 $z_j^*=(Gy_1^*,\ Gy_2^*,\ \cdots,\ Gy_I^*)$，$j=0,\ 1,\cdots,J$，$Z^*=(z_0^*,\ z_1^*,\ \cdots,\ z_J^*)$，于是 Z^* 是序列 $\{Z^{(d)}\}$ 的极限，在式 (10-178) 两边取极限，得

$$Gy_i^*=P_i(Z^*) \qquad (10\text{-}179)$$

现在证明异步并行迭代的收敛性，设 $Gy_i^{(0)}(i=1,2,\cdots,I)$，$Z^{(0)}=(z_0^{(0)},\ z_1^{(0)},\ \cdots,\ z_J^{(0)})$ 分别由算法 10-6 的步 1 得到的迭代初值，构造自然数的递增序列 $\{m_k\}(k=1,\ 2,\ \cdots)$ 满足条件

$$\bigcup_{m_k\leqslant S(l)<l<m_{k+1}} I(l)=\{1,\ 2,\ \cdots,\ I\} \qquad (10\text{-}180)$$

其中 $m_0=0$，这表示在 m_k 和 m_{k+1} 之间，每个整体变量 Gy_i 至少更新一次，并且其中至少有一次更新用到的信息均是在 m_k 次迭代后更新得到的信息。由假定 10-6，记使 $i\in I(l)$ 成立的最

小自然数 l 为 l_i，可取 $m_1=\max\limits_{i=1,\ 2,\ \cdots,\ I} l_i+1$，设已得到$m_k$，由于 $S(l)\to\infty$，$(l\to\infty)$，可找到自然数 l_s，使当 $l\geqslant l_s$ 时，都有 $S(l)\geqslant m_k$，取 l_i 为满足 $l>l_s$ 且使 $i\in I(l)$ 成立的最小的自然数 l，取

$$m_{k+1}=\max_{i=1,\ 2,\ \cdots,\ I} l_i+1$$

由这种构造得到的序列 $\{m_k\}$ 将使式（10-180）成立。

记 $Z(l,\ i)=(z_0(l,\ i),\ z_1(l,\ i),\ \cdots,z_J(l,\ i))$，若对于 $0<l<m_1$ 的 l，$i\in I(l)$，Gy_i 得到第一次更新，则有

$$\begin{aligned}\|Gy_i^{(1)}-Gy_i^*\|&=\|P_i(Z(l,\ i))-P_i(Z^*)\|\\&\leqslant\alpha\|Z(l,\ i)-Z^*\|=\alpha\|Z^{(0)}-Z^*\|\end{aligned}\tag{10-181}$$

由式（10-175），（10-176）和式（10-181），对于所有 $l=1,2,\cdots$ 和 $i=1,\ 2,\ \cdots I$ 都有

$$\|Z(l,\ i)-Z^*\|\leqslant\|Z(l-1,\ i)-Z^*\|\tag{10-182}$$

因此由式（10-181），有

$$\|Gy_i^{(l)}-Gy_i^*\|\leqslant\alpha\|Z^{(0)}-Z^*\|\tag{10-183}$$

并且以后的迭代值将保持这个不等式。对于成立 $S(l)\geqslant m_1$ 的 l，不等式

$$\|Gy_k^{i_k(l,i,j)}-Gy_k^*\|\leqslant\alpha\|Z^{(0)}-Z^*\|\tag{10-184}$$

将对所有 k，$i=1,\ 2,\ \cdots,\ I$，$j=0,\ 1,\ \cdots,J$ 都成立。因此对于满足 $m_1\leqslant S(l)<l<m_2$ 的 l，若 $i\in I(l)$，则有

$$\begin{aligned}\|Gy_i^{(l)}-Gy_i^*\|&=\|P_i(Z(l,\ i))-P_i(Z^*)\|\\&\leqslant\alpha\|Z(l,\ i)-Z^*\|\leqslant\alpha^2\|Z^{(0)}-Z^*\|\end{aligned}\tag{10-185}$$

并且在以后的迭代中，这个估计式将一直保持，特别当 $l\geqslant m_2$ 时，对所有 $i=1,\ 2,\ \cdots,\ I$ 时，都有

$$\|Gy_i^{(l)}-Gy_i^*\|\leqslant\alpha^2\|Z^{(0)}-Z^*\|$$

现在设当 $l\geqslant m_k$ 时，成立估计式

$$\|Gy_i^{(l)}-Gy_i^*\|\leqslant a^k\|Z^{(0)}-Z^*\| \quad i=1,\ 2,\ \cdots,\ I \tag{10-186}$$

于是对于满足 $m_k\leqslant S(l)<l$ 的 l，若 $i\in I(l)$，有

$$\begin{aligned}\|Gy_i^{(l)}-Gy_i^*\| &= \|P_iZ(l,\ i)-P_i(Z^*)\|\leqslant a\|Z(l,\ i)-Z^*\|\\ &= a\max_{0\leqslant j\leqslant J}\max_{K=1,\ 2,\ \cdots,\ I}\|Gy'^{k(l,i,j)}-Gy_k^*\|\\ &\leqslant a^{k+1}\|Z^{(0)}-Z^*\|\end{aligned}$$

这就推得当 $l\to\infty$ 时，$\|Gy_i^{(l)}-Gy_i^*\|\to 0$，因而 $Gy_i^{(l)}\to Gy_i^*$，从而证明了异步并行迭代的收敛性。

若函数 f_i，F_i 和 g_i 对变量满足李普希兹条件，可以证明假定 10-5 对于常用的数值积分方法（如龙格-库塔方法）和插值方法（如拉格朗日方法）是成立的，简单的进程控制方法也易使假定 10-6 成立。例如按照固定的次序循环地将 $I+J$ 个过程段不断地分配给 p_r 台处理机运行，并且每个过程段总会在有限时间内运行完，更新相应的整体变量，则假定 10-6 中的条件都能成立。

在 VAX-11/780 计算机上模拟算法 10-6 的计算过程。下面的例子给出这种计算过程对一个简化导弹运动模型的数字仿真的计算结果。

〔例10-4〕 假定地球为不转的圆球，在发射点直角坐标系中，简化的导弹平面运动模型为

$$\begin{aligned}&\frac{dV_x}{dt}=\dot{w}_x+g_x \quad V_x(0)=v_{x0}\\ &\frac{dV_y}{dt}=\dot{w}_y+g_y \quad V_y(0)=v_{y0}\\ &\frac{dx}{dt}=v_x \quad x(0)=x_0\\ &\frac{dy}{dt}=v_y \quad y(0)=y_0\end{aligned} \tag{10-187}$$

式中 v_x，v_y，x，y 分别是导弹质心运动在发射点直角坐标系中的速度向量和位置向量的分量。$\dot{w}_x$，$\dot{w}_y$ 是作用在导弹上除地球

引力以外的各种力产生的合加速度向量在坐标系中的分量，g_x，g_y是作用在导弹上的地球引力加速度在坐标系中的分量。用到的计算公式为

$$\dot{w}_x=\dot{w}_{x1}\cos\varphi-\dot{w}_{y1}\sin\varphi$$

$$\dot{w}_y=\dot{w}_{x1}\sin\varphi+\dot{w}_{y1}\cos\varphi$$

$$\dot{w}_{x1}=\frac{1}{m}(P-C_xqS_M)\quad C_x=0.51\quad S_M=3.2$$

$$\dot{w}_{y1}=\frac{1}{m}C_y^\alpha aqS_M\quad C_y^\alpha=2.40642066$$

$$P=P_{b0}\dot{G}+S_a(P_0-P)\quad P_{b0}=265\quad \dot{G}=175$$

$$p_0=10336\quad S_a=0.5$$

$$\alpha=\varphi-\theta$$

$$\theta=\mathrm{tg}^{-1}\frac{v_y}{v_x}$$

$$g_x=-g_0\left(\frac{R}{r}\right)^2\cdot\frac{x}{r}\quad g_0=9.81\quad R=6371110$$

$$g_y=-g_0\left(\frac{R}{r}\right)^2\frac{R+y}{r}$$

$$r=(x^2+(R+y)^2)^{1/2}$$

$$h=r-R$$

$$q=\frac{1}{2}\rho v^2$$

$$v^2=v_x^2+v_y^2$$

$$m=\frac{1}{g_0}\cdot(G_0-\dot{G}t)\quad G_0=16000$$

$$p=p_0\left(1-\frac{h}{44300}\right)^{5.256}$$

$$\rho=\rho_0\left(1-\frac{h}{44300}\right)^{4.256}\qquad \rho_0=0.125$$

$$\varphi=\begin{cases}1.5707963 & t\leqslant 4\text{秒}\\ 1.5707963-0.018007363(t-4)\\ \qquad+0.000071457791(t-4)^2 & t\geqslant 4\text{秒}\end{cases}$$

为了应用算法 10-6，我们将方程组（10-187）分解成两个子组

$$\begin{cases} \dfrac{dV_x}{dt}=\dot{w}_x+g_x \\ \dfrac{dx}{dt}=v_x \end{cases} \tag{10-188}$$

$$\begin{cases} \dfrac{dV_y}{dt}=\dot{w}_y+g_y \\ \dfrac{dy}{dt}=V_y \end{cases} \tag{10-189}$$

即取式（10-170）中的 $I=2$，$y_1=(V_x,\ x)$，$y_2=(V_y,\ y)$。从计算公式知道方程组（10-187）的右函数的主要计算集中在计算 $\dot{w}_{x1}$，$\dot{w}_{y1}$，所以我们取 $g_1=(\dot{w}_{x1},\ \dot{w}_{y1})$，应用三个处理机 P_x，P_y，P_g，其中 P_x 机积分式（10-188），P_y 机积分式（10-189），P_g 机计算 g_{1c}。

我们在 VAX-11/780 计算机上设置三个模块分别模拟 P_x，P_y，P_g 处理机，应用不同的进程控制进行试验。表 10-6 中列出其中一组试验结果，在试验中，P_x，P_y 机用到的 $\dot{w}_{x1}$，$\dot{w}_{y1}$ 的信息是 P_g 机的前两步的计算结果。P_x，P_y 机均采用步长 $h=0.125$ 的龙格-库塔积分方法，插值公式采用三点拉格朗日插值公式。作为比较，我们在表 10-7 中，列出应用传统的串行龙格-库塔公式步长 $h=0.125$ 的计算结果。将表 10-6 与表 10-7 进行比较，可知算法 10-6 是可行的。

表10-6　问题（10-187）的插值分解异步并行迭代算法计算结果

t	v_x	x	v_y	y
10	10.85879040	21.37198257	195.8977509	968.8706655
20	81.47441864	428.0340271	386.6062312	3896.641113
30	214.7883606	1858.430575	565.1720581	8655.095703

表10-7　问题（10-187）的串行龙格-库塔方法计算结果

t	v_x	x	v_y	y
10	10.85878849	21.37198448	195.8977356	968.8708496
20	81.47443390	428.0340576	386.6058655	3896.640869
30	214.7883148	1858.439209	565.1720581	8655.089844

10.5.3　插值分解异步并行向前算法

插值分解异步并行迭代算法每前进一步可得到若干个点上解的近似值，但是它要求进行异步迭代，这可能会增加计算的复杂性。现在我们讨论插值分解异步并行向前算法，它不需要进行异步迭代。

设问题（10-1）已分割成式（10-170），并且其右函数已分解成式（10-171）的形式。若在式（10-170）的第 i 个子组计算时，已知其它子组的解 $y_1(t)$, $y_2(t)$, …, $y_{i-1}(t)$, $y_{i+1}(t)$, … $y_I(t)$ 的近似函数 $\tilde{y}_1(t)$, $\tilde{y}_2(t)$, …, $\tilde{y}_{i-1}(t)$, $\tilde{y}_{i+1}(t)$, …, $\tilde{y}_I(t)$，并且知道函数 $g_j(t)=g_j(y_1(t), y_2(t), \cdots, y_I(t), t)$ 的近似函数 $\tilde{g}_j(t)$（$j=1, 2, \cdots, J$），将这些近似函数代入式（10-170）的第 i 个子组，得到其近似微分方程组

$$\frac{dy_i}{dt}=A_i(y_i, t), \quad y_i(t_0)=y_{i0} \tag{10-190}$$

其中

$$\begin{aligned} A_i(y_i, t)=F_i(t, &\tilde{g}_1(t), \tilde{g}_2(t), \cdots, \\ &\tilde{g}_J(t), \tilde{y}_1(t), \tilde{y}_2(t), \cdots, \tilde{y}_{i-1}(t), y_i, \\ &\tilde{y}_{i+1}(t), \cdots, \tilde{y}_I(t)) \end{aligned} \tag{10-191}$$

选取向前步的数值积分方法 ME_i 和数值积分步长 h_{il}，对组（10-190）进行数值积分。用 T_{yi} 表示对式（10-190）进行数值积分的格点集

$$T_{yi}=\{t_{yil} | t_{yil}=t_{yil-1}+h_{il}, \quad l=1, 2, \cdots, L_i, \ t_{i0}=t_0\}$$

记用上述数值积分得到的式（10-190）的解在格点 t_{yil} 处的近似值为 y_{il}。若已知 y_{i0}，y_{i1}，…，y_{il}，称用上面描述的方法得到 y_{il+1} 的计算为一个过程段，记为 P_{yil+1}。为了与其它组交换信息，设置整体变量 G_{yi}，它储存格点集 T_{yi} 中已达到的格点和相应的 y_{il} 及其导数值 y'_{il}。整体变量 G_{yi} 的构造由表 10-8 表示，其中 l_i 表示数值积分已达到的格点数，K_i 表示在 G_{yi} 中保留的当前格点前面的格点数。当过程段 $P_{yil_{j+1}}$ 结束时，对整体变量 G_{yi} 进行更新。设置整体变量 G_{yi} 的作用有两个：一个作用是为过程段 $P_{yil_{j-1}}$ 提供起始值，例如若方法 ME_i 是 K 步法，则在 G_{yi} 中至少必须保存 $t_{il_{j+1}}$ 前的 k 个点，即 K_i 至少取为 K；另外一个作用是由它保存的关于 $y_i(t)$ 的信息形成近似函数 $\tilde{y}_i(t)$。提供给其它过程段使用。

表10-8　整体变量 G_{yi} 的结构

$t_{yil_j-K_j}$	$t_{yil_j-K_j+1}$	$t_{yil_j-K_j+2}$	…	t_{yil_j-1}	t_{yil_j}
$y_{il_j-K_j}$	$y_{il_j-K_j+1}$	$y_{il_j-K_j+2}$	…	y_{il_j-1}	y_{il_j}
$y'_{il_j-K_j}$	$y'_{il_j-K_j+1}$	$y'_{il_j-K_j+2}$	…	y'_{il_j-1}	y'_{il_j}

设置整体变量 G_{gj}，与插值分解异步并行迭代算法一样，其构造由表 10-9 给出。计算点 $g_{jm_{j+1}}$ 的过程段记为 $P_{gjm_{j+1}}$，过程段 $P_{gjm_{j+1}}$ 的计算结束，对整体变量 G_{gj} 进行更新。量 $h_{gjm}=t_{gjm}-t_{gjm-1}$ 称作 G_{gj} 的步长。

表10-9　整体变量 G_{gj} 的构造

$t_{gjmj-rj}$	$t_{gjmj-rj+1}$	…	t_{gjmj}
g_{jm_j-rj}	$g_{jmj-rj+1}$	…	g_{jmj}

根据上面的构造，在每个时刻，我们有 $I+J$ 个可运行的过

程段，其中每个过程段运行结束将使相应的整体变量的值往前推进一步。按照一定的进程控制，将这 $I+J$ 个过程段分配给 P_r 台处理机进行并行计算，就得到插值分解异步并行向前算法，应用这种算法，每前进一步不需在各个过程段之间进行异步迭代，但是需要设置一个起步阶段。这是因为在开始计算时，整体变量 G_{yi}，G_{gj} 中的信息尚未构造，从而无法进行构造满足一定精度的近似函数 $\tilde{y}_i(t)$，$\tilde{g}_j(t)$。起步阶段可用 10.5.2 的插值分解异步并行迭代算法，它不需要起步阶段，也可以应用其它的串行算法。由于这部分计算仅占整个计算的极小部分，因而具体算法的选取不会影响算法的效能。

〔**例10-5**〕 仍考虑例10-4 中的简化导弹运动模型（10-187）和相应的分解。但采用这里构造的插值分解异步并行向前算法，在每个时刻有三个过程段 P_x，P_y 和 P_g，它们分别积分式（10-188）、（10-189）和计算 $\dot{w}_{x1}$，$\dot{w}_{y1}$ 的值，相应整体变量中的格点记为 t_{xi}，t_{yj}，t_{gk}，它们满足递推式

$$
\begin{aligned}
t_{xi} &= t_{x_{i-1}} + h_x \\
t_{yj} &= t_{y_{j-1}} + h_y \\
t_{g_k} &= t_{g_{k-1}} + h_g
\end{aligned}
$$

h_x，h_y，h_g 分别为计算步长。我们在 VAX-11/780 计算机上应用多处理机系统并行算法模拟器对上述算法进行模拟，多处理机系统处理机的台数 $P_r=2$。进程控制是这样的，设当前已计算到整体变量的格点分别为 t_{x_i}，t_{y_j}，t_{g_k}，记 $t_{min}=\{t_{x_i}, t_{y_j}, t_{g_k}\}$，用字母 a 代表 x，y，g 中的任何一个，则若有空闲处理机，按下面原则安排运行的过程段：

（1）正在运行的过程段不再安排；

（2）若 $t_{a_l} \geqslant t_{min}+2h_a$，则过程段 P_a 不安排运行，其中 l 代表 i，j，k 中相应于 a 的足标；

（3）在可以安排运行的程序段中任选相应于 t_a 最小的过程段运行。

若按这原则无过程段可以运行，则此处理机只能空闲，等待其它处理机上的过程段运行完后才有可能安排过程段在这处理机上运行。

表10-10列出一组模拟结果，其中取 $h_x=0.0625, h_y=0.125, h_z=0.0625$。与表10-7中的串行计算结果相比，插值分解异步并行向前算法是可行的。

表10-10 问题（10-187）的插值分解异步并行向前算法计算结果

t	v_x	x	v_y	y
10	10.85879230	21.37195587	195.8980408	968.8743896
20	81.47442627	428.0338135	386.6050720	3896.644531
30	214.7882538	1858.436279	565.1723022	8655.078125

应用模拟器 PARSIM 的结果表明，若采用例10-5中的进程控制方法，当各个过程段 $P_{yjl_{j+1}}$ 和 $P_{zjm_{j+1}}$ 的计算量近似相同时，将能达到很高的效率。而若计算量相差很大，处理机空闲等待时间会比较大，降低了使用多处理机系统的效率。

10.6 块 方 法

在使用龙格-库塔方法时，为了由 t_{i-1} 计算 t_i 处的方程的数值解，它同时计算了在 t_{i-1} 和 t_i 之间的一些中间点上的值。当然在这些点上的值的精度比在 t_i 处的精度要差一点。将这种思想推广，构造相应的计算方法，使得这些点为我们需要的格点，而在这些格点上解的近似值也具有所需要的精度。这种方法与传统的方法不同，每向前计算一步，得到的不是一个新格点上的值，而是一组格点上的值，块方法就是具有这种性质的方法。在并行计算机系统上实现这类方法时可以使得若干个点上的右函数值进行并行计算。

本节讨论的块方法与10.2节中讨论的预估校正并行算法有一定的相似性。10.2节讨论的方法也可以逐次确定一个块的所有格点上的解的近似值，但是这两者是有差别的。在10.2节的方法的每一个循环中，预估值是对一个块中的格点计算的，而校正值是对另外一个块的格点计算的。预估值的块总是超前校正值所在的块，因此对于这种方法，必须经过两个循环的计算才能确定一个块的所有格点上的预估校正值。本节讨论的方法通过一个循环的计算就能确定一个块上的解的近似值，这样可以使步长变简单一些。

10.3节中的方法也可以在一个计算循环中确定一个块的格点上解的近似值，但是在方法的推导思想上与计算的格式上与本节讨论的方法是不同的。

10.6.1 块单步方法

块方法的特征是这种方法的每一次应用，都产生一个等距格点块上解的近似值，即沿 t 轴向前推进一个格点块，这一节讨论的方法在向前推进时只依赖前面一个块上最后一点的值。

对于步长 h，令 $t_k=t_0+kh$，要求构造初值问题（10-1）的解 $y(t)$ 在格点序列 $\{t_k\}$ 上的近似 $\{y_k\}$。在得到值 y_k，$k=nr$ 后，求解方程

$$\sum_{j=1}^{r} a_{ij}y_{k+j}=e_iy_k+hd_if_n+h\sum_{j=1}^{r} b_{ij}f_{k+j}$$

$$i=1,\ 2,\ \cdots,\ r \qquad (10\text{-}192)$$

可以同时得到 r 个格点上的值 y_{k+1}，y_{k+2}，$\cdots$，y_{k+r}，方程(10-192)中的 a_{ij}，e_i，d_i，b_{ij} 都是常数，$f_{k+j}=f(t_{k+j},\ y_{k+j})$，这样构造的方法称作 r 点块方法。由于方程（10-192）中只用到上一块的最后一个点上的值，又称 r 点块单步法。记

$$A=(a_{ij}),\quad B=(b_{ij}),\quad e=(e_1,\ e_2,\ \cdots,\ e_r)^T,$$

$$d=(d_1,\ d_2,\ \cdots,\ d_r)^T$$

和

$$Y_m=(y_{k+1},\ y_{k+2},\ \cdots,\ y_{k+r})^T,$$
$$F(Y_m)=(f_{k+1},\ f_{k+2},\ \cdots,\ f_{k+r})^T$$

则方法（10-192）可表成

$$AY_m=hBF(Y_m)+ey_k+hdf_k \tag{10-193}$$

我们假定矩阵A为单位矩阵。而只讨论形状为

$$Y_m=hBF(Y_m)+ey_k+hdf_k \tag{10-194}$$

的方法，因为必要时可以在式（10-193）的两边左乘A^{-1}，即可化为式（10-194）。

对于公式（10-192）、（10-193）、（10-194）中的表示，我们作如下的说明。首先，若y和f是标量，则这些公式按通常的向量运算；但若y和f均是n维向量时，则这些公式的每一个都代表n个公式，其中每个公式都是将y和f用其相应的分量代替后得到的。这种说明对10.6.2和10.6.3也适用。

下面我们讨论确定方程（10-194）中的系数B，e，d的方法，首先将问题（10-1）化成等价的积分方程

$$y(t)=y_k+\int_{t_k}^{t} f(t,\ y(t))dt \tag{10-195}$$

然后用过点$(t_k,\ f(t_k,\ y_k))$，$(t_{k+1},\ f(t_{k+1},\ y_{k+1}))$，…，$(t_{k+r},\ f(t_{k+r},\ y_{k+r}))$的$r$次拉格朗日插值多项式$L(t)$近似被积函数$f(t,\ y(t))$，取积分

$$y_{k+1}=y_k+\int_{t_k}^{t_{k+1}} L(t)dt \tag{10-196}$$

为$y(t_{k+1})$的近似将式（10-196）中的积分直接积出就可得到形式为式（10-194）的方程，从而确定了其中的系数B，e，d.

特别有

$$e=(1,\ 1,\ \cdots,\ 1)^T$$

应用这种方法确定的公式称作牛顿-柯特斯（Newton-Cotes）公式。我们将 $r=1,\ 2,\ \cdots,\ 8$ 的公式（10-194）的 B 和 d 的值列在下面

$$r=1,\ d_1=\frac{1}{2},\quad b_{11}=\frac{1}{2}$$

$r=2$

i	d_i分子	b_{i1}分子	b_{i2}分子	分母
1	5	8	−1	12
2	1	4	1	3

$r=3$

i	d_i分子	b_{i1}分子	b_{i2}分子	b_{i3}分子	分母
1	9	19	−5	1	24
2	1	4	1	0	3
3	3	9	9	8	8

$r=4$

i	d_i分子	b_{i1}分子	b_{i2}分子	b_{i3}分子	b_{i4}分子	分母
1	251	646	−264	106	−19	720
2	29	124	24	4	−1	90
3	27	102	72	42	−3	80
4	14	64	24	64	14	45

$r = 5$

i	d_i分子	b_{i1}分子	b_{i2}分子	b_{i3}分子	b_{i4}分子	b_{i5}分子	分母
1	475	1427	−798	482	−173	27	1440
2	28	129	14	14	− 6	1	90
3	51	219	114	114	−21	3	160
4	14	64	24	64	14	0	45
5	95	375	250	250	375	95	288

$r = 6$

i	d_i分子	b_{i1}分子	b_{i2}分子	b_{i3}分子	b_{i4}分子	b_{i5}分子	b_{i6}分子	分 母
1	19087	65112	−46461	37504	−20211	6312	−863	60480
2	1139	5640	33	1328	−807	264	−37	3780
3	685	3240	1161	2176	−729	216	−29	2240
4	286	1392	384	1504	174	48	− 8	945
5	3715	17400	6375	16000	11625	5640	−275	12096
6	41	216	27	272	27	216	41	140

$r = 7$

i	d_i分子	b_{i1}分子	b_{i2}分子	b_{i3}分子	t 分子	b_{i5}分子	b_{i6}分子	b_{i7}分子	分母
1	36799	139849	−121797	123133	−88547	41499	−11351	1357	120960
2	1107	5864	−639	2448	−1927	936	−261	32	3780
3	1325	6795	1377	5927	−3033	1377	−373	45	4480
4	278	1448	216	1784	−106	216	−64	8	945
5	7155	36725	6975	41625	13625	17055	−2475	275	24192
6	41	216	27	272	27	216	41	0	140
7	5257	25039	9291	20923	20923	9261	25039	5257	17280

$r = 8$

i	d_i分子	b_{i1}分子	b_{i2}分子	b_{i3}分子	b_{i4}分子	b_{i5}分子	b_{i6}分子	b_{i7}分子	b_{i8}分子	分母
1	1070017	4467094	4604594	5595358	−5033120	3146338	−1291214	312874	−33953	3628800
2	32377	182584	−42494	120088	−116120	74728	−31154	7624	−833	113400
3	12881	70902	3438	79934	−56160	34434	−14062	3402	−369	44800
4	4063	22576	244	32752	−9080	9232	−3956	976	−107	14175
5	41705	230150	7550	318350	−4000	170930	−49150	11450	−1225	145152
6	401	2232	18	3224	360	2664	158	72	−9	1400
7	149527	816634	48706	1085938	54880	736078	522046	223174	−8183	518400
8	3956	23552	−3712	41984	−18160	41984	−3712	23552	3956	14175

文献〔34〕证明 $r=1$，2，…，8 的牛顿-柯特斯公式是收敛的，并且是 A-稳定的，这些公式的局部截断误差的阶由表10-11给出，当 r 是偶数时，每一块的最后一点的误差阶比表10-10中的提高一阶。

表10-11　牛顿-柯特斯公式（8-194）的局部截断误差阶

r	1	2	3	4	5	6	7	8
误差阶 $p(r)$	3	4	5	6	7	8	9	10

方程（10-194）是 Y_m 的非线性方程，求解它需要一个迭代过程，采用简单迭代法，我们得到下面的算法。

算法10-7（简单迭代法计算过程）

步 1　给出 y_{k+1}，y_{k+2}，…，y_{k+r} 的迭代初值 $y_{k+1}^{(0)}, y_{k+2}^{(0)}, \cdots, y_{k+r}^{(0)}$，组成 $Y_m^{(0)}=(y_{k+1}^{(0)}, y_{k+2}^{(0)}, \cdots, y_{k+r}^{(0)})^T$，并令 $l=0$。

步 2　计算

$$Y_m^{(l+1)}=hBF(Y_m^{(l)})+ey_k+hdf_k \qquad (10\text{-}197)$$

步 3　判别 $Y_m^{(l+1)}$ 是否满足精度，若不满足精度，则令 $l=l+1$，转步 2；若满足精度，则令 $Y_m=Y_m^{(l+1)}$，这 r 个点的块迭代结束。

在用公式（10-197）进行迭代时，r 个右端函数值 $f(t_{k+j}, y_{k+j}^{(l)})(j=1, 2, \cdots r)$ 可以进行并行计算。若右端函数的计算很复杂时，这种迭代可达到较高的并行度。

容易证明，若 $Y_m^{(l)}$ 的精度阶低于公式（10-194）的精度阶 $p(r)$，则 $Y_m^{(l+1)}$ 的精度阶将比 $Y_m^{(l)}$ 的精度阶提高一阶。因此若 $Y_m^{(0)}$ 的精度阶为 p_0，则用式（10-197）迭代 $p(r)-p_0$ 次就能达到要求的阶，再迭代将不能提高精度阶，但可改善其数值稳定性。

迭代式（10-197）收敛的条件为

$$\left\| hB\frac{\partial F(Y_m)}{\partial Y_m} \right\| < 1 \qquad (10\text{-}198)$$

当问题（10-1）为刚性问题时，条件（10-198）将限制数值积分的步长，为了去掉这个限制，可以采用牛顿迭代，但这需要计算函数 $f(t, y)$ 的雅可比矩阵，和求矩阵 $\left(I - hB\dfrac{\partial F(Y_m)}{\partial Y_m}\right)$ 的逆矩阵。当 n 很大时，这将花费很大的计算量。

10.6.2 并行块预估校正方法

在10.6.1中，应用 r 点块单步法，算法10-7每运行一次可得到 r 个格点上组（10-1）的解的近似值。在算法10-7的步 1 中，要求给出迭代初值。最简单的初值给出方法为

$$y_{k+j}^{(0)} = y_k + jhf_k \quad j = 1, 2, \cdots, r$$

但是它的精度阶太低，文献〔35〕〔39〕均给出 $r = 2$ 的较高精度的预估值。其基本思想类似于线性多步公式的预估校正方法，即利用上一块的所有结果来预估当前块各个格点上的值。现在我们来构造与公式（10-194）一致的块预估校正方法，简称 BPC 记

$$\begin{aligned}\bar{Y}_{m-1} &= (y_{k-1}, y_{k-2}, \cdots, y_{k-r})^T F(\bar{Y}_{m-1}) \\ &= (f_{k-1}, f_{k-2}, \cdots, f_{k-r})^T\end{aligned}$$

其中 y_{k-j}，f_{k-j} 均是前面已计算的值。将公式（10-194）改写成

$$Y_m = hBF(Y_m^p) + ey_k + hdf_k \tag{10-199}$$

其中 $Y_m^p = (y_{k+1}^p y_{k+2}^p, \cdots, y_{k+r}^p)$ 是预估值，由公式

$$Y_m^p = h\bar{B}F(Y_{m-1}) + \bar{A}Y_{m-1} + \bar{e}y_k + h\bar{d}f_k \tag{10-200}$$

给出。其中 $\bar{e}$，$\bar{d}$ 均是 r 维向量，$\bar{A}$，$\bar{B}$ 均是 $r \times r$ 矩阵。由公式（10-200）、（10-199）组成块预估校正方法。称式（10-200）为预估公式，而称式（10-199）为校正公式。块预估校正方法的并行计算格式为

算法10-8（并行块预估校正算法）

步 0　给出 $y_0, y_1, \cdots, y_r, f_0, f_1, \cdots, f_r$，令 $k = r$，$m = 1$。

步 1　按公式（10-200）计算预估值 Y_m^p

步 2　并行计算 $f^{p}_{k+j}=f(t_{k+j},\ y^{p}_{k+j})$，$j=1,\ 2,\ \cdots,\ r$

步 3　按公式（10-199）计算校正值 Y_m。

步 4　若 t_{mr+r} 为求解的终端，则停止计算，否则并行计算 $f_{k+j}=f(t_{k+j},\ y_{k+j})$（$j=1,\ 2,\ \cdots,\ r$），令 $k=mr+r$，$m=m+1$，转步 1。

在这个算法中步 2 和步 4 中右端函数可以在不同的处理机中进行并行计算，由于各个处理机的运算量是相同的，仅是函数的变量值不同，因此若各个处理机相同时，本质上不存在同步问题。

公式（10-200）中的 $\bar{e}$，$\bar{d}$，$\overline{A}$，$\overline{B}$ 可采用待定系数法来确定。即确定这些量使公式（10-200）对于函数 $y(t)=(t-t_k)^l$（$l=0,\ 1,\ 2,\ \cdots,\ r+1$）是精确的。这样，可得到关系式

$$\begin{cases}\bar{e}=(I-\overline{A})\bar{u}\\ \bar{d}=(I+\overline{A})u-\overline{B}\bar{u}\\ \overline{B}=(W\bar{l}+\overline{A}W)U^{-1}\end{cases}\qquad(10\text{-}201)$$

其中 $\bar{u}$ 是每个分量都是 1 的 r 维向量，u 是第 l 个分量为 l 的 r 维向量，矩阵 $U=[u^1,\ u^2,\ \cdots,\ u^r]$，$u^l$ 是将 u 的每个分量用其 l 次幂代替得到的向量，矩阵 $W=[w^2,\ w^3,\ \cdots,\ w^{r+1}]$，$w^l=\left(-\frac{1}{l}\right)u^l$，由范德蒙（Vandermonde）矩阵的非奇性推得矩阵 U 的逆矩阵 U^{-1} 存在，$\bar{l}$ 是 $r\times r$ 对角线矩阵，其第 l 个对角线元素为 $(-1)^l$。I 为单位矩阵。

选取公式（10-201）中的矩阵 $\overline{A}$ 可以改进整个求解过程的品质。文献[35]中的选取方法是选取 $\overline{A}$ 的元素使舍入误差达到极小值，应用文献[43]提出的误差模型，推得 $\overline{A}$ 的所有的元素均是 $1/(r+1)$，我们称这样得到的 BPC 方法为等权预估方法（简称 EWP）。

[例10-6]　$r=2$ 的等权预估方法（EWP）

$\bar{u}=(1,\ 1)^T$，$u=(1,\ 2)^T$，

$$U=\begin{bmatrix}1 & 1\\ 2 & 4\end{bmatrix},\quad W=\begin{bmatrix}-\frac{1}{2} & \frac{1}{3}\\ 2 & -\frac{8}{3}\end{bmatrix},\quad U^{-1}=\begin{bmatrix}2 & -\frac{1}{2}\\ -1 & \frac{1}{2}\end{bmatrix}$$

$$\overline{A}=\begin{bmatrix}-\frac{1}{3} & \frac{1}{3}\\ \frac{1}{3} & -\frac{1}{3}\end{bmatrix}\qquad \overline{B}=\begin{bmatrix}-\frac{2}{3} & \frac{1}{2}\\ -\frac{72}{12} & \frac{29}{12}\end{bmatrix}$$

$$\bar{e}=\left(\frac{1}{3},\ -\frac{1}{3}\right)^T\quad \bar{d}=\left(\frac{13}{6},\ \frac{79}{12}\right)^T$$

预估公式为

$$y_{k+1}^p=\frac{1}{3}(y_{k-2}+y_{k-1}+y_k)+\frac{h}{6}(3f_{k-2}-4f_{k-1}+13f_k)$$

(10-202)

$$y_{k+2}^p=-\frac{1}{3}(y_{k-2}+y_{k-1}+y_k)+\frac{h}{12}(29f_{k-2}-72f_{k-1}+79f_k)$$

另外一种选取 $\overline{A}$ 的思想是为了提高算法的稳定性品质，为此，文献〔40〕提出直接选 $\overline{A}$ 为零矩阵的方法，称作零权预估方法（简称 NWP）。

〔例10-7〕 $r=2$ 的零权预估方法（NWP）

向量 $\bar{u}$，u，矩阵 U，W，U^{-1} 与例10-6中相同。取 $\overline{A}=0$，则

$$\overline{B}=W\bar{1}U^{-1}=\begin{bmatrix}-\frac{4}{3} & -\frac{5}{12}\\ -\frac{20}{3} & \frac{7}{3}\end{bmatrix}$$

$$\bar{d}=\left(\frac{23}{12}\quad -\frac{19}{3}\right)^T\quad \bar{e}=(1,\ 1)^T$$

预估公式为

$$y_{k+1}^p=y_k+\frac{h}{12}(5f_{k-2}-16f_{k-1}+23f_k)$$

(10-203)

$$y_{k+2}^p=y_k+\frac{h}{3}(7f_{k-2}-20f_{k-1}+19f_k)$$

对EWP方法和NWP方法进行稳定性分析可以看出，NWP方法确实比EWP方法具有较好的稳定性品质。我们将块预估校正方法（10-200）、（10-199）应用到单个方程 $y'(t)=\lambda y(t)$，$\lambda<0$，不管初值 $y(0)$ 取何值，当 t 增加时，此方程的任何解都将收敛到零。因此，要求由数值方法生成的该方程的近似解也收敛到零，这种要求为算法的基本稳定性特征。当块预估校正方法中的系数向量和矩阵取定后，只能调整公式中的积分步长 h 使算法满足这种要求。我们先推导保证块预估校正方法具有基本稳定性特征的条件。

记：$\tilde{Y}_m=(y_{mr},\ y_{mr+1},\ y_{mr+2},\ \cdots,\ y_{(m+1)r})^T$

由公式（10-200）、（10-199）推得

$$\tilde{Y}_m=\Phi\tilde{Y}_{m-1} \tag{10-204}$$

其中

$$\Phi=(\Phi_1+\mu\Phi_2)(\Phi_3+\mu\Phi_4)\,\psi \qquad \mu=\lambda h$$

$$\Phi_1=\left[\begin{array}{c|c} 1 & 0 \\ \hline e & 0 \end{array}\right] \qquad \Phi_2=\left[\begin{array}{c|c} 0 & 0 \\ \hline d & B \end{array}\right]$$

$$\Phi_3=\left[\begin{array}{c|c} 1 & 0 \\ \hline \bar{e} & \bar{A} \end{array}\right] \qquad \Phi_4=\left[\begin{array}{c|c} 0 & 0 \\ \hline \bar{d} & \bar{B} \end{array}\right]$$

ψ 是 $(r+1)\times(r+1)$ 矩阵，它是将单位矩阵的列向量按相反次序排列得到的。若 Φ 的特征值的模都小于1，则能保证当 $m\to\infty$ 时，有 $\tilde{Y}_m\to 0$。我们寻找最大的 $H\geqslant 0$，使当 $\mu\in(-H,\ 0)$ 时这个条件成立，数 H 称作积分公式的稳定性界。

对于 $r=2,\ 3,\ \cdots,\ 8$，表10-12列出EWP和NWP方法相应的 H 值，以及它们的比值。这些结果表明从数值稳定性的角度来看，NWP方法优于EWP方法，文献〔40〕中的数组试验结果也证实了这一点。

算法10-7步1中的 $Y_m^{(3)}$ 可采用式（10-200）给出的 $\tilde{Y}_m^p$。应

表10-12　NWP方法和EWP方法的稳定性界及比值

方法 \ r (H)	2	3	4	5	6	7	8
NWP	0.576	0.326	0.222	0.168	0.135	0.114	0.098
EWP	0.439	0.311	0.179	0.149	0.088	0.070	0.042
NWP/EWP	1.31	1.05	1.24	1.13	1.53	1.63	2.33

用式（10-197）进行迭代时，若取 $Y_m=Y_m^{(1)}$ 得到的算法与算法10-8等价。我们还可以取 $Y_m=Y_m^{(2)}$，…，$Y_m=Y_m^{(l)}$。得到相应的预估和 l 次校正算法。

由于公式（10-200）和公式（10-194）的局部截断误差分别为

$$T_i^p=C_i^p h^{r+2}y^{(r+2)}(\xi_i^p)\quad i=mr+j,$$
$$j=1,\ 2,\ \cdots,\ r\quad \xi_i^p\in[t_{mr-r},\ t_{mr+r}]\qquad(10\text{-}205)$$
$$T_i=C_i h^{r+2}y^{(r+2)}(\xi_i)\quad i=mr+j,\ j=1,\ 2,\ \cdots,\ r$$
$$\xi_i\in[t_{mr},\ t_{mr+r}]\qquad(10\text{-}206)$$

其中 C_i^p，C_i 是常数，因此公式（10-200）与公式（10-194）是同阶的。校正次数增多不能提高计算的误差阶，但可以改善其数值稳定性。表10-13给出 $r=2$ 时应用EWP公式预估而分别校正 $l=1$，2，3，4的方法的稳定性界 H 的值。随 l 增大，H 的值也增大。

表10-13　$r=2$ EWP公式预估、校正 l 次的稳定性界

l	1	2	3	4
H	0.439	0.694	0.934	1.293

在上面对块方法进行讨论时，假定积分步长 h 在整个积分过程中是固定的。但是为了使计算更加合理有效，应该在计算过程中能根据局部截断误差的变化调整积分步长 h，使这个误差不会大得不能接受，也不会过分小得使计算量太大。调整积分步长的

策略通常先估计比值 $R_i=|e_i/\hat{\varepsilon}_i|$，其中 e_i 是 y_i 的局部截断误差的估计，$\hat{\varepsilon}_i$ 是允许的局部截断误差的界。如果已得到 r 个新的值 $y_i(i=mr+j,\ j=1,\ 2,\ \cdots,\ r)$，估计 $R=\max\{R_i\}$，若 $R\leqslant 1$，r 个新计算值可接受。如果 $R>1$，表示误差太大，需要缩小积分步长，重新进行计算。根据沙姆品（Shampine）和瓦茨（Watts）对局部误差性质的分析，为了达到要求的误差界，下一次计算“最好的”步长应该取成

$$h_{新}=\sigma h_{旧},\quad \sigma=\left(\frac{1}{R}\right)^{\gamma},\quad \gamma=\frac{1}{P(r)}$$

这里下一次计算是指缩小步长重新计算（若 $R>1$）或计算下一块上点的值（若 $R\leqslant 1$）。

对于预估校正方法，局部截断误差的估计通常取成

$$\varepsilon_i=\mu_i(y_i-y_i^p) \tag{10-207}$$

其中 μ_i 为常数，μ_i 的最简单的取法为取 $\mu_i=1$，另外一种取法可以由估计式（10-205）、（10-206）得到。假定前一块得到的所有值是精确的，对于 $i=mr+j$，$j=1,\ 2,\ \cdots r$，由式(10-205)、(10-206)

$$y(t_i)=y_i^p+C_i^p h^{r+2}y^{(r+2)}(\xi_i^p)=y_i+C_i h^{r+2}y^{(r+2)}(\xi_i)$$

因此

$$y_i-y_i^p=C_i^p h^{r+2}y^{(r+2)}(\xi_i^p)-C_i h^{r+2}y^{(r+2)}(\xi_i)$$

如果假定 $y^{(r+2)}(t)$ 在区间 $[t_{mr-r},\ t_{mr+r}]$ 上不变，在 t_i 处的局部截断误差可用 $y_i-y_i^p$ 表示成

$$e_i=\frac{C_i}{C_i^p-C_i}(y_i-y_i^p)\quad i=mr+1,\ \cdots,\ mr+r \tag{10-208}$$

于是可取式（10-207）中的 $\mu_i=C_i/(C_i^p-C_i)$。

为了应用式（10-208）还需要确定常数 C_i 和 C_i^p。由于常数 C_i 和 C_i^p 与 $y(t)$ 无关，我们设 $y(t)=t^{r+2}$ 为问题（10-1）的解。将式（10-194）应用到第一块的点，有

$$Y=(r+2)h^{r+2}Bu^{r+1}$$

另外有

$$Y^*-Y=(r+2)!h^{r+2}C$$

其中

$$Y^*=(y(t_1),\ y(t_2),\ \cdots,\ y(t_r))^T$$
$$C=(C_1,\ C_2,\ \cdots,\ C_r)^T$$

由于 $Y^*=h^{r+2}u^{r+2}$，推出

$$(r+2)!h^{r+2}C=h^{r+2}u^{r+2}-(r+2)h^{r+2}Bu^{r+1}$$

从而得到C的显式表示

$$C=\frac{1}{(r+2)!}[u^{r+2}-(r+2)Bu^{r+1}] \tag{10-209}$$

类似地，对公式（10-200）进行处理，得到：

$$C^p=\frac{1}{(r+2)!}[(I-(-1)^r\overline{A})u^{r+2}$$
$$+(-1)^r(r+2)\overline{B}u^{r+1}] \tag{10-210}$$

其中

$$C^p=(C_1^p,\ C_2^p,\ \cdots,\ C_r^p)^T$$

经调整步长后，新块中的计算步长与前面块中的步长可能不同，但是，在推导预估公式（10-200）时，假定这两块中的步长是相同的，这样，在步长发生变化时，即步长由h变成σh时，预估公式（10-200）将不能应用，应该对其系数向量和矩阵作相应的变动。下面以NWP预估公式为例指出这种变动的步骤。

将式（10-200）中的第i个方程记成（$\overline{A}=0$）

$$y_{mr+i}^p=y_k+h\left(\sum_{j=0}^{r}b_{ij}f_{mr-j}\right) \tag{10-211}$$

其中$b_{i0}=\overline{d}_i$。令$P(t)$是对点$(t_{mr-j},f_{mr-j})(j=0,1,2,\cdots,r)$构造的拉格朗日插值多项式

$$P(t)=\sum_{j=0}^{r}f_{mr-j}v_j(s) \tag{10-212}$$

其中

$$s=(t-t_{mr})/h,\quad v_j(s)=\prod_{\substack{l=0\\l\neq j}}^{r}\frac{(s+l)}{(l-j)}$$

应用多项式$P(t)$，公式（10-211）可以改写成

$$y^p_{mr+i}=y_{mr}+\int_{t_{mr}}^{t_{mr+i}}P(t)dt \tag{10-213}$$

记

$$V_j(s)=\int_0^s v_j(s)ds$$

则由式（10-213），将式（10-214）中的系数b_{ij}取成

$$b_{ij}=V_j(i\sigma)\quad i=1,\ 2,\ \cdots,\ r,$$
$$j=0,\ 1,\ \cdots,\ r \tag{10-214}$$

就能适应步长由h调整到σh的变化。

10.6.3 并行多块方法

将线性多步方法的思想应用到块方法，可以得到并行多块方法。首先给出k-块r-点方法的概念。对于问题（10-1）和自然数m、r，仍记

$$Y_m=(y_{mr+1},\ y_{mr+2},\ \cdots,\ y_{mr+r})^T$$
$$F(Y_m)=(f_{mr+1},\ f_{mr+2},\ \cdots,\ f_{mr+r})^T$$

一般的k-块r-点方法是指形式为

$$Y_m=\sum_{i=1}^{K}A_iY_{m-i}+h\sum_{i=0}^{K}B_iF(Y_{m-i}) \tag{10-215}$$

的矩阵有限差分方程，其中A_i和B_i都是适当选定的$r\times r$矩阵，式（10-215）中的运算具有式（10-194）后面的说明。

如果$r=1$，则式（10-215）是线性k-步方法，当$B_0=0$时，称式（10-215）是显式方法，否则是隐式的。

令

$$Z_m=(y(t_{mr+1}),\ y(t_{mr+2}),\ \cdots,\ y(t_{mr+r}))^T$$

其中$y(t)$是问题（10-1）的解。称量

$$E_m=Z_m-\sum_{i=1}^{K}A_iZ_{m-i}-h\sum_{i=0}^{K}B_iF(Z_{m-i}) \qquad (10-216)$$

为方法（10-215）在第m个块处的局部截断误差。

在确定方法（10-215）中的系数矩阵A_i，B_i时，我们总是先给定局部截断误差E_m应达到的精度阶，然后选取A_i和B_i中的元素。选取的方法有许多，常用的有泰勒展开方法，多项式插值方法和待定系数法。由于A_i和B_i均是$r\times r$矩阵，所含可选择的元素比要达到的精度阶所必需的量要多得多，因此可只用其部分元素，而直接令其余的元素为零。在例10-8到例10-11中给出k-块r-点方法的一些例子及相应局部截断误差的主项，其中a_i，b_i可任意选取。

〔**例10-8**〕 $k=2$，$r=2$ 显式部分块信息方法

$$\begin{aligned}Y_m=&\begin{bmatrix}0 & 1+a_1+a_2\\ 0 & 1+a_3+a_4\end{bmatrix}Y_{m-2}+\begin{bmatrix}-a_1 & -a_2\\ -a_3 & -a_4\end{bmatrix}Y_{m-1}\\&+\frac{h}{12}\begin{bmatrix}0 & 9+5a_1+4a_2\\ 0 & 32+5a_3+4a_4\end{bmatrix}F(Y_{m-2})\\&+\frac{h}{12}\begin{bmatrix}8a_1+16a_2 & 27-a_1+4a_2\\ -64+8a_3+16a_4 & 80-a_3+4a_4\end{bmatrix}F(Y_{m-1})\end{aligned} \qquad (10-217)$$

$$\text{局部截断误差}=\frac{1}{72}\begin{bmatrix}(27+3a_1)h^4y^{(4)}\\ (192+3a_3)h^4y^{(4)}\end{bmatrix} \qquad (10-218)$$

〔**例10-9**〕 $K=1$，$r=2$ 隐式低阶方法

$$\begin{aligned}Y_m=&\begin{bmatrix}1+a_1 & -a_1\\ 1+a_2 & -a_2\end{bmatrix}Y_{m-1}\\&+\frac{h}{72}\begin{bmatrix}56+46a_1-24b_1 & 72b_1\\ 27+27a_2 & 81+57a_2\end{bmatrix}F(Y_{m-1})\\&+\frac{h}{72}\begin{bmatrix}120+42a_1-72b_1 & -32-16a_1+24b_1\\ 81-15a_2 & 27+3a_2\end{bmatrix}F(Y_m)\end{aligned} \qquad (10-219)$$

$$\text{局部截断误差}=\frac{1}{720}\begin{bmatrix}(320+190a_1-240b_1)h^4y^{(4)}\\(-27-19a_2)h^5y^{(5)}\end{bmatrix} \tag{10-220}$$

〔例10-10〕 $K=2$ $\gamma=2$ 显式全块信息方法

$$Y_m=\begin{bmatrix}1+a_1+a_2+a_3 & -a_1\\ 1+a_4+a_5+a_6 & -a_4\end{bmatrix}Y_{m-2}+\begin{bmatrix}-a_2 & -a_3\\ -a_5 & -a_6\end{bmatrix}Y_{m-1}$$
$$+\frac{h}{24}\begin{bmatrix}9a_1+8a_2+9a_3 & 64+19a_1+32a_2+27a_3\\ -55+9a_4+8a_5+9a_6 & 275+19a_4+32a_5+27a_6\end{bmatrix}F(Y_{m-2})$$
$$+\frac{h}{24}\begin{bmatrix}-32-5a_1+8a_2+27a_3 & 64+a_1+9a_3\\ -325-5a_4+8a_5+27a_6 & 225+a_4+9a_6\end{bmatrix}F(Y_{m-1}) \tag{10-221}$$

$$\text{局部截断误差}=\frac{1}{720}\begin{bmatrix}(224-19a_1-8a_2-27a_3)h^6y^{(6)}\\(2125-19a_4-8a_5-27a_6)h^5y^{(6)}\end{bmatrix} \tag{10-222}$$

〔例10-11〕 $K=1$ $\gamma=2$ 隐式高阶方法

$$Y_m=\begin{bmatrix}1+a_1 & -a_1\\ 1+a_2 & -a_2\end{bmatrix}Y_{m-1}$$
$$+\frac{h}{24}\begin{bmatrix}8+9a_1 & 32+19a_1\\ 9+9a_2 & 27+19a_2\end{bmatrix}F(Y_{m-1})$$
$$+\frac{h}{24}\begin{bmatrix}8-5a_1 & a_1\\ 27-5a_2 & 9+a_2\end{bmatrix}F(Y_m) \tag{10-223}$$

$$\text{局部截断误差}=\frac{1}{720}\begin{bmatrix}(-8-19a_1)h^5y^{(5)}\\(-27-19a_2)h^5y^{(5)}\end{bmatrix} \tag{10-224}$$

这里要指出，如果取 $a_1=a_2=a_3=a_4=-\frac{1}{3}$，则方法(10-217)给出例10-6的预估公式(10-202)。如果取 $a_1=a_3=0$，$a_2=a_4=-1$，则方法(10-217)给出例10-7的预估公式(10-203)。

为了构成并行多块方法的计算格式，选取显式的多块方法作为预估方法，选取隐式的多块方法作为校正方法，并且选定校正的次数 L，将预估公式的系数矩阵记为 A_i^p、B_i^p，步数记为 K^p。于是当已知 Y_{m-i}，$F(Y_{m-i})$，$i=1$，2，$\cdots$，K^p 时，计算 Y_m 的并行多块方法的计算格式为

算法10-9（并行多块方法）

步 1　计算

$$Y_m^{(l)0}=\sum_{i=1}^{K^p}A_i^pY_{m-i}+h\sum_{i=1}^{K^p}B_i^pF(Y_{m-i})$$

并令 $l=0$。

步 2　并行计算

$$f_{mr+j}^{(l)}=f(t_{mr+j},\ y_{mr+j}^{(l)}),\quad j=1,\ 2,\ \cdots,\ r$$

并构成 $F(Y_m^{(l)})=(f_{mr+1}^{(l)},\ f_{mr+2}^{(l)},\ \cdots,\ f_{mr+r}^{(l)})^T$，如果 $l=L$，则令

$$Y_m=Y_m^{(l)}\quad F(Y_m)=F(Y_m^{(l)})$$

并且停止这一块的计算；否则转步 3。

步 3　计算

$$Y_m^{(l+1)}=\sum_{i=1}^{K}A_iY_{m-i}+h\sum_{i=1}^{K}B_iF(Y_{m-i})+hB_0F(Y_m^{(l)})$$

令 $l=l+1$，转步 2。

在这个算法中，r 个右端函数可以并行计算，其它的运算主要是向量运算，因而具有较高的并行程度。算法10-9实际是线性多步方法计算中的 $PE(CE)^L$ 格式。

对于并行多块方法来说，稳定性问题是其实际应用的最大的限制，需要寻找公式中的自由参数提高它的稳定性品质。为了进行稳定性分析，我们假定 $K^p=K+1$，记

$$U_m^{(l)}=\begin{bmatrix}Y_m^{(l)}\\Y_{m-1}\\\vdots\\Y_{m-k}\end{bmatrix}\qquad V_m^{(l)}=\begin{bmatrix}F(Y_m^{(l)})\\F(Y_{m-1})\\\vdots\\F(Y_{m-k})\end{bmatrix}$$

并令

$$\widetilde{A}^p=\begin{bmatrix} A_1^p & A_2^p\cdots A_K^p & A_{K+1}^p \\ I & 0 \quad 0 & 0 \\ \vdots & 0 & 0 \\ 0 & I & 0 \end{bmatrix}$$

$$\widetilde{B}^p=\begin{bmatrix} B_1^p & B_2^p\cdots B_K^p & B_{K+1}^p \\ 0 & 0 \cdots 0 & 0 \\ \vdots & & \vdots \\ 0 & & 0 \end{bmatrix}$$

$$A=\begin{Bmatrix} A_1 & A_2\cdots A_K & 0 \\ I & 0 \quad 0 & 0 \\ 0 & & \vdots \\ \vdots & 0 & 0 \\ 0 & \cdots \quad I & 0 \end{Bmatrix}$$

$$B=\begin{bmatrix} B_0 & B_1\cdots B_{K-1} & B_K \\ 0 & 0 \quad 0 & 0 \\ \vdots & & \vdots \\ 0 & \cdots & 0 \end{bmatrix}$$

量 A^p，B^p，A，B 都是 $(K+1)r\times(K+1)r$ 矩阵。由算法10-9得

$$U_m^{(0)}=A^pU_{m-1}^{(L)}+hB^pV_{m-1}^{(L)} \qquad (10\text{-}225)$$

$$U_m^{(l+1)}=AU_{m-1}^{(l)}+hBV_m^{(l)} \qquad (10\text{-}226)$$

将式（10-225）、（10-226）应用到单个方程 $y'(t)=\lambda y(t)$，$\lambda<0$ 得到

$$U_m^{(0)}=(A^p+\mu B^p)U_{m-1}^{(l)}$$

$$U_m^{(l+1)}=AU_{m-1}^{(l)}+\mu BU_m^{(l)} \qquad \mu=\lambda h$$

因此可以得到递推式

$$U_m^{(l)}=T(\mu,\ A,\ B,\ A^p,\ B^p)U_{m-1}^{(l)} \qquad (10\text{-}227)$$

其中

$$T(\mu,\ A,\ B,\ A^p,\ B^p)=[I+(\mu B)+\cdots+(\mu B)^{k-1}]A+(\mu B)^k(A^p+\mu B^p)$$

若矩阵 A，B，A^p，B^p 给定，递推式（10-227）绝对稳定的充要条件是，量 μ 使不等式

$$\rho(T(\mu,\ A,\ B,\ A^p,\ B^p))<1 \qquad (10-228)$$

成立，其中 $\rho(T)$ 表示矩阵 T 的谱半径。也就是说矩阵 $T(\mu, A, B, A^p, B^p)$ 的所有特征值的模都小于 1。令 $H_a=\max\{H \mid H>0\}$，当 $\mu\in(-H,\ 0)$ 时，式（10-228）成立，它是递推式（10-227）的稳定性界。

对于给定的矩阵 A，B，A^p，B^p，对所有 $0\leqslant\theta\leqslant 2\pi$，求出使

$$S(T(Z,\ A,\ B,\ A^p,\ B^p))=e^{i\theta} \qquad (10-229)$$

成立的所有复数 Z，其中 $S(T)$ 表示矩阵 T 的特征值。满足式（10-229）的所有 z 在复平面上形成的曲线称作稳定区域的边界，由此曲线与负实轴的交点就能确定 H_a 的值。

当给定要求的精度阶后，系数矩阵 A，A，A^p，B^p 内往往含有许多待定的元素，我们可以选取这些元素使 H_a 尽可能大。在表10-14列出以式（10-217）为预估公式和式（10-219）为校正公式的 PECE 格式的并行多块方法的一些参数及相应的 H_a 值，在表10-15列出以式（10-221）作为预估公式和式（10-223）作为校正公式的 PECE 格式的并行多块方法的一些参数及相应

表 10-14

预估公式（10-217）				校正公式（10-219）			PECE
a_1	a_2	a_3	a_4	a_1	a_2	b_1	H_a
-1/3	-1/3	-1/3	-1/3	-1.0	-1.0	5/12	0.439
-1/3	-1/3	-1/3	-1/3	-1.0	0.0	5/12	0.848
2.0	-7/3	-1/3	-1/3	-1.0	-1/5	7/12	0.955
7/3	-8/3	-1/3	-1/3	-1.0	-3/5	11/12	1.130
7/3	-8/3	-1/3	-1/3	-1.0	-3/5	1.0	1.249

的 H_μ 值。

表 10-15

预估公式 (10-221)						校正公式(10-223)		PECE
a_1	a_2	a_3	a_4	a_5	a_6	a_1	a_2	H_μ
-1.000	-0.667	-0.333	-1.00	-0.667	-0.667	-0.500	-1.000	0.311
-1.000	-0.667	-0.333	-1.00	-0.667	-0.667	-1.000	-0.500	0.603
-1.233	-0.567	-0.333	-1.133	-0.567	-0.567	-0.999	-0.200	0.680
-1.333	-0.567	-0.133	-1.333	-0.467	-0.567	-1.000	-1.000	0.692
-1.233	-0.467	-0.033	-1.333	-0.767	-0.367	-1.000	-1.000	0.740
-1.230	-0.463	-0.0277	-1.330	-0.77212	-0.370	-1.010	-0.0452	0.810

作为比较，我们考虑四阶的亚当斯-巴什福兹-莫尔顿公式对(简称 ABM公式)，对 ABM 的 PECE 格式。它对应的 H_μ 值 $H_{ABM}=1.285$，将表10-15最后一行的方法与其相比，似乎由于稳定性限制，并行多块方法是不如四阶 ABM 方法的。但是由于 r 点的多块方法每执行一个循环向前推进的时间长度为 rh，考虑 $r=2$ 的情形，假定问题 (10-1) 要在区间 $[t_0, t_0+T]$ 上求解，并且所用的方法都采用允许的最大步长，于是采用 ABM 方法需调用 T/H_{ABM} 次，共计算 $2T/H_{ABM}$ 个右端函数值。采用表10-15最后一行所对应的并行多块方法需调用 $T/(2H_B)$ 次 ($H_B=0.810$)，共计算 $4T/(2H_B)$ 个右端函数值，虽然并行多块方法计算的右函数值的个数比 ABM 方法多，但由于其计算的并行性，应用两个处理机仅需 T/H_B 个右函数的计算时间。所以若右函数的计算量非常大，使得算法10-9中其它计算所占的比例极小，算法运行所用的时间可用右函数的计算时间估计。于是这个并行多块方法的运行时间与 ABM 方法的运行时间之比近似为 $H_{ABM}/(2H_B)=0.7932$，并行多块方法优于 ABM 方法。实际上对于非刚性问题，出于计算精度的考虑，不能采用稳定性允许的最大步长。而在稳定性允许下，相同阶的方法通常可采用相近的步长，这时并行多块方法的运行时间与 ABM 方法运行时间之比

可接近 0.5。

10.7 多重内插法

1964年，尼弗吉尔脱（Nievergelt）在文献[42]中提出多重内插方法，它的基本思想为：令问题（10-1）的解为 $y(t)$，用点 $t_0=0<t_1<t_2<\cdots<t_m=a$ 将求解区间[0，a]分成 m 个子区间，并构造初值问题

$$\frac{dy}{dt}=f(t,\ y),\ y(t_i)=y_i\ i=0,\ 1,\ \cdots,\ m-1 \tag{10-230}$$

令其解为 $y(t;\ t_i,\ y_i)$，则 $y(t_{i+1};\ t_i,\ y_i)$ 仅是 y_i 的函数，记此函数为 $\Psi_i(y)$，即 $\Psi_i(y_i)=y(t_{i+1};\ t_i,\ y_i)$。若已构造函数 $\Psi_i(y)$（$i=0,\ 1,\ \cdots,\ m-1$），则对任意的 y_0，由

$$y_1=\Psi_0(y_0),\ y_2=\Psi_1(y_1),\ \cdots,\ y_m=\Psi_{m-1}(y_{m-1}) \tag{10-231}$$

通过计算得到 $y(t)$ 在格点 t_1，t_2，$\cdots$，t_m 上的值，通常很难得到函数 $\Psi_i(y)$ 的表示式，可以采用插值的方法得到它的近似表示式。

这种方法的计算步骤是这样的：给定问题（10-230）的 y_i 的 K_i 个值 $y_i^{(k)}$，$k=1,\ 2,\ \cdots,\ K_i$。数值积分初值问题

$$\frac{dy}{dt}=f(t,\ y)\ y(t_i)=y_i^{(k)} \tag{10-232}$$

得到函数值 $\Psi_i(y_i^{(k)})$ 的近似值 $\tilde{y}_{i+1}^{(k)}$，通过 $(y_i^{(k)},\ y_{i+1}^{(k)})(k=1,\ 2,\ \cdots,\ K_i)$ 构造插值函数 $\tilde{\Psi}_i(y)$。

对于 $i=0,\ 1,\ \cdots,\ m-1$，$k=1,\ 2,\ \cdots,\ K_i$，初值问题（10-232）可在各个处理机上进行并行计算，因此这方法具有很高的并行性，但是由于插值误差的传播，使这方法很难实际应用。

多重内插算法的构造思想是许多并行算法构造的出发点。

第十一章　常微分方程边值问题的并行数值计算方法

数值求解常微分方程边值问题的方法很多。例如有里兹-伽略金（Ritz-Галеркин）方法、不变嵌入法、打靶法和有限差分法等。据凯勒（Keller）在文献〔4〕中进行的理论比较得出的结论，目前在这些方法中，有限差分法和多点打靶法最为有效，本章主要讨论这两类方法的并行计算问题。

11.1　一般打靶法的并行计算

考虑常微分方程边值问题

$$\frac{dy}{dt}=f(t,\ y)\quad a\leqslant t\leqslant b \tag{11-1}$$

$$B(a,\ y(a),\ b,\ y(b))=0 \tag{11-2}$$

其中$y\in R^n$，$f(t,\ y)\in R^n$，$B(a,\ y(a),\ b,\ y(b))\in R^n$，$t$是自变量，$a$和$b$是给定的常数，问题（11-1）、(11-2)是两点边值问题，它的定解条件给在点a和b处。假定函数f和B对其变量充分光滑，并且问题（11-1）、(11-2）具有唯一的孤立解$y^*(t)$，即沿$y^*(t)$的问题（11-1）、(11-2）的线性扰动问题

$$\frac{dz}{dt}=A(t,\ y^*)z$$

$$B_a(y^*)z(a)+B_b(y^*)z(b)=0$$

具有唯一解，即零解。$A(t,\ y)$、$B_a(y)$、$B_b(y)$都是$n\times n$矩阵

$$A(t,\ y)=\frac{\partial f(t,\ y)}{\partial y}\Big|_{y=y(t)}$$

$$B_a(y)=\frac{\partial B(a,\ y(a),\ b,y(b))}{\delta y(a)}$$

$$B_b(y)=\frac{\partial B(a,\ y(a),b,\ y(b))}{\partial y(b)}$$

在上述假定下，本节讨论的打靶法及其数值实现可用来计算问题（11-1)、(11-2）的解。特别它保证方程（11-16）将具有唯一根。打靶法的基本思想就是将问题（11-1)、(11-2）的求解归化成该方程的求根问题，目前非线性方程求根的方法已比较完善，而在求解过程中所需要的微分方程的数值解可进行并行计算，因此打靶法是在并行处理机系统上求解常微分方程边值问题的有效方法之一。

本节研究一般打靶法中的并行计算问题，下节考虑如何提高计算过程中的并行效率。

11.1.1 线性问题的一般打靶法

考虑一般的一阶线性两点边值问题

$$y'=A(t)y+f(t)\qquad a\leqslant t\leqslant b \tag{11-3}$$

$$B_a y(a)+B_b y(b)=\beta \tag{11-4}$$

其中$A(t)$，B_a，B_b都是$n\times n$矩阵，y、$f(t)$、β是n维列向量。$A(t)$是t的矩阵函数。$f(t)$是t的向量值函数。不作特殊说明时，假定$A(t)$、$f(t)$对其变元t具有所需要的光滑性。问题（11-3)、(11-4）本身是经常出现的，而且它常常作为求解一般边值问题（11-1)、(11-2）的子问题出现。

为了求解问题（11-3)、(11-4)，我们作下面的构造：任意选取y_0，求解初值问题

$$y'=A(t)y+f(t),\ \ y(a)=y_0 \tag{11-5}$$

记式（11-5）的解为$y_0(t)$，再对每一个单位向量$e_\nu=(0,0,\cdots,0,1,0,\cdots)^T$，其中1在第$\nu$个分量处，求解线性齐次常微分方程初值问题

$$y'=A(t)y\quad y(a)=e_\nu,\quad \nu=1,\ 2,\ \cdots,\ n \tag{11-6}$$

将其解记为 $y_v(t)$。作 $n \times n$ 矩阵

$$Y(t)=[y_1(t),\ y_2(t),\ \cdots,\ y_n(t)] \qquad (11-7)$$

于是问题 (9-3)、(9-4) 的解 $y(t)$ 可以表示成

$$y(t)=y_0(t)+\sum_{v=1}^{n}\xi_v y_v(t)$$

$$=y_0(t)+Y(t)\xi \qquad (11-8)$$

其中 ξ_v，$v=1,\ 2,\ \cdots,\ n$ 是待定的系数，$\xi=(\xi_1,\ \xi_2,\ \cdots,\ \xi_n)^T$。将式 (11-8) 代入 (11-4)，并稍加整理得

$$(B_a+B_bY(b))\xi=\beta-B_ay_0-B_by_0(b)$$

记

$$Q=B_a+B_bY(b) \qquad (11-9)$$

则 ξ 由线性方程组

$$Q\xi=\beta-B_ay_0-B_by_0(b) \qquad (11-10)$$

确定。

由此，立即可得下面的定理

定理11-1 如果 $A(t)$，$f(t)\in C^p[a,\ b]$，则问题 (11-3)、(11-4) 具有唯一解 $y(t)\in C^{p+1}[a,\ b]$ 的充分必要条件是矩阵 Q 是非奇的。

证明 由于式 (11-3) 的每个解 $y(t)\in C^{p+1}$ 可以表示成式 (11-8)，因而问题 (11-3)、(11-4) 的唯一可解性等价于问题 (11-10) 的唯一可解性。证完。

实际上，我们只要求 $A(t)$，$f(t)$ 是分段光滑的，定理这时候也成立，只需将其中光滑性的部分作适当的改变。下面我们假定 Q 是非奇的，因而问题 (11-3)、(11-4) 具有唯一解。

由上面的构造可以看出 $n+1$ 个初值问题 (11-5)(11-6) 是相似的，它们的计算可并行进行。

算法 11-1 线性二点边值打靶法

步 0 选取 y_0 和 e_v，$v=1,\ 2,\ \cdots,\ n$

步 1 并行求解 $n+1$ 个初值问题 (11-5)、(11-6)。在(11-6) 中取 $f(t)\equiv 0$，则这 $n+1$ 个初值问题形状一致。

步 2　构造矩阵Q，并求解线性方程组（11-10），得ξ。

步 3　由式（11-8）构造解$y(t)$。

实际求解时，步 1 中必然要采用数值求解常微分方程初值问题的算法。从而步 2 构造的矩阵Q是近似的，步 3 构造的解$y(t)$也是近似的。下面我们来说明只要数值积分的步长适当小，可以保证矩阵Q的近似仍是非奇异的，并给出$y(t)$的近似的误差估计。

设数值求解式（11-5）、(11-6）的离散格点$\{t_j\}$为$t_0=a$，$t_j=t_{j-1}+h_j$，$1\leqslant j\leqslant M$，$t_M=b$，$h=\max\limits_j h_j\leqslant\theta\min\limits_j h_j$其中$\theta$是固定的正数。设采用的数值积分方法是$m$阶的稳定方法（如四阶龙格-库塔方法或亚当斯方法）。若$A(t)$和$f(t)$是充分光滑的，将数值近似解表示成$\{u_{v,j}\}_{j=0}^{M}$，$v=0,1,\cdots,n$，则有估计

$$\|y_v(t_j)-u_{v,j}\|\leqslant Eh^m\quad 0\leqslant j\leqslant M \tag{11-11}$$

若$A(t)$、$f(t)$是分段光滑的，只要将间断点包含在格点集$\{t_j\}$中，这个结果也成立。记$U_j=(u_{1,j},\cdots,u_{n,j})$和$Q_h=B_a+B_bU_M$，则方程（11-10）由近似方程

$$Q_h\xi_h=\beta-B_au_{0,0}-B_bu_{0,M} \tag{11-12}$$

代替。由式（11-11）推得$\|Q-Q_h\|=O(h^m)$，再由Q的非奇异性和巴拿赫扰动定理推得当h充分小时，Q_h是非奇异的，并有$Q_h^{-1}=Q^{-1}+O(h^m)$。由式（11-12）

$$\xi_h=Q_h^{-1}(\beta-B_au_{0,0}-B_bu_{0,M})$$

问题（11-3）、(11-4）的数值解为

$$u_j=u_{0,j}+U_j\xi_h$$

由巴拿赫扰动定理，我们还可得到

$$\|Q^{-1}-Q_h^{-1}\|=O(h^m)$$

式（11-11）、(11-12）推得

$$\|y(t_j)-u_j\|=O(h^m)$$

这表明用数值积分求解代替算法 11-1 中的精确求解可得到精度为$O(h^m)$的近似解。

许多实际问题通过线性变换可以将$y(a)$或$y(b)$的一些分量从边界条件式(11-4)中分离出来，不损失一般性。我们考虑式(11-4)的下述形式

$$y_{(a)}^{i}=y_a^i \quad i=1,\ 2,\ \cdots,\ q \tag{11-13}$$

$$B_{aq}y(a)+B_{bq}y(b)=\beta_q \tag{11-14}$$

其中y^i表示向量y的第i个分量B_{aq}，B_{bq}是$(n-q)\times n$矩阵，β_q是$n-q$维列向量，$y_a^i(i=1,\ 2,\ \cdots,\ q)$给定。由这种条件看出$y(a)$的前$q$个分量已知，仅需确定后$n-q$个分量。为此将算法11-1中的$y_b$选成

$$y_c=(y_a^1,\ y_a^2,\ \cdots,\ y_a^q,\ y_0^{q+1},\ \cdots,\ y_0^n)^T$$

其中y_0^{q+1}，$\cdots$，y_0^n可任取，而e_v取$v=q+1$，$\cdots$，n，数值积分相应的$n-q+1$个方程，其解为$u_{v,j}(v=0,\ q+1,\cdots,n)$作矩阵$U_{jq}=(u_{q+1,j},\ \cdots,\ u_{n,j})$，于是要求的数值解可表成

$$u_j=u_{0,j}+U_{jq}\xi_q$$

其中ξ_q是$n-q$维列向量，ξ_q由方程

$$(B_{aq}E_q+B_{bq}U_{Mq})\xi_q=\beta_q-B_{aq}y_0-B_{bq}u_{0,M}$$

求出，其中$E_q=(e_{q+1},\ \cdots,\ e_n)$。这种处理表明在式(11-13)、(11-14)的情形求解的初值问题的个数可减少q个。

11.1.2 非线性问题的打靶法

利用打靶法求解非线性边值问题的思想是很容易想到的。考虑问题(11-1)、(11-2)，若已知初始值$y(a)=s$，积分初值问题

$$y'(t)=f(t,\ y(t)) \quad y(a)=s \tag{11-15}$$

得到解$y(t;\ s)$。其中s表示解对初值s的依赖关系。将$y(t;\ s)$代入式(11-2)，得到s应满足的方程

$$\phi(s)\equiv B(a,\ s,\ b,\ y(b;\ s))=0 \tag{11-16}$$

如果$s=s^*$是方程(11-16)的一个根，则$y(t)=y(t;\ s^*)$是问题(11-1)、(11-2)的一个解。反过来，对于问题(11-1)、(11-2)的任意一个解$y(t)$，则$s=y(a)$将是方程(11-16)

的一个根。这样将边值问题的求解归化成求根的问题。

为了采用牛顿法的思想解方程（11-16）的根，可以有许多方法。下面讨论其中的两种算法。一种称作拟线性化算法，另一种称作直接打靶法。

我们先讨论拟线性化算法。

给定初值 $s=s_0$，由问题（11-15）求得解 $y_0(t)=y(t;s_0)$。假定 $y_0(t)$ 不满足边界条件（11-2），而问题（11-1）、（11-2）的解为 $y^*(t)$。记 $\Delta y^*(t)=y^*(t)-y_0(t)$。由式（11-1）、（11-2）得

$$\Delta y^*(t)=f(t,\ y_0(t)+\Delta y^*(t))-f(t,\ y_0(t)) \tag{11-17}$$

$$B(a,\ s_0+\Delta y^*(a),\ b,\ y_0(b)+\Delta y^*(b))=0 \tag{11-18}$$

将式（11-17）、（11-18）按 $\Delta y^*(t)$ 展开成泰勒级数，并只保留一次项，得到式（11-17）、（11-18）的近似

$$\Delta y'(t)=\frac{\partial f(t,\ y_0(t))}{\partial y(t)}\Delta y(t) \tag{11-19}$$

$$\frac{\partial B}{\partial y(a)}\Delta y(a)+\frac{\partial B}{\partial y(b)}\Delta y(b)=-B(a,\ y_0(a),\ b,\ y_0(b)) \tag{11-20}$$

其中 $\frac{\partial B}{\partial y(a)}$、$\frac{\partial B}{\partial y(b)}$ 的变量为 $(a,\ y_0(a),\ b,\ y_0(b))$。记 $A(t)=\frac{\partial f(t,\ y_0(t))}{\partial y(t)}$，$B_a=\frac{\partial B}{\partial y(a)}$，$B_b=\frac{\partial B}{\partial y(b)}$，则式（11-19）、（11-20）为线性两点边值问题（11-3）、（11-4）的形式，可由算法11-1求解。设其解为 $\Delta y(t)$。令 $s_1=s_0+\Delta y(a)$，重复上述过程。

算法 11-2 拟线性化算法

步 0 选取初值 s_0，令 $i=0$

步 1 用常微分方程初值问题的并行算法计算初值问题（11-15）的数值解 $y_i(t)$，其中 $s=s_i$。

步 2　构造线性两点边值问题 (11-19)、(11-20)，其中 s_0 换成 s_i。

步 3　调用算法 11-1，求问题 (11-19)、(11-20) 的解 $\Delta y(t)$。

步 4　判别 $\Delta y(a)$ 是否满足精度。若满足，停止计算，令 $y_i(t)$ 为问题 (11-1)、(11-2) 的近似解；否则令 $s_{i+1}=s_i+\Delta y(a)$，$i=i+1$，转步 1。

现在讨论直接打靶法。

类似于上面的推导，给定 s_0，计算 $y_0(t)=y(t;s_0)$，问题 (11-1)、(11-2) 的解 $y^*(t)$ 对应的初值为 $s^*=y^*(a)$，$y^*(t)=y(t;s^*)$。由方程 (11-16) 得，

$$\phi(s^*)=\phi(s_0+(s^*-s_0))=B(a,\ s_0+(s^*-s_0),\ b,\ y\cdot(b;\ s_0+(s^*-s_0)))=0$$

对 $\Delta s=s^*-s_0$ 展开成泰勒级数，并保留一次项，得方程

$$\left(B_a+B_b\frac{\partial y}{\partial s}\right)\Delta s=-B(a,\ y_0(a),\ b,\ y_0(b)) \tag{11-21}$$

其中

$$B_a=\frac{\partial B(a,\ y_0(a),\ b,\ y_0(b))}{\partial y_0(a)}$$

$$B_b=\frac{\partial B(a,\ y_0(a),\ b,\ y_0(b))}{\partial y_0(b)} \tag{11-22}$$

$$\frac{\partial y}{\partial s}=\frac{\partial y(b,\ s_0)}{\partial s_0}$$

由式 (11-21) 解出 Δs，于是令 $s_1=s_0+\Delta s$，可重复进行迭代。

式 (11-22) 中的偏导数的计算可采用两种方法，一种方法是解析法，即推导 $\frac{\partial y}{\partial s}$ 的计算公式，进行计算。可以证明 $\frac{\partial y}{\partial s}$ 恰好是用算法 11-1 求解边值问题 (11-19)、(11-20) 时得到的矩阵 $Y(b)$。上面得到的迭代过程就是算法 11-2。另一种算法采用差商代替式 (11-22) 中的导数。更直接地，可以采用差商直接代替 $\phi(s)$ 对其变元的偏导数，于是我们得到下面的差商打靶法。

算法 11-3 差商打靶法

步 0 选取初值 s_0，令 $i=0$。

步 1 确定差商步长 λ_i，并构造计算差商的变量

$$s_i^v = s_i + \lambda_i e_v,\quad v = 1,\ 2,\ \cdots,\ n$$

其中 e_v 为单位向量 $(0,\ \cdots,\ 0,\ 1,\ 0,\ \cdots,\ 0)^T$。

步 2 对于 $s=s_i$，$s=s_i^v$，并行求解初值问题 (11-15)，得到解 $y(t,\ s_i)$ 和 $y(t,\ s_i^v)$，$v=1,\ 2,\ \cdots,\ n$。

步 3 并行计算

$$\Phi(s_i) \text{ 和 } \Phi(s_i^v)\ v = 1,\ 2,\ \cdots,\ n$$

步 4 计算矩阵 $\tilde{\Phi}'(s_i)$，其第 v 个列向量为

$$(\Phi(s_i^v) - \Phi(s_i))/\lambda_i$$

步 5 求解线性方程组

$$\tilde{\Phi}'(s_i)(s_{i+1} - s_i) = -\Phi(s_i)$$

得 s_{i+1}。

步 6 判别 $\|s_{i+1} - s_i\|$ 是否满足要求的精度。若满足，停止计算，$y_i(t) = y(t;\ s_i)$ 即为所求的解；否则令 $i = i+1$，转步 1。

算法 11-2 和算法 11-3 均是牛顿法线性展开思想的应用。当 s_0 适当选取时，这两个算法得到的序列 $\{s_i\}$ 将很快达到要求的精度。

11.2 并行打靶法

上一节我们讨论了一般打靶法的并行计算问题。打靶法的基本思想是将边值问题化成初值问题，然后对初值进行迭代，使边界条件得到满足。这种一般打靶法的并行计算主要是让同种类型的若干个初值问题进行并行计算。因此，若不考虑求解初值问题本身的并行效率，则上一节中讨论的算法的并行效率主要依赖于同种类型的初值问题的个数。为提高并行的效率，设法将初值问题的个数增加。并行打靶法可以达到这个目的。

最早构造并行打靶法的目的主要是为了克服初值问题数值求

解过程中的不稳定现象。对于有些初值问题，它的解对于初值的选取是很敏感的，在计算中引进的误差将随着求解的过程逐渐扩大，最终将精确解掩盖掉，甚至产生上溢的现象。为了克服这种现象引起的困难，可以缩短数值求解初值问题的区间，使当误差随着求解还没有扩大到影响需要的精度时，计算已达到区间的终点。最早构造的这种类型的方法称作多重打靶法，现在称它为并行打靶法。在并行处理计算机问世后，并行打靶法更显出了它的生命力。它不仅能在计算中很好的控制不稳定现象，还可以大大提高计算速度。

并行打靶法的基本思想是把原来的求解区间分割成若干个子区间，在这些子区间上同时数值积分，选取每个子区间起点处解的值，以满足边界条件和分点处的连续性条件。

11.2.1 线性问题的并行打靶法

考虑线性问题 (11-3)、(11-4)。选择一些结点把区间〔a，b〕分成许多小区间。设结点集为$\langle\tau_j\rangle$，

$$\tau_0 = a < \tau_1 < \cdots < \tau_j = b$$

由 11.1 节中的讨论，在第 j 个区间〔τ_{j-1}，τ_j〕上，问题 (11-3)、(11-4) 的解 $y(t)$ 可以表示成

$$y(t) = y_j(t) = v_j(t) + V_j(t)\xi_j \quad \tau_{j-1} \leqslant t \leqslant \tau_j \tag{11-23}$$

其中 $v_j(t)$ 和 $V_j(t)$ 分别是初值问题

$$v_j'(t) = A(t)v_j(t) + f(t) \quad v_j(\tau_{j-1}) = v_j^0 \quad \tau_{j-1} \leqslant t \leqslant \tau_j \tag{11-24}$$

和

$$V_j'(t) = A(t)V_j(t) \quad V_j(\tau_{j-1}) = I \quad \tau_{j-1} \leqslant t \leqslant \tau_j \tag{11-25}$$

的解。其中 $V_j(t)$ 是 $n \times n$ 矩阵，I 是 $n \times n$ 单位矩阵，$v_j(t)$ 和 v_j^0 均是 n 维列向量。实际计算时 v_j^0 可适当指定。由解 $y(t)$ 在内部结点上的连续性要求有

$$y_j(\tau_{j-1}) = y_{j-1}(\tau_{j-1})$$

于是由式 (11-23)

$$v_j^0+\xi_j=v_{j-1}^0+V_{j-1}(\tau_{j-1})\xi_{j-1}\quad j=2,\ \cdots,\ J \tag{11-26}$$

边界条件 (11-4) 可写成

$$B_a[v_1^0+\xi_1]+B_b[v_J(b)+V_J(b)\xi_J]=\beta \tag{11-27}$$

将式 (11-26)、(11-27) 合并成矩阵形式

$$\begin{bmatrix} B_a & \cdots & B_bV_J(b) \\ -V_1(\tau_1) & I & 0 \\ & -V_2(\tau_2) & I \\ 0 & \ddots & \ddots \\ & -V_{J-1}(\tau_{J-1}) & I \end{bmatrix} \begin{bmatrix} \xi_1 \\ \xi_2 \\ \xi_3 \\ \vdots \\ \xi_J \end{bmatrix}$$

$$= \begin{bmatrix} \beta-[B_av_1^0+B_bv_J(b)] \\ v_1(\tau_1)-v_2^0 \\ v_2(\tau_2)-v_3^0 \\ \vdots \\ v_{J-1}(\tau_{J-1})-v_J^0 \end{bmatrix} \tag{11-28}$$

求解方程组 (11-28)，得解 $\{\xi_j\}_1^J$。把它代入式 (11-23) 即得问题 (11-3)、(11-4) 的解。

很明显 11.1 节中讨论的是 $J=1$ 的情形。

将上述推导总结成下面的算法

算法 11-4 线性问题的并行打靶法

步 0 选取结点 τ_0，τ_1，$\cdots$，τ_J 和向量 v_1^0，v_2^0，$\cdots$，v_J^0

步 1 对 $j=1$，2，$\cdots$，J 并行求解初值问题 (11-24)(11-25)。

步 2 构造并求解线性方程组 (11-28)，得 ξ_1，ξ_2，$\cdots$，ξ_J

步 3 由式 (11-23) 构造解 $y(t)$。

由构造容易看到，并行打靶法将可并行求解的初值问题的个数变成原来的 J 倍，所以并行效率提高了，缺点是增加了求解的线性方程组的维数。将上述两点进行折衷，并加上求解的稳定性要求，来确定较好的结点数 J 的值。

11.2.2 非线性问题的并行打靶法

考虑问题 (11-1)、(11-2)。与线性问题类似，选取结点集 $\{\tau_j\}_{j=0}^{J}$，在第 j 个子区间 $[\tau_{j-1}, \tau_j]$ 上，问题 (11-1)、(11-2) 的解 $y(t)=y_j(t)$ 是初值问题

$$y_j'(t)=f(t, y_j(t)) \quad y_j(\tau_{j-1})=s_j \quad \tau_{j-1}\leqslant t\leqslant\tau_j \tag{11-29}$$

的解，s_j 应取成 $y(\tau_{j-1})$。实际求解时 s_j 是未知的。将式(11-29)的解仍记成 $y_j(t)=y_j(t; s_j)$。由解在内部结点上的连续性要求，有

$$y_j(\tau_{j-1})=y_{j-1}(\tau_{j-1}; s_{j-1}) \quad j=2, \cdots, J \tag{11-30}$$

再由边界条件 (11-2)，得到条件

$$B(a, s_1, b, y_J(b, s_J))=0 \tag{11-31}$$

把上述边界条件和连续性条件写成下面的向量形式

$$\Phi(\hat{s})=\begin{bmatrix} B(a, s_1, b, y_J(b, s_J)) \\ s_2-y_1(\tau_1, s_1) \\ \vdots \\ s_J-y_{J-1}(\tau_{J-1}, s_{J-1}) \end{bmatrix}=0 \qquad \hat{s}=\begin{bmatrix} s_1 \\ s_2 \\ \vdots \\ s_J \end{bmatrix} \tag{11-32}$$

上面的处理可叙述成求初值向量 $\hat{s}$ 使 $\Phi(\hat{s})=0$。这种并行打靶过程也可以看作人工地扩大了的两点边值问题的一般打靶法。为此，我们作下面的构造。在每个子区间 $[\tau_{j-1}, \tau_j]$ 上引进新的独立变量 $\hat{t}$

$$\hat{t}\equiv(t-\tau_{j-1})/\Delta_j \quad \Delta_j\equiv\tau_j-\tau_{j-1} \quad 1\leqslant j\leqslant J \tag{11-33}$$

记 $z_j(\hat{t})\equiv z(\tau_{j-1}+\hat{t}\Delta_j)$ 和

$$f_j(\hat{t}, z_j(\hat{t}))\equiv\Delta_j f(\tau_{j-1}+\hat{t}\Delta_j, z(\tau_{j-1}+\hat{t}\Delta_j))$$

于是式 (11-29)、(11-32) 可以写成

$$\hat{y}'(\hat{t})=\hat{f}(\hat{t}, \hat{y}(\hat{t})) \quad 0\leqslant\hat{t}\leqslant 1 \quad \hat{y}\equiv\begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_J \end{pmatrix} \quad \hat{f}\equiv\begin{pmatrix} f_1 \\ f_2 \\ \vdots \\ f_J \end{pmatrix} \tag{11-34}$$

$$\hat{g}(\hat{y}(0),\hat{y}(1))\equiv\begin{pmatrix} B(a,\ y_1(0),\ b,\ y_J(1)) \\ y_2(0)-y_1(1) \\ \vdots \\ y_J(0)-y_{J-1}(1) \end{pmatrix}=0 \tag{11-35}$$

问题（11-34）、（11-35）等价于问题（11-1）、(11-2)。它是人工地扩大了的边值问题，对于问题（11-34）、(11-35）的一般打靶法就是对问题（11-1）、(11-2）的并行打靶法。用算法 11-2 或算法 11-3 求解问题（11-34）、(11-35）时，可并行求解的初值问题的个数为式（11-1）、(11-2）的 J 倍。并行效率提高了，积分区间的长度可简单地看成缩短原来的 J 分之一。直观地可看成将整个区间的计算逐段分配给各个处理机独立地进行。但要求解的非线性方程组（11-35）的维数增加了。求解（11-35）可将其化成优化问题，用第七章中的并行优化算法来求解。下一节我们将给出这样处理的例子，它将非线性动力系统的最优控制问题化成两点边值问题，然后再用并行打靶法进行求解。

11.3 最优控制问题的并行计算

高阶的复杂的非线性系统的最优化计算，往往需要大量的计算机运算时间。利用并行处理机将可以使所需要的计算时间大大的缩短，使复杂的最优化计算可以应用到高阶非线性系统的控制器的设计中去。

11.3.1 最优控制问题

考虑非线性控制过程

$$\frac{dy(t)}{dt}=f(t,\ y(t),\ u(t))\ t\in[t_0,\ t_f] \tag{11-36}$$

其中 $y\in R^n$ 是系统的状态变量，$y(t)$ 的初始条件

$$y(t_0)=y_0$$

已知。$u\in R^m$ 是系统的控制。最优控制问题是要确定控制 u

(t)，$t\in[t_0, t_f]$和相应的轨道$y(t)$，使指标

$$J=\Phi(y(t_f), t_f)+\int_{t_0}^{t_f} L(y(t), u(t), t)dt \tag{11-37}$$

达极小值。构造哈密顿（Hamilton）函数

$$H(y(t), u(t), \lambda(t), t)=L(y(t), u(t), t)+\lambda^T(t)f(t, y(t), u(t)) \tag{11-38}$$

其中$\lambda\in R^n$，称作共轭变量。于是最优性的必要条件为

$$\partial H/\partial u(t)=0 \tag{11-39}$$

$$\lambda'(t)=-\partial H/\partial y(t)=g(y(t), \lambda(t), u(t)) \tag{11-40}$$

$$y'(t)=\partial H/\partial\lambda(t)=f(t, y(t), u(t)) \tag{11-41}$$

$$y(t_0)=y_0 \tag{11-42}$$

$$\lambda(t_f)=\partial\Phi(y(t_f), t_f)/\partial y(t_f)=\lambda_f(y(t_f), t_f) \tag{11-43}$$

式中t_f是控制过程结束的时刻，我们考虑t_0，t_f给定的情形。由式（11-39）解出$u(t)$，将它记为

$$u(t)=h(y(t), \lambda(t), t)$$

如果共轭变量的初始状态值$\lambda(t_0)$已知，则求解方程组(11-39)～(11-42)将能得到所要的解。一般来说$\lambda(t_0)$是未知的。于是组（11-39)～(11-43）组成常微分方程两点边值问题。应用打靶法的思想，求$\lambda(t_0)$，使由组（11-39)～(11-42）得到的$\lambda(t_f)$满足方程（11-43)。这是一个求非线性方程组根的问题。将其化成优化问题，求$\lambda(t_0)$使误差函数

$$E=\left\| \lambda(t_f)-\frac{\partial}{\partial y(t_f)}\Phi(y(t_f), t_f)\right\|^2 \tag{11-44}$$

达极小值。这个优化问题与上面的最优控制问题是不同的。这个问题优化的参数是$\lambda(t_0)$，它是常参数，不随时间变化。而前面的最优控制问题优化的是随时间变化的控制变量$u(t)$。显然，若找到的$\lambda(t_0)$，使由式（11-44）定义的$E=0$，则相应的$\lambda(t)$、$u(t)$、$y(t)$是问题（11-39)～(11-43）的解。

当最优控制问题具有约束时，可将式（11-39）换成其它的公式：例如要求控制满足约束条件

$$u(t)\in\Omega(y(t),\ t)$$

其中 $\Omega(y(t),\ t)$ 是依赖于时间 t 和状态 $y(t)$ 的一个集合。应用邦特列雅金（Понтрягин）极大值原理，将满足式（11-39）的 $u(t)$ 换成 $u(t)=h(y(t),\ \lambda(t),\ t)$，它使 u 的函数（其中 $y(t)$，$\lambda(t)$，t 看成固定值）

$$H(u)=H(y(t),\ u,\ \lambda(t),\ t)$$

在集合 $\Omega(y(t),\ t)$ 上达极小值。经过这样处理后，有约束的最优控制问题的计算格式与无约束问题的格式就一致了。

11.3.2 最优控制问题的并行打靶法

一般说来，如果 $\lambda(t_0)$ 的初始值选得比较好，使式（11-44）定义的误差函数比较小，则用第七章的并行优化算法（如并行变尺度法等），将能很快收敛到 E 的极小值点 0，从而得到最优控制问题的解。但实际问题往往很难选取 $\lambda(t_0)$ 的值，因而需要通过并行打靶来缩小计算时间。另外，若 $\lambda(t_0)$ 较好，但如果所考虑的系统对初始共轭变量的扰动非常敏感，即使收敛，速度仍很慢。这时也可以采用并行打靶法来节省计算时间，同时减小这些敏感性的影响。

下面我们描述并行打靶法与第七章的并行优化算法的组合，来求解最优控制问题。

记

$$z=\begin{pmatrix} y \\ \lambda \end{pmatrix}$$

$$R(z(t),\ u(t),\ t)=\begin{bmatrix} f(t,\ y(t),\ u(t)) \\ g(y(t),\lambda(t),\ u(t)) \end{bmatrix}$$

于是由式（11-40）、(11-41)

$$z'(t)=R(z(t),\ u(t),\ t) \qquad (11\text{-}45a)$$

选取 $2n\times 2n$ 矩阵 A 和 B，使有

$$A\begin{bmatrix} y(t_0) \\ \cdots \\ \lambda(t_0) \end{bmatrix}=\begin{bmatrix} y(t_0) \\ \cdots \\ 0 \end{bmatrix} \qquad B\begin{bmatrix} y(t_f) \\ \cdots \\ \lambda(t_f) \end{bmatrix}=\begin{bmatrix} 0 \\ \cdots \\ \lambda(t_f) \end{bmatrix}$$

于是问题 (11-39)～(11-43) 的边界条件可写成

$$Az(t_0)+Bz(t_f)=\alpha \qquad (11\text{-}45\,b)$$

其中 α 是由边界值组成的 $2n$ 维向量，即有

$$\alpha=\begin{bmatrix} y_0 \\ \cdots \\ \lambda_f(y(t_f),\ t_f) \end{bmatrix}$$

将区间 $[t_0,\ t_f]$ 分成 N 个子区间 $t_0<t_1<\cdots<t_N=t_f$，于是问题 (11-39)～(11-43) 可以写成

$$z_j'=R_j(z_j,\ t) \quad t\in[t_j,\ t_{j+1}] \qquad (11\text{-}46\,a)$$

$$Az_0(t_0)+Bz_{N-1}(t_f)=\alpha \quad j=0,\ 1,\ \cdots,\ N-1 \qquad (11\text{-}46\,b)$$

其中

$$z_j(t)=\begin{cases} z(t) & t\in[t_j,\ t_{j+1}] \\ 0 & \text{其它} \end{cases}$$

$$R_j(z_j,\ t)=R(z_j(t),\ u(t),\ t)$$

由于 $z(t)$ 在分点上的连续性，必须加上条件

$$z_{j-1}(t_j)=z_j(t_j) \quad j=1,\ 2,\ \cdots,\ N-1 \qquad (11\text{-}47)$$

将式 (11-46)、(11-47) 合并，得到非线性边值问题

$$Z'(t)=F(Z(t),\ t) \qquad (11\text{-}48\,a)$$

$$PZ_a+QZ_r=\gamma \qquad (11\text{-}48\,b)$$

其中 $Z(t)$，$F(Z(t),\ t)$，Z_a，Z_r，γ 均是 $2nN$ 维列向量，定义如下：

$$Z(t)=\begin{bmatrix} z_0(t) \\ \hdashline z_1(t) \\ \hdashline \vdots \\ \hdashline z_{N-1}(t) \end{bmatrix}$$

$$F(Z(t),\ t)=\begin{bmatrix} R_0(z_0(t),\ t) \\ \hline R_1(z_1(t),\ t) \\ \hline \vdots \\ \hline R_{N-1}(z_{N-1}(t),\ t) \end{bmatrix}$$

$$Z_s=\begin{bmatrix} z_0(t_0) \\ \hline z_1(t_1) \\ \hline \vdots \\ \hline z_{N-1}(t_{N-1}) \end{bmatrix} \qquad Z_r=\begin{bmatrix} z_0(t_1) \\ \hline z_1(t_2) \\ \hline \vdots \\ \hline z_{N-1}(t_f) \end{bmatrix} \qquad \gamma=\begin{bmatrix} y_0 \\ \lambda_f \\ \cdots \\ 0 \end{bmatrix}$$

P和Q均是 $2nN\times 2nN$ 矩阵，有形式

$$P=\begin{bmatrix} A & & & 0 \\ & I & & \\ & 0 & \ddots & \\ & & & I \end{bmatrix} \qquad Q=\begin{bmatrix} 0 & 0\cdots B \\ -I & 0\cdots 0 \\ 0 & -I\cdots 0 \\ 0\cdots 0 & -I \quad 0 \end{bmatrix}$$

按这种构造，误差函数为

$$E=\|PZ_s+QZ_r-\gamma\|^2 \tag{11-49}$$

现在的问题是在约束条件（9-48 a）下求 Z_s 使 E 达极小值。

将并行打靶法和并行优化算法组合，得到下面的算法步：

步 0　选取初始的 Z_s。

步 1　以 Z_s 为初始值，应用常微分方程初值问题的并行算法在 N 个区间上同时积分，得 Z_r 的值。

步 2　计算误差函数 $E=\|PZ_s+QZ_r-\gamma\|^2$

步 3　如果 E 充分小，则过程停止。否则，用并行优化算法更新 Z_s 的值，转步 1 。

11.3.3　算　　例

考虑由非线性微分方程

$$\dot{y}_1(t)=y_2(t)$$

$$\dot{y}_2(t)=a(t)[1-y_1^2(t)]y_2(t)-y_1(t)+u(t)$$

$$t\in[t_0,\ t_f] \tag{11-50}$$

描述的范德波尔（Vanderpol）系统，其中 $y(t_0)=y_0$ 是已知的，

$a(t)$ 是参数，它反映特殊系统的动力学。我们取 $a(t)\equiv 1$，$t\in[t_0, t_f]$。希望寻找最优控制 $u(t)$，使指标

$$J=10(y_1^2(t_f)+y_2^2(t_f))+\int_{t_0}^{t_f}[y_1^2(t)+y_2^2(t)+u^2(t)]dt \tag{11-51}$$

达极小值。相应的非线性两点边值问题由状态方程 (11-50)，共轭变量方程

$$\begin{aligned}\dot{\lambda}_1(t)&=\lambda_2(t)[2a(t)y_1(t)y_2(t)+1]-2y_1(t)\\ \dot{\lambda}_2(t)&=-\lambda_1(t)-\lambda_2(t)a(t)[1-y_1^2(t)]-2y_2(t)\end{aligned} \tag{11-52}$$

和边界条件

$$y(t_0)=y_0=[y_{10},\ y_{20}]^T \tag{11-53}$$

$$\lambda(t_f)=\lambda_f=[20y_1(t_f),\ 20y_2(t_f)]^T \tag{11-54}$$

组成。最优控制为

$$u(t)=-\frac{1}{2}\lambda_2(t)\quad t\in[t_0, t_f] \tag{11-55}$$

取 $t_0=0$，$t_f=5$，$y(0)=[0,\ 1]^T$。初始共轭变量 $\lambda(0)$ 的选取影响方程 (11-50)、(11-52) 的求解的稳定性。例如，若 $\lambda(0)$ 取在 $[0.5,\ 0.5]^T$ 的附近，则数值求解不会出现问题。但当 $\lambda(0)$ 取 $[0.5,\ 1.0]^T$ 时，向前积分 (11-50)、(11-52) 时会出现发散的现象。采用并行打靶的方法将会克服这种现象引起的数值求解的困难。

将积分区间 $[0,\ 5]$ 分成两个子区间。取 $t_1=1.6$，$t_2=t_f=5$。误差函数 (11-49) 中的 P 和 Q，Z_a，Z_r，γ 分别为

$$P=\left(\begin{array}{cc:cccc}0&0&0&0&0&0\\0&0&0&0&0&0\\\hdashline 0&0&1&0&0&0\\0&0&0&1&0&0\\0&0&0&0&1&0\\0&0&0&0&0&1\end{array}\right)$$

$$Q=\left(\begin{array}{cccc:cc} 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ \hdashline -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 \end{array}\right)$$

$Z_*=[\lambda_1(0),\ \lambda_2(0),\ y_1(1.6),\ y_2(1.6),\ \lambda_1(1.6),\ \lambda_2(1.6)]^T$

$Z_r=[y_1(1.6),\ y_2(1.6),\ \lambda_1(1.6),\ \lambda_2(1.6),\ \lambda_1(5.0),\lambda_2(5.0)]^T$

$\gamma=[\lambda_1(5.0),\ \lambda_2(5.0),\ 0,\ 0,\ 0,\ 0]^T$

由于$y(t_0)=y_0$是已知的，所以在Z_*中不需要放进去。这样，Z_*中未知量的个数可减少2个。

取Z_*的初始值为

$Z_*=[0.43019,\ 5.1156,\ 0.3189,\ -0.2012,\ 1.871,-0.7814]^T$

应用并行打靶法和并行优化算法进行迭代使误差函数E达极小值。

Z_*的这种取值接近于最优值。初始共轭变量收敛到最优值的结果在表11-1中给出。作为比较，表中还列出应用串行的DFP变尺度法得到的结果。

当Z_*的初始值离最优值远时，应用并行打靶法不出现

表11-1　$u(t)$无约束，$y_0=(0.0,\ 1.0)^T$，
$Z_{*0}=(0.43019,\ 5.1156,\ 0.3189,\ -0.2012,$
$1.871,\ -0.7814)^T$并行打靶法结果

极小化算法	DFP算法（串行）	并行变尺度算法	查赞-麦连科算法
E的求值次数	37	19	32
E的梯度求值次数	5	3	—
迭代次数	5	3	4
E的最后值	0.945×10^{-5}	0.243×10^{-5}	0.129×10^{-4}
最优共轭向量$\lambda(0)$值	0.43247 5.1186	0.43902 5.1236	0.43019 5.1167

$y(t)$、$\lambda(t)$ 发散的现象，在区间〔0，5〕上这些解保持有界。

11.4 自动变阶变步长的有限差分方法

从这节开始，我们讨论有限差分方法。有限差分法是求解常微分方程边值问题的一种常用的方法。从并行的效率看，它能达到相当高的并行程度。有限差分法求解常微分方程边值问题的过程是这样的：先确定求解区间中的格点，用有限个格点代替连续区间。接着用差分近似来代替微分方程中出现的导数，将微分方程离散成差分方程。这些方程与定解条件一起通常得到确定解在这些格点上的近似值的非线性方程组，最后求解这个方程组，得到所需要的解的近似值。

有限差分法与打靶法相比，要求更多的存储量。但当微分方程化成初值问题求解不稳定时，有限差分方法会特别有用的。事实上，当并行打靶法中 $\Delta_j=\tau_j-\tau_{j-1}$ 很小时，它归化成特殊的有限差分方法。

有限差分法求得的解通过延期校正，可以在精度上得到很大的提高。这一节叙述一个自动变阶变步长的差分方法。它以低阶的梯形公式为基础，通过延期校正估计近似解中的误差，然后将近似解逐次更新。由于计算过程中得到误差的一个估计，从而可自动变阶变步长。

考虑非线性一阶方程组

$$y'(t)=f(t,\ y(t))\quad t\in(a,\ b) \tag{11-56}$$

要求它的解 $y(t)$ 满足多点边界条件

$$g(y(\tau_1),\ y(\tau_2),\ \cdots,\ y(\tau_N))=0 \tag{11-57}$$

$$a\leqslant\tau_1<\tau_2<\cdots<\tau_N\leqslant b$$

向量 $y(t)$、$f(t,\ y)$ 和 g 均在 R^n 中取值。在区间〔a，b〕上构造格点集 $\pi=\{t_j\}$：

$$t_0=a,\ t_j=t_{j-1}+h_j\quad j=1,\ 2,\ \cdots,\ J\quad t_J=b$$

$$h=\max h_j \tag{11-58}$$

要求格点集 π 包含点集 $\{\tau_1, \tau_2, \cdots, \tau_N\}$。记 $\tau_i = t_{j_i}$。

对式（11-56）进行离散化的基本差分格式采用梯形公式

$$\Phi_\pi(u)_j \equiv h_j^{-1}(u_j - u_{j-1}) - \frac{1}{2}[f(t_{j-1}, u_{j-1}) + f(t_j, u_j)] = 0 \tag{11-59}$$

再要求 u_{j_i} 满足方程

$$g(u_{j_1}, u_{j_2}, \cdots, u_{j_N}) = 0 \tag{11-60}$$

将满足方程（11-59）、(11-60) 的序列 $\{u_0, u_1, \cdots, u_J\}$ 称为多点边值问题（11-56）、(11-57) 的解 $y^*(t)$ 的近似，即将 u_j 作为 $y^*(t_j)$ 的近似值。

记

$$U = (u_0, u_1, \cdots, u_J)^T$$

于是方程组（11-59）、(11-60) 可写成向量的形式

$$\Phi_\pi(U) = 0 \tag{11-61}$$

其中

$$\Phi_\pi(U) = \begin{bmatrix} \Phi_\pi(U)_0 \\ \Phi_\pi(U)_1 \\ \Phi_\pi(U)_2 \\ \vdots \\ \Phi_\pi(U)_J \end{bmatrix}$$

$$\Phi_\pi(U)_0 = g(u_{j_1}, u_{j_2}, \cdots, u_{j_N})$$

为了下面叙述的完整性，我们列举一些必要的理论结果。

我们称问题（11-56）、(11-57) 的解 $y^*(t)$ 为孤立解，如果沿 $y^*(t)$ 的问题（11-56）、(11-57)，线性扰动问题具有唯一解，即零解。我们假定问题（11-56）、(11-57) 的解 $y^*(t)$ 是孤立的。于是对于充分小的 h_0 和所有 $h \leqslant h_0$，有

（1）差分方程（11-59）、(11-60) 在 $\{y^*(t_i), i = 0, 1, \cdots, J\}$ 的邻域中具有唯一解。这个解可用牛顿法求解。对适当选取的初始值，用牛顿法求解将具有平方的收敛速度。

（2）$\max_j \|u_j - y^*(t_j)\| = O(h^2)$

（3）误差$u_j-y^*(t_j)$具有渐近展开式

$$u_j-y^*(t_j)=\sum_{\nu=1}^{m}h^{2\nu}e_\nu(t_j)+O(h^{2m+2})$$

$$j=0,\ 1,\ \cdots,\ J$$

（4）具有稳定性条件：存在与h无关的常数c和r，使当$\|U-V\|\leqslant r$时，有

$$\|U-V\|\leqslant c\,\|\Phi_s(U)-\Phi_s(V)\| \tag{11-62}$$

$\Phi_s(U)$的弗里希（Eréchet）导数（雅可比矩阵）$\Phi_s'(U)$具有形式

$$\Phi_s'(U)=\left(\begin{array}{c|ccccc} G_0 & G_1, & \cdots\cdots & G_J \\ \hline S_1 & R_1 & 0\cdots & 0 \\ 0 & S_2 & R_2\cdots & 0 \\ \vdots & & \ddots & \ddots \\ 0 & 0 & S_J & R_J \end{array}\right) \tag{11-63}$$

它的每一个子矩阵都是$n\times n$方阵，并有

$$G_j=\frac{\partial g}{\partial u_j}\qquad j=0,\ 1,\ \cdots,\ J$$

$$S_j=-\left[\frac{1}{h_j}I+\frac{1}{2}f'_{j-1}\right]$$

$$R_j=\frac{1}{h_j}I-\frac{1}{2}f'_j\qquad j=0,\ 2,\ \cdots,\ J$$

$$[f'_j]_{st}=\frac{\partial f_s}{\partial u^t}(t_j,\ u_j)$$

其中f_s是向量函数$f(\cdot,\ \cdot)$的第s个分量，而u^t是向量u的第t个分量。

现在将求解方程（11-61）的并行计算过程叙述如下。为了与后面的叙述一致，我们将方程（11-61）推广成

$$\Phi_s(U)=\eta \tag{11-64}$$

其中η是给定的$n(J+1)$维列向量。方程（11-61）对应$\eta=0$的方程（11-64）。

算法 9-5

步 0　选取初值$U^{(0)}=(u_0^{(0)}, u_1^{(0)}, \cdots, u_J^{(0)})^T$和迭代精度$TOL_u$。令$l=0$。

步 1　并行计算右函数值

$$f(t_j, u_j^{(e)}) \qquad j=0, 1, \cdots, J$$

步 2　并行计算

$$\Phi_n(U^{(e)})_j=h_j^{-1}(u_j^{(e)}-u_{j-1}^{(e)})-\frac{1}{2}[f(t_{j-1}, u_{j-1}^{(e)})+f(t_j, u_j^{(e)})]$$
$$j=1, 2, \cdots, J$$

步 3　计算

$$\Phi_n(U^{(e)})_0=g(u_{j1}^{(e)}, u_{j2}^{(e)}, \cdots, u_{jN}^{(e)})$$

步 4　计算矩阵$\Phi_n'(U^{(e)})$

步 5　求解线性方程组

$$\Phi_n'(U^{(e)})\Delta=\eta-\Phi_n(U^{(e)})$$

步 6　计算

$$U^{(e+1)}=U^{(e)}+\Delta$$

步 7　判别不等式是否成立

$$\|\Delta\|\leqslant TOL_u$$

若此不等式成立，停止计算，并令$U^{(e+1)}$为方程（11-64）的近似解；若不等式不成立，则令$e=e+1$，转步 1。

上面已指出，对适当选取的初值$U^{(0)}$，算法11-5将收敛，并且具有平方的收敛速度。算法11-5的步 1 和步 2 在并行处理机上能高效的实现。步 4 中矩阵$\Phi_n'(U^{(e)})$的计算主要集中在计算函数$f(t, u)$对u的雅可比矩阵的计算上，对$j=0, 1, \cdots, J$也是可以并行计算的。在实际实现这个算法时，为了节省计算量，不必对每个e均计算$\Phi_n'(U^{(e)})$，而可以固定采用某一个矩阵$\tilde{\Phi}_n'$。它可以是对$U^{(0)}$算得的矩阵$\Phi_n'(U^{(0)})$，或者可以是另外一次计算中得到的矩阵，只要$\tilde{\Phi}_n'$与$\Phi_n'(U^{(e)})$有一定的近似程度都可以用。

由方程（11-61）确定的解$U=(u_0, u_1, \cdots, u_J)^T$与精确解$Y^*=(y^*(t_0), y^*(t_1), \cdots, y^*(t_J))^T$的误差为$O(h^2)$的量。从精

度阶来看误差比较大，为了提高误差的阶，下面利用误差的延期校正原理推导一个逐次提高阶的迭代格式。

11.4.1 误差的延期校正方法

问题（11-56）、（11-57）可以写成一般的非线性算子方程

$$F(y)=0 \tag{11-65}$$

的形式，边界条件已含在F中。而函数$y^*(t)$是它的唯一解。通过离散化方法，我们将方程归化成离散方程（11-61）。它的解U是解y^*限制在格点集上的量y^*的近似。称

$$\varepsilon = U - Y^* \tag{11-66}$$

为全局误差。于是，由方程（11-61）得到的解U有$\varepsilon = O(h^2)$，即近似解U的精度阶为2。误差延期校正的思想是根据计算得到的近似解去估计计算过程中所引进的误差，然后将估计得到的误差从计算过程中修正掉，从而得到更精确的近似解。这样逐次提高可得到愈来愈精确的解。

由此可以知道，误差延期校正过程由下面三部分组成：数值解U，误差估计算子Ψ和计算更新解$\bar{U}$的方法Φ。即对给定的数值解U，由算子Ψ得到误差估计$\Psi(U)$，然后由

$$\Phi(\bar{U})=\Psi(U) \tag{11-67}$$

计算更新解$\bar{U}$。希望每完成一次校正过程，精度阶能提高一个固定的量。事实上，文献[53]给出了下面的重要结果。

定理11-2 设下列条件成立

（1）问题（11-65）具有孤立解。

（2）方程

$$\Phi(U)=0$$

的解U有渐近表示式

$$U=Y^*+O(h^p)$$

（3）算子$\Phi(\cdot)$在Y^*处是稳定的，即存在正常数s和r，使对任何满足

$$\|\Phi(Y^{(l)})-\Phi(Y^*)\|\leqslant r \qquad l=1,\ 2$$

的$Y^{(1)}$和$Y^{(2)}$，都有

$$\|Y^{(1)}-Y^{(2)}\|\leqslant s\,\|\Phi(Y^{(1)})-\Phi(Y^{(2)})\|$$

（4）局部截断误差估计算子$\Psi(\cdot)$满足

$$\Psi(Y^*)=\Phi(Y^*)+O(h^{r+p})$$

并且对于任意的$Y^{(1)}$，$Y^{(2)}$，有

$$\|\Psi(Y^{(1)})-\Psi(Y^{(2)})\|\leqslant kh^r\|Y^{(1)}-Y^{(2)}\|$$

则方程（11-67）的解$\bar{U}$有

$$\bar{U}=Y^*+O(h^{r+p})$$

证明 由方程（11-67），

$$\begin{aligned}\Phi(\bar{U})-\Phi(Y^*)&=\Psi(U)-\Phi(Y^*)\\&=\Psi(U)-\Psi(Y^*)+\Psi(Y^*)\\&\quad-\Phi(Y^*)\end{aligned}$$

因此

$$\begin{aligned}\|\Phi(\bar{U})-\Phi(Y^*)\|&\leqslant kh^r\|U-Y^*\|+\|\Psi(Y^*)-\Phi(Y^*)\|\\&=O(h^{r+p})\end{aligned}$$

由定理的稳定性条件推得定理的结论成立。

由方程（11-61）确定的解U有

$$U=Y^*+O(h^2)$$

下面我们构造一个校正过程，使得到的更新解$\bar{U}$有：

$$\bar{U}=Y^*+O(h^4)$$

然后用$\bar{U}$代替U，构造新的更新解。一般，若数值解U有渐近估计式

$$U=Y^*+O(h^p)$$

则由下述的校正过程得到的更新解$\bar{U}$有

$$\bar{U}=Y^*+O(h^{p+2})\qquad(11\text{-}68)$$

即每校正一次精度阶提高2。

我们先来给出误差估计算子Ψ的构造方法。利用泰勒级数展开，可以得到离散方程（11-59）的局部截断误差$T_n(y^*)\equiv\Phi_n(y^*)$的渐近展开式。事实上，有

$$T_n(y^*)_j=-\sum_{v=1}^{L}\frac{v}{2^{2v-1}(2v+1)}y^{*(2v+1)}_{j-(1/2)}\frac{h_j^{2v}}{(2v)!}+O(h^{2L+2}) \tag{11-69}$$

其中

$$y^{*(2v+1)}_{j-(1/2)}=\frac{d^{2v+1}}{dt^{2v+1}}y^*(t)\Big|_{t=t_{j-1}+\frac{1}{2}h_j}$$

$$=\frac{d^{2v}}{dt^{2v}}f(t,\ y^*(t))\Big|_{t=t_{j-1}+\frac{1}{2}h_j}$$

$$j=1,\ 2,\ \cdots,\ J$$

为简单起见，可将式（11-69）记成

$$T_n(y^*)_j=\sum_{v=1}^{L}T_v(y^*(\xi_j))h_j^{2v}+O(h^{2L+2}) \tag{11-70}$$

其中函数$T_v(y)$是式（11-69)中h_j^{2v}的系数。

用$F_{n,k}(y^*)_j$表示展开式（11-69）的前k项之和。对于每个$1\leqslant j\leqslant J$，令i_j是唯一的使不等式

$$\tau_{i_j}<t_{j-1}+\frac{1}{2}h_j<\tau_{i_j+1}$$

成立的足标。对于任意充分光滑的n维向量值函数$\mu(t)$，构造线性算子

$$\Psi_j^{(k)}(\mu)\equiv\sum_{i=0}^{2K}W_{j,i}\mu'_{i-p_j+i}\qquad j=1,\ 2,\ \cdots,\ J \tag{11-71}$$

其中$\mu'_{i-p_j+i}=\mu'(t_{i-p_j+i})$。$p_j$选成使不等式

$$\tau_{i_j}\leqslant t_{j-p_j+i}\leqslant\tau_{i_j+1} \tag{11-72}$$

对$i=0,\ 1,\ \cdots,\ 2k$均成立。(11-71）中的权函数$W_{j,i}$选成使对任意充分光滑的函数$\mu(t)$有

$$\Psi_j^{(k)}(\mu)=F_{n,k}(\mu)_j+O(h^{2k+2}) \tag{11-73}$$

显然（11-72）对构造的格点集π提出了要求：在每两个边界点之间至少应有$2k-1$个格点。当处理分段光滑的函数时，若

将不光滑点放到边界点集中去，则这个要求是非常需要的，可以避免构造$\Psi_j^{(k)}(\cdot)$的格点之间包含不光滑点，从而保证得到所需要的精度。

满足（11-73）的线性算子$\Psi_j^{(k)}(\cdot)$是很容易构造的，只要将插值微分的公式应用到向量函数$\mu'(t)$的格点上的值的各个分量上，然后代入$F_{n,k}(\cdot)_j$的导数项，经整理就能得到所需要的算子。

对于任意的$n(J+1)$维列向量$U=(u_0, u_1, \cdots, u_J)^T$定义

$$S_j^{(k)}(U) \equiv \sum_{i=0}^{2k} W_{j,i} f(t_{j-p_j+i}, u_{j-p_j+i})$$

其中权系数$W_{j,i}$与（11-71）中的相同。于是可得

$$S_j^{(k)}(Y^*) = \Psi_j^{(k)}(y^*)$$

如果$Y^{(k-1)}=(y_0^{(k-1)}, y_1^{(k-1)}, \cdots, y_J^{(k-1)})^T$是$Y^*$的$O(h^{2k})$精确的离散解，并且其误差$e_{k-1} \equiv Y^{(k-1)} - Y^*$具有$h$的偶次幂的渐近展开式，则有展开式

$$y_i^{(k-1)} - y^*(t_i) = e_k(t_i)h^{2k} + e_{k+1}(t_i)h^{2k+2} + O(h^{2k+4}) \tag{11-74}$$

其中$e_k(t)$和$e_{k+1}(t)$是光滑的，与格点集π无关，由$S_j^{(k)}$的构造，将有

$$S_j^{(k)}(y^*) = \Psi_j^{(k)}(y^*) = F_{n,k}(y^*)_j + O(h^{2k+2})$$

因此

$$\begin{aligned}
&\Phi_n(y^*)_j - S_j^{(k)}(Y^{(k-1)}) \\
&\quad = F_{n,k+1}(y^*)_j - S_j^{(k)}(y^* + e_k h^{2k} + e_{k+1}h^{2k+2}) \\
&\qquad + O(h^{2k+4}) \\
&\quad = F_{n,k+1}(y^*)_j - S_j^{(k)}(y^*) - S_j^{(k)}(e_k)h^{2k} \\
&\qquad - S_j^{(k)}(e_{k+1})h^{2k+2} + O(h^{2k+4}) \\
&\quad = F_{n,k+1}(y^*)_j - F_{n,k}(y^*)_j - S_j^{(k)}(e_k)h^{2k} + O(h^{2k+2}) \\
&\quad = T_{k+1}(y^*)_j h^{2k+2} - S_j^{(k)}(e_k)h^{2k} + O(h^{2k+2})
\end{aligned} \tag{11-75}$$

由于式 (11-73)，$S_j^{(k)}(e_k)=T_1(e_k)h^2+O(h^4)$，所以由上式得到

$$S_j^{(k)}(y^{(k-1)})=\Phi_n(Y^*)_j+O(h^{2k+2})$$

即$S_j^{(k)}(y^{(k-1)})$是$\Phi_n(Y^*)_j$的$O(h^{2k+2})$的近似，因而也是$F_{n,k}(y^*)_j$的$O(h^{2k+2})$近似，构造差分方程

$$\Phi_n(U)=S^{(k)}(Y^{(k-1)}) \tag{11-76}$$

其中 $S^{(k)}(\cdot)=(S_0^{(k)}(\cdot),\ S_1^{(k)}(\cdot),\ \cdots,\ S_j^{(k)}(\cdot))^T$，由式(11-75)，方程 (11-76) 的局部截断误差的阶为$O(h^{2k+2})$。由文献〔28〕，方程 (11-76) 的解 $Y^{(k)}$是 Y^* 的$O(h^{2k+2})$精确的离散解。

由$S_j^{(k+1)}(\cdot)$的构造，我们有

$$\begin{aligned}S_j^{k+1}(Y^{(k)})&=\Phi_n(Y^*)_j+O(h^{2k+4})\\&=F_{n,k+1}(y^*)_j+O(h^{2k+4})\end{aligned}$$

类似于式 (11-75) 的第二个等式

$$\begin{aligned}&S_j^{(k+1)}(y^{(k)})-S_j^{(k)}(Y^{(k-1)})\\&\quad=F_{n,k+1}(y^*)_j-S_j^{(k)}(y^*)-S_j^{(k)}(e_k)h^{2k}\\&\qquad+O(h^{2k+4})\end{aligned}$$

将其与式 (11-75) 比较，我们得到

$$\begin{aligned}S_j^{(k+1)}(Y^{(k)})-S_j^{(k)}(Y^{(k-1)})=\Phi_n(Y^*)_j-S_j^{(k)}(Y^{(k-1)})\\+O(h^{2k+4})\end{aligned}$$

即我们得到引理11-1。

引理11-1 量$S_j^{(k+1)}(Y^{(k)})-S_j^{(k)}(Y^{(k-1)})$是方程 (11-76)的局部截断误差的精度阶为$2k+4$的估计式。

这里要指出，为得到引理11-1，文献〔50〕要求构造$S_j^{(k)}(\cdot)$使$S_j^{(k)}(y^*)$是$F_{n,k}(y^*)_j$的$2k+4$阶的近似，而这里只需要构造$S_j^{(k)}(\cdot)$使$S_j^{(k)}(y^*)$是$F_{n,k}(y^*)_j$的$2k+2$阶的近似。

应用引理11-1，误差$\varepsilon_k=\equiv Y^{(k)}-Y^*$满足方程

$$\begin{aligned}\Phi_n(Y^{(k)}-\varepsilon_k)-S^{(k)}(Y^{(k-1)})=S^{(k+1)}(Y^{(k)})-S^{(k)}(Y^{(k-1)})\\+O(h^{2k+4})\end{aligned}$$

将$\Phi_n(Y^{(k)}-\varepsilon_k)$在$Y^{(k)}$处线性展开，并利用式 (11-76) 和 $e_k=O(h^{2k+2})$，得

$$\Phi_x'(Y^{(k)})e_k = S^{(k)}(Y^{(k-1)}) - S^{(k+1)}(Y^{(k)}) + O(h^{2k+4})$$

解出e_k

$$e_k = \Phi_x'(Y^{(k)})^{-1}[S^{(k)}(Y^{(k-1)}) - S^{(k+1)}(Y^{(k)})] + \Phi_x'(Y^{(k)})^{-1}O(h^{2k+4})$$

记其右边的第一项为Δ_k，则有

$$\Delta_k = e_k + O(h^{2k+4}) \tag{11-77}$$

它是误差e_k的精度为$O(h^{2k+4})$的估计，可以作为检验全局误差是否满足精度要求的检测量。

由上所述，方程（11-76）的解$Y^{(k)}$的精度阶比$Y^{(k-1)}$提高2，并且其误差e_k具有只含h的偶次幂的渐近展开式，所以$\Phi_x(\cdot)$可作为误差延期校正过程中的更新方法Φ。

这样，误差延期校正方法从$S^{(0)}(\cdot) \equiv 0$开始，逐次对$k=0, 1, 2, \cdots$求解方程（11-76）。每求一次，精度阶提高2，而在求解过程中，格点集和方程都是不变的，仅是方程（11-76）的右端项发生变化，这是它与理查森外插过程的不同处。

将上面的推导总结成下面的算法

算法11-6 误差延期校正算法

步0 取$\eta = S^{(0)} = 0$，选取初值$U^{(0)} = (u_0^{(0)}, u_1^{(0)}, \cdots, u_J^{(0)})^T$，令$k=0$。

步1 由算法11-5对选定的η和$U^{(0)}$计算方程（11-64）的解U，记$Y^{(k)} = U$。

步2 计算$S^{(k+1)}(Y^{(k)})$

步3 若$k=0$，令$\Delta_k = 0$，转步4。否则求解线性方程组

$$\Phi_x'(Y^{(k)})\Delta_k = S^{(k)}(Y^{(k-1)}) - S^{(k+1)}(Y^{(k)}) \tag{11-78}$$

并判别$\|\Delta_k\|$是否小于预给的精度参数。若小，则停止计算，否则转步4。

步4 令

$$\eta = S^{(k+1)}(Y^{(k)}) \quad U^{(0)} = Y^{(k)} + \Delta_k \quad k = k+1$$

转步1。

11.4.2 线性方程组的直接求解方法

在算法11-5和算法11-6中需要求解形如

$$\Phi_x'(U)x = b \tag{11-79}$$

的线性方程组，它可以利用求解线性方程组的并行算法来求解。矩阵 $\Phi_x'(U)$ 的稀疏性，可帮助我们构造求解它的算法。下面简单叙述系数矩阵的形状为（11-63）的块系统的直接解法。将 Φ_x' 进行分块，得

$$\Phi_x' = \begin{bmatrix} A & B \\ \cdots & \cdots \\ C & D \end{bmatrix} \begin{matrix} \}n \\ \}n*J \end{matrix} \tag{11-80}$$

$$\underbrace{\quad}_{n} \quad \underbrace{\quad}_{n*J}$$

考虑分块向量

$$x = \begin{bmatrix} x_0 \\ \cdots \\ \hat{x} \end{bmatrix} = \begin{pmatrix} x_0 \\ \vdots \\ x_1 \\ x_2 \\ \vdots \\ x_J \end{pmatrix} \qquad b = \begin{bmatrix} b_0 \\ \cdots \\ \hat{b} \end{bmatrix} = \begin{pmatrix} b_0 \\ \vdots \\ b_1 \\ b_2 \\ \vdots \\ b_J \end{pmatrix}$$

于是方程组（11-79)可写成

$$\begin{bmatrix} A & B \\ \cdots & \cdots \\ C & D \end{bmatrix} \begin{bmatrix} x_0 \\ \cdots \\ \hat{x} \end{bmatrix} = \begin{bmatrix} b_0 \\ \cdots \\ \hat{b} \end{bmatrix} \tag{11-81}$$

它可用消去法求解。将其写成显式，有

$$x_0 = (A - BD^{-1}C)^{-1}(b_0 - BD^{-1}\hat{b}) \tag{11-82}$$

$$D\hat{x} = (\hat{b} - Cx_0) \tag{11-83}$$

它的计算的主要工作量是求解系数矩阵为二对角块矩阵 D 的线性方程组，即要计算 $D^{-1}C$、$D^{-1}\hat{b}$ 和由方程（11-83）计算 $\hat{x}$。记

$$\tilde{C} = [C \vdots \hat{b}], \quad \tilde{V} = [V \vdots W]$$

其中 $V = D^{-1}C$，$W = D^{-1}\hat{b}$。于是 $\tilde{C}$ 和 $\tilde{V}$ 满足关系式

$$D\tilde{V} = \tilde{C}$$

将矩阵 $\tilde{C}$ 和 $\tilde{V}$ 每 n 行分成一块，并将这些块分别记作 $\tilde{V}_j$，$\tilde{C}_j$　$j = 1, 2, \cdots, J$。由 D 的特殊形式，得到。

$$\tilde{V}_j=R_j^{-1}(\tilde{C}_j-S_j\tilde{V}_{j-1}) \quad j=1,\ \cdots,\ J \tag{11-84}$$

其中 $\tilde{V}_0\equiv 0$。当然可以不求R_j^{-1}，而用消去法求$\tilde{V}_j$。

利用$\tilde{V}$，式（11-82）、(11-83）归化成线性方程组

$$\begin{aligned}(A-BV)x_0&=(b_0-BW)\\ \hat{x}&=W-Vx_0\end{aligned} \tag{11-85}$$

在大多数情况下，$n\times(n\times J)$ 矩阵B是非常稀疏的，最多有$n\times n$个非零元素。

11.4.3 自适应变阶变步长方法

对于给定的格点集，由算法11-6可以自动的改变近似解的精度阶来满足给定的误差。但是由于格点集π给定后，只提高精度阶不能保证计算精度一定能满足需要。另外，当精度阶提高时，算子$\Psi_j^{(k)}(\cdot)$和$S_j^{(k)}(\cdot)$的误差也增加。由于这些原因，只提高精度阶是不合适的。当精度阶提高，计算误差没有明显的改进时，可以通过缩小步长h_j来减小计算的误差。下面将描述一个自动变阶和变步长的算法。

取定一个基本格点集$\pi=\{t_j,\ j=0,\ 1,\ \cdots,\ J\}$，它包含边界点$\tau_i$，$i=1,\ 2,\ \cdots,\ N$，给定误差$TOL$。定义足标$j_i$为

$$t_{j_i}=\tau_i \quad i=1,\ 2,\ \cdots,\ N$$

记

$$\nu_i=j_{i+1}-j_i+1 \quad i=1,\ 2,\ \cdots,\ N-1$$

$$\nu=\min_i\nu_i$$

用$\pi_{1/2}$表示将π加密使包含所有的中点$t_{j-1}+\frac{1}{2}h_j$所得到的格点集。这时J变成$2J$。

算法11-7 自适应变阶变步长算法

步 0 给定基本格点集π和误差TOL。取小于1的正常数C。令$k=1$。

步 1 判别是否有

$$\nu < 2k+2 \tag{11-86}$$

如果不等式（11-86）成立，转步 8；否则转步 2。

步 2　若 $k=1$，令$S^{(k-1)}(\cdot)\equiv 0$，并计算$S^{(k)}(\cdot)$的权系数$W_{j,i}$，转步 3；若 $k>1$，计算$S^{(k)}(\cdot)$ 的权系数$W_{j,i}$。

步 3　令

$$\eta = S^{(k-1)}(Y^{(k-2)})$$

当 $k=1$ 时取 $\eta=0$，由算法11-5计算

$$\Phi_\eta(Y^{(k-1)}) = S^{(k-1)}(Y^{(k-2)})$$

的解$Y^{(k-1)}$，并计算$S^{(k)}(Y^{(k-1)})$。

步 4　求解方程组

$$\Phi'_\eta(Y^{(k-1)})\Delta_{k-1} = S^{(k-1)}(Y^{(k-2)}) - S^{(k)}(Y^{(k-1)})$$

得Δ_{k-1}。

步 5　如果有

$$\|\Delta_{k-1}\| \leqslant TOL$$

则停止计算。令$Y^{(k-1)}$为问题（11-56）、（11-57）的近似解；否则转步 6。

步 6　令NEWERROR$=\|\Delta_{k-1}\|$。如果 $k=1$，转步 7；否则如果下式成立

$$\text{NEWERROR} \leqslant C * \text{OLDERROR} \tag{11-87}$$

转步 7，若不成立，转步 8。

步 7　令 OLDERROR = NEWERROR，$K = k+1$，转步 1。

步 8　加密格点，计算格点集 $\pi_{1/2}$，并令 $\pi = \pi_{1/2}$，$k=1$，转步 1。

这个算法所采用的变阶变步长的策略是尽量在给定的网点集上通过提高精度阶来满足所要求的误差值，若提高阶时误差减小的速度太慢，或者阶提高到插值的格点不够时，则加密格点集，式（11-87）中的C表示在给定的格点上阶提高时误差减小的最低速度。若实际计算中减小的速度低于这个值，则加密格点。算法中尽量不加密格点，因为加密格点将使计算量成倍的增加。由

此看出，这种策略比起单纯加密格点会有很大的好处。

除求解第一个步 3 中的非线性方程组时需给出迭代初值外，其它可取$Y^{(k-2)}$作为迭代的初值。当加密格点时，可用粗糙格点上的解插值得到加密点上的值作为初值。

上述算法对式（11-56)的右端函数具有对 t 的间断点时也是可以处理的，只要将间断点包含在基本格点集中，例如可包含在 τ_i 中。若$\tau_i=t_{ji}$是一个间断点，对式（11-59)作下面的改变

$$\Phi_n(u)_{j_i}=\frac{1}{h_{j_i}}(u_{j_i}-u_{j_{i-1}})-\frac{1}{2}[f(t_{j_{i-1}},\ u_{j_{i-1}})+f^-(t_{j_i},\ u_{j_i})]=0$$

$$\Phi_n(u)_{j_{i+1}}=\frac{1}{h_{j_{i+1}}}(u_{j_{i+1}}-u_{j_i})-\frac{1}{2}[f^+(t_{j_i},\ u_{j_i})+f(t_{j_{i+1}},\ u_{j_{i+1}})]=0$$

最后我们指出，算法11-6的步 3，算法11-7的步 4 中的线性方程组的求解的计算量是比较小的，因为这时$S^{(k)}(Y^{(k-1)})$，$S^{(k+1)}(Y^{(k)})$是已知的，而且在采用算法11-5求解方程

$$\Phi_n(Y^{(k)})=S^{(k)}(Y^{(k-1)})$$

时，量$\Phi_n'(Y^{(k)})$ 或者它的因子分解都是已经计算好的。

11.5 纽面洛夫方法

考虑二阶常微分方程两点边值问题

$$y''(t)=f(t,\ y(t))\quad t\in(t_0,\ t_F)\tag{11-88}$$

$$y(t_0)=y_0,\quad y(t_F)=y_F\tag{11-89}$$

为简单起见假定 y（t）是充分光滑的实变量 t 的实值函数，函数 $f(\cdot,\ \cdot)$ 和数 t_0、t_F、y_0、y_F 均给定。下面讨论的方法可直接推广到 y 是 m 维向量值的情形。式（11-88)、(11-89）是只含偶数阶导数的$2m$阶方程。边界条件（11-89）也可以推广到一般的非线性边界条件。

为了得到问题（11-88)、(11-89）的解$y^*(t)$ 的近似，对于给定的自然数$N>1$，令

$$h=(t_F-t_0)/N \quad t_i=t_0+ih \quad i=0,\ 1,\ \cdots,\ N$$

要求计算格点 t_i 上解 $y^*(t_i)$ 的近似值 y_i（$i=1,\ \cdots, N-1$）。

记 $Y=(y_0,\ y_1,\ \cdots,\ y_N)^T$。$y^*(t_i)$ 的近似值 y_i，$i=1,\ \cdots,\ N-1$ 由边界条件式 (11-89)和差分方程

$$F_h(Y)_i\equiv h^{-2}(y_{i-1}-2y_i-y_{i+1})-\frac{1}{12}[f(t_{i-1},\ y_{i-1})+10f(t_i,\ y_i)+f(t_{i+1},\ y_{i+1})]=0$$
$$i=1,\ 2,\ \cdots,\ N-1 \qquad (11\text{-}90)$$

确定的方法称作纽面洛夫（Numerov）方法。

应该指出，用点 t_i 进行离散化时最好象11.4节中一样采用不等距的格点，象上述采用等距格点，可能会有许多缺点。例如：如果解 $y^*(t)$ 在区间 $[t_0,\ t_F]$ 的一些区域中变化非常缓慢，而在另外一些区域中变化非常快。于是为了保证求解的精度，必须采用非常密的格点。边界层问题就具有这种现象。下面为了讨论方便，采用等距格点，当然在许多情况下，它也是很好的。

在式 (11-90) 的左边用 $y^*(t_i)$ 代替 y_i，得到在 t_i 处的局部截断误差

$$F_h(y^*)_i=\sum_{K=2}^{k}a_kf^{(2k)}(t_i,\ y^*(t_i))\frac{h^{2k}}{(2k)!}+O(h^{2k+2})$$
$$i=1,\ 2,\ \cdots,\ n-1 \qquad (11\text{-}91)$$

其中 $f^{(2k)}(t,\ y^*(t))$ 表示 $f(t,\ y^*(t))$ 对 t 的 $2k$ 阶全导数，即有

$$f^{(2k)}(t,\ y^*(t))=\frac{d^{2k}}{dt^{2k}}f(t,\ y^*(t))$$

常数 a_k 为

$$a_k=\frac{1}{6}-\frac{1}{(k+1)(2k+1)}$$

式 (11-90) 求得的解 y_i 将有

$$y_i=y^*(t_i)+O(h^4)$$

当 $h\to 0$ 时，它对 i 是一致的。

类似于11.4节，采取延期校正方法可逐次得到精度阶愈来愈高的近似解，逐次的误差阶为$O(h^4)$，$O(h^8)$，$O(h^{12})$，…。为此，用$F_{h,m}(y^*)_i$表示$F_h(y^*)_i$的前$2m$项之和，而令S_m表示这和的误差阶为$O(h^{4m+4})$的差分近似。详细地讲，选取权$W_{ij}^{(m)}$，对任意的函数g（t）作和

$$S_m(g)(t_i)=\sum_j W_{ij}^{(m)}\, g(t_j) \tag{11-92}$$

而当g为充分光滑的函数时有

$$S_m(g)(t_i)=\sum_{K=2}^{2m+1}\frac{a_k g^{(2k)}(t_i)h^{2k}}{(2k)!}+O(h^{4m+4}) \tag{11-93}$$

将由式（11-90）求得的解Y记为$Y^{(0)}$，并记$Y^{(m)}=(y_0^{(m)},\ y_1^{(m)},\ \cdots,\ y_N^{(m)})^T$，由

$$F_h(Y^{(m)})_i=\sum_j W_{ij}^{(m)} f(t_j,\ y_j^{(m-1)}) \quad 1\leqslant i\leqslant N-1 \tag{11-94}$$

$$y_0^{(m)}=y_0 \quad y_N^{(m)}=y_F \tag{11-95}$$

逐次求解$Y^{(1)}$，$Y^{(2)}$，…。在适当的假定下，应用文献〔49〕中的方法可证明有

$$y_i^{(m)}=y^*(t_i)+O(h^{4m+4})$$

对i是一致成立的。

由$Y^{(m-1)}$通过式（11-94）、（11-95）得到$Y^{(m)}$的过程称作进行一步延期校正过程，m称作解$Y^{(m)}$的级。

为了估计近似解$Y^{(m)}$的误差，求解$e^{(m)}=(e_0^{(m)},\ e_1^{(m)},\ \cdots,\ e_N^{(m)})$的线性方程组

$$\begin{aligned}h^{-2}[e_{i-1}^{(m)}-2e_i^{(m)}+e_{i+1}^{(m)}]-\frac{1}{12}[&f_y(t_{i-1},\ y_{i-1}^{(m)})e_{i-1}^{(m)}\\&+10f_y(t_i,\ y_i^{(m)})e_i^{(m)}+f_y(t_{i+1},\ y_{i+1}^{(m)})e_{i+1}^{(m)}]\\=&S_m[Y^{(m-1)}]_i-S_{m+1}[Y^{(m)}]_i\end{aligned}$$

$$i=1,\ 2,\ \cdots,\ N-1 \tag{11-96}$$

$$e_0^{(m)}=0,\ e_N^{(m)}=0 \tag{11-97}$$

可推得

$$e_i^{(m)} = Y_i^{(m)} - y^*(t_i) + O(h^{(m+3)}) \qquad (11\text{-}98)$$

对 i 一致成立。所以 $e^{(m)}$ 可以作为 $Y^{(m)}$ 的误差的渐近估计式。当 $\|e^m\|$ 小于事先给定的允许误差值 TOL 时，取 $Y = Y^{(m)}$ 作为 $y^*(t)$ 的满足精度的近似值。

式（11-96）中的 $S_m[Y^{(m-1)}]_i$ 为

$$S_m[Y^{(m-1)}]_i = \sum_j W_{ij}^{(m)} f(t_j, y_j^{(m-1)})$$

下面来描述组面洛夫方法的延期校正过程的一个实现。输入数据是 t_0，t_F，y_0，y_F，初始格点距 h（或 $N = (t_F - t_0)/h$）的初始值、允许误差值 TOL。想通过延期校正过程求近似解 Y，使 $\max_i |y_i - y^*(t_i)| \leqslant TOL$ 成立。计算过程是这样的：先在初始格点距 h 下计算 $Y^{(m)}$（$m = 0$，1，…）。在某一个 $Y^{(m)}$ 计算后计算 $e^{(m)}$。检验 $\max_i |e_i^m| \leqslant TOL$ 是否成立，如果成立，取 $Y^{(m)}$ 为解 Y。如果不成立，若有

$$\max_i |e_i^{(m)}| < c \max_i |e_i^{(m-1)}|$$

则继续进行误差延期校正步，由 $Y^{(m)}$ 得到 $Y^{(m+1)}$。否则将格点距 h 缩小一半，即将 N 加倍，对新的 h 重新计算 $Y^{(0)}$，c 为小正数。例如：取

$$c < 0.1$$

由上所述，算法的计算过程与算法11-7是完全类似的，其中的计算并行性处理可与算法11-7一样进行。

下面我们叙述实现这个算法时的一些处理

（1）避免无用的高阶校正。由 $Y^{(m)}$ 得到 $Y^{(m+1)}$ 的 m 级校正步是根据 $Y^{(m)}$ 的误差渐近地按 Ah^{4m+4} 变化的假定下进行的，其中 A 是某个常数。但是在进行校正时，我们常常不验证这个假定是否成立，但往往会出现在校正后发现计算无用的现象，希望能尽可能避免这种现象的发生。

如果在 m 级的误差确实按 Ah^{4m+4} 随 h 变化，则在 $h = (t_f - t_0)/N$ 处的误差估计 $e^{(m)}(h)$ 和在 $2h = (t_f - t_0)/(N/2)$ 处的误差估

计$e^m(2h)$之间近似地有

$$r_{mj}=e_j^m(2h)/e_i^m(h)\approx 2^{4m+4} \tag{11-99}$$

其中j和i之间的关系为$i=2j\left(j=1,\ 2,\cdots,\frac{N}{2}-1\right)$。因此，若对格点距$h=(t_f-t_0)/N$已得到$Y^{(m)}$后，要确定是否进行校正步由$Y^{(m)}$得到$Y^{(m+1)}$，如果这时已有对格点距$2h$的结果$Y^{(m)}$，则可以计算由式（11-99）确定的$r_{mj}$。当$r_{mj}$与$2^{4m+4}$相近，即有

$$(\ln r_m)/(4m+4)\approx \ln 2\approx 0.693 \tag{11-100}$$

时，求$Y^{(m+1)}$是合理的。当然实际计算得到的r_{mj}并不需要近似式比较好的成立。文献[51]指出只需有$0.55\leqslant(\ln r_m)/(4m+4)\leqslant 0.85$就可以了，于是可得下面的粗略的估计：$9\leqslant r_0\leqslant 27$，$81\leqslant r_1\leqslant 729$，$729\leqslant r_2\leqslant 20000$，$6561\leqslant r_3\leqslant 500000$，$60000\leqslant r_4\leqslant 14000000$，$500000\leqslant r_5\leqslant 400000000$。

当然，在第一次计算出m级的结果时，相应的$2h$的结果是没有的，因此无法计算r_m来进行检验。这表明，如果在进行延期校正之前要能进行检验，则对于每个N的值只能往前推进一个级。

算法11-7没有进行这种检验，而是检验式（11-87）是否成立。在实际计算时，可将这两种检验组合起来。若有相应m的$2h$的结果，检验r_m的值。若没有$2h$的结果，取适当的c的值。如$c=0.1$，检验式（11-87）。但限制只进行一次，这样对于每个N可以往前推进两个级，限制较小，同时又可避免进行无用的高阶校正。

（2）避免进行不必要的低阶计算。在11.4节和本节上述的算法中，当计算达到m级后，发现计算精度没有达到，同时进行高阶校正无效时，将格点距$h=(t_f-t_0)/N$二等分，然后再从零级开始计算，记新的值为Y_N，而旧的值为Y_{0c}。通常在相同的格点上$Y_N^{(0)}$的值的精度没有$Y_0^{(m)}$的好，在算法11-7中可由$Y_0^{(m)}$的值插出$\tilde{Y}_N$作为求$Y_N^{(0)}$的初值。但实际上，只要插值的精度阶比较高，$\tilde{Y}_N$的精度比$Y_N^{(0)}$的高。于是为了节省计算量，计算可以不从零级开始，为了确定从哪一级开始计算比较好，我们首先假定

$\tilde{Y}_N$ 与y^*之间的误差$\tilde{Y}_N-y^*$的量级与误差$y_0^{(m)}-y^*$是一样的。用量$\|e_0^{(m)}\|_\infty=\max_i|e_0^{(m)}|$ 作为$\tilde{Y}_N$的误差的估计。由于对旧的步长，我们已计算到m级，因而已得到估计$e_0^{(0)}, e_0^{(1)}, \cdots, e_0^{(m-1)}$。新的步长相应的误差可以取为 $\|e_0^{(i)}\|_\infty/2^{4i+4}\equiv\tilde{e}_N^{(i)}$。将 $\|e_0^{(m)}\|_\infty$ 与 $\tilde{e}_N^{(i)}$ 进行比较，取满足 $\|e_0^{(m)}\|_\infty<\tilde{e}_N^{(i)}$ 的最大的 i，并将 $\tilde{Y}_N$ 作为$Y_N^{(i)}$的值，于是由延期校正方法从 $Y_N^{(i)}$ 计算 $Y_N^{(i+1)}$。从而避免直接计算 $Y_N^{(0)}$，$Y_N^{(1)}$，$\cdots$，$Y_N^{(i)}$，即避免了不必要的低阶计算。

（3）非线性方程的求解。在所用的格点距 h 的每一级的计算中，都需要求解非线性方程组（11-90）或非线性方程组（11-94）、（11-95）。为了讨论方便，我们描述应用到非线性方程

$$G(Z)=0 \tag{11-101}$$

的方法。在11.4节和本节中所用的方法均是标准的牛顿方法

$$Z_{k+1}=Z_k-[G'(Z_k)]^{-1}G(Z_k) \tag{11-102}$$

按通常的求解方法，在牛顿法迭代的每一步和对每一个格点距的所有级，G' 均需要重新计算，这是很费时间的。另外一个问题是迭代的初值Z_0的选取可以加速迭代的收敛，下面讨论对这个方法的一些改进。

当然，从理论上来看，式（11-102）中的每一个 G' 若都重新计算，可保证得到二次收敛性：$\|Z^*-Z_{k+1}\|_\infty<d\|Z^*-Z_k\|_\infty^2$，其中 Z^*为（11-101）的解。但实际求解时可以简化。假设 $i=0$ 时的方程（11-90）和 $i\neq 0$ 时的方程（11-94）、（11-95）都表示成式（11-101）。如果在 $i=0$ 时，通过式（11-102）已求得式（11-101）的解$Y^{(0)}$，或者对一般的 i，通过对格点距为 $2h$的值进行插值已求得格点距为 h 的点上的值 $Y^{(i)}$，则在同样的新格点距 h 的点上求 $j>i$ 级上的值 $Y^{(j)}$时，可以通过迭代

$$Z_{k+1}=Z_k-[G'(Y^{(j)})]^{-1}G(Z_k) \tag{11-103}$$

求得，其中$G'(Y^{(j)})$ 不需要重新计算。

由于$Y_k^{(r)}=y^*(t_k)+O(h^{4r+4})$，$k=1$，$2$，$\cdots$，$N-1$，所以至少有$Y^{(i)}-Y^{(j)}=O(h^4)$。这样迭代式（11-102）的平方收

敛性换成线性收敛性

$$\|Z^*-Z_{k+1}\|_\infty \leqslant dh^4\|Z^*-Z_k\|_\infty \tag{11-104}$$

由于$Y_k^{(j)}=y^*(t_k)+O(4^{4j+4})$，而$Y_k^{(j-1)}=y^*(t_k)+O(h^{4(j-1)+4})=y^*(t_k)+O(h^{4j+4})/h^4$。所以

$$h^4\|Y^{(j+1)}-Y^{(j)}\|_\infty=O(h^{4j+4}) \tag{11-105}$$

若由式（11-103）迭代计算$Y^{(j)}$，而初值$Z_0=Y^{(j-1)}$，则若在式（11-104）中取$Z^*=Y^{(j)}$，由式（11-104）、（11-105），得

$$\|Y^{(j)}-Z_1\|_\infty \leqslant dh^4\|Y^{(j)}-Y^{(j-1)}\|_\infty=O(h^{4j+4})$$

这表示由式（11-103）迭代一步得到的Z_1就可以作为$Y^{(j)}$，这样可节省大量的计算。

11.6 求解两点边值问题的高阶差分方法

前面两节中我们应用的基本差分公式的精度阶比较低。对于大型的问题或者精度要求很高的中等问题，应用低阶公式的缺点主要是存储量要求太大。当然前两节采用延期校正方法可逐次提高精度阶。但每提高一次均需要求解一个大型的非线性方程组。计算量也是很大的。研究高阶的有限差分方法将可以适当的克服这些缺点。这一节我们描述一个推导高阶有限差分公式的一个一般的模式，再给出几个具体的公式，最后讨论求解公式导出的非线性方程组的一些方法。由此得到的算法都可并行实现。

考虑常微分方程边值问题

$$\frac{dy}{dt}=f(t,\ y) \tag{11-106}$$

$$g(y(a),\ y(b))=0 \tag{11-107}$$

设格点是等距的，格点距为h，$t_0=a$，$t_i=t_0+ih$。$t_N=b$。数值积分式（11-106）的公式采用m点洛巴托（Lobatto）型公式

$$y_{i+1}-y_i=h\sum_{j=0}^{m-1}\beta_j\{f(t_{i+\alpha_j},\ y_{i+\alpha_j})+f(t_{i-\alpha_j},\ y_{i-\alpha_j})\},\quad \alpha_0=\frac{1}{2} \tag{11-108}$$

其中 $v=i+\frac{1}{2}$，$t_{v\pm\alpha_j}=t_v\pm\alpha_j h$，$t_v=\frac{1}{2}(t_i+t_{i+1})$，$\beta_j$ 是要选取的权系数，α_j 是正数，由它确定右函数 f 的计算点 $t_v\pm\alpha_j h$。下面的处理的基本思想是由 y_i，y_{i+1}，$f(t_i, y_i)$，$f(x_{i+1},y_{i+1})$ 和 $f(t_{v\pm\alpha_k}, \bar{y}_{v\pm\alpha_k})$，$k<j$ 的线性组合，构造量 $\bar{y}_{v\pm\alpha_j}$，并用 $\bar{y}_{v\pm\alpha_j}$ 近似代替 $y_{v\pm\alpha_j}$。这样我们得到一种特殊结构的隐式龙格-库塔方法。在这种构造中，只需要将 y_i，y_{i+1} 看成需要求解的量，而 $\bar{y}_{v\pm\alpha_j}$ 可以由它逐次构造出来。这种构造又有点象并行打靶法，y_i，y_{i+1} 是子区间的两个端点上的值，$\bar{y}_{v\pm\alpha_j}$ 看成是子区间内部格点上的值，但这些点在构造上是不同的。

令 A_j，γ_{jk} 和 μ_{jk}，$0\leqslant k<j$ 是要选取的常数。我们选 $A_0=1$ 而 $\gamma_{0k}=\mu_{0k}=0$。对于 $j=0$，1，2…，m，递推地定义函数

$$W_j(t, z, u, v)=A_j v+(1-A_j)u + z\sum_{K=0}^{j-1}\{\gamma_{jk}\phi_k(t, z, u, v) +\mu_{jk}\phi_k(t, -z, v, u)\} \tag{11-109a}$$

$$\phi_j(t, z, u, v)=f(t+\alpha_j z, W_j(t, z, u, v)) \tag{11-109b}$$

取

$$\bar{y}_{v+\alpha_j}=W_j(t_v, h, y_i, y_{i+1}) \tag{11-110}$$

$$\bar{y}_{v-\alpha_j}=W_j(t_v, -h, y_{i+1}, y_i) \tag{11-111}$$

用 $\bar{y}_{v\pm\alpha_j}$ 代替式（11-108）中的 $y_{v\pm\alpha_j}$，得到的公式可写成

$$y_{i+1}-y_i-h\sum_{j=0}^{m-1}\beta_j\{\phi_j(t_v, h, y_i, y_{i+1})+\phi_j(t_v, h, y_{i+1}, y_i)\}=0 \tag{11-112}$$

其中 $j=0$ 的项为 $\beta_0\{f(t_{i+1}, \varphi_{i+1})+f(t_i, y_i)\}$。设 $y(t)$ 为微分方程（11-106）的解。将 $y\left(t_v-\frac{1}{2}h\right)$ 和 $y\left(t_v+\frac{1}{2}h\right)$ 分别

代入式（11-112）中的 y_i 和 y_{i+1}，得到积分公式（11-112）的局部截断误差 $TE(h, t_v)$，显然有 $TE(h, t_v)=TE(-h, t_v)$。式（11-112）可以写成形式

$$\frac{y_{i+1}-y_i}{h}=\Psi(t_v, h, y_i, y_{i+1})$$

$$\equiv\sum_{j=0}^{m-1}\beta_j\{\phi_j(t_v, h, y_i, y_{i+1})+\phi_j(t_v, -h, y_{i+1}, y_i)\}$$

函数 $\Psi(t_v, h, y_i, y_{i+1})$ 满足对称性

$$\Psi(t_v, h, y_i, y_{i+1})=\Psi(t_v, -h, y_{i+1}, y_i)$$

由文献〔31〕的结果推得下面的定理

定理11-3 由公式（11-108）求得的问题（11-106）、（11-107）的近似解 $\{y_n\}$，如果它精确地满足边界条件（11-107），或者它与精确解 $y(a)$，$y(b)$ 之间的误差可展开成仅含 h 的偶次幂的幂级数，则全局误差 $y_n-y(t_n)$ 可展开成仅含 h 的偶次幂的幂级数。

下面简短地列出形式为式（11-108）的4阶、6阶和8阶的公式。$m=1$ 的公式是通常的2阶梯形公式。

11.6.1 四阶公式

在公式（11-108）中取 $m=2$，给出基本的积分公式

$$y_{i+1}-y_i=h\{\beta_0(y'_{i+1}+y'_i)+\beta_1(y'_{v+a1}+y'_{v-a1})\}\quad v\equiv i+\frac{1}{2}$$

作 y_i，y_{i+1}，y'_i，y'_{i+1} 的线性组合

$$\bar{y}_{v+a1}=\left(\frac{1}{2}+3\alpha_1/2-2\alpha_1^3\right)y_{i+1}+\left(\frac{1}{2}-3\alpha_1/2+2\alpha_1^3\right)y_i$$
$$+h\left(-\frac{1}{8}-\alpha_1/4+\alpha_1^2/2+\alpha_1^3\right)y'_{i+1}$$
$$+h\left(\frac{1}{8}-\alpha_1/4-\alpha_1^2/2+\alpha_1^3\right)y'_i \qquad (11\text{-}113)$$

$$\bar{y}_{v-a1}=\left(\frac{1}{2}-3\alpha_1/2+2\alpha_1^3\right)y_{i+1}+\left(\frac{1}{2}+3\alpha_1/2-2\alpha_1^3\right)y_i$$

$$+h\left(-\frac{1}{8}+\alpha_1/4+\alpha_1^2/2-\alpha_1^3\right)y'_{i+1}$$

$$+h\left(\frac{1}{8}+\alpha_1/4-\alpha_1^2/2-\alpha_1^3\right)y'_i$$

用 $\bar{y}_{i\pm\alpha_1}$ 代替 $y_{i\pm\alpha_1}$，得到公式

$$y_{i+1}-y_i=h\{\beta_0(y'_{i+1}+y'_i)+\beta_1(f(t_{i+\alpha_1},\bar{y}_{i+\alpha_1})+f(t_{i-\alpha_1},\bar{y}_{i-\alpha_1}))\} \tag{11-114}$$

如果取

$$\beta_0=(1-12\alpha_1^2)/(6-24\alpha_1^2),\quad \beta_1=1/(3-12\alpha_1^2) \tag{11-115}$$

则公式（11-114）的精度阶为 4，并且这个公式的局部截断误差的主项为

$$\frac{(1-4\alpha_1^2)}{576}h^5\frac{\partial f}{\partial y}(t_i,\ y_i)y_i^{(V)}+\frac{1}{5760}(-2+40\alpha_1^2)h^5y_i^{(V)} \tag{11-116}$$

α_1 的选取依赖于求解的问题的特征。如果想使右函数计算的次数减少，可以选取 $\alpha_1=0$，由式（11-115）、（11-113），这时公式（11-114）有

$$y_{i+1}-y_i=\frac{h}{6}(f_{i+1}+f_i)+\frac{2}{3}hf\left(t_{i+1/2},\ \frac{y_{i+1}+y_i}{2}-\frac{h}{8}(f_{i+1}-f_i)\right) \tag{11-117}$$

若右函数 $f(t,\ y)$ 的计算需要的时间比起其它的运算要少，可以选 α_1 使函数

$$|10-40\alpha^2|+|-2+40\alpha^2|$$

达极小时。容易验证，取 α_1 满足 $2\leqslant 40\alpha_1^2\leqslant 10$ 将可以达到这个目的。但不能选 α_1 使式（11-116）的第一项为零。因这时 $\alpha_1^2=\frac{1}{4}$，使 β_0 和 β_1 均变成无限大。

11.6.2 六阶公式

公式（11-108）中取$m=3$，得到公式

$$y_{i+1}-y_i=h\{\beta_0(y'_{i+1}+y'_i)+\beta_1(y'_{\nu+a_1}+y'_{\nu-a_1})+\beta_2(y'_{\nu+a_2}+y'_{\nu-a_2})\} \tag{11-118}$$

如果取

$$\beta_0=\frac{\frac{1}{10}-\frac{2}{3}a_1^2-\frac{2}{3}a_2^2+2a_1^2a_2^2}{(1-4a_1^2)(1-4a_2^2)}$$

$$\beta_1=\frac{\frac{1}{60}-\frac{1}{3}a_2^2}{(1-4a_1^2)(a_1^2-a_2^2)} \tag{11-119}$$

$$\beta_2=\frac{\frac{1}{3}a_1^2-\frac{1}{60}}{(1-4a_2^2)(a_1^2-a_2^2)}$$

这个公式的精度阶为6，并且局部离散化误差的主项为

$$T=\frac{h^7y^{(7)}}{46080}\left\{\frac{-\frac{1}{35}+\frac{8}{21}(a_1^2+a_2^2)-\frac{48}{7}a_1^2a_2^2+\frac{128}{5}a_1^4a_2^2}{(1-4a_1^2)}\rightarrow\right.$$

$$\left.\leftarrow\frac{-\frac{256}{3}a_1^4a_2^4-\frac{16}{15}(a_1^4+a_1^2a_2^2+a_2^4)}{(1-4a_2^2)}\right\} \tag{11-120}$$

我们用式（11-113）给出的$\bar{y}_{\nu\pm a_1}$代替公式（11-118）中的$y_{\nu\pm a_1}$用$\bar{y}_{\nu+a_1}$代替$y_{\nu+a_1}$的误差的主项为

$$-\frac{h^4}{4!}\left(a_1^2-\frac{1}{4}\right)^2y^{(7)}_{\nu+a_1}+a_1\frac{h^5}{5!}\left(2a_1^2-\frac{1}{2}\right)^2y^{(7)}_{\nu+a_1}$$

$$-\frac{h^6}{6!}\left(a_1^2-\frac{1}{4}\right)^2\left(10a_1^2+\frac{1}{2}\right)y^{(7)}_{\nu+a_1} \tag{11-121}$$

因此，若在式（11-118）中，$y_{\nu\pm a_2}$取精确值，而$y_{\nu\pm a_1}$取近似值$\bar{y}_{\nu\pm a_1}$，得到的公式的局部截断误差的主项为

$$T-h\beta_1\left\{\frac{\partial f_a}{\partial y}\left(-\frac{h^4}{4!}\left(\alpha_i^2-\frac{1}{4}\right)^2 y_a^{(\mathrm{IV})}+\alpha_1\frac{h^5}{5!}\left(2\alpha_i^2-\frac{1}{2}\right)^2 y_a^{(\mathrm{V})}\right.\right.$$
$$\left.-\frac{h^6}{6!}\left(\alpha_i^2-\frac{1}{4}\right)^2\left(10\alpha_i^2+\frac{1}{2}\right)y_a^{(\mathrm{VI})}\right)$$
$$+\frac{\partial f_{-a}}{\partial y}\left[-\frac{h^4}{4!}\left(\alpha_i^2-\frac{1}{4}\right)^2 y_{-a}^{(\mathrm{IV})}-\alpha_1\frac{h^5}{5!}\left(2\alpha_i^2-\frac{1}{2}\right)^2 y_{-a}^{(\mathrm{V})}\right.$$
$$\left.\left.-\frac{h^6}{6!}\left(\alpha_i^2-\frac{1}{4}\right)^2\left(10\alpha_i^2+\frac{1}{2}\right)y_{-a}^{(\mathrm{VI})}\right]\right\} \qquad (11-122)$$

其中 $y_a\equiv y(t_{\nu+a})$，$\frac{\partial f_a}{\partial y}\equiv\frac{\partial f}{\partial y}\Big|_{(t_\nu+a,y_a)}$

将式（11-122）在点 $(t_\nu,\ y(t_\nu))$ 处展开，可得到局部截断误差的形式

$$DE=T-2\beta_1\left(\alpha_i^2-\frac{1}{4}\right)^2\frac{h^5}{4!}\frac{\partial f}{\partial y}y^{(\mathrm{IV})}+O(h^7) \qquad (11-123)$$

由此看出，为了使得到的公式仍具有精度阶 6，我们必须作 $y_{\nu\pm a_2}$ 的差分近似，使它所产生的误差项恰好能与式（11-123）中出现的 h^5 项抵消。由式（11-122）还可看到，通过适当的选取 α_2 不会使式（11-118）的离散误差有大的减小。因此为了使每次迭代时，在每个格点处节省一次函数求值，我们取 $\alpha_2=0$，由式(11-109)、(11-110)、(11-111）看到，我们要寻找的 y_ν 的有限差分近似 $\bar{y}_\nu$ 有如下形式

$$y_\nu\cong\bar{y}_\nu=A_2y_{i+1}+(1-A_2)y_i+h(\gamma_{20}y'_{i+1}+\mu_{20}y'_i$$
$$+\gamma_{21}f(t_{\nu+a_1},\bar{y}_{\nu+a_1})+\mu_{21}f(t_{\nu-a_1},\bar{y}_{\nu-a_1})) \qquad (11-124)$$

记式（11-124）的右边为 $G(y_i,\ y_{i+1},\ y_{\nu\pm a_1})$。为了得到6阶公式，需要选取 A_2，γ_{20}，μ_{20}，γ_{21}，和 μ_{21} 的值，使有

$$G(y(t_i),\ y(t_{i+1}),\ y(t_{\nu\pm a_1}))=y(t_\nu)+Rh^4y^{(\mathrm{IV})}+O(h^6)$$

其中 $\beta_2R=(\beta_1/4!)\left(\alpha_i^2-\frac{1}{4}\right)^2$。这个要求导出线性方程组

$$A_2=\frac{1}{2},\quad \gamma_{20}=-\mu_{20},\quad \gamma_{21}=-\mu_{21},\quad -\frac{1}{16}+\gamma_{20}/2+\gamma_{21}\alpha_1=0$$

$$\frac{1}{32}+\gamma_{20}/2+4\gamma_{21}\alpha_i^2=(\beta_1/\beta_2)\left(\alpha_i^2-\frac{1}{4}\right)^2/4!$$

取$\alpha_1=\frac{1}{4}$，最后导出下面的6阶积分公式

$$y_{i+1}-y_i=h\left\{\frac{7}{90}(y'_{i+1}+y'_i)+\frac{16}{45}\left[f(t_{i+1/4},\ \bar{y}_{i+1/4})\right.\right.$$
$$\left.\left.+f(t_{i+3/4},\bar{y}_{i+3/4})\right]+\frac{2}{15}f(t_{i+1/2},\ \bar{y}_{i+1/2})\right\} \quad (11\text{-}125)$$

其中

$$\bar{y}_{i+1/4}=\frac{5}{32}y_{i+1}+\frac{27}{32}y_i+h\left(\frac{9}{64}y'_i-\frac{3}{64}y'_{i+1}\right)$$
$$\bar{y}_{i+3/4}=\frac{27}{32}y_{i+1}+\frac{5}{32}y_i+h\left(-\frac{9}{64}y'_{i+1}+\frac{3}{64}y'_i\right)$$
$$\bar{y}_{i+1/2}=\frac{1}{2}(y_{i+1}+y_i)+\frac{5}{24}h(y'_{i+1}-y'_i)$$
$$-\frac{2}{3}h\left\{f(t_{i+3/4},\ \bar{y}_{i+3/4})-f(t_{i+1/4},\ \bar{y}_{i+1/4})\right\}$$

11.6.3 八阶公式

8阶的公式可类似2，4，6阶公式的推导。下面只列出8阶的公式

$$y_{i+1}-y_i=h\sum_{j=0}^{4}\beta_j\{f(t_{v+a_j},\ \bar{y}_{v+a_j})+f(t_{v-a_j},\bar{y}_{v-a_j})\},$$
$$v=i+\frac{1}{2} \quad (11\text{-}126)$$

其中

$$\beta_0=\frac{1}{48},\ \beta_1=0,\ \beta_2=\frac{392}{3000},\ \beta_3=\frac{441}{2000},\ \beta_4=\frac{16}{125}$$

$$\alpha_0=\frac{1}{2},\ \alpha_1=\frac{1}{8},\ \alpha_2=\sqrt{(5/28)},\ \alpha_3=\sqrt{(5/84)},\ \alpha_4=0$$

$$\bar{y}_{v+a_j}=A_1y_{i+1}+(1-A_1)y_i+h\{\gamma_{10}f(t_{i+1},\ y_{i+1})$$
$$+\mu_{10}f(t_i,\ y_i)\} \quad (11\text{-}127)$$
$$\bar{y}_{v-a_j}=(1-A_1)y_{i+1}+A_1y_i-h\{\mu_{10}f(t_{i+1},\ y_{i+1})$$
$$+\gamma_{10}f(t_i,\ y_i)\}$$

$$A_1=0.68359375,\quad \gamma_{10}=-0.146484375,\quad \mu_{10}=0.087890625$$

$$\begin{aligned}\bar{y}_{v+a_2}&=A_2y_{i+1}+(1-A_2)y_i+h\{\gamma_{20}f(t_{i+1},\ y_{i+1})\\&\quad+\mu_{20}f(t_i,\ y_i)+\gamma_{21}f(t_{v+a_1},\ \bar{y}_{v+a_1})\\&\quad+\mu_{21}f(t_{v-a_1},\ \bar{y}_{v-a_1})\}\\\bar{y}_{v-a_2}&=A_2y_i+(1-A_2)y_{i+1}-h\{\mu_{20}f(t_{i+1},\ y_{i+1})\\&\quad+\gamma_{20}f(t_i,\ y_i)+\mu_{21}f(t_{v+a_1},\ \bar{y}_{v+a_1})\\&\quad+\gamma_{21}f(t_{v-a_1},\ \bar{y}_{v-a_1})\}\end{aligned}\qquad(11\text{-}128)$$

$$A_2=0.90768108762415,\quad \gamma_{20}=-0.049584693699578$$

$$\mu_{20}=0.010959523987495$$

$$\gamma_{21}=0.0049918972431044,\quad \mu_{21}=0.048529312209089$$

$$\begin{aligned}\bar{y}_{v+a_3}&=A_3y_{i+1}+(1-A_3)y_i+h\{\gamma_{30}f(t_{i+1},\ y_{i+1})\\&\quad+\mu_{30}f(t_i,\ y_i)+\gamma_{31}f(t_{v+a1},\ \bar{y}_{v+a1})\\&\quad+\mu_{31}f(t_{v-a_1},\ \bar{y}_{v-a_1})+\gamma_{32}f(t_{v+a_2},\ \bar{y}_{v+a_2})\\&\quad+\mu_{32}f(t_{v-a_2},\ \bar{y}_{v-a_2})\}\\\bar{y}_{v-a_3}&=(1-A_3)y_{i+1}+A_3y_i-h\{\mu_{30}f(t_{i+1},\ y_{i+1})\\&\quad+\gamma_{30}f(t_i,\ y_i)+\mu_{31}f(t_{v+a_1},\ \bar{y}_{v+a_1})\\&\quad+\gamma_{31}f(t_{v-a_1},\ \bar{y}_{v-a_1})+\mu_{32}f(t_{v+a_2},\ \bar{y}_{v+a_2})\\&\quad+\gamma_{32}f(t_{v-a_2},\ \bar{y}_{v-a_2})\}\end{aligned}\qquad(11\text{-}129)$$

$$A_3=1.0174389430282,\quad \gamma_{30}=0.028248623573731$$

$$\mu_{30}=-0.0058153723950256,\quad \gamma_{31}=-0.053267047106879$$

$$\mu_{31}=-0.0016671478879304,\quad \gamma_{32}=-0.24568952822503$$

$$\mu_{32}=0.0047265472500833$$

$$\begin{aligned}\bar{y}_v&=A_4(y_{i+1}+y_i)+h\{\gamma_{40}[f(t_{i+1},\ y_{i+1})-f(t_i,\ y_i)]\\&\quad+\gamma_{41}[f(t_{v+a_1},\ \bar{y}_{v+a_1})-f(t_{v-a_1},\ \bar{y}_{v-a_1})]\\&\quad+\gamma_{42}[f(t_{v+a_2},\ \bar{y}_{v+a_2})-f(t_{v-a_2},\ \bar{y}_{v-a_2})]\\&\quad+\gamma_{43}[f(t_{v+a_3},\ \bar{y}_{v+a_3})-f(t_{v-a_3},\ \bar{y}_{v-a_3})]\}\end{aligned}\qquad(11\text{-}130)$$

$$A_4=0.5,\quad \gamma_{40}=-0.039127604166667,\quad \gamma_{41}=\frac{1}{15},\quad \gamma_{42}=0,$$

$$\gamma_{43}=-0.21014254500503$$

11.6.4 自适应格点延期校正并行迭代算法

考虑式（11-107）为分离的情形，即假定有形状

$$g(v,\ w)=\begin{bmatrix} g_1(v) \\ g_2(w) \end{bmatrix}=0 \tag{11-131}$$

其中 $g_1(v)$ 是 p 维向量，$g_2(w)$ 是 q $(=(s-p))$ 维向量记

$$Y=\begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_N \end{bmatrix} \tag{11-132}$$

于是方程组（11-107）、(11-108) 可写成形式

$$\Phi(Y)\equiv\begin{Bmatrix} g_1(y_0) \\ L_h(y_1) \\ L_h(y_2) \\ \vdots \\ L_h(y_N) \\ g_2(y_N) \end{Bmatrix}=0 \tag{11-133}$$

其中$L_h(y_{i+1})$ 是将式（11-108）的右边部分移到左边并除以 h 后得到的表达式。如果应用拟牛顿法求解这个方程，若已得到第 ν 次迭代结果 $Y^{(\nu)}$，则第$\nu+1$ 次迭代结果$Y^{(\nu+1)}$的线性方程组为

$$A^{(\nu)}[Y^{(\nu+1)}-Y^{(\nu)}]=-\Phi(Y^{(\nu)})\quad \nu=0,\ 1,\ 2,\cdots \tag{11-134}$$

其中

$$A^{\nu}\equiv\begin{pmatrix} M_a^{(\nu)} & & & & 0 \\ -L_1^{(\nu)} & R_1^{(\nu)} & & & \\ & -L_2^{(\nu)} & R_2^{(\nu)} & \ddots & \\ 0 & & & -L_N^{(\nu)} & R_N^{(\nu)} \\ & & & & M_b^{(\nu)} \end{pmatrix}$$

$$M_a^{(\nu)}\equiv\frac{\partial g_1}{\partial y}(y_0^{(\nu)}),\quad M_b^{(\nu)}\equiv\frac{\partial g_2}{\partial y}(y_N^{(\nu)}),\ t_j=t_{j-1}+h,$$

$$b-a=Nh$$

$A^{(\nu)}$ 中的 $L_j^{(\nu)}$ 和 $R_j^{(\nu)}$ 对于公式（11-117）、（11-123）和(11-124）取不同的形状。对公式（11-117）为

$$L_j^{(\nu)}=-\frac{1}{h}\left[I+\frac{1}{2}hJ_{j-1}^{(\nu)}+\frac{1}{12}h^2(J_{j-1}^{(\nu)})^2\right]$$

$$R_j^{(\nu)}=-\frac{1}{h}\left[I-\frac{1}{2}hJ_j^{(\nu)}+\frac{1}{12}h^2(J_j^{(\nu)})^2\right] \tag{11-135}$$

对公式（11-123）为

$$L_j^{(\nu)}=\frac{1}{h}\left[I+\frac{1}{2}hJ_{j-1}^{(\nu)}+\frac{1}{10}h^2(J_{j-1}^{(\nu)})^2+\frac{1}{120}h^3(J_{j-1}^{(\nu)})^3\right]$$

$$R_j^{(\nu)}=\frac{1}{h}\left[I-\frac{1}{2}hJ_j^{(\nu)}+\frac{1}{10}h^2(J_j^{(\nu)})^2-\frac{1}{120}h^3(J_j^{(\nu)})^3\right] \tag{11-136}$$

对公式（11-124）为

$$L_j^{(\nu)}=\frac{1}{h}\left[I-\frac{1}{2}hJ_{j-1}^{(\nu)}+\frac{1}{9}h^2(J_{j-1}^{(\nu)})^2+\frac{1}{72}h^3(J_{j-1}^{(\nu)})^3\right.$$
$$\left.+\frac{1}{1008}h^4(J_{j-1}^{(\nu)})^4+\frac{1}{30240}h^5(J_{j-1}^{(\nu)})^5\right]$$

$$R_j^{(\nu)}=\frac{1}{h}\left[I-\frac{1}{2}J_j^{(\nu)}+\frac{1}{9}h^2(J_j^{(\nu)})^2-\frac{1}{72}h^3(J_j^{(\nu)})^3\right. \tag{11-137}$$
$$\left.+\frac{1}{1008}h^4(J_j^{(\nu)})^4-\frac{1}{30240}h^5(J_j^{(\nu)})^5\right]$$

公式中 $J_j^{(\nu)}\equiv\frac{\partial f}{\partial y}(t_j,\ y_j^{(\nu)})$。在推导这些公式时作了简化，在 $L_j^{(\nu)}$ 中凡是雅可比矩阵全用 $J_{j-1}^{(\nu)}$ 代替，在 $R_j^{(\nu)}$ 中，凡是雅可比矩阵全用 $J_j^{(\nu)}$ 代替，这样可保证在每一个格点处仅需计算一个雅可比矩阵。实际求解时，可以改成若干个格点计算一个雅可比矩阵，不同的 ν 采用同一个值，于是若干个 $L_j^{(\nu)}$ 和 $R_j^{(\nu)}$ 是相同的，可节省大量的计算。

下面以公式（11-124）为例构造一个完整的并行算法。对其它的公式可类似构造。

算法11-8

步 0　构造格点 $t_0, t_1, \cdots, t_N$，$t_0 = a$，$t_N = b$，$t_j = t_{j-1} + h$，构造迭代初值 $y_0^{(0)}$，$y_1^{(0)}$，$\cdots$，$y_N^{(0)}$。给出精度控制参数 e。令 $\nu = 0$

步 1　并行计算 $f(t_j, y_j^{(\nu)})$ $j = 0, 1, \cdots, N$。

步 2　按公式（11-127）并行计算 $\bar{y}^{(\nu)}_{j+\frac{1}{2}+\alpha_1}$ 和 $\bar{y}^{(\nu)}_{j+\frac{1}{2}-\alpha_1}$，$j = 0, 1, \cdots, N-1$。

步 3　并行计算 $f(t_{j+\frac{1}{2}+\alpha_1}, \bar{y}^{(\nu)}_{j+\frac{1}{2}+\alpha_1})$ 和 $f(t_{j+\frac{1}{2}-\alpha_1}, \bar{y}^{(\nu)}_{j+\frac{1}{2}-\alpha_1})$ $j = 0, 1, \cdots, N-1$。

步 4　按公式（11-128）并行计算 $\bar{y}^{(\nu)}_{j+\frac{1}{2}+\alpha_2}$ 和 $\bar{y}^{(\nu)}_{j+\frac{1}{2}-\alpha_2}$ $j = 0, 1, \cdots, N-1$。

步 5　并行计算 $f(t_{j+\frac{1}{2}+\alpha_2}, \bar{y}^{(\nu)}_{j+\frac{1}{2}+\alpha_2})$ 和 $f(t_{j+\frac{1}{2}-\alpha_2}, \bar{y}^{(\nu)}_{j+\frac{1}{2}-\alpha_2})$ $j = 0, 1, \cdots, N-1$。

步 6　按公式（11-129）并行计算 $\bar{y}^{(\nu)}_{j+\frac{1}{2}+\alpha_3}$ 和 $\bar{y}^{(\nu)}_{j+\frac{1}{2}-\alpha_3}$ $j = 0, 1, \cdots, N-1$。

步 7　并行计算 $f(t_{j+\frac{1}{2}+\alpha_3}, \bar{y}^{(\nu)}_{j+\frac{1}{2}+\alpha_3})$ 和 $f(t_{j+\frac{1}{2}-\alpha_3}, \bar{y}^{(\nu)}_{j+\frac{1}{2}-\alpha_3})$ $j = 0, 1, \cdots, N-1$。

步 8　按公式（11-130）并行计算 $\bar{y}^{(\nu)}_{j+1/2}$，$j = 0, 1, \cdots, N-1$。

步 9　并行计算 $f(t_{j+1/2}, \bar{y}^{(\nu)}_{j+1/2})$，$j = 0, 1, \cdots, N-1$。

步10　并行计算 $L_n(y_j^{(\nu)})$，$j = 1, 2, \cdots, N$。

步11　计算 $g_1(y_0^{(\nu)})$，$g_2(y_N^{(\nu)})$，$M_a^{(\nu)}$，$M_b^{(\nu)}$。

步12　并行计算 $J_j^{(\nu)}$，$j = 0, 1, \cdots, N$。

步13　并行计算 $L_j^{(\nu)}$，$R_j^{(\nu)}$，$j = 1, \cdots, N$。

步14　求解线性方程组（11-134）得 $y^{(\nu+1)}$。

步15　判别是否有

$$\|y^{(\nu+1)} - y^{(\nu)}\| \leqslant \varepsilon$$

若不等式成立，停止迭代，$y^{(\nu-1)}$ 即为要求的解；若不等式不成立，令 $y^{(\nu)}=y^{(\nu+1)}$，$\nu=\nu+1$，转步 1。

由这算法看出，主要计算集中在步 1～步 14，它们可并行计算。

利用方程（11-134）的矩阵 $A^{(\nu)}$ 的特殊形状，可设计有效的算法。若 $g(v, w)$ 不是分离的形状（11-131），则 $A^{(\nu)}$ 取形状

$$A^{(\nu)}=\begin{pmatrix} M_0^{(\nu)} & 0 & \cdots & 0 & M_b^{(\nu)} \\ -L_1^{(\nu)} & R_1^{(\nu)} & 0 & \cdots & 0 \\ & -L_2^{(\nu)} & R_2^{(\nu)} & & \\ 0 & \ddots & \ddots & & 0 \\ & & & -L_N^{\nu} & R_N^{\nu} \end{pmatrix}$$

算法11-8是一个固定精度阶的方法，虽然在步15给出相邻两次迭代精度的比较，但它未给出$y^{(\nu)}$与问题（11-106)、(11-107)的解$y(t)$之间的误差（全局截断误差)。因此，即使算法11-8已求得满足其精度比较的解，这个解与$y(t)$之间的误差可能是很大的，需要给出其全局截断误差的估计，当得到全局截断误差的估计后，也希望能得到更精确的解，由 11.4 节和 11.5 节知道，应用误差延期校正方法可达到上述的目的，我们对本节导出的 4，6，8 阶公式构造误差延期校正迭代格式，由于其主要的计算包含在算法 11-8（或类似的其它阶的算法）中。而且算法11-8中的计算是可以并行进行的，因此下面构造的算法也将具有较高的并行度。

对于给定的格点集

$$\pi: a=t_0<t_1<t_2<\cdots<t_N=b$$

问题（11-106)、(11-107）的解$y(t)$在这些格点上的值$y(t_i)$组成向量$Y^*=(y(t_0), y(t_1), \cdots, y(t_N))^T$。为后面变格点的需要，令 $h_i=t_{i+1}-t_i$，于是式（11-133）的 $L_4(y_{i+1})$ 中的 h 为

h_i，分别记 $L_h(y_{i+1})$ 应用公式 (11-117)，(11-125)，(11-126) 的 $\Phi(Y)$ 为 $\Phi_2(Y)$，$\Phi_3(Y)$，$\Phi_4(Y)$，而记应用公式

$$L_h(y_{i+1})=\frac{1}{h_i}\left[y_{i+1}-y_i-\frac{h_i}{2}(y'_i+y'_{i+1})\right] \qquad (11\text{-}138)$$

的 $\Phi(Y)$ 为 $\Phi_1(Y)$。于是对每个 i，$\Phi_i(Y)$ 的精度阶是 $2i$。

按照误差延期校正过程的格式，取更新解的方法为 $\Phi_1(\cdot)$，而取误差估计算子 Ψ 分别为

$$\Psi(\cdot)=\Phi_1(\cdot)-\Phi_i(\cdot) \quad i=2,\ 3,\ 4$$

构造下面的算法。

算法11-9

步 1　求解方程

$$\Phi_1(Y)=0$$

得解 $Y^{(1)}$，并令 $i=1$。

步 2　计算 $-\Phi_{i+1}(Y^{(i)})$，求解方程

$$\Phi_1(Y)=\Phi_1(Y^{(i)})-\Phi_{i+1}(Y^{(i)})$$

得解 $Y^{(i+1)}$。

步 3　判断不等式

$$\|Y^{(i+1)}-Y^{(i)}\|\leqslant\delta$$

是否成立；若成立，则停止计算，否则转步 4。

步 4　若 $i+1=4$，则停止计算，否则令 $i=i+1$，转步 2。

算法11-9中的 δ 为精度控制参数

我们还可以取 $\Phi_2(\cdot)$ 作为更新解的方法 Φ，而取误差估计算子 Ψ 为

$$\Psi(\cdot)=\Phi_2(\cdot)-\Phi_i(\cdot) \quad i=3,\ 4$$

将算法11-9作明显的改变后，可以构造相应的算法。

由算法11-9的步 4 看到，当算法11-9运行结束时，得到的近似值 $Y^{(4)}$ 可能不满足要求的精度，这时需要减小格点集 π 的格点距 h_i。在11.4节中采取将格点距 h_i 都缩小一半的方式，但是对许多问题采用这种策略将是不合理的，因为解 $y(t)$ 在区间 $[a,b]$ 中的变化不是均匀的。下面推导一种改变格点距 h_i 的策略。

由定理11-2

$$(Y^{(3)})_i-(Y^*)_i=(Y^{(3)})_i-(Y^{(4)})_i+O(h_i^6)$$
$$=A_ih_i^6+O(h_i^6) \qquad (11\text{-}139)$$

因此有近似等式（误差为$O(h_i^6)$）

$$\|(Y^{(3)})_i-(Y^*)_i\|=\|(Y^{(3)})_i-(Y^{(4)})_i\|=\|A_ih_i^6\| \qquad (11\text{-}140)$$

并且假定其中A_i与h_i无关，如果要求新的格点距h_i^*恰好满足精度要求，应有

$$\|A_ih_i^{*6}\|=\delta$$

由式（11-139）、（11-140）推得

$$h_i^*=h_i(\delta/\|(Y^{(3)})_i-(Y^{(4)})_i\|)^{1/6} \qquad (11\text{-}141)$$

根据式（11-141）给出的h_i^*对区间$[t_i,\ t_{i+1}]$加进新的格点。若采取等分区间，则分点数$NT(j)$不应小于数$E(j)=(\|(Y^{(3)})_i-(Y^{(4)})_i\|/\delta)^{1/6}$的整数部分，例如取

$$NT(j)=\lfloor 1.2(\|(Y^{(3)})_i-(Y^{(4)})_i\|/\delta)^{1/6}\rfloor \qquad (11\text{-}142)$$

其中记号$\lfloor\ \rfloor$表示取整数部分。若式（11-142)给出的$NT(j)=0$，表示格点距h_i不需要减小。

当格点距变化后，重新调用算法11-9，进行计算，直到满足精度为止。

第十二章　偏微分方程的并行计算

12.1　椭圆型方程差分格式的并行计算

本节主要以拉普拉斯（Laplace）方程为模型，讨论各种迭代方法的并行计算。

考虑矩形区域G：$\{0 \leqslant x \leqslant L,\ 0 \leqslant y \leqslant M\}$内，拉普拉斯方程的第一边值问题：

$$\Delta u \equiv \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0 \quad (x,\ y) \in G \tag{12-1}$$

$$u|_{\Gamma} = \varphi(x,\ y) \quad (x,\ y) \in \Gamma \tag{12-2}$$

其中Γ是G的边界。

假设离散网格是正方形的。

将线段$[0,\ L]$$l$等分，即步长$h = L/l$；将线段$[0,\ M]$$m$等分，$m = [M/h]$。

使用五点差分格式离散化，可得线性代数方程组

$$\begin{cases} u_{i,j} = \dfrac{1}{4}(u_{i+1,j} + u_{i-1,j} + u_{i,j+1} + u_{i,j-1}) \\ \qquad i = 1,\ 2,\ \cdots,\ l-1 \quad j = 1,\ 2,\ \cdots,\ m-1 \\ u_{i,j} = \varphi_{i,j} = \varphi(ih,\ jh) \\ i = 0,\ l \quad j = 0,\ 1,\ \cdots,\ m \\ j = 0,\ m \quad i = 0,\ 1,\ \cdots,\ l \end{cases} \tag{12-3}$$

其中$u_{i,j}$是网格结点$(i,\ j)$处$u(ih,\ jh)$的近似值。

该方程组的矩阵-向量表示为

$$A \cdot \boldsymbol{u} = \boldsymbol{b} \tag{12-4}$$

其中A是$(l-1)\times(m-1)$阶块三对角方阵，$\boldsymbol{u}$是按照自左至右、自下而上排列的$(l-1)\times(m-1)$维内部结点向量，$\boldsymbol{b}$是$(l-1)\times(m-1)$维向量。

$$A=\begin{pmatrix} D_1 & E_2 & & & 0 \\ E_2 & D_2 & E_3 & & \\ & \ddots & \ddots & \ddots & E_{l-1} \\ 0 & & & E_{l-1} & D_{l-1} \end{pmatrix}$$

$$u=(u_1,\ u_2,\ \cdots,\ u_{l-1})^T$$

$$b=(b_1,\ b_2,\ \cdots,\ b_{l-1})^T$$

$$D_i=\begin{pmatrix} 1 & -\frac{1}{4} & & & 0 \\ -\frac{1}{4} & 1 & -\frac{1}{4} & & \\ & \ddots & \ddots & \ddots & -\frac{1}{4} \\ 0 & & & -\frac{1}{4} & 1 \end{pmatrix}$$

和

$$E_i=\begin{pmatrix} -\frac{1}{4} & & & & \\ & -\frac{1}{4} & & 0 & \\ & & \ddots & & \\ 0 & & & & -\frac{1}{4} \end{pmatrix}$$

分别是 $m-1$ 阶的三对角阵和对角阵

$$u_i=(u_{i,1},\ u_{i,2},\ \cdots,\ u_{i,m-1})^T$$

是第 i 列上自下而上排列的内部结点向量

$$b_i=\frac{1}{4}(\varphi_{i,0},\ 0,\ \cdots,\ 0,\ \varphi_{i,m})^T$$

$$i=2,\ 3,\ \cdots,\ l-2$$

$$\boldsymbol{b}_1=\frac{1}{4}[\varphi_{0,1}+\varphi_{1,0},\ \varphi_{0,2},\ \cdots,\ \varphi_{0,m-2},\ \varphi_{0,m-1}+\varphi_{1,m}]^T$$

$$\boldsymbol{b}_{l-1}=\frac{1}{4}[\varphi_{l,1}+\varphi_{l-1,0},\ \varphi_{l,2},\cdots,\ \varphi_{l,m-2},\ \varphi_{l,m-1}+\varphi_{l-1,m}]^T$$

一般稀疏矩阵A的阶数小于300时，使用直接方法求解线代数方程组（12-4）是比较有效的；当A的阶数较大时，一般使用迭代方法求解的效果更佳。

下面，介绍几种典型迭代方法的并行计算。

12.1.1 简单迭代及其超松弛方法的并行计算

简单雅可比（Jacobi）迭代方法的迭代计算公式为

$$u_{i,j}^{(k+1)}=\frac{1}{4}(u_{i+1,j}^{(k)}+u_{i-1,j}^{(k)}+u_{i,j+1}^{(k)}+u_{i,j-1}^{(k)}) \tag{12-5}$$

$$i=1,\ 2,\ \cdots,\ l-1;\ j=1,\ 2,\ \cdots,\ m-1$$

其相应的矩阵-向量表示为

$$D\cdot\boldsymbol{u}^{(k+1)}=(L+R)\cdot\boldsymbol{u}^{(k)}+\boldsymbol{b} \tag{12-6}$$

其中D是矩阵A的对角线元素组成的对角阵，$-L$、$-R$分别是A的严格下，上三角阵。

五点格式下，$D=I$是单位阵，$R=L^T$，所以式（12-6）可以改写成

$$\boldsymbol{u}^{(k+1)}=(L+L^T)\boldsymbol{u}^{(k)}+\boldsymbol{b} \tag{12-7}$$

简单迭代的超松弛方法（JOR方法）的迭代计算公式为

$$u_{i,j}^{(k+1)}=(1-\omega)u_{i,j}^{(k)}+\frac{\omega}{4}(u_{i+1,j}^{(k)}+u_{i-1,j}^{(k)}+u_{i,j+1}^{(k)}+u_{i,j-1}^{(k)}) \tag{12-8}$$

$$i=1,\ 2,\ \cdots,\ l-1;\ j=1,\ 2,\ \cdots,\ m-1$$

其相应的矩阵-向量表示为

$$\boldsymbol{u}^{(k+1)}=((1-\omega)I+\omega(L+L^T))\cdot\boldsymbol{u}^{(k)}+\omega\cdot\boldsymbol{b} \tag{12-9}$$

其中ω为松弛因子。

JOR方法的并行计算格式如下

$$u_i^{(k+1)}=(1-\omega)u_i^{(k)}+\frac{\omega}{4}(u_{i+1}^{(k)}+u_{i-1}^{(k)}+u_{i+1}^{(k)}+u_{i-1}^{(k)}),$$

(12-10)

$i=1,\ 2,\ \cdots,\ l-1$

其中 $u_i^{(e)}$（$e=k,\ k+1;\ t=i-1,\ i,\ i+1$）是第 t 列上的 $m-1$ 维内点向量，$u_{i-1}^{(k)}$ 和 $u_{i+1}^{(k)}$ 分别是第 i 列网格上下 $m-1$ 个结点上的向量和上 $m-1$ 个结点上的向量。

并行格式（12-10）每次计算的向量长度是 $m-1$。简单迭代及其超松弛方法都是适于并行计算的。

12.1.2 逐次超松弛迭代方法的并行计算

逐次超松弛迭代方法（SOR方法）的迭代计算公式为

$$u_{i,j}^{(k+1)}=(1-\omega)u_{i,j}^{(k)}+\frac{\omega}{4}(u_{i-1,j}^{(k+1)}+u_{i,j-1}^{(k+1)}+u_{i+1,j}^{(k)}+u_{i,j+1}^{(k)})$$

(12-11)

$i=1,\ 2,\ \cdots,\ l-1;\ j=1,\ 2,\ \cdots,\ m-1$

其相应的矩阵-向量表示为

$$u^{(k+1)}=((1-\omega)I+\omega(I-L)^{-1}L^T)u^{(k)}+\omega(I-L)^{-1}b \qquad (12\text{-}12)$$

SOR方法的迭代计算是按自下而上、自左至右的顺序进行的，这种有序计算不能直接改造成适于并行计算的格式。

“红-黑”排序超松弛方法不仅具有SOR方法相同的收敛速度，而且具有适于并行计算的特点。

“红-黑”排序法的基本思想是使用“红”与“黑”二种颜色标识网格结点，在五点格式下，任何一个标识“红”点，其相邻的四点都标识为“黑”点；反之，任何一个“黑”点相邻的四点都标识为“红”点。每次迭代先归类计算所有的“红”点，然后，归类计算所有的“黑”点。反之亦然。

假设“0”表示“红”点“×”表示“黑”点，“红-黑”排序的图示如下：

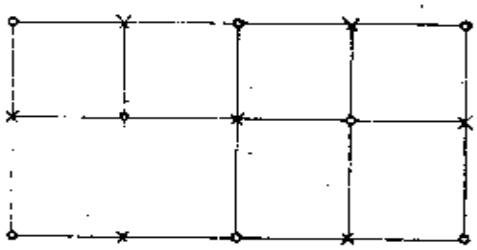


图12-1 “红-黑”排序图

下面，给出“红-黑”排序超松弛方法的并行计算格式。

“红”点的并行计算格式

$$u_{2i-1}^{(k+1)1}=(1-\omega)u_{2i-1}^{(k)1}+\frac{\omega}{4}(u_{2i}^{(k)1}+u_{2i-2}^{(k)1}+u_{2i-1+}^{(k)1}+u_{2i-1-}^{(k)1}) \tag{12-13}$$

$$i=1,\ 2,\ \cdots,\ l_1$$

$$u_{2i}^{(k+1)2}=(1-\omega)\cdot u_{2i}^{(k)2}+\frac{\omega}{4}(u_{2i+1}^{(k)2}+u_{2i-1}^{(k)2}+u_{2i+}^{(k)2}+u_{2i-}^{(k)2}) \tag{12-14}$$

$$i=1,\ 2,\ \cdots,\ l_2$$

其中 $l_1=[l/2]$，$l_2=[(2l-1)/4]$

$$u_{2i-1}^{(k+1)1}=[u_{2i-1,1}^{(k+1)},\ u_{2i-1,3}^{(k+1)},\ \cdots,\ u_{2i-1,2m_1-1}^{(k+1)}]^T$$

$$u_{2i-1}^{(k)1}=[u_{2i-1,1}^{(k)},\ u_{2i-1,3}^{(k)},\ \cdots,\ u_{2i-1,2m_1-1}^{(k)}]^T$$

是m_1维“红”点向量，$m_1=[m/2]$

$$u_{2i}^{(k+1)2}=[u_{2i,2}^{(k+1)},\ u_{2i,4}^{(k+1)},\ \cdots,\ u_{2i,2m_2}^{(k+1)}]^T$$

$$u_{2i}^{(k)2}=[u_{2i,2}^{(k)},\ u_{2i,2}^{(k)},\ \cdots,\ u_{2i,2m_2}^{(k)}]^T$$

是m_2维“红”点向量，$m_2=[(2m-1)/4]$

$$u_{2i}^{(k)1}=[u_{2i,1}^{(k)},\ u_{2i,3}^{(k)},\ \cdots,\ u_{2i,2m_1-1}^{(k)}]^T$$

$$u_{2i-2}^{(k)1}=[u_{2i-2,1}^{(k)},\ u_{2i-2,3}^{(k)},\ \cdots,\ u_{2i-2,2m_1-1}^{(k)}]^T$$

$$u_{2i-1+}^{(k)1}=[u_{2i-1,2}^{(k)},\ u_{2i-1,4}^{(k)},\ \cdots,\ u_{2i-1,2m_1}^{(k)}]^T$$

$$u_{2i-1-}^{(k)1}=[u_{2i-1,0}^{(k)},\ u_{2i-1,2}^{(k)},\ \cdots,\ u_{2i-1,2m_1-2}^{(k)}]^T$$

是m_1维“黑”点向量

$$u_{2l+1}^{(k)2}=[u_{2l+1,2}^{(k)},\ u_{2l+1,4}^{(k)},\ \cdots,\ u_{2l+1,2m_2}^{(k)}]^T$$

$$u_{2l-1}^{(k)2}=[u_{2l-1,2}^{(k)},\ u_{2l-1,4}^{(k)},\ \cdots,\ u_{2l-1,2m_2}^{(k)}]^T$$

$$u_{2l+}^{(k)2}=[u_{2l,3}^{(k)},\ u_{2l,5}^{(k)},\ \cdots,\ u_{2l,2m_2+1}^{(k)}]^T$$

$$u_{2l-}^{(k)2}=[u_{2l,1}^{(k)},\ u_{2l,3}^{(k)},\ \cdots,\ u_{2l,2m_2-1}^{(k)}]^T$$

是m_2维“黑”点向量。

“黑”点的并行计算格式类似于“红”点计算。

“红-黑”方法可以看成是“二色”方法，进一步介绍“多色”排序方法。

“多色”排序法就是使用多种颜色标识网格结点，然后，将同一种颜色标识的网格结点进行归类计算。

“P色”排序法的矩阵-向量表示为

$$\begin{pmatrix} \overline{D}_1 & \overline{B}_{1,2} & \cdots & \overline{B}_{1,p} \\ \overline{B}_{2,1} & \overline{D}_2 & \ddots & \overline{B}_{2,p} \\ & \ddots & \ddots & \vdots \\ & & & \overline{B}_{p-1,p} \\ & & & \\ \overline{B}_{p,1} & \overline{B}_{p,2} & \cdots & \overline{B}_{p,p-1} \quad \overline{D}_p \end{pmatrix} \begin{pmatrix} \overline{u}_1 \\ \overline{u}_2 \\ \vdots \\ \overline{u}_p \end{pmatrix} = \begin{pmatrix} \overline{b}_1 \\ \overline{b}_2 \\ \vdots \\ \overline{b}_p \end{pmatrix} \tag{12-15}$$

其中$\overline{D}_i$是对角阵，$\overline{u}_i$是所有标识第i种颜色的网格结点上的向量。

“P色”排序法的迭代计算公式为

$$\overline{u}_i^{(k+1)}=\overline{D}_i^{-1}\left(\overline{b}_i-\sum_{j<i}\overline{B}_{i,j}\cdot\overline{u}_j^{(k+1)}-\sum_{j>i}\overline{B}_{i,j}\cdot\overline{u}_j^{(k)}\right)$$
$$i=1,\ 2,\ \cdots,\ P \tag{12-16}$$

“P色”排序超松弛方法的迭代计算公式为

$$\overline{u}_i^{(k+1)}=(1-\omega)\overline{u}_i^{(k)}+\omega\overline{D}_i^{-1}\left(\overline{b}_i-\sum_{j<i}\overline{B}_{i,j}\overline{u}_j^{(k+1)}-\sum_{j>i}\overline{B}_{i,j}\cdot\overline{u}_j^{(k)}\right) \tag{12-17}$$

$i=1,\ 2,\ \cdots,\ P$

显然“P色”法是适合于并行计算的。当$P=2$时，即为“红-黑”法。

“红-黑”排序法求解拉普拉斯方程和泊松（Poisson）方程的五点差分格式的计算效果是很好的，也适于并行计算。但是对于含混合偏导项的一般椭圆型方程和更高阶的差分格式，使用“红-黑”排序法的效果不佳。这时，只有酌情使用“多色”排序法来计算。

〔例12-1〕，求解含混合偏导项的椭圆型方程

$$\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial x\partial y}+\frac{\partial^2 u}{\partial y^2}=0$$

使用9点差分格式和“四色”排序法计算，比五点差分格式下的“红-黑”排序法要好。

12.1.3 线迭代及其超松弛方法的并行计算

简单线迭代方法的迭代计算公式为

$$u_{i,j}^{(k+1)}=\frac{1}{4}(u_{i,j+1}^{(k+1)}+u_{i,j-1}^{(k+1)}+u_{i-1,j}^{(k)}+u_{i+1,j}^{(k)}) \tag{12-18}$$

$i=1,\ 2,\ \cdots,\ l-1;\ j=1,\ 2,\ \cdots,\ m-1$

其相应的矩阵-向量表示为

$$D\cdot u_i^{(k+1)}=\frac{1}{4}(u_{i-1}^{(k)}+u_{i+1}^{(k)})+b_i \tag{12-19}$$

$i=1,\ 2,\ \cdots,\ l-1$

其中$u_t^{(e)}$（$e=k,\ k+1;\ t=i-1,\ i,\ i+1$）是第$t$列上内部结点上的向量；

$$D=\begin{pmatrix} 1 & -\frac{1}{4} & & & 0 \\ -\frac{1}{4} & 1 & -\frac{1}{4} & & \\ & \ddots & \ddots & \ddots & \\ & & & & -\frac{1}{4} \\ 0 & & & -\frac{1}{4} & 1 \end{pmatrix}$$

是 $m-1$ 维方阵；

$$\boldsymbol{b}_i=\frac{1}{4}(\varphi_{i,0},\ 0,\ \cdots,\ 0,\ \varphi_{i,m})^T$$

是 $m-1$ 维向量。

简单线迭代超松弛（JLOR）方法的迭代计算公式为

$$u_{i,j}^{(k+1)}=(1-\omega)\cdot u_{i,j}^{(k)}+\frac{\omega}{4}(u_{i,j-1}^{(k+1)}+u_{i,j-1}^{(k+1)}+u_{i-1,j}^{(k)}+u_{i+1,j}^{(k)}) \tag{12-20}$$

$i=1,\ 2,\ \cdots,\ l-1;\ j=1,\ 2,\ \cdots,\ m-1$

其相应的矩阵-向量表示为

$$D_\omega\cdot\boldsymbol{u}_i^{(k+1)}=(1-\omega)\boldsymbol{u}_i^{(k)}+\frac{\omega}{4}(\boldsymbol{u}_{i-1}^{(k)}+\boldsymbol{u}_{i+1}^{(k)})+\omega\boldsymbol{b}_i \tag{12-21}$$

$i=1,\ 2,\ \cdots,\ l-1$

其中

$$D_\omega=\begin{pmatrix} 1 & -\frac{\omega}{4} & & & 0 \\ -\frac{\omega}{4} & 1 & -\frac{\omega}{4} & & \\ & \ddots & \ddots & \ddots & \\ & & & & -\frac{\omega}{4} \\ 0 & & & -\frac{\omega}{4} & 1 \end{pmatrix}$$

迭代公式（12-19）和（12-21）给出的 $l-1$ 个 $m-1$ 阶的方程组彼此是相互独立的，所以，只要选择相同的算法求解每个方程组，在 i 方向可以组织并行计算，并行计算的向量长度为 $l-1$。

逐次超松弛线迭代（SLOR）方法的迭代计算公式为

$$u_{i,j}^{(k+1)}=(1-\omega)u_{i,j}^{(k)}+\frac{\omega}{4}(u_{i,j-1}^{(k+1)}+u_{i,j+1}^{(k+1)}+u_{i-1,j}^{(k+1)}+u_{i+1,j}^{(k)}) \tag{12-22}$$

$i=1,\ 2,\ \cdots,\ l-1;\ j=1,\ 2,\ \cdots,\ m-1$

其相应的矩阵-向量表示为

$$D_\omega\cdot u_i^{(k+1)}=\frac{\omega}{4}u_{i-1}^{(k+1)}+\left[(1-\omega)u_i^{(k)}+\frac{\omega}{4}u_{i+1}^{(k)}+\omega b_i\right]$$

$$i=1,\ 2,\ \cdots,\ l-1 \tag{12-23}$$

其中

$$u_0^{(k+1)}=[\varphi_{0,1},\ \varphi_{0,2},\ \cdots,\ \varphi_{0,m-1}]^T$$

式（12-23）是一个向量递归计算。对于每个 i（$i=1,\ 2,\ \cdots,\ l-1$），即计算一个三对角方程组。关于计算三对角方程组的并行算法有“倍增法”、“并行追赶法”、“循环奇偶约化法”等。

12.1.4 交替方向隐式迭代法的并行计算

交替方向隐式迭代（ADI）方法的计算公式为

$$\begin{cases} u_{i,j}^{(2k+1)}=u_{i,j}^{(2k)}+\alpha_k(u_{i-1,j}^{(2k+1)}+u_{i+1,j}^{(2k+1)}-2\cdot u_{i,j}^{(2k+1)}) \\ \qquad\qquad +\alpha_k(u_{i,j-1}^{(2k)}+u_{i,j-1}^{(2k)}-2\cdot u_{i,j}^{(2k)}) \\ u_{i,j}^{(2k+2)}=u_{i,j}^{(2k+1)}+\alpha_k(u_{i,j+1}^{(2k+2)}+u_{i,j-1}^{(2k+2)}-2\cdot u_{i,j}^{(2k+2)}) \\ \qquad\qquad +\alpha_k(u_{i-1,j}^{(2k+1)}+u_{i-1,j}^{(2k+1)}-2\cdot u_{i,j}^{(2k+1)}) \end{cases} \tag{12-24}$$

$i=1,\ 2,\ \cdots,\ l-1;\ j=1,\ 2,\ \cdots,\ m-1$

其中 α_k 是迭代加速参数。

该迭代方法相当于沿行方向（i 方向）进行一次线迭代，再沿列方向（j 方向）进行一次线迭代，二步合并成一次完整的迭代。

式（12-24）相应的矩阵-向量表示为

$$H_{a_k}\cdot u_j^{(2k+1)}=a_k u_{j+1}^{(2k)}+(1-2a_k)u_j^{(2k)}+a_k u_{j-1}^{(2k)}+b_j,$$
$$j=1,\ 2\cdots,\ m-1 \tag{12-25}$$

$$G_{a_k}\cdot u_j^{(2k+2)}=a_k\cdot u_{j+1}^{(2k+1)}+(1-2a_k)\cdot u_j^{(2k+1)}+a_k\cdot u_{j-1}^{(2k+1)}+b_i \tag{12-26}$$
$$i=1,\ 2,\ \cdots,\ l-1$$

其中 u_{i-1}、u_i、u_{i+1} 分别是第 $j-1$、j、$j+1$ 行上内部结点上的向量；u_{i-1}、u_i、u_{i+1} 分别是第 $i-1$、i、$i+1$ 列上内部结点上的向量；

$$H_{a_k}=\begin{pmatrix} 1+2a_k & -a_k & 0 & & \\ -a_k & 1+2a_k & -a_k & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & -a_k \\ 0 & & & -a_k & 1+2a_k \end{pmatrix}$$

是 $l-1$ 阶三对角阵；

$$G_{a_k}=\begin{pmatrix} 1+2a_k & -a_k & 0 & & \\ -a_k & 1+2a_k & -a_k & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & -a_k \\ 0 & & & -a_k & 1+2a_k \end{pmatrix}$$

是 $m-1$ 阶三对角阵；

$$b_i=a_k\cdot[\varphi_{0,j},\ 0,\ \cdots,\ 0,\ \varphi_{l,j}]^T$$

是 $l-1$ 维向量；

$$b_i=a_k\cdot[\varphi_{i,0},\ 0,\ \cdots,\ 0,\ \varphi_{i,m}]^T$$

是 $m-1$ 维向量。

式（12-25）的 $m-1$ 个 $l-1$ 阶方程组彼此是独立的，选择相同的算法求解每个方程组，沿 j 方向展开是适于并行计算的，并行计算的向量长度为 $m-1$。

式（12-26）的 $l-1$ 个 $m-1$ 阶方程组彼此也是独立的，选择相同的算法求解每个方程组，沿 i 方向展开是适于并行计算的，并行计算的向量长度为 $l-1$。

当然，如果对于式（12-25）和（12-26）中的每个方程选择不同的算法求解，也是可以进行异步并行计算的。

12.1.5 预条件迭代方法的并行计算

将式（12-4）给出的矩阵 A 分解

$$A=D-L-R$$

其中 $D=I$ 是 A 的对角线元素组成的单位对角阵(单位阵)，$-L$、$-R$ 分别是 A 的严格下、上三角阵。

预条件迭代方法的一般迭代公式为

$$C\cdot\boldsymbol{u}^{(k+1)}=C\cdot\boldsymbol{u}^{(k)}+\alpha(\boldsymbol{b}-A\boldsymbol{u}^{(k)}) \tag{12-27}$$

其中 α 为加速收敛因子。

$$C=(I-L)(I-R)$$

令

$$\boldsymbol{d}^{(k)}=C\boldsymbol{u}^{(k)}+\alpha(\boldsymbol{b}-A\boldsymbol{u}^{(k)})$$

并将迭代公式（12-27）分成二个半迭代步来执行，可得预条件迭代的另一种形式

$$\begin{cases}(I-L)\boldsymbol{u}^{(k+1/2)}=\boldsymbol{d}^{(k)}\\(I-R)\boldsymbol{u}^{(k+1)}=\boldsymbol{u}^{(k+1/2)}\end{cases} \tag{12-28}$$

进一步给出公式（12-28）的等价形式

$$\begin{cases}D\boldsymbol{u}_1^{(k+1/2)}=\boldsymbol{d}_1^{(k)}\\D\boldsymbol{u}_i^{(k+1/2)}=\dfrac{1}{4}\cdot\boldsymbol{u}_{i-1}^{(k+1/2)}+\boldsymbol{d}_i^{(k)}\quad i=2,\ 3,\cdots,\ l-1\end{cases} \tag{12-29a}$$

和

$$\begin{cases}D\boldsymbol{u}_{l-1}^{(k+1)}=\boldsymbol{u}_{l-1}^{(k+1/2)}\\D\boldsymbol{u}_i^{(k+1)}=\dfrac{1}{4}\boldsymbol{u}_{i+1}^{(k+1)}+\boldsymbol{u}_i^{(k+1/2)}\quad i=l-2,\ l-3,\cdots,1\end{cases} \tag{12-29b}$$

其中 u_i 是第 i 列上的内部结点上的向量；

$$D=\begin{pmatrix} 1 & -\frac{1}{4} & & & 0 \\ -\frac{1}{4} & 1 & -\frac{1}{4} & & \\ & \ddots & \ddots & \ddots & \\ & & & & -\frac{1}{4} \\ 0 & & & -\frac{1}{4} & 1 \end{pmatrix}$$

是 $m-1$ 阶的三对角阵。

式（12-29）的计算归结为三对角方程组的计算。可以使用“循环奇偶约化法”等并行算法来计算。

12.2 椭圆型方程有限元方法的并行计算

本节以泊松方程的第一边值问题为模型，介绍有限元分块并行算法。

假设并行机的处理机台数（或等效并行台数）为 s。并假设并行机具有间接访问数组的功能。

考虑有界区域 G 上，泊松方程的第一边值问题

$$\Delta u\equiv-\left(\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}\right)=f(x,\ y)\quad (x,\ y)\in G \tag{12-30}$$

$$u\mid\Gamma=\varphi(x,\ y)\quad (x,\ y)\in\Gamma \tag{12-31}$$

其中 Γ 是有界区域 G 的边界。

泊松方程第一边值问题对应的泛函为

$$F[u]=\iint_G\left[\left(-\frac{\partial u}{\partial x}\right)^2+\left(-\frac{\partial u}{\partial y}\right)^2-2f\cdot u\right]dxdy \tag{12-32}$$

其中 $u\in M=\{u\mid u\in H'_\varphi(G)\}$，$H'_\varphi(G)=\{u\mid u\in H'(G),\ u\mid\Gamma=\varphi\}$，$H'(G)$ 是一阶索伯列夫空间。

取 $u_0\in H'_\varphi(G)$，求 $u\in H'_\varphi(G)$，使得

$$F[u]=\inf_{v-u_0\in H_0^1(G)}\{F[v]\} \tag{12-33}$$

问题（12-33）的解，即为泊松方程第一边值问题的弱解。

当 $u\in J=\{u\mid u\in C_{G+\Gamma}^1,\ u\mid\Gamma=\varphi\}$ 时，问题（12-33）的解，就是泊松方程第一边值问题的解。

变分方法将泊松方程第一边值问题的求解，变成了求解方程（12-32）的极小泛函问题。

有限元方法求解方程（12-32）的极小近似解的计算，大致可分成二步：

第一步，以装配形成总刚度阵 K 和总向量 f 为目标。

第二步，求解线代数方程组

$$K\cdot u=f \tag{12-34}$$

其中 u 是使得泛函 $F[u]$ 达到极小的近似解向量。

本节着重介绍总刚度阵 K 和向量 f 的并行计算和并行装配。

12.2.1 有限元分块并行算法的基本思想

在单元刚度阵和向量对总刚度阵 K 和总向量 f 的装配迭加中，由于相邻单元之间存在着相关性，所以，直接的并行装配迭加是不行的。为了提高 K 和 f 装配迭加的并行性，同时又避免装配迭加中的相关性。这里给出了分块并行计算和装配迭加的方法。

分块并行算法的思想就是从所有单元组成的单元集合中，构造若干单元子集合，每个单元子集合含 s 个单元（s 是处理机台数），要求这 s 个单元是彼此无关的，即彼此是不相邻的。然后，对每个单元子集合并行计算 s 个不相关单元的刚度阵和向量，并将这 s 个不相关单元的刚度阵和向量并行地装配迭加到总刚度阵 K 和总向量 f 中去。如此继续下去，直到所有单元子集合都完成了对 K 和 f 的并行计算和并行装配迭加。

12.2.2 构造不相关的单元子集合

将区域G分成s块，$H^{(l)}$表示第l个单元块，每个单元块进一步剖分成m个单元。这样区域G共剖分成$m\cdot s$个单元。

构造不相关的单元子集合如下：

$$P_i=\{P_i^{(1)},\ P_i^{(2)},\ \cdots,\ P_i^{(j)}\}\quad i=1,\ 2,\ \cdots,\ m$$

其中$P_i^{(l)}$是第l个单元块中的第i个单元，且P_i中的s个单元是互不相邻的。

〔例12-2〕 假设处理机台数$s=4$，区域G是正方形的，一种不相关的分块划分和单元子集合的图示如下：

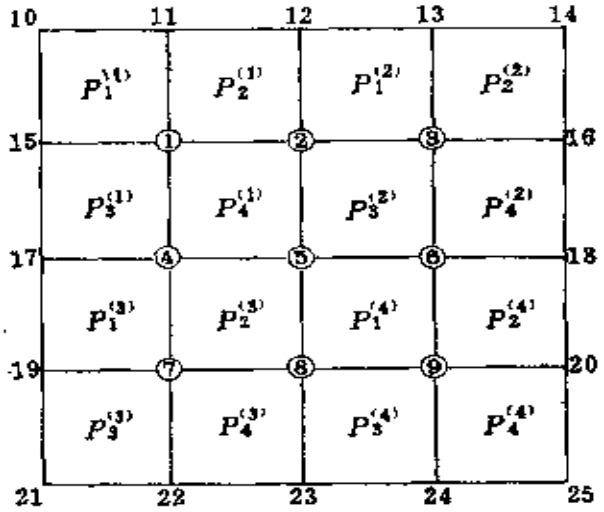


图12-2 不相关单元分块划分图

下面，我们给出整体的结点编号和不相关单元子集合原则下结点编号的关系。

假设每个单元都含有q个结点。

单元$P_i^{(l)}$的结点顺序为

$$P_i^{(l)}=\{d_{i,1}^{(l)},\ d_{i,2}^{(l)},\ \cdots,\ d_{i,q}^{(l)}\}$$

$$i=1,2,\cdots,m;\quad l=1,2,\cdots,s$$

其中$d_{i,t}^{(l)}$表示第l个单元块中第i个单元的第t个结点，其在整体结点编号中的位置记为$e_{i,t}^{(l)}$。

构造单元子集合 P_i 的结点集合如下：

$$D_{i,t}=\{d_{i,t}^{(1)},\ d_{i,t}^{(2)},\ \cdots,\ d_{i,t}^{(n)}\}$$

$$i=1,2,\cdots,m;\quad t=1,2,\cdots,q$$

$D_{i,t}$ 相应的 x 方向坐标向量为

$$x_{i,t}=[x_{i,t}^{(1)},\ x_{i,t}^{(2)},\ \cdots,\ x_{i,t}^{(n)}]^T$$

$D_{i,t}$ 的 y 方向坐标向量为

$$y_{i,t}=[y_{i,t}^{(1)},\ y_{i,t}^{(2)},\ \cdots,\ y_{i,t}^{(n)}]^T$$

12.2.3 并行构造单元的线性插值函数

以三角形单元为例，介绍在每个单元上，并行构造变分问题解 $u=u(x,\ y)$ 的线性插值函数。

单元 $P_i^{(l)}$ 上的线性插值函数为

$$u_i^{(l)}(x,\ y)=u_{i,1}^{(l)}\cdot N_{i,1}^{(l)}(x,y)+u_{i,2}^{(l)}\cdot N_{i,2}^{(l)}(x,y)+u_{i,3}^{(l)}\cdot N_{i,3}^{(l)}(x,y)\qquad(12\text{-}35)$$

$$(x,\ y)\in P_i^{(l)}$$

其中 $u_{i,t}^{(l)}=u(d_{i,t}^{(l)})(t=1,\ 2,\ 3)$ 是未知函数 $u(x,\ y)$ 在结点 $d_{i,t}^{(l)}$ 处的值，

$$N_{i,t}^{(l)}(x,y)=\frac{1}{2\Delta_i^{(l)}}(a_{i,t}^{(l)}+b_{i,t}^{(l)}\cdot x+C_{i,t}^{(l)}\cdot y)$$

$$t=1,\ 2,\ 3;\qquad(12\text{-}36)$$

$$\begin{cases}a_{i,1}^{(l)}=x_{i,2}^{(l)}\cdot y_{i,3}^{(l)}-x_{i,3}^{(l)}\cdot y_{i,2}^{(l)}\\ a_{i,2}^{(l)}=x_{i,3}^{(l)}\cdot y_{i,1}^{(l)}-x_{i,1}^{(l)}\cdot y_{i,3}^{(l)}\\ a_{i,3}^{(l)}=x_{i,1}^{(l)}\cdot y_{i,2}^{(l)}-x_{i,2}^{(l)}\cdot y_{i,1}^{(l)}\\ b_{i,1}^{(l)}=y_{i,2}^{(l)}-y_{i,3}^{(l)}\\ b_{i,2}^{(l)}=y_{i,3}^{(l)}-y_{i,1}^{(l)}\\ b_{i,3}^{(l)}=y_{i,1}^{(l)}-y_{i,2}^{(l)}\\ c_{i,1}^{(l)}=x_{i,3}^{(l)}-x_{i,2}^{(l)}\\ c_{i,2}^{(l)}=x_{i,1}^{(l)}-x_{i,3}^{(l)}\\ c_{i,3}^{(l)}=x_{i,2}^{(l)}-x_{i,1}^{(l)}\end{cases}\qquad(12\text{-}37)$$

和

$$\Delta_i^{(l)}=\frac{1}{2}(x_{i,2}^{(l)}\cdot y_{i,3}^{(l)}+x_{i,3}^{(l)}\cdot y_{i,1}^{(l)}+x_{i,1}^{(l)}\cdot y_{i,2}^{(l)}$$
$$-x_{i,2}^{(l)}y_{i,1}^{(l)}-x_{i,3}^{(l)}y_{i,2}^{(l)}-x_{i,1}^{(l)}y_{i,3}^{(l)}) \qquad (12\text{-}38)$$

$\Delta_i^{(l)}$ 是单元 $P_i^{(l)}$ 的面积。

式（12-35）至（12-38）沿 l 方向展开是适于并行计算的，其并行计算格式如下：

$$u_i(x,y)=u_{i,1}*N_{i,1}(x,y)+u_{i,2}*N_{i,2}(x,y)+u_{i,3}$$
$$*N_{i,3}(x,y) \qquad (12\text{-}39)$$
$$i=1,\ 2,\ \cdots,\ m$$

其中

$$N_{i,t}(x,y)=\left(\frac{1}{2\Delta_i}\right)*(a_{i,t}+b_{i,t}\cdot x+c_{i,t}\cdot y)$$
$$t=1,\ 2,\ 3$$
$$a_{i,1}=x_{i,2}*y_{i,3}-x_{i,3}*y_{i,2}$$
$$a_{i,2}=x_{i,3}*y_{i,1}-x_{i,1}*y_{i,3}$$
$$a_{i,3}=x_{i,1}*y_{i,2}-x_{i,2}*y_{i,1}$$
$$b_{i,1}=y_{i,2}-y_{i,3}$$
$$b_{i,2}=y_{i,3}-y_{i,1}$$
$$b_{i,3}=y_{i,1}-y_{i,2}$$
$$c_{i,1}=x_{i,3}-x_{i,2}$$
$$c_{i,2}=x_{i,1}-x_{i,3}$$
$$c_{i,3}=x_{i,2}-x_{i,1}$$
$$\Delta_i=\frac{1}{2}*(x_{i,2}*y_{i,3}+x_{i,3}*y_{i,1}+x_{i,1}*y_{i,2}$$
$$-x_{i,2}*y_{i,1}-x_{i,3}*y_{i,2}-x_{i,1}*y_{i,3})$$

这里的运算符表示对应分量计算。

12.2.4 单元刚度阵和向量的并行计算

仍以三角形单元为例，经过单元分析可得

$$F[u_i^{(l)}]=\iint_{P_i^{(l)}}\left[\left(\frac{\partial u_i^{(l)}}{\partial x}\right)^2+\left(\frac{\partial u_i^{(l)}}{\partial y}\right)^2-2f\cdot u_i^{(l)}\right]dxdy$$

$$= u_l^{(l)T} \cdot K_l^{(l)} \cdot u_l^{(l)} - 2u_l^{(l)T} \cdot f_l^{(l)} \tag{12-40}$$

$$i = 1,\ 2,\ \cdots,\ m;\quad l = 1,\ 2,\ \cdots,\ s$$

其中 $u_l^{(l)} = [u_{l,1}^{(l)},\ u_{l,2}^{(l)},\ u_{l,3}^{(l)}]^T$ 是单元未知向量。

$$f_l^{(l)} \triangleq [f_{l,1}^{(l)},\ f_{l,2}^{(l)},\ f_{l,3}^{(l)}]^T$$

$$= \frac{\Delta_l^{(l)}}{3}[\tilde{f}_{l,1}^{(l)},\ \tilde{f}_{l,2}^{(l)},\ \tilde{f}_{l,3}^{(l)}]^T$$

是单元向量，“$\triangleq$”表示记为，$\tilde{f}_{l,i}^{(l)}$ 是 $f(x, y)$ 在结点 $d_{l,i}^{(l)}$ 处的值。

$$K_l^{(l)} \triangleq \begin{bmatrix} k_{l,1,1}^{(l)} & k_{l,1,2}^{(l)} & k_{l,1,3}^{(l)} \\ k_{l,2,1}^{(l)} & k_{l,2,2}^{(l)} & k_{l,2,3}^{(l)} \\ k_{l,3,1}^{(l)} & k_{l,3,2}^{(l)} & k_{l,3,3}^{(l)} \end{bmatrix}$$

$$= \frac{1}{4\Delta_l^{(l)}}\left(\begin{bmatrix} b_{l,1}^{(l)} \cdot b_{l,1}^{(l)} & b_{l,1}^{(l)} \cdot b_{l,2}^{(l)} & b_{l,1}^{(l)} \cdot b_{l,3}^{(l)} \\ b_{l,2}^{(l)} \cdot b_{l,1}^{(l)} & b_{l,2}^{(l)} \cdot b_{l,2}^{(l)} & b_{l,2}^{(l)} \cdot b_{l,3}^{(l)} \\ b_{l,3}^{(l)} \cdot b_{l,1}^{(l)} & b_{l,3}^{(l)} \cdot b_{l,2}^{(l)} & b_{l,3}^{(l)} \cdot b_{l,3}^{(l)} \end{bmatrix}\right.$$

$$\left. + \begin{bmatrix} c_{l,1}^{(l)} \cdot c_{l,1}^{(l)} & c_{l,1}^{(l)} \cdot c_{l,2}^{(l)} & c_{l,1}^{(l)} \cdot c_{l,3}^{(l)} \\ c_{l,2}^{(l)} \cdot c_{l,1}^{(l)} & c_{l,2}^{(l)} \cdot c_{l,2}^{(l)} & c_{l,2}^{(l)} \cdot c_{l,3}^{(l)} \\ c_{l,3}^{(l)} \cdot c_{l,1}^{(l)} & c_{l,3}^{(l)} \cdot c_{l,2}^{(l)} & c_{l,3}^{(l)} \cdot c_{l,3}^{(l)} \end{bmatrix}\right)$$

$K_l^{(l)}$ 是单元刚度阵。

单元刚度阵 $\{K_l^{(l)}\}$ 的元素沿 l 方向的并行计算格式为

$$k_{i,m,l} = \left(\frac{1}{4\Delta_l}\right) * (b_{i,t} * b_{t,j} + c_{i,t} * c_{t,j}), \tag{12-41}$$

$$i = 1,\ 2,\ \cdots,\ m;\quad t,\ j = 1,\ 2,\ 3$$

单元向量 $\{f_l^{(l)}\}$ 沿 l 方向的并行计算格式为

$$f_{i,t} = \frac{\Delta_l}{3} * \tilde{f}_{i,t} \tag{12-42}$$

$$i = 1,\ 2,\ \cdots,\ m;\quad t = 1,\ 2,\ 3$$

12.2.5 总体装配总刚度阵和总向量的并行计算

总体合成

$$\sum_i \sum_l F[u_l^{(l)}] = \sum_i \sum_l (u_l^{(l)T} \cdot K_l^{(l)} u_l^{(l)} - 2u^{(l)T} \cdot f_l^{(l)})$$

$$=u^T\cdot K\cdot u-2u^T\cdot f \qquad (12\text{-}43)$$

其中总刚度阵K和总向量f分别是单元刚度阵$\langle K_i^{(b)}\rangle$和向量$\langle f_i^{(b)}\rangle$按编号迭加而成的。

我们将K和f的装配迭加分成m步来执行，每步完成一个不相关单元子集合对K和f的装配迭加。

下面，给出第i个单元子集合对K和f的装配迭加。

假设剖分单元后，共有N个结点，其中内部结点的个数为n。再假设K是按列的次序装配的。

已知单元刚度阵$K_i^{(b)}$的元素$k_{t,j}^{(b)}$（t，$j=1$，2，3）在K中的位置是第$e_{t,i}^{(b)}$行和第$e_{t,j}^{(b)}$列；单元向量$f_i^{(b)}$的元素$f_{t,i}^{(b)}$（$t=1$，2，3）在f中的位置是$e_{t,i}^{(b)}$。

并行装配迭加的计算步骤如下：

1. 并行计算间接控制向量

$$Q_{t,s,j}=N*Q_{t,j}+Q_{t,s}-1 \qquad t,\ j=1,\ 2,\ 3$$

其中

$$Q_{t,s}=[e_{t,1}^{(1)},\ e_{t,1}^{(2)},\ \cdots,\ e_{t,1}^{(s)}]^T$$

$$Q_{t,j}=[e_{t,1}^{(1)},\ e_{t,1}^{(2)},\ \cdots,\ e_{t,1}^{(s)}]^T$$

2. 在向量$Q_{t,s,j}$的间接控制下，在K中取向量$g_{t,s,j}$；在向量$Q_{t,s}$的间接控制下，在f中取向量$h_{t,s}$。

3. 并行迭加计算

$$\bar{g}_{t,s,j}=k_{t,s,j}+g_{t,s,j} \qquad t,\ j=1,\ 2,\ 3$$

$$\bar{h}_{t,s}=f_{t,s}+h_{t,s} \qquad t=1,\ 2,\ 3$$

4. 在向量$Q_{t,s,j}$和$Q_{t,s}$的间接控制下，将$\bar{g}_{t,s,j}$和$\bar{h}_{t,s}$分别送回到K和f中。

经过m步上述的并行装配迭加以后，就得到了总刚度阵K和总向量f。

12.3 抛物型方程差分格式的并行计算

本节主要介绍一维热传导方程的并行计算。

在区域G：$\{0\leqslant x\leqslant 1,\ 0\leqslant t\leqslant T\}$上，考虑一维热传导

方程的第一边值问题

$$Lu\equiv\frac{\partial u}{\partial t}-\frac{\partial^2 u}{\partial x^2}=0 \qquad 0<x<1,\quad 0<t<T \tag{12-44}$$

$$u|_{t=0}=\varphi(x) \qquad 0\leqslant x\leqslant 1 \tag{12-45}$$

$$u(0,t)=\psi_1(t) \quad u(1,t)=\psi_2(t) \quad 0\leqslant t\leqslant T \tag{12-46}$$

其中$\varphi(0)=\psi_1(0)$，$\varphi(1)=\psi_2(0)$。

将空间线段$[0,1]$ l 等分，即空间步长$h=1/l$；将时间线段$[0,T]$ m 等分，即时间步长$\tau=T/m$。

关于空间变量x，使用二阶中心差商，可得

$$\frac{du_k(t)}{dt}=\frac{u_{k+1}(t)-2u_k(t)+u_{k-1}(t)}{h^2} \tag{12-47}$$

$$k=1,\ 2,\ \cdots,\ l-1$$

其中$u_k(t)$是$u(kh,\ t)$的近似值。

公式（12-47）的矩阵-向量表示为

$$\frac{d\boldsymbol{u}(t)}{dt}=B\cdot\boldsymbol{u}(t)+\boldsymbol{b}(t) \tag{12-48}$$

其中$\boldsymbol{u}(t)=[u_1(t),\ u_2(t),\ \cdots,\ u_m(t)]^T$是$m$维未知向量；

$$A=-\frac{1}{h^2}\begin{pmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & 0 \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & -1 \\ & 0 & & -1 & 2 \end{pmatrix},$$

是m阶三对角阵；

$$\boldsymbol{b}(t)=\frac{1}{h^2}[\psi_1(t),\ 0,\ \cdots,\ 0,\ \psi_2(t)]^T$$

是m维向量。

12.3.1 显式格式的并行计算

关于公式（12-47）的时间变量 t，使用向前差商，可得显式格式

$$u_{k,j+1}=ru_{k+1,j}+(1-2r)u_{k,j}+ru_{k-1,j} \qquad (12\text{-}49)$$

$$k=1,\ 2,\ \cdots,\ l-1;\quad j=0,\ 1,\ \cdots,\ m-1$$

其中 $u_{k,j}$ 是网格结点（k，j）处 $u(kh,\ j\tau)$ 的近似值，$r=\tau/h^2$。

公式（12-49）的矩阵-向量表示为

$$\boldsymbol{u}_{j+1}=A_1\boldsymbol{u}_j+\boldsymbol{d}_j \qquad (12\text{-}50)$$

$$j=0,\ 1,\ 2,\ \cdots,\ m-1$$

其中 $\boldsymbol{u}_t$（$t=j$，$j+1$）是第 t 时间层上的内部结点上的向量；

$$A_1=\begin{pmatrix} 1-2r & r & & & 0 \\ r & 1-2r & r & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & r \\ 0 & & & r & 1-2r \end{pmatrix}$$

是 $l-1$ 阶三对角阵；

$$\boldsymbol{d}_j=r\,[\psi_{1,j},\ 0,\ \cdots,\ 0,\ \psi_{2,j}]^T$$

是 $l-1$ 维向量。

显式格式的截断误差为 $0(\tau+h^2)$。

当 $0<r\leqslant\frac{1}{2}$ 时，显式格式是稳定的。

显式格式的并行计算格式为

$$\boldsymbol{u}_{j+1}=r\boldsymbol{u}_{j+}+(1-2r)\boldsymbol{u}_j-r\boldsymbol{u}_{j-} \qquad (12\text{-}51)$$

$$j=0,\ 1,\ 2,\ \cdots,\ m-1$$

其中

$$\boldsymbol{u}_{j+}=[u_{2,j},\ u_{3,j},\ \cdots,\ u_{l,j}]^T$$

$$\boldsymbol{u}_{j-}=[u_{0,j},\ u_{1,j},\ \cdots,\ u_{l-2,j}]^T$$

12.3.2 隐式格式的并行计算

关于式（12-47）的时间变量 t，使用向后差商，可得隐式格式

$$-ru_{k-1,j+1}+(1+2r)u_{k,j+1}-ru_{k+1,j+1}=u_{k,j} \qquad (12\text{-}52)$$

$$k=1,\ 2,\ \cdots,\ l-1;\ j=0,\ 1,\ 2,\ \cdots,\ m-1$$

其相应的矩阵-向量表示为

$$A_2\boldsymbol{u}_{j+1}=\boldsymbol{u}_j+\boldsymbol{d}_{j+1} \qquad (12\text{-}53)$$

$$j=0,\ 1,\ 2,\ \cdots,\ m-1$$

其中

$$A_2=\begin{pmatrix} 1+2r & -r & & & 0 \\ -r & 1+2r & -r & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & -r \\ 0 & & & -r & 1+2r \end{pmatrix}$$

是 $l-1$ 阶三对角阵。

隐式格式（12-53）的计算，归结为计算三对角方程组，可以使用“循环奇偶约化法”等并行算法计算。

隐式格式的截断误差为 $O(\tau+h^2)$，且是无条件稳定的。

12.3.3 六点格式的并行计算

关于式（12-47）的时间变量 t，在点 $\left(k,\ j+\frac{1}{2}\right)$ 处使用中心差商，可得六点格式：

$$-\frac{r}{2}u_{k+1,j+1}+(1+r)u_{k,j+1}-\frac{r}{2}u_{k-1,j+1}$$

$$=\frac{r}{2}u_{k+1,j}+(1-r)u_{k,j}+\frac{r}{2}u_{k-1,j} \qquad (12\text{-}54)$$

$$k=1,\ 2,\ \cdots,\ l-1;\ j=0,\ 1,\ 2,\ \cdots,\ m-1$$

其相应的矩阵-向量表示为

$$(I+A_2)u_{j+1}=(I+A_1)u_j+d_j+d_{j+1} \qquad (12\text{-}55)$$
$$j=0,\ 1,\ 2,\ \cdots,\ m-1$$

六点格式（12-55）的等式右端部分是适于并行计算的，而$(I+A_2)$是$l-1$阶的三对角阵，所以六点格式的计算也是三对角方程组的计算问题。可以使用“循环奇偶约化法”等并行算法计算。

六点格式的截断误差是$O(\tau^2+h^2)$，且是无条件稳定的。

12.3.4 交替群显式格式的并行计算

交替群显式格式（AGE格式）是在“左-右”格式（LR格式）和“右-左”格式（RL格式）的基础上，发展而得到的一种显式格式。为此，首先介绍LR格式和RL格式。

LR格式为

$$-ru_{k+1,j+1}+(1+r)u_{k,j+1}=(1-r)u_{k,j}+ru_{k-1,j}$$
$$k=1,2,\cdots,l-1;\quad j=0,1,2,\cdots,m-1 \qquad (12\text{-}56)$$

RL格式为

$$-ru_{k,j+1}+(1+r)u_{k+1,j+1}=(1-r)u_{k+1,j}+ru_{k+2,j}$$
$$k=0,1,\cdots,l-2;\quad j=0,1,2,\cdots,m-1 \qquad (12\text{-}57)$$

假设l是偶数。

将LR格式和RL格式结合起来，可得

$$\begin{bmatrix}u_{k,j+1}\\ u_{k+1,j+1}\end{bmatrix}=\frac{1}{1+2r}\begin{bmatrix}r(1+r)u_{k-1,j}+(1-r^2)u_{k,j}\\ r^2u_{k-1,j}+r(1-r)u_{k,j}\end{bmatrix}$$
$$+\begin{bmatrix}r(1-r)u_{k+1,j}+r^2u_{k+2,j}\\ (1-r^2)u_{k+1,j}+r(1+r)u_{k+2,j}\end{bmatrix} \qquad (12\text{-}58)$$

将右边界条件代入LR格式（12-56），可得

$$u_{l-1,j+1}=\frac{1}{1+r}[r\psi_{x,j+1}+(1-r)u_{l-1,j}+ru_{l-2,j}] \qquad (12\text{-}59)$$

将左边界条件代入 RL 格式（12-57），可得

$$u_{1,j+1}=\frac{1}{1+r}\left[r\psi_{1,j+1}+(1-r)u_{1,j}+ru_{2,j}\right] \quad (12\text{-}60)$$

我们将式（12-58）和式（12-59）结合，就得到了群显式右格式（GER 格式）

$$(I+r\cdot G_1)\boldsymbol{u}_{j+1}=(I-rG_2)\boldsymbol{u}_j+\boldsymbol{b}_j^{(1)} \quad (12\text{-}61)$$

$$j=0,1,2,\cdots,m-1$$

其中

$$G_1=\begin{bmatrix} G^{(1)} & & & 0 \\ & G^{(2)} & & \\ & & \ddots & \\ & & & G^{\left(\frac{l-2}{2}\right)} \\ 0 & & & \\ & & & 1 \end{bmatrix}$$

$$G_2=\begin{bmatrix} 1 & & & 0 \\ & G^{(1)} & & \\ & & G^{(2)} & \\ & & & \ddots \\ 0 & & & G^{\left(\frac{l-2}{2}\right)} \end{bmatrix}$$

$$\boldsymbol{b}_j^{(1)}=r\cdot[\psi_{1,j},\ 0,\ \cdots,\ 0,\ \psi_{2,j+1}]^T$$

且有

$$G^{(i)}=\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \qquad i=1,\ 2,\ \cdots,\ \frac{1}{2}(l-2)$$

将式（12-58）和（12-60）结合，就得到了群显式左格式（GEL 格式）

$$(I+rG_2)\boldsymbol{u}_{j+1}=(I-rG_1)\boldsymbol{u}_j+\boldsymbol{b}_j^{(2)} \quad (12\text{-}62)$$

$$j=0,\ 1,\ 2,\ \cdots,\ m-1$$

其中

$$\boldsymbol{b}_j^{(2)}=r\,[\psi_{1,j+1},\ 0,\ \cdots,\ 0,\ \psi_{2,j}]^T$$

进一步将 GER 格式和 GEL 格式结合起来，可得 AGE 格式

$$\begin{cases}(I+rG_1)u_{2j-1}=(I-rG_2)\cdot u_{2j-2}+b_{2j-1}^{(1)}\\(I+rG_2)u_{2j}=(I-rG_1)\cdot u_{2j-1}+b_{2j}^{(2)}\end{cases}\qquad(12\text{-}63)$$

$$j=1,2,\cdots,\frac{m}{2}$$

AGE 格式的截断误差为 $O(\tau+h^2)$。

当 $r>0$ 时，AGE 格式是稳定的。

AGE 格式（12-63）非常适于并行计算的。对于每个 j，AGE 格式是通过 GER 格式和 GEL 格式来完成。下面，介绍 GER 格式的更直观的并行计算格式。

GER 格式的并行计算格式

$$\begin{cases}u_{j+1}=\dfrac{r(1+r)}{1+2r}u_{j-}+\dfrac{1-r^2}{1+2r}u_j+\dfrac{r(1-r)}{1+2r}u_{j+}\\\qquad+\dfrac{r^2}{1+2r}u_{j++}\\u_{j+1+}=\dfrac{r^2}{1+2r}u_{j-}+\dfrac{r(1-r)}{1+2r}u_j+\dfrac{1-r^2}{1+2r}u_{j+}\\\qquad+\dfrac{r(1+r)}{1+2r}u_{j++}\\u_{l-1,j+1}=\dfrac{r}{1+r}\cdot\psi_{2,j+1}+\dfrac{r}{1+r}\cdot u_{l-2,j}+\dfrac{1-r}{1+r}\cdot u_{l-1,j}\end{cases}\qquad(12\text{-}64)$$

其中

$$\begin{aligned}u_{j+1}&=[u_{1,j+1},\ u_{3,j+1},\ \cdots,\ u_{l-3,j+1}]^T\\u_{j+1+}&=[u_{2,j+1},\ u_{4,j+1},\ \cdots,\ u_{l-2,j+1}]^T\\u_{j-}&=[u_{0,j},\ u_{2,j},\ \cdots,\ u_{l-4,j}]^T\\u_j&=[u_{1,j},\ u_{3,j},\ \cdots,\ u_{l-3,j}]^T\\u_{j+}&=[u_{2,j},\ u_{4,j},\ \cdots,\ u_{l-2,j}]^T\\u_{j++}&=[u_{3,j},\ u_{5,j},\ \cdots,\ u_{l-1,j}]^T\end{aligned}$$

都是 $(l-2)/2$ 维向量。

GEL 格式和 GER 格式基本上类似，其并行计算格式也差不多。

12.4 输运方程差分格式的并行计算

考虑最简单的输运方程

$$\frac{\partial u}{\partial t}+c\frac{\partial u}{\partial x}=0 \quad -\infty<x<+\infty \quad 0<t\leqslant T \tag{12-65}$$

$$u(x,0)=\varphi(x) \quad -\infty<x<+\infty \tag{12-66}$$

其中C是输运波的速度。

取网格

$$\begin{cases} x_i=i\cdot h & i=0,\ \pm 1,\ \pm 2,\ \cdots \\ t_j=j\cdot \tau & j=0,\ 1,\ 2,\ \cdots,\ m \end{cases}$$

其中h和τ分别是x和t方向的步长。

12.4.1 拉克斯格式的并行计算

方程（12-65）关于t的偏导数采用向前差商，关于x采用中心差商，可得拉克斯（Lax）差分格式

$$u_{i,j+1}=u_{i,j}-\frac{c\tau}{2h}(u_{i+1,j}-u_{i-1,j}) \tag{12-67}$$

$$i=0,\pm 1,\pm 2,\cdots;\quad j=0,1,2,\cdots,m-1$$

拉克斯格式的截断误差为$O(\tau+h^2)$。

当$|c\tau/h|\leqslant 1$时，拉克斯格式是稳定的。

拉克斯格式的并行计算格式为

$$\boldsymbol{u}_{j+1}=\boldsymbol{u}_j-\frac{c\tau}{2h}(\boldsymbol{u}_{j+}-\boldsymbol{u}_{j-}) \tag{12-68}$$

$$j=0,\ 1,\ 2,\ \cdots,\ m-1$$

其中$\boldsymbol{u}_j$、$\boldsymbol{u}_{j+}$、$\boldsymbol{u}_{j-}$都是第j个时间层上的向量，对于任意的整数k，$\boldsymbol{u}_{j+}$和$\boldsymbol{u}_{j-}$的第k个分量分别是$\boldsymbol{u}_j$的第$k+1$和$k-1$个分量。

12.4.2 偏心格式的并行计算

方程（12-65）关于t和x的偏导数都用向前差商代替，可

得右偏心差分格式

$$u_{i,j+1}=\left(1+\frac{c\tau}{h}\right)\cdot u_{i,j}-\frac{c\tau}{h}\cdot u_{i+1,j} \tag{12-69}$$

$$i=0,\pm 1,\pm 2,\cdots;\quad j=0,1,2,\cdots,m-1$$

方程（12-65）关于 t 采用向前差商，关于 x 采用向后差商，则可得左偏心差分格式

$$u_{i,j+1}=\left(1-\frac{c\tau}{h}\right)u_{i,j}+\frac{c\tau}{h}\cdot u_{i-1,j} \tag{12-70}$$

$$i=0,\pm 1,\pm 2,\cdots;\quad j=0,1,2,\cdots,m-1$$

偏心格式的截断误差为 $O(\tau+h)$。

当 $c<0$ 和 $-1\leqslant c\tau/h\leqslant 0$ 时，右偏心格式稳定。

当 $c>0$ 和 $0\leqslant c\tau/h\leqslant 1$ 时，左偏心格式是稳定的。

右偏心格式的并行计算格式为

$$u_{j+1}=\left(1+\frac{c\tau}{h}\right)u_j-\frac{c\tau}{h}\cdot u_{j+} \tag{12-71}$$

$$j=0,\ 1,\ 2,\ \cdots,\ m-1$$

左偏心格式的并行计算格式为

$$u_{j+1}=\left(1-\frac{c\tau}{h}\right)u_j+\frac{c\tau}{h}u_{j-} \tag{12-72}$$

$$j=0,\ 1,\ 2,\ \cdots,\ m-1$$

12.4.3 拉克斯-文卓夫格式的并行计算

拉克斯-文卓夫（Lax-wendroff）格式的一般形式为

$$u(x,\ t+\Delta t)=u(x,\ t)+u_t(x,\ t)\cdot\Delta t+\frac{\Delta t^2}{2}u_{tt}(x,\ t) \tag{12-73}$$

从方程（12-67）可得

$$\begin{cases} u_t(x,\ t)=-cu_x(x,\ t) \\ u_{tt}(x,\ t)=c^2\cdot u_{xx}(x,\ t) \end{cases} \tag{12-74}$$

以（i，j）为中心点，对 $u_x(x,\ t)$ 采用一阶中心差商，对 $u_{xx}(x,\ t)$ 采用二阶中心差商。则可得方程（12-65）的具

有二阶精度的拉克斯-文卓夫格式

$$u_{i,j+1}=u_{i,j}-\frac{c\tau}{2h}(u_{i+1,j}-u_{i-1,j})+\frac{1}{2}\left(\frac{c\tau}{h}\right)^2(u_{i+1,j}-2u_{i,j}+u_{i-1,j}) \tag{12-75}$$

$$i=0,\pm1,\pm2,\cdots;\quad j=0,1,2,\cdots,m-1$$

拉克斯-文卓夫格式的截断误差为$O(\tau^2+h^2)$。

当$|c|\tau/h\leqslant1$时，拉克斯-文卓夫格式是稳定的。

拉克斯-文卓夫格式的并行计算格式为

$$u_{j+1}=a\cdot u_{j-}+b\cdot u_j+c\cdot u_{j+} \tag{12-76}$$

$$j=0,1,2,\cdots,m-1$$

其中

$$a=\frac{c\tau}{2h}+\frac{1}{2}\left(\frac{c\tau}{h}\right)^2$$

$$b=1-\left(\frac{c\tau}{h}\right)^2$$

$$c=-\frac{c\tau}{2h}+\frac{1}{2}\left(\frac{c\tau}{h}\right)^2$$

12.4.4 中子输运方程的并行计算

讨论使用卡尔逊的S_N格式求解中子输运方程的并行计算。

在输运近似下，一维球几何定常中子输运方程的多群形式为

$$\mu\cdot\frac{\partial\varphi_g}{\partial r}+\frac{1-\mu^2}{r}\cdot\frac{\partial\varphi_g}{\partial\mu}+\left(\Sigma_g^{tr}+\frac{\lambda}{v_g}\right)\varphi_g=Q_g$$

$$-1\leqslant\mu\leqslant1,\quad 0<r<r_M \tag{12-77}$$

当$-1\leqslant\mu\leqslant0$时，$\varphi_g(r_M,\ \mu)=0$ (12-78)

当$0<\mu\leqslant1$时，$\varphi_g(0,\ \mu)=\varphi_g(0,\ -\mu)$ (12-79)

其中

$$Q_g=\chi_g\cdot\sum_{l=1}^{G}(v\cdot\Sigma_l^f\psi_l)+\sum_{l=1}^{g}(\Sigma_{l\to g}^s\psi_l)+S_g$$

S_g为独立源，

$$\psi_g(r)=\frac{1}{2}\int_{-1}^{1}\psi_g(r,\ \mu)d\mu$$

将线段〔0，r_M〕剖分为K份

$$0=r_0<r_1<\cdots<r_k<r_{k+1}<\cdots<r_K=r_M$$

记$\Delta r_{k+1/2}=r_{k+1}-r_k$；

将线段〔-1，1〕$2N$等分

$$-1=\mu_0<\mu_1<\cdots<\mu_{j-1}<\mu_j<\cdots<\mu_{2N}=1$$

记$\Delta\mu=\mu_j-\mu_{j-1}=1/N$。

在点$(r_{k+1/2},\ \mu_{j-1/2})$处，方程(12-77)的迭代差分格式为

$$a_{g,j,k+1/2}^{(n+1)}\varphi_{g,j,k+1}^{(n+1)}-b_{g,j,k+1/2}^{(n+1)}\varphi_{g,j,k}^{(n+1)}+c_{g,j,k+1/2}^{(n+1)}\varphi_{g,j-1,k+1}^{(n+1)}$$
$$-d_{g,j,k+1/2}^{(n+1)}\varphi_{g,j-1,k}^{(n+1)}=\theta_j Q_{g,k+1/2}^{(n)} \tag{12-80}$$
$$j=0,1,\cdots,2N;\quad k=0,1,\cdots,K-1$$

其中$\varphi^{(n+1)}$表示第$n+1$次迭代未知值。

$$\begin{cases} a_{g,j,k+1/2}^{(n+1)}=\dfrac{\alpha_j}{\Delta r_{k+1/2}}+\dfrac{\beta_j}{\bar{r}_{k+1/2}}+\left(\Sigma_{g,k+1/2}^{tr}+\dfrac{\lambda^{(n+1)}}{v_{g,k+1/2}}\right) \\ b_{g,j,k+1/2}^{(n+1)}=\dfrac{\alpha_j}{\Delta r_{k+1/2}}-\dfrac{\beta_j}{\bar{r}_{k+1/2}}-\left(\Sigma_{g,k+1/2}^{tr}+\dfrac{\lambda^{(n+1)}}{v_{g,k+1/2}}\right) \\ c_{g,j,k+1/2}^{(n+1)}=\dfrac{\gamma_j}{\Delta r_{k+1/2}}-\dfrac{\beta_j}{\bar{r}_{k+1/2}}+\left(\Sigma_{g,k+1/2}^{tr}+\dfrac{\lambda^{(n+1)}}{v_{g,k+1/2}}\right) \\ d_{g,j,k+1/2}^{(n+1)}=\dfrac{\gamma_j}{\Delta r_{k+1/2}}+\dfrac{\beta_j}{\bar{r}_{k+1/2}}-\left(\Sigma_{g,k+1/2}^{tr}+\dfrac{\lambda^{(n+1)}}{v_{g,k+1/2}}\right) \end{cases} \tag{12-81}$$

$$Q_{g,k+1/2}^{(n)}=\chi_g\sum_{l=1}^{G}(\nu\Sigma_{l,k+1/2}^{f}\psi_{l,k+1/2}^{(n)})$$
$$+\sum_{l=1}^{g}(\Sigma_{l\to g,k+1/2}^{s}\psi_{l,k+1/2})+s_{g,k+1/2} \tag{12-82}$$

$$\psi_{g,k+1/2}^{(n)}=\frac{\Delta\mu}{8}\sum_{j=1}^{2N}(\varphi_{g,j-1,k}^{(n)}+\varphi_{g,j,k}^{(n)}+\varphi_{g,j-1,k+1}^{(n)}+\varphi_{g,j,k+1}^{(n)}) \tag{12-83}$$

$$\begin{cases} \alpha_j=\begin{cases} -2, & \text{当 } j=0 \text{ 时} \\ 2(\mu_{j-1}+2\Delta\mu/3), & \text{否则} \end{cases} \\ \beta_j=\begin{cases} 0, & \text{当 } j=0 \text{ 时} \\ \dfrac{2}{\Delta\mu}[1-\mu_{j-1}\mu_j-(\Delta\mu)^2/3], & \text{否则} \end{cases} \\ \gamma_j=2(\mu_j-2\Delta\mu/3) \\ \theta_j=\begin{cases} 2, & \text{当 } j=0 \text{ 时} \\ 4, & \text{否则} \end{cases} \end{cases} \tag{12-84}$$

$$\begin{cases} \Delta r_{k+1/2}=r_{k+1}-r_k \\ \bar{r}_{k+1/2}=\dfrac{r_{k+1}^2+r_{k+1}\cdot r_k+r_k^2}{2r_{k+1}+r_k} \\ r'_{k+1/2}=\dfrac{r_{k+1}^2+r_{k+1}\cdot r_k+r_k^2}{2r_k+r_{k+1}} \end{cases} \tag{12-85}$$

$$\lambda^{(n+1)}=\frac{\lambda^{(n-1)}(Q_{1,1/2}^{(n)}\cdot Q_{1,1/2}^{(n-2)}-Q_{1,1/2}^{(n-1)}\cdot Q_{1,1/2}^{(n-2)})+\lambda^{(n)}(Q_{1,1/2}^{(n-1)}\cdot Q_{1,1/2}^{(n-2)}-Q_{1,1/2}^{(n-1)}\cdot Q_{1,1/2}^{(n-1)})}{Q_{1,1/2}^{(n)}\cdot Q_{1,1/2}^{(n-2)}-Q_{1,1/2}^{(n-1)}\cdot Q_{1,1/2}^{(n-1)}} \tag{12-86}$$

$\lambda^{(n+1)}$ 为本次迭代的本征值。

迭代差分公式（12-80）的并行计算步骤如下：

1. 迭代前的系数计算

（1）将式（12-84）沿 j 方向并行计算可得

$$\boldsymbol{\alpha}=[\alpha_0,\ \alpha_1,\ \cdots,\ \alpha_N,\ \alpha_{N+1},\ \cdots,\ \alpha_{2N}]^T$$
$$\boldsymbol{\beta}=[\beta_0,\ \beta_1,\ \cdots,\ \beta_N,\ \beta_{N+1},\ \cdots,\ \beta_{2N}]^T$$
$$\boldsymbol{\gamma}=[\gamma_0,\ \gamma_1,\ \cdots,\ \gamma_N,\ \gamma_{N+1},\ \cdots,\ \gamma_{2N}]^T$$
$$\boldsymbol{\theta}=[2,\ 4,\ \cdots,\ 4,\ 4,\ \cdots,\ 4]^T$$

（2）将式（12-85）沿 k 方向并行计算可得

$$\Delta\boldsymbol{r}=[\Delta r_{1/2},\ \Delta r_{1+1/2},\ \cdots,\ \Delta r_{k-1/2}]^T$$
$$\bar{\boldsymbol{r}}=[\bar{r}_{1/2},\ \bar{r}_{1+1/2},\ \cdots,\ \bar{r}_{k-1/2}]^T$$
$$\boldsymbol{r}'=[r'_{1/2},\ r'_{1+1/2},\cdots,\ r'_{k-1/2}]^T$$

（3）并行计算

$$a_{g,k+1/2}=\frac{\alpha}{\Delta r_{k+1/2}}+\frac{\beta}{\bar{r}_{k+1/2}}+\Sigma_{g,k+1/2}^{tr}$$

$$b_{g,k+1/2}=\frac{\alpha}{\Delta r_{k+1/2}}-\frac{\beta}{r'_{k+1/2}}-\Sigma_{g,k+1/2}^{tr}$$

$$c_{g,k+1/2}=\frac{\gamma}{\Delta r_{k+1/2}}-\frac{\beta}{\bar{r}_{k+1/2}}+\Sigma_{g,k+1/2}^{tr}$$

$$d_{g,k+1/2}=\frac{\gamma}{\Delta r_{k+1/2}}+\frac{\beta}{r'_{k+1/2}}-\Sigma_{g,k+1/2}^{tr}$$

$$k=0,\ 1,\ 2,\ \cdots,\ K-1$$

2. 迭代计算，对于 $n=1,\ 2,\ 3,\ \cdots$；

（1）根据式（12-86），计算本次迭代的本征值。

（2）将式（12-83）沿 k 方向并行计算可得

$$\psi_g^{(n)}=[\psi_{g,1/2}^{(n)},\ \psi_{g,1+1/2}^{(n)},\ \cdots,\ \psi_{g,k-1/2}^{(n)}]^T$$

（3）将式（12-82）沿 k 方向并行计算可得

$$Q_g^{(n)}=[Q_{g,1/2}^{(n)},\ Q_{g,1+1/2}^{(n)},\ \cdots,\ Q_{g,k-1/2}^{(n)}]^T$$

（4）并行计算

$$a_{g,k+1/2}^{(n+1)}=a_{g,k+1/2}+\lambda^{(n+1)}/v_{g,k+1/2}$$

$$b_{g,k+1/2}^{(n+1)}=b_{g,k+1/2}-\lambda^{(n+1)}/v_{g,k+1/2}$$

$$c_{g,k+1/2}^{(n+1)}=c_{g,k+1/2}+\lambda^{(n+1)}/v_{g,k+1/2}$$

$$d_{g,k+1/2}^{(n+1)}=d_{g,k+1/2}-\lambda^{(n+1)}/v_{g,k+1/2}$$

$$k=0,\ 1,\ 2,\ \cdots,\ K-1$$

（5）在 $-1\leqslant\mu\leqslant 0$ 的范围内，递推计算

$$\varphi_{g,j,k}^{(n+1)}=\frac{a_{g,j,k+1/2}^{(n+1)}}{b_{g,j,k+1/2}^{(n+1)}}\cdot\varphi_{g,j,k+1}^{(n+1)}+\frac{c_{g,j,k+1/2}^{(n+1)}}{d_{g,j,k+1/2}^{(n+1)}}\cdot\varphi_{g,j-1,k+1}^{(n+1)}-\frac{d_{g,j,k+1/2}^{(n+1)}}{b_{g,j,k+1/2}^{(n+1)}}\cdot\varphi_{g,j-1,k}^{(n+1)}-\frac{\theta_j\cdot Q_{g,k+1/2}^{(n)}}{b_{g,j,k+1/2}^{(n+1)}}$$

$$j=0,1,\cdots,N;\quad k=K-1,K-2,\cdots,1,0 \tag{12-87}$$

式中

$$\varphi_{g,-1,k}^{(n+1)}=0\qquad k=K,\ K-1,\ \cdots,\ 1,\ 0$$

$$\varphi_{g,j,k}^{(n+1)}=0\qquad j=0,\ 1,\ \cdots,\ N$$

（6）在 $0<\mu\leqslant 1$ 的范围内，递推计算

$$\varphi^{(n+1)}_{g,i,k+1}=\frac{b^{(n+1)}_{g,i,k+1/2}}{a^{(n+1)}_{g,i,k+1/2}}\cdot\varphi^{(n+1)}_{g,i,k}-\frac{c^{(n+1)}_{g,i,k+1/2}}{a^{(n+1)}_{g,i,k+1/2}}\cdot\varphi^{(n+1)}_{g,i-1,k+1}$$

$$+\frac{d^{(n+1)}_{g,i,k+1/2}}{a^{(n+1)}_{g,i,k+1/2}}\cdot\varphi^{(n+1)}_{g,i-1,0}+\frac{\theta_i\cdot Q^{(n)}_{g,k+1/2}}{a^{(n+1)}_{g,i,k+1/2}} \tag{12-88}$$

$$j=N+1,\ N+2,\ \cdots,\ 2N;\ k=0,\ 1,\ 2,\ \cdots,\ K-1$$

其中

$$\varphi^{(n+1)}_{g,N+I,0}=\varphi^{(n+1)}_{g,N-I,0}\quad I=0,\ 1,\ 2,\ \cdots,\ N$$

这是根据中心对称条件（12-79）得到的。

对于每个固定的 j，式（12-87）和（12-88）都是一阶线性递归计算式，可以使用倍增法，分段法、循环加倍法等并行算法来计算。

针对上面介绍的中子输运方程求解问题，我们分别在YH 机、757 机和 M-170 机上，编制了并行的或串行的程序。

下面，给出 16 群定常问题，在不同的机器上，使用不同程序的运行时间表（CPU 时间单位：秒）。

表12-1　16群定常S_N程序的运行时间

题号	网格点规模	YH 机汇编程序	YH机向量FT程序	YH机标量FT程序	757机向量FT程序	757机标量FT程序	M-170机标量FT程序
1	4×10	0.416	1.8407	2.895	14.65	27.79	16.427
2	16×100	7.7483	16.0945	64.369	161.05	557.94	453.523
3	16×100	52.8518	106.4597	428.572	1107.3		3065.147
4	16×100	392.5774	615.6364	2561.279	6552.0		

12.5 可压缩流体动力学欧拉方法的并行计算

质点网格法（PIC 法）和多流体网格法是数值流体动力学中二个较好的计算方法。它们都属于欧拉方法，能计算大扰动、大变形的问题，并兼备了拉格朗日方法的某些优点。

这二个方法都分成差分计算和输运计算二个阶段，其中二个方法的差分计算是相同的；不同的是，PIC 法采用离散质点输运，而多流体法采用连续的质量输运。

本节介绍了这二个方法及其并行计算。

12.5.1 可压缩流体动力学方程

在柱坐标系中，考虑无粘性、无热传导的可压缩流体的偏微分方程组

$$\begin{cases} \dfrac{\partial \rho}{\partial t}+\dfrac{\partial \rho u}{\partial z}+\dfrac{\partial r\rho v}{r\partial r}=0 \\ \rho\left(\dfrac{\partial u}{\partial t}+u\dfrac{\partial u}{\partial z}+v\dfrac{\partial u}{\partial r}\right)=-\dfrac{\partial p}{\partial z} \\ \rho\left(\dfrac{\partial v}{\partial t}+u\dfrac{\partial v}{\partial z}+v\dfrac{\partial v}{\partial r}\right)=-\dfrac{\partial p}{\partial r} \\ \rho\left(\dfrac{\partial e}{\partial t}+u\dfrac{\partial e}{\partial z}+v\dfrac{\partial e}{\partial r}\right)=-p\left(\dfrac{\partial u}{\partial z}+\dfrac{\partial rv}{r\partial r}\right) \\ 0<z<Z_M,\quad 0<r<R_M,\quad t>0 \end{cases} \tag{12-89}$$

其中 z 和 r 分别是柱坐标系中的轴向和径向变量，u 和 v 是轴向和径向的分速度，e 是比内能，ρ 是密度，$p=F_\beta(\rho,\ e)$ 是压力，它是依物质记号 β 而定的已知函数。

将轴向线段〔0，Z_M〕I 等分，即轴向步长 $\Delta z=Z_M/I$；将径向线段〔0，R_M〕J 等分，其径向步长 $\Delta r=R_M/J$。从而将矩形区域$\{0\leqslant z\leqslant Z_M,\ 0\leqslant r\leqslant R_M\}$离散成矩形网格域。

对于每个网格编号（i，j），编号顺序是自左至右（$1\leqslant i\leqslant I$）和自下而上（$1\leqslant j\leqslant J$）的。网格（i，j）的中心

位置是 $z_i=\left(i-\frac{1}{2}\right)\Delta z$ 和 $r_j=\left(j-\frac{1}{2}\right)\Delta r$。矩形网格的图示如下：

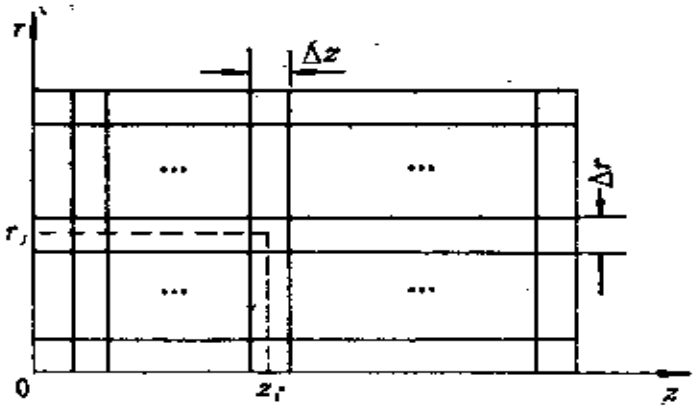


图12-3 矩形网格划分图

假定每个网格中最多含有二种物质，用 θ 来标识（$\theta=1$，2）。

下面，给出网格及其网格（i，j）上的一些物理量的定义

空网格：不含任何物质的网格；

纯网格：只含第一种物质的网格；

混合网格：含二种物质的网格；

自由面网格：本网格含有物质，但相邻网格中有空网格。

$\beta_{\theta,i,j}$：网格（i，j）上第 θ 种物质的物质记号（简记为物质记号）。

$M_{\theta,i,j}$：质量；

$\rho_{\theta,i,j}$：密度；

$e_{\theta,i,j}$：比内能；

$E_{\theta,i,j}$：比能；

$(\mathrm{ME})_{\theta,i,j}$：总能；

$Q_{i,j}$：总内能；

$M_{i,j}$：总质量；

$\sigma_{i,j}$：第一种物质所占的体积份额；

$p_{i,j}$：压力；

$u_{i,j}$：轴向分速度；

$v_{i,j}$：径向分速度；

$X_{i,j}$：轴向分动量；

$Y_{i,j}$：径向分动量；

$V_j = 2\pi r_j \cdot \Delta z \cdot \Delta r$：网格（$i$，$j$）的体积；

$A_j^i = 2\pi r_j \cdot \Delta r$：网格（$i$，$j$）右边界的面积；

$A_{j+1/2}^i = 2\pi r_{j+1/2} \cdot \Delta z$：网格（$i$，$j$）上边界的面积

接着，给出上述物理量的一般向量表示形式

ω_i：表示第 i 列网格的中心位置处物理量 ω 组成的向量；

ω_j：表示第 j 行网格中心位置处的向量；

$\omega_{i+1/2}$：表示第 i 列网格右边界处的向量；

$\omega_{j+1/2}$：表示第 j 行网格上边界处的向量。

例如：$\boldsymbol{u}_{i+1/2} = [u_{i+1/2,1}, u_{i+1/2,2}, \cdots, u_{i+1/2,J}]^T$，其中 $u_{i+1/2,j}$ 是网格（i，j）右边界处的轴向分速度。

关于初始条件，已知 $t = 0$ 时，每个网格上的所有物理量。

关于边界条件有三类

1．固壁边界条件

在固壁边界上，法向速度和压力梯度为零，下边界为对称轴，按固壁条件处理。

2．连续边界条件

允许物质从计算区域的某些边界连续流入或流出，这种边界二边的力学状态完全相同。

3．自由面条件

忽略大气压，与真空相邻的物质界面（自由面）上的压力为零。

12.5.2 差分的并行计算

差分计算阶段忽略输入项，计算单个网格内能的中间值 $\bar{u}$，$\bar{v}$ 和 $\bar{\varepsilon}$。

忽略输运项，可得偏微分方程组

$$\begin{cases} \rho \dfrac{\partial u}{\partial t} = -\dfrac{\partial (p+q)}{\partial z} \\ \rho \dfrac{\partial v}{\partial t} = -\dfrac{\partial r\rho}{r\partial r} + \dfrac{p}{r} - \dfrac{\partial q}{\partial r} \\ \rho \dfrac{\partial e}{\partial t} = -\dfrac{p}{r}\dfrac{\partial rv}{\partial r} - \dfrac{\partial rqv}{r\partial r} + v\dfrac{\partial q}{\partial r} \\ \qquad\qquad - p\dfrac{\partial u}{\partial z} - \dfrac{\partial qu}{\partial z} + u\dfrac{\partial q}{\partial z}, \end{cases} \tag{12-90}$$

其中 q 为引入的人工粘性项。

为了使得方程（12-90）的差分计算适于并行，首先采取如下三步处理：

1. 在计算区域的边界四周虚设一圈网格，使网格域扩充为 $(I+2)\times(J+2)$。

对于固壁边界和对称轴，虚设网格的速度与相邻边界网格速度成镜面反射，其它力学量相同。对于连续边界，虚设网格的所有力学量与相邻边界网格的量相同。这样边界网格就变成了内网格。

2. 对于每个网格（i，j）形成二个特征量 $g_{1,i,j}$（简记为 $g_{i,j}$）和 $g_{2,i,j}$。

对于空网格，有 $g_{1,i,j}=g_{2,i,j}=0$；

对于纯网格，有 $g_{1,i,j}=1.0$，$g_{2,i,j}=0$；

对于混合网格，有 $g_{1,i,j}=g_{2,i,j}=1.0$。

3. 在差分计算中，凡是可能为零的分母都加上小量 δ。

这样，方程（12-90）的差分计算可以分成三步来执行。

1. 压力的并行计算

$$\boldsymbol{p}_i = F_{pi}(\boldsymbol{\rho}_i,\ \boldsymbol{e}_i) \qquad i=1,\ 2,\ \cdots,\ I$$

其中

$$\rho_{i,j}=M_{1,i,j}/V_i,\quad e_{i,j}=e_{1,i,j},\quad \beta_{i,j}=\beta_{1,i,j}。$$

上式只能计算得到纯网格与空网格上的压力。由于混合网格的数量一般不超过总网格数的 5 %，而且计算复杂，宜采用标量

计算。

2. 网格边界值的并行计算

（1）网格边界压力的并行计算

$$p_{i+1/2}=\frac{g_i * p_{i+1}+g_{i+1} * p_i}{g_i+g_{i+1}+\delta} \quad i=1,\ 2,\ \cdots,\ I$$

$$p_{j+1/2}=\frac{g_j * p_{j+1}+g_{j+1} * p_j}{g_j+g_{j+1}+\delta} \quad j=1,\ 2,\ \cdots,\ J$$

（2）网格边界粘性的并行计算

$$q_{i+1/2}=f\cdot\frac{M_i^{(n)}+M_{i+1}^{(n+1)}}{V}(g_i * u_{i+1}^{(n)}-g_{i+1} * u_i^{(n)}-|g_i * u_{i+1}^{(n)}-g_{i+1} * u_i^{(n)}|)^2$$

$$i=1,\ 2,\ \cdots,\ I$$

$$q_{j+1/2}=f\cdot\left(\frac{M_j^{(n)}}{V_j}+\frac{M_{j+1}^{(n)}}{V_{j+1}}\right)(g_j * v_{j+1}^{(n)}-g_{j+1} * v_j^{(n)}-|g_j * v_{j+1}^{(n)}-g_{j+1} * v_j^{(n)}|)^2$$

$$j=1,\ 2,\ \cdots,\ J$$

其中 f 为粘性系数。

（3）网格边界速度的并行计算

$$\bar{u}_{i+1/2}=\frac{u_i^{(n)}+\tilde{u}_i+u_{i+1}^{(n)}+\tilde{u}_{i+1}}{2(g_i+g_{i+1}+\delta)}$$

$$i=1,\ 2,\ \cdots,\ I$$

$$\bar{v}_{j+1/2}=\frac{v_j^{(n)}+\tilde{v}_j+v_{j+1}^{(n)}+\tilde{v}_{j+1}}{2(g_j+g_{j+1}+\delta)}$$

$$j=1,\ 2,\ \cdots,\ J$$

3. 中间值的并行计算

（1）轴向分速度中间值的并行计算

$$\tilde{u}_i=u_i^{(n)}-\frac{\Delta t\cdot g_i * A^*}{M_i^{(n)}+\delta}(p_{i+1/2}-p_{i-1/2}+q_{i+1/2}-q_{i-1/2})$$

$$i=1,\ 2,\ \cdots,\ I$$

其中 $\tilde{u}_{i,j}$ 是网格（i，j）的轴向分速度中间值。

（2）径向分速度中间值的并行计算

$$\tilde{\boldsymbol{v}}_j=\boldsymbol{v}_j^{(n)}-\frac{\Delta t\cdot \boldsymbol{g}_j}{\boldsymbol{M}_j^{(n)}+\delta}\left[A_{j+1/2}^r(\boldsymbol{p}_{j+1/2}-\boldsymbol{p}_j)+A_j^r(\boldsymbol{q}_{j+1/2}-\boldsymbol{q}_{j-1/2})+A_{j-1/2}^r(\boldsymbol{p}_j-\boldsymbol{p}_{j-1/2})\right]$$

$$j=1,\ 2,\ \cdots,\ J$$

其中 $\tilde{v}_{i,j}$ 是网格（i，j）的径向分速度中间值。

（3）比内能中间值的并行计算

$$\tilde{\boldsymbol{e}}_{\theta,i}=\boldsymbol{e}_{\theta,i}^{(n)}+\frac{\boldsymbol{g}_{\theta,i}*(\Delta\boldsymbol{Q})_i}{\boldsymbol{V}*\boldsymbol{\rho}_{\theta,i}+\delta}$$

$$\theta=1,\ 2;\quad i=1,\ 2,\ \cdots,\ I$$

其中

$$(\Delta Q)_{i,j}=(\Delta Qu)_{i,j}+(\Delta Qv)_{i,j}$$

$$(\Delta\boldsymbol{Qu})_i=\boldsymbol{A}^2\cdot\Delta t\cdot\left[\bar{\boldsymbol{u}}_{i-1/2}*(\boldsymbol{p}_i+\boldsymbol{q}_{i-1/2})+\bar{\boldsymbol{u}}_i*(\boldsymbol{q}_{i+1/2}-\boldsymbol{q}_{i-1/2})-\bar{\boldsymbol{u}}_{i+1/2}*(\boldsymbol{p}_i+\boldsymbol{q}_{i+1/2})\right]$$

$$(\Delta\boldsymbol{Qv})_j=\Delta t\left[\bar{\boldsymbol{v}}_{j-1/2}*(A_{j-1/2}^r\boldsymbol{p}_j+A_{j-1}^r\boldsymbol{q}_{j-1/2})+\bar{\boldsymbol{v}}_j*(A_{j+1/2}^r\boldsymbol{q}_{j+1/2}-A_{j-1/2}^r\boldsymbol{q}_{j-1/2})-\bar{v}_{j+1/2}*(A_{j+1/2}^r\boldsymbol{p}_j+A_{j+1}^r\boldsymbol{q}_{j+1/2})\right]$$

12.5.3 输运的并行计算

在输运计算阶段，只是在输运修改时，PIC 法和多流体法的计算不同。

输运计算也分成三步执行。

1．网格总量的并行计算

$$\boldsymbol{X}_i=\sum_{\theta=1}^{2}\tilde{\boldsymbol{u}}_i*\boldsymbol{M}_{\theta,i}^{(n)}$$

$$\boldsymbol{Y}_i=\sum_{\theta=1}^{2}\tilde{\boldsymbol{v}}_i*\boldsymbol{M}_{\theta,i}^{(n)}$$

$$(ME)_{\theta,i}=M_{\theta,i}^{(n)} * \widetilde{E}_{\theta,i}$$

$$\theta=1,\quad 2;\quad i=1,\quad 2,\quad \cdots,\quad I$$

其中

$$\widetilde{E}_{\theta,i}=\widetilde{e}_{\theta,i}+\frac{1}{2}(\widetilde{u}_i^2+\widetilde{v}_i^2)$$

2. 输运修改的并行计算

（1）PIC 法的输运修改

① 计算质点运动

假设 t^n 时刻，质点 K 的坐标是 $(Z_K^{(n)},\ R_K^{(n)})$。

质点运动分三步计算

a. 求出质点 K 所在的网格 $(i,\ j)$

$$i=[Z_K^{(n)}]+1 \qquad j=[R_K^{(n)}]+1.$$

b. 计算质点 K 的有效速度

$$u_K=\frac{\sum_{m_1}\sum_{n=0}^{1} A_{i'+m,j'+n}\cdot\widetilde{u}_{i'+m,j'+n}}{\sum_{m_1}\sum_{n=0}^{1} A_{i'+m,j'+n}\cdot g_{i'+m,j'+n}}$$

$$v_K=\frac{\sum_{m_2}\sum_{n=0}^{1} A_{i'+m,j'+n}\cdot\widetilde{v}_{i'+m,j'+n}}{\sum_{m_2}\sum_{n=0}^{1} A_{i'+m,j'+n}\cdot g_{i'+m,j'+n}}$$

其中

$$A_{i',j'}=b_z\cdot b_r \qquad b_z=[Z_K^{(n)}+0.5]$$

$$A_{i',j'+1}=b_z'\cdot b_r \qquad b_z'=1-b_z$$

$$A_{i'+1,j'}=b_z\cdot b_r' \qquad b_r=[R_K^{(n)}+0.5]$$

$$A_{i'+1,j'+1}=b_z'\cdot b_r' \qquad b_r'=1-b_r$$

$$i'=[Z_K^{(n)}]$$

$$j'=[R_K^{(n)}]$$

c．求出 $t^{n+1}=t^n+\Delta t$ 时刻，质点 K 的位置

$$Z_K^{(n+1)}=Z_K^{(n)}+\frac{u_K'\cdot\Delta t}{\Delta z}\qquad R_K^{(n+1)}=R_K^{(n)}+\frac{v_K'\cdot\Delta t}{\Delta r}$$

和质点 K 所在的网格 $(i_1,\ j_1)$

$$i_1=[Z_K^{(n+1)}]+1,\quad j_1=[R_K^{(n+1)}]+1$$

上的质点运动计算，对所有的质点都是可以并行计算的。

b．修正网格总量

修改流出网格 $(i,\ j)$ 和流入网格 $(i_1,\ j_1)$ 的质量、动量、和能量。由于每个时间步只有 1 — 2 %的质点通过网格边界，宜采用标量计算。

（2）多流体法的输运修改

① 质量流的并行计算

计算通过网格 $(i,\ j)$ 和 $(i+,\ j)$（对所有的 j）公共边界（面积为 A_i^r）的质量流 $(\Delta M)_{\theta,i+1/2,j}$，计算分四步执行。

a．输运体积的并行计算

通过网格 $(i,\ j)$ 和 $(i+1,\ j)$（对所有 j）公共边界输运流体的体积的并行计算为

$$\widetilde{V}_{i+1/2}=\Delta t\cdot A^r*\widetilde{u}_{i+1/2}|$$

其中

$$\widetilde{u}_{i+1/2}=\frac{\widetilde{u}_i+\widetilde{u}_{i+1}}{(g_i+g_{i+1}+\delta)\left[1+\frac{\Delta t}{\Delta z}(g_i*\widetilde{u}_{i+1}-g_{i+1}*\widetilde{u}_i)\right]}$$

为边界速度向量。

b．贡献和接受网格位置的并行计算

贡献网格所在的列数向量

$$\xi_1=\begin{cases}i, & \text{当 } \widetilde{u}_{i+1/2}\geqslant 0 \text{ 时}\\ i+1, & \text{否则}\end{cases}$$

即当 $\tilde{u}_{i+1/2,j}\geqslant 0$ 时，ξ_1 的第 j 个分量为 i，否则为 $i+1$。可以使用向量归并方法计算。

贡献网格的位置向量

$$\eta_1=\xi_1*(J+2)+N$$

其中

$$N=[1,\ 2,\ \cdots,\ J+2]^T$$

接受网格所在的列数向量

$$\xi_2=\begin{cases} i+1, & \text{当}\widetilde{u}_{i+1/2}\geqslant 0\text{时} \\ i, & \text{否则} \end{cases}$$

接受网格的位置向量

$$\eta_2=\xi_2*(J+2)+N$$

为了记法简便，${}^*\omega$ 表示以贡献网格位置向量 η_1 作为间接地址访问主存得到的向量；ω^* 表示 η_2 作为间接地址访问主存得到的向量。

c．输运体积份额的并行计算

假定贡献网格第一种物质输运体积的份额为 $x_{i+1/2,j}$，第二种物质的份额为 $1-x_{i+1/2,j}$。

$$x_0=\begin{cases} \sigma_j^*, & \text{当}{}^*\beta_{1,i}=\beta_{2,i}^*\text{时} \\ 1-\sigma_j^*, & \text{否则} \end{cases}$$

其中 σ_j^* 为接受网格第一种物质在网格中所占的体积份额向量。

$$x_1=\begin{cases} 1.0, & \text{当}{}^*\beta_{1,i}+{}^*\beta_{2,i}\neq\beta_{1,i}^*+\beta_{2,i}^*\text{时} \\ x_0, & \text{否则} \end{cases}$$

$$x_2=\begin{cases} 0, & \text{当}{}^*\beta_{1,i}=\beta_{1,i}^*\text{或}{}^*\beta_{1,i}=\beta_{2,i}^*\text{时} \\ x_1, & \text{否则} \end{cases}$$

$$x_3=\min\left\{x_2,\ \frac{{}^*M_{1,i}}{{}^*\rho_{1,i}*\widetilde{V}_{i+1/2}+\delta}\right\}$$

$$x_{i+1/2}=\max\left\{x_3,\ 1-\frac{{}^*M_{2,i}}{{}^*\rho_{2,i}*\widetilde{V}_{i+1/2}+\delta}\right\}$$

d．质量流的并行计算

$$(\Delta M)_{1,i+1/2}=x_{i+1/2}*{}^{*}\rho_{1,i}*\widetilde{V}_{i+1/2}$$

$$(\Delta M)_{2,i+1/2}=(1-x_{i+1/2})*{}^{*}\rho_{2,i}*\widetilde{V}_{i+1/2}$$

$$(\Delta M)_{i+1/2}=(\Delta M)_{1,i+1/2}+(\Delta M)_{2,i+1/2}$$

② 修改网格总量的并行计算

$${}^{*}X_i={}^{*}X_i-{}^{*}\widetilde{u}_i*(\Delta M)_{i+1/2}$$

$$X_i^{*}=X_i^{*}+{}^{*}\widetilde{u}_i*(\Delta M)_{i+1/2}$$

$${}^{*}Y_i={}^{*}Y_i-{}^{*}\widetilde{v}_i*(\Delta M)_{i+1/2}$$

$$Y_i^{*}=Y_i^{*}+{}^{*}\widetilde{v}_i*(\Delta M)_{i+1/2}$$

$${}^{*}M_{\theta,i}={}^{*}M_{\theta,i}-(\Delta M)_{\theta,i+1/2}$$

$$M^{*}_{K\theta,i}=M^{*}_{K\theta,i}+(\Delta M)_{\theta,i+1/2}$$

$${}^{*}(ME)_{\theta,i}={}^{*}(ME)_{\theta,i}-{}^{*}\widetilde{E}_{\theta,i}*(\Delta M)_{\theta,i+1/2}$$

$$(ME)^{*}_{K\theta,i}=(ME)^{*}_{K\theta,i}+{}^{*}\widetilde{E}_{\theta,i}*(\Delta M)_{\theta,i+1/2}$$

其中

$$K_\theta=\begin{cases}1, & \text{当}{}^{*}\beta_{\theta,i}=\beta_{1,i}^{*}\text{时}\\ 2, & \text{否则}\end{cases}$$

关于通过网格（i，j）和（i，$j+1$）（对所有的 i）公共边界（面积为 $A^{*}_{j+1/2}$）的质量流和修改网格总量的计算，与上面的计算类似。

3. 重新分配的并行计算

重新分配是用输运修改得到的网格动量和总能量，计算 $t^{n+1}=t^{n}+\Delta t$ 时刻网格的速度和比内能。

$$u_i^{(n+1)}=\frac{X_i^{(n+1)}}{M_{1,i}^{(n+1)}+M_{2,i}^{(n+1)}+\delta}$$

$$v_i^{(n+1)}=\frac{Y_i^{(n+1)}}{M_{1,i}^{(n+1)}+M_{2,i}^{(n+1)}+\delta}$$

$$e_{\theta,i}^{(n+1)}=\frac{(ME)_{\theta,i}^{(n+1)}}{M_{\theta,i}^{(n+1)}+\delta}-\frac{1}{2}(u_i^{(n+1)2}+v_i^{(n+1)2})$$

$\theta=1,\ 2,\ i=1,\ 2,\ \cdots,\ I$

12.5.4 并行计算的效率

通过对算法的并行改造，这二个方法的绝大部分计算实现了向量化，对个别不宜并行计算的部分也合理地采用串、并行混合计算。在编制 YH 机向量汇编程序时，主要采用了下面一些措施来提高计算速度。

1．避免了向量存访指令的体碰头现象。

2．定义一些全局量，合理安排各种寄存器，尽可能减少访问主存的次数。

3．充分利用多功能部件和采用链接技术，提高向量指令运算的并行度。

4．合理编制和安排指令，尽量减少寄存器和功能部件的等待时间，尽量将标量运算指令插入向量运算指令之间，使标量计算时间被向量运算所吸收。

5．利用条件汇编，减少转移指令。

6．使用好一些特殊指令。例如：使用位串寄存器 VM 与向量归并指令使二分叉的计算向量化；使用压缩指令，将跑出计算系统的质点去掉，减少存储量与计算量；使用向量指令的循环特性，加快 $\sum_{i=1}^{N} X_i$ 和 $\prod_{i=1}^{N} X_i$ 型算式的计算等。

由于较好地发挥了 YH 机的特点，编制的向量汇编程序的并行计算效果比较好，计算速度也比较高。

以网格数为 38×24 的二维问题的计算为例，比较 YH 机和某百万次/每秒以上的机器的汇编程序的计算速度，YH 机上的 PIC 向量汇编程序的计算速度快 56 倍，多流体向量汇编程序的速度快 44 倍。如果网格数更多的话，整个问题的并行计算速度将更快。

12.6 理想气体超音速轴对称钝体绕流的并行计算

有轴对称钝头飞行器，处于超音速气流中，假设气体具有无粘性、无热传导等性质，计算确定飞行器周围的流场。一般情况下，钝体飞行器周围流场中，钝头附近为亚跨音速区，钝体后身附近为超音速区，以一个激波为其分界面。本节主要介绍使用分激波的稳定方法求解钝头附近的亚跨音速区流动的并行计算。

12.6.1 亚跨音速区的流动方程

为了简化方程，突出并行计算特点，考虑无粘性、无热传导和没有外力作用的理想气体下的零攻角轴对称钝体绕流的气体动力学方程组

$$\begin{cases} \dfrac{\partial R}{\partial t}+u\cdot\dfrac{\partial R}{\partial r}+\dfrac{v}{r}\cdot\dfrac{\partial R}{\partial\theta}+\dfrac{\partial u}{\partial r}+\dfrac{\partial v}{r\partial\theta} \\ \qquad +\dfrac{2u+v\cdot\cot\theta}{r}=0 \\ \dfrac{\partial u}{\partial t}+u\cdot\dfrac{\partial u}{\partial r}+\dfrac{v}{r}\cdot\dfrac{\partial u}{\partial\theta}-\dfrac{v^2}{r}+\dfrac{p}{\rho}\cdot\dfrac{\partial P}{\partial r}=0 \\ \dfrac{\partial v}{\partial t}+u\cdot\dfrac{\partial v}{\partial r}+\dfrac{v}{r}\cdot\dfrac{\partial v}{\partial\theta}+\dfrac{u\cdot v}{r}+\dfrac{p}{r\rho}\cdot\dfrac{\partial P}{\partial\theta}=0 \\ \dfrac{\partial S}{\partial t}+u\cdot\dfrac{\partial S}{\partial r}+\dfrac{v}{r}\cdot\dfrac{\partial S}{\partial\theta}=0 \end{cases} \tag{12-91}$$

其中

$R=\ln\rho$，$P=\ln p$，$S=P-\kappa R$，$\kappa=1.4$；u、v是速度分量，ρ是密度，p是压强。

物体的物面方程为

$$r-b(\theta)=0 \tag{12-92}$$

物体的脱体激波方程为

$$r - SH(\theta, t) = 0 \tag{12-93}$$

方程（12-91）的计算区域如下图所示：

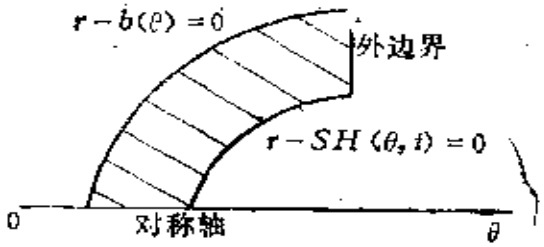


图12-4 曲边扇形面计算区域图

为了给钝体后身计算提供精确的初值，选取$\theta_M = \frac{\pi}{2} - 2 \cdot \Delta\theta$，其中$\Delta\theta$是钝头附近计算$\theta$方向的等分间隔。

为简化计算区域，坐标变换如下：

$$\begin{cases} x = \dfrac{r - b(\theta)}{SH(\theta, t) - b(\theta)} \\ y = \pi - \theta \\ t = t \end{cases}$$

变换后的矩形计算区域如下：

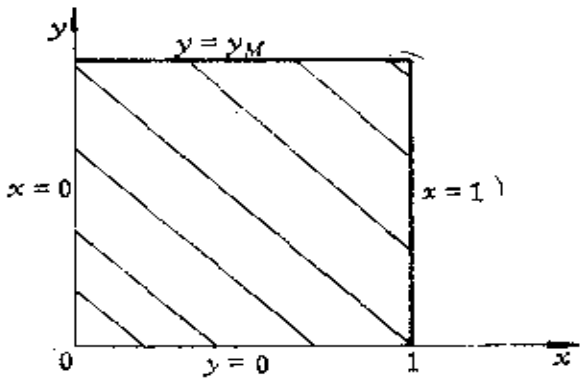


图12-5 矩形计算区域图

其中$x = 0$对应于物面，$x = 1$对应于激波，$y = 0$对应于对称轴，$y = y_M = \frac{\pi}{2} + 2 \cdot \Delta y$对应于外边界。

经过坐标变换后，方程（12-91）变成

$$
\begin{cases}
\dfrac{\partial R}{\partial t}+A\cdot\dfrac{\partial R}{\partial y}+F\cdot\dfrac{\partial R}{\partial x}+B\cdot\dfrac{\partial u}{\partial x}+D\cdot\dfrac{\partial v}{\partial y} \\
\qquad +E\cdot\dfrac{\partial v}{\partial x}+H=0 \\
\dfrac{\partial u}{\partial t}+A\cdot\dfrac{\partial u}{\partial y}+F\cdot\dfrac{\partial u}{\partial x}+K\cdot\dfrac{\partial P}{\partial x}+I=0 \\
\dfrac{\partial v}{\partial t}+A\cdot\dfrac{\partial v}{\partial y}+F\cdot\dfrac{\partial v}{\partial x}+J\cdot\dfrac{\partial P}{\partial y} \\
\qquad +L\cdot\dfrac{\partial P}{\partial x}+M=0 \\
\dfrac{\partial S}{\partial t}+A\cdot\dfrac{\partial S}{\partial y}+F\cdot\dfrac{\partial S}{\partial x}=0
\end{cases}
$$

$$0<x<1,\quad 0<y<y_M \tag{12-94}$$

其中

$$
\begin{cases}
A=v\cdot D,\quad D=-\dfrac{1}{r},\quad r=b+x\cdot\delta, \\
\delta=SH(\theta,\ t)-b(\theta) \\
F=B\left(u-x\cdot\dfrac{\partial SH}{\partial t}-A\cdot C\right),\quad B=\dfrac{1}{\delta}, \\
C=(x-1)\dfrac{\partial b}{\partial\theta}-x\cdot\dfrac{\partial SH}{\partial\theta} \\
E=-B\cdot C\cdot D,\quad H=D(v\cdot\operatorname{ctg}y-2u), \\
K=G\cdot\dot{B} \\
G=e^{P-R},\quad I=D\cdot v^2,\quad J=G\cdot D, \\
L=G\cdot E,\quad M=-A\cdot v
\end{cases}
\tag{12-95}
$$

将 x 方向线段〔0，1〕l 等分，即步长 $\Delta x=1/l$；将 y 方向线段〔0，y_M〕$m+2$ 等分，即步长 $\Delta y=\pi/2m$。这样网格划分后，在 x 方向为 $l+1$ 个点和在 y 方向为 $m+3$ 个点。

把 y 方向的分点数作为向量长度，这样整个计算区域可以分为一个激波向量、一个物面向量和 $l-1$ 个内点向量。由于在微

波上（$x=1$）要满足激波前后关系式，在物面上（$x=0$）要满足物面附着流条件，因此，内点、激波点、和物面点的计算是不同的，下面将分别予以介绍。

12.6.2 内点的并行计算

对于方程组（12-94）采用拉克斯-文卓夫格式

$$f(t^n+\Delta t)=f(t^n)+\Delta t\left(\frac{\partial f}{\partial t}\right)\Big|_{t=t^n}+\frac{\Delta t^2}{2}\left(\frac{\partial^2 f}{\partial t^2}\right)\Big|_{t=t^n}$$

其中 f 分别表示各流动元 R、u、v、S，$\left(\frac{\partial f}{\partial t}\right)\Big|_{t=t^n}$ 和 $\left(\frac{\partial^2 f}{\partial t^2}\right)\Big|_{t=t^n}$ 分别是 f 关于 t 的一阶和二阶偏导数在 $t=t^n$ 处的值。

显然，$\frac{\partial f}{\partial t}$ 可以由方程（12-94）直接移项得到。而 $\frac{\partial^2 f}{\partial t^2}$ 必须由方程（12-94）分别对变量 t，x，y 求偏导数才能得到。

$$\left.\begin{aligned}
\frac{\partial^2 R}{\partial t\partial\alpha}=&-\left(\frac{\partial A}{\partial\alpha}\cdot\frac{\partial R}{\partial y}+A\frac{\partial^2 R}{\partial y\partial\alpha}\right)-\left(\frac{\partial F}{\partial\alpha}\cdot\frac{\partial R}{\partial x}\right.\\
&\left.+F\cdot\frac{\partial^2 R}{\partial x\partial\alpha}\right)-\left(\frac{\partial B}{\partial\alpha}\cdot\frac{\partial u}{\partial x}+B\frac{\partial^2 u}{\partial x\partial\alpha}\right)\\
&-\left(\frac{\partial D}{\partial\alpha}\cdot\frac{\partial v}{\partial y}+D\frac{\partial^2 v}{\partial y\partial\alpha}\right)\\
&-\left(\frac{\partial E}{\partial\alpha}\cdot\frac{\partial v}{\partial x}+E\cdot\frac{\partial^2 v}{\partial x\partial\alpha}\right)-\frac{\partial H}{\partial\alpha}\\
\frac{\partial^2 u}{\partial t\partial\alpha}=&-\left(\frac{\partial A}{\partial\alpha}\cdot\frac{\partial u}{\partial y}+A\cdot\frac{\partial^2 u}{\partial y\partial\alpha}\right)\\
&-\left(\frac{\partial F}{\partial\alpha}\cdot\frac{\partial u}{\partial x}+F\cdot\frac{\partial^2 u}{\partial x\partial\alpha}\right)\\
&-\left(\frac{\partial K}{\partial\alpha}\cdot\frac{\partial P}{\partial x}+K\cdot\frac{\partial^2 P}{\partial x\partial\alpha}\right)-\frac{\partial I}{\partial\alpha}\\
\frac{\partial^2 v}{\partial t\partial\alpha}=&-\left(\frac{\partial A}{\partial\alpha}\cdot\frac{\partial v}{\partial y}+A\cdot\frac{\partial^2 v}{\partial y\partial\alpha}\right)
\end{aligned}\right\}$$

$$-\left(\frac{\partial F}{\partial \alpha}\cdot\frac{\partial v}{\partial x}+F\cdot\frac{\partial^2 v}{\partial x\partial\alpha}\right)$$
$$-\left(\frac{\partial J}{\partial \alpha}\cdot\frac{\partial P}{\partial y}+J\cdot\frac{\partial^2 P}{\partial y\partial\alpha}\right)$$
$$-\left(\frac{\partial L}{\partial \alpha}\cdot\frac{\partial P}{\partial x}+L\cdot\frac{\partial^2 P}{\partial x\partial\alpha}\right)-\frac{\partial M}{\partial\alpha}$$
$$\frac{\partial^2 S}{\partial t\partial\alpha}=-\left(\frac{\partial A}{\partial\alpha}\cdot\frac{\partial S}{\partial y}+A\cdot\frac{\partial^2 S}{\partial y\partial\alpha}\right)$$
$$-\left(\frac{\partial F}{\partial\alpha}\cdot\frac{\partial S}{\partial x}+F\frac{\partial^2 S}{\partial x\partial\alpha}\right)$$

(12-96)

其中 α 分别表示 t 、 x 、 y 。

对于 $\frac{\partial^2 f}{\partial x^2}$ 、 $\frac{\partial^2 f}{\partial y^2}$ 、 $\frac{\partial^2 f}{\partial x\partial y}$ 、 $\frac{\partial f}{\partial x}$ 和 $\frac{\partial f}{\partial y}$ （ $f=R$ 、 u 、 v 、 S ）等项都使用具有二阶精度的差分公式逼近。

拉克斯-文卓夫格式是适合于并行计算的，具有二阶精度的格式，且满足 $C-F$ 条件

$$\Delta t\leqslant 0.667\cdot\frac{1}{V+a}\min(\Delta r,\ r\cdot\Delta\theta)$$

时，格式稳定，其中

$$V=\sqrt{u^2+v^2},\quad a=\sqrt{\kappa\frac{p}{\rho}}$$

下面，介绍另一种二阶三层格式

$$f(t^n+\Delta t)=f(t^n)+\Delta t\cdot\left(\frac{\partial f}{\partial t}\right)\Big|_{t=t^n}+\frac{\Delta t^2}{2}$$
$$\cdot\frac{\left(\frac{\partial f}{\partial t}\right)\Big|_{t=t^n}-\left(\frac{\partial f}{\partial t}\right)\Big|_{t=t_n-\Delta t}}{\Delta t_{n-1}}$$
$$+e_x\cdot\Delta x^2\cdot\left(\frac{\partial^2 f}{\partial x^2}\right)\Big|_{t=t^n}+e_y\cdot\Delta y^2\cdot\left(\frac{\partial^2 f}{\partial y^2}\right)\Big|_{t=t^n}$$

其中 Δt_{n-1} 是上个时间步长，ε_x 和 ε_y 是人为粘性系数，$\varepsilon_x \cdot \Delta x^2 \cdot \left(\frac{\partial^2 f}{\partial x^2}\right)\Big|_{t=t^n} + \varepsilon_y \cdot \Delta y^2 \cdot \left(\frac{\partial^2 f}{\partial y^2}\right)\Big|_{t=t^n}$ 是引入的人为粘性项。

二阶三层格式的稳定条件是

$$\Delta t \leqslant 0.33 \cdot \frac{1}{V+a} \min(\Delta r,\ r \cdot \Delta\theta)$$

比较这二种计算格式，其稳定性基本上一样，拉克斯-文卓夫格式稍好一点。计算量方面，二阶三层格式的计算量约比拉克斯-文卓夫格式减少 50%，在 YH 机上，二阶三层格式的计算速度提高 40%；在内存需要方面，拉克斯-文卓夫格式只需要二个时间层的数据，而二阶三层格式需要三个时间层的数据。对于不同类型和特点的计算机，二种格式的计算效果可能不同，所以要根据计算机的特点选择格式。

12.6.3 激波点的并行计算

激波点的计算是轴对称钝体绕流计算的关键。这里主要介绍摩提（Mortti）的特征线方法。

物体的激波方程为

$$r - SH(\theta,\ t) = 0$$

假设 $t^{n+1} = t^n + \Delta t$ 时刻，激波点 Q 的极坐标为（r，θ），那么点 Q 处激波的单位外法线向量为

$$\boldsymbol{N} = \frac{1}{\nu}\left(1,\ -\frac{\partial SH}{r\partial\theta}\right)$$

其中

$$\nu = \sqrt{1 + \left(\frac{\partial SH}{SH \cdot \partial\theta}\right)^2}$$

在 Q 点处建立直角坐标系（ξ，η），其中 ξ 方向即为 $\boldsymbol{N}$ 方向，而 η 方向的单位向量为

$$\tau = \frac{1}{\nu}\left(\frac{-\partial SH}{SH \cdot \partial\theta},\ -1\right)$$

[illegible]点 A 的极坐标为 (r_A, θ_A)，在 Q 为原点的直[illegible]坐标系中的坐标为 (ξ, η)。满足如下关系式：

$$[illegible]=\xi^2+L_2^2$$

$$[illegible]=\frac{\pi}{2}-\sin^{-1}\left(-\frac{L_1\cos y+L_2\sin y}{r_A}\right)$$

其中

$$L_1=SH+\frac{\xi}{v}-\frac{\eta}{SH}\cdot\frac{\partial SH}{\partial\theta}$$

$$L_2=-\frac{1}{v}\left(\eta+\frac{\xi}{SH}\cdot\frac{\partial SH}{\partial\theta}\right)$$

假设在极坐标系中的速度分量为 u、v 和在直角坐标系中的速度分量为 $\tilde{u}$，$\tilde{v}$。满足如下关系式

$$\begin{cases}\tilde{u}=\dfrac{1}{v}(A_1\cdot u+A_2\cdot v)\\ \tilde{v}=\dfrac{1}{v}(B_1\cdot u+B_2\cdot v)\end{cases}$$

其中

$$\begin{cases}A_1=\sigma_1\cdot\cos\theta+\sigma_2\cdot\sin\theta,\quad A_2=-\sigma_1\sin\theta+\sigma_2\cdot\cos\theta\\ B_1=-\sigma_2\cdot\cos\theta+\sigma_1\cdot\sin\theta,\quad B_2=\sigma_2\cdot\sin\theta+\sigma_1\cdot\cos\theta\\ \sigma_1=\dfrac{\sin y}{SH}\cdot\dfrac{\partial SH}{\partial\theta}-\cos y,\quad \sigma_2=\dfrac{\cos y}{SH}\cdot\dfrac{\partial SH}{\partial\theta}+\sin y\end{cases}$$

在激波点 Q 处建立了直角坐标系（ξ，η），它的运动方程为

$$\begin{cases}\dfrac{\partial R}{\partial t}+\tilde{u}\cdot\dfrac{\partial R}{\partial\xi}+\tilde{v}\cdot\dfrac{\partial R}{\partial\eta}+\dfrac{\partial\tilde{u}}{\partial\xi}+\dfrac{\partial\tilde{v}}{\partial\eta}=0\\ \dfrac{\partial\tilde{u}}{\partial t}+\tilde{u}\cdot\dfrac{\partial\tilde{u}}{\partial\xi}+\tilde{v}\cdot\dfrac{\partial\tilde{u}}{\partial\eta}+\dfrac{p}{\rho}\cdot\dfrac{\partial S}{\partial\xi}+a^2\cdot\dfrac{\partial R}{\partial\xi}=0\\ \dfrac{\partial\tilde{v}}{\partial t}+\tilde{u}\cdot\dfrac{\partial\tilde{v}}{\partial\xi}+\tilde{v}\cdot\dfrac{\partial\tilde{v}}{\partial\eta}+\dfrac{p}{\rho}\cdot\dfrac{\partial S}{\partial\eta}+a^2\cdot\dfrac{\partial R}{\partial\eta}=0\\ \dfrac{\partial S}{\partial t}+\tilde{u}\cdot\dfrac{\partial S}{\partial\xi}+[illegible]=\dfrac{\partial S}{\partial\eta}=0\end{cases}$$

(12-97)

其中 a 是音速。

从方程（12-97）可得二条特征线

$$\frac{d\xi}{dt}=\tilde{u}\pm a \tag{12-98}$$

和特征关系式

$$\frac{d\tilde{u}}{dt}\pm\frac{a}{\kappa}\cdot\frac{dP}{dt}=\mp a\left(\frac{\partial\tilde{v}}{\partial\eta}+\tilde{\omega}\right) \tag{12-99}$$

其中

$$P=S+\kappa\cdot R$$

$$\kappa=1\cdot 4$$

$$\tilde{\omega}=\begin{cases}\Omega_A, & \text{对称轴上}\\ \dfrac{1}{r\cdot\sin\theta}(u\cdot\sin\theta+v\cdot\cos\theta), & \text{非对称轴上}\end{cases}$$

由激波前后关系式，可得激波后流动元 u、v、R、S 的计算公式

$$\begin{cases} u=\dfrac{1}{v^2}\left(\dfrac{A}{SH}\cdot\dfrac{\partial SH}{\partial\theta}+C\right)\\ v=-\dfrac{u}{SH}\cdot\dfrac{\partial SH}{\partial\theta}+A\\ \rho=\dfrac{(\kappa+1)V_{n,\infty}^2}{(\kappa-1)V_{n,\infty}^2+2\kappa}\\ p=\dfrac{2V_{n,\infty}^2}{\kappa+1}-\dfrac{\kappa-1}{\kappa+1}\end{cases} \tag{12-100}$$

其中

$$A=v_\infty+\frac{u_\infty}{SH}\cdot\frac{\partial SH}{\partial\theta}$$

$$C=\frac{\partial SH}{\partial t}+\frac{v\cdot V_{n,\infty}}{\rho}$$

$$V_{n,\infty}=\frac{1}{v}\left(u_\infty-\frac{\partial SH}{\partial t}-\frac{v_\infty}{SH}\cdot\frac{\partial SH}{\partial\theta}\right)$$

$$u_-=-\sqrt{\kappa}\cdot M_-\cdot\cos y$$

$$v_-=-\sqrt{\kappa}\cdot M_-\cdot\sin y$$

M_∞ 为来流马赫数，u_∞、v_∞ 为激波前的速度分量，$V_{n\infty}$ 是激波前法向速度。

$x=1$ 上对应的 $m+3$ 个网格点组成了一个激波点向量 $\boldsymbol{Q}$。下面计算步骤中没有特别说明的向量都是该激波点向量相应的向量表示。

关于激波点向量 $\boldsymbol{Q}$ 的计算步骤如下：

1．并行计算 $t^n+\Delta t$ 时刻，激波点向量 $\boldsymbol{Q}$ 的激波位置：

$$\boldsymbol{SH}^{(n+1)}=\boldsymbol{SH}^{(n)}+\Delta t\left(\frac{\partial \boldsymbol{SH}}{\partial t}\right)\Big|_{t=t^n}+\frac{\Delta t^2}{2}\cdot\left(\frac{\partial^2 \boldsymbol{SH}}{\partial t^2}\right)_{t=t^n}$$

2．并行计算激波位置对 y 方向的导数。

3．假设激波移动速度 $\boldsymbol{SH}_t^{(n+1)}$。

4．根据计算公式（12-100），并行计算激波后，激波点向量 $\boldsymbol{Q}$ 对应的流动元向量 $\boldsymbol{u}$、$\boldsymbol{v}$、$\boldsymbol{R}$、$\boldsymbol{S}$。

5．通过激波点向量 $\boldsymbol{Q}$ 在（ξ，t）平面上作特征线向量 $\frac{d\xi}{dt}=\boldsymbol{u}+\boldsymbol{a}$，并假定交（$\xi$，$t$）平面上的点向量 $\boldsymbol{A}$。利用平均法反复迭代求出精确的交点向量 $\boldsymbol{A}$ 的坐标和流动参数向量。

6．将特征线二端点向量 $\boldsymbol{Q}$ 和 $\boldsymbol{A}$ 的流动元向量代入特征关系式（12-99），并并行求出新的激波移动速度向量。

7．判别新、旧激波移动速度是否一致，若不一致，需要重新调整激波移动速度，然后，重复计算第 4 至 6 步，直到满足特征关系式为止。

12.6.4 物面点的并行计算

无粘性理想气体条件下，物面是等熵的，即 $\frac{\partial S}{\partial t}=0$，所以物面点的计算归为求解流动元 u、v、R。

物体的物面方程为

$$r-b(\theta)=0$$

假设物面点Q的极坐标为（r、θ），那么Q处的单位内法线向量为

$$N=\frac{1}{\nu}\left(-1,\ \frac{\partial b}{b\cdot\partial\theta}\right)$$

其中

$$\nu=\sqrt{1+\left(\frac{\partial b}{b\cdot\partial\theta}\right)^2}$$

在Q点处建立直角坐标系（ξ，η），其中ξ方向为N方向，η方向的单位向量为

$$\tau=\frac{1}{\nu}\left(\frac{\partial b}{b\cdot\partial\theta},\ 1\right)$$

假设空间任一点A的极坐标为（r_A，θ_A），直角坐标为（ξ，η），满足关系式

$$\begin{cases} r_A=\sqrt{L_1^2+L_2^2} \\ \theta_A=\dfrac{\pi}{2}-\sin^{-1}\left(-\dfrac{L_1\cos\gamma+L_2\sin\gamma}{r_A}\right) \end{cases}$$

其中

$$L_1=b-\frac{\eta}{\nu\cdot b}\cdot\frac{\partial b}{\partial\gamma}-\frac{\xi}{\nu}$$

$$L_2=\frac{\eta}{\nu}-\frac{\xi}{\nu\cdot b}\cdot\frac{\partial b}{\partial\gamma}$$

假设在极坐标系中的速度分量为u、v，在直角坐标系中的速度分量为$\tilde{u}$，$\tilde{v}$。满足关系式：

$$\tilde{u}=\frac{1}{\nu}\left(-u-\frac{v}{b}\cdot\frac{\partial b}{\partial\gamma}\right)$$

$$\tilde{v}=\frac{1}{\nu}\left(v-\frac{u}{b}\cdot\frac{\partial b}{\partial\gamma}\right)$$

由物面附着流动条件，可知$\tilde{u}=0$，所以有

$$\begin{cases} u=-\dfrac{v}{b}\cdot\dfrac{\partial b}{\partial\gamma} \\ v=-\dfrac{1}{\mu}\tilde{v} \end{cases}$$

由于物面等熵和物面附着流条件，在直角坐标系（ξ，η）上的运动方程为

$$\frac{\partial \tilde{v}}{\partial t}+\tilde{v}\cdot\frac{\partial \tilde{v}}{\partial \eta}+\frac{p}{\rho}\cdot\frac{\partial P}{\partial \eta}=0 \qquad (12\text{-}101)$$

其中

$$\frac{\partial \tilde{v}}{\partial \eta}=-\frac{1}{v^2\cdot b}\left(\frac{\partial v}{\partial y}-\frac{1}{b}\cdot\frac{\partial b}{\partial y}\cdot\frac{\partial u}{\partial y}\right)$$

$$\frac{\partial P}{\partial \eta}=-\frac{\kappa}{v\cdot b}\cdot\frac{\partial R}{\partial y}$$

关于方程（12-101）可以采用拉克斯-文卓夫格式并行计算物面点向量的速度分向量u和v（向量长为$m+3$）。

关于物面点向量的流动元向量R也采用特征线方法并行计算。

12.6.5 并行计算的效率

上面给出了内点、激波点和物面点，这三类计算点的并行格式。整个轴对称钝体绕流的计算基本上实现了并行化。

以（x，y）的网格点规模为 8 ×38为例，在零攻角轴对称钝体绕流计算钝锥外形中，计算一个M马赫数，一般稳定用步数为1200步左右。在 25 万次至 30 万次/每秒的计算机上用手编程序大约需要 50 分钟；在YH机上，使用并行算法，用汇编程序大约需要13秒钟。在有攻角的情况下，计算一个马赫数，一个攻角的稳定步数是1800左右。在百万次/每秒的计算机上用手编程序大约需要 5 小时；在 YH 机上，使用并行算法，用汇编程序大约需要 6 分钟左右。

如果网格点规模更大的话，整个问题的并行计算速度将更快。

第十三章　蒙特卡罗方法及其并行计算

40年代诞生的蒙特卡罗（Monte Carlo）方法（MC方法），借助数字计算机可以实现它所要求的大量的随机抽样试验，因而能够广泛地应用于模拟数学物理、工程技术、生产管理等各个领域中的实际问题。本章将简要介绍这一方法并讨论有关问题的并行计算。

13.1　蒙特卡罗方法概述

13.1.1　基本内容

利用MC方法解决一个给定的问题时，基本想法是首先建立与该问题有相似性的概率统计模型，利用这种相似性使得概率统计模型的某些特征等于给定问题的解，然后对模型进行随机模拟或统计抽样，再利用所得结果求出这些特征的统计估计值作为给定问题的近似解。解的精确度可用估计值的标准误差表示。

MC方法可以解决各种确定性和随机性问题，在求解过程中一般包含如下几个内容：

（1）对求解的问题建立简单而又便于实现的概率统计模型，使所求的解恰好是所建立模型的概率分布或数学期望。

（2）根据概率统计模型的特点，设计和使用降低方差和费用的各种方法，加速模拟结果的收敛。

（3）建立对随机变量的抽样方法，其中包括建立产生伪随机数的方法。

（4）给出获得所求解的统计估计值及其方差或标准误差的方法。

13.1.2 蒙特卡罗方法的特点

MC方法的一个突出的表现是结果精度和收敛过程具有一些独有的特点。

通常，MC方法是用某个随机变量X的简单子样X_1，X_2，…，X_N的算术平均值

$$\bar{X}_N=\frac{1}{N}\sum_{n=1}^{N}X_n$$

作为所求解I的近似值。由柯尔莫哥罗夫强大数定理可知，当数学期望$E(X)=I$时，算术平均值$\bar{X}_N$以概率1收敛到I，即

$$P\left(\lim_{N\to\infty}\bar{X}_N=I\right)=1$$

按照中心极限定理，对于任何$\lambda_\alpha>0$有

$$P\left(|\bar{X}_N-I|<\frac{\lambda_\alpha\sigma}{\sqrt{N}}\right)\approx\frac{2}{\sqrt{2\pi}}\int_0^{\lambda_\alpha}e^{-\frac{1}{2}t^2}dt=1-\alpha$$

式中的λ_α称为正态差，α称为显著水平，$1-\alpha$称为置信水平，λ_α与α是一一对应的，σ为随机变量X的标准差。由此可知，不等式

$$|\bar{X}_N-I|<\frac{\lambda_\alpha\sigma}{\sqrt{N}}$$

近似地以概率$1-\alpha$成立，这表明，$\bar{X}_N$收敛到I的速度的阶为$O(1/\sqrt{N})$。

若$\sigma\neq 0$，则MC方法的误差ε为

$$\varepsilon=\frac{\lambda_\alpha\sigma}{\sqrt{N}}\qquad(13\text{-}1)$$

这表明，MC方法的误差ε是由σ和$\sqrt{N}$决定的。在固定σ的情况下，要想将精确度提高一位数字，就要增加100倍的工作量。反之，在固定误差ε和抽样产生一个X的平均费用C不变情况下，如果σ减少10倍，则可减少100倍的工作量。若费用C是随

着方法的改变而改变时，由于$N=(\lambda_\alpha\sigma/\varepsilon)^2$，$NC=(\lambda_\alpha/\varepsilon)^2\sigma^2C$，因此MC方法的效率是与$\sigma^2C$成正比的。这说明提高MC方法的效率，既不是简单地增加抽样数N，也不是简单地减少标准差σ，而应该是在减小标准差的同时兼顾考虑费用大小，使方差σ^2与费用C的乘积尽量小。

从上述的讨论中，很容易看出MC方法的一些优点以及它与一般数值方法的不同点

（1）MC方法及其程序结构简单

例如用平均值方法计算定积分

$$I=\int_0^1 g(x)dx$$

的数值。其计算步骤为

① 产生均匀分布在〔0，1〕上的随机数r_n；

② 计算$g(r_n)$；

③ 用平均值$\bar{I}=\frac{1}{N}\sum_{n=1}^{N}g(r_n)$作为$I$的近似值

这个例子体现了MC方法计算积分是通过大量的简单的重复抽样实现的。

（2）收敛的概率性和收敛速度与问题维数无关。

对于MC方法，虽然不能断言其误差不超过某个值，但能指出其误差以接近1的概率不超过某个界限。

MC方法的收敛速度与一般数值方法相比是很慢的，其主阶仅为$O(1/\sqrt{N})$。因此，MC方法不宜于求解精度要求很高的问题。

由式（13-1）可见，MC方法的误差ε只与标准差σ和样本容量N有关，而与样本中元素所在空间无关，即MC方法的收敛速度与问题维数无关。这决定了MC方法对多维问题的适用性。

（3）MC方法的适应性强

MC方法的广泛适应性是不可忽视的，在解题时受问题条件限制的影响较小。例如要计算s维空间中任意一个区域D_s上的积分

$$I=\int_{D_s}\cdots\int g(x_1,\ x_2,\ \cdots,\ x_s)dx_1dx_2\cdots dx_s$$

无论D_s的形状如何特殊，只要能给出描述D_s的几何条件，就总可以用平均值方法给出I的近似估计值

$$\bar{I}=\frac{D_s}{N}\sum_{n=1}^{N}g(x_1^{(n)},\ x_2^{(n)},\ \cdots,\ x_s^{(n)})$$

其中点$(x_1^{(n)},\ x_2^{(n)},\ \cdots,\ x_s^{(n)})\in D_s$，且均匀分布在$D_s$上。

13.1.3 随机变量抽样

用MC方法模拟实际过程时，需要产生各种概率分布的随机变量的抽样值。随机变量抽样指的就是在已知分布的总体中产生简单子样。令$F(x)$表示已知分布，ξ_1，ξ_2，…，ξ_N表示在总体$F(x)$中产生的容量为N的简单子样。按照简单子样的定义，ξ_1，ξ_2，…，ξ_N相互独立，具有相同的分布$F(x)$。通常把〔0，1〕上均匀分布随机变量的抽样值称为随机数。其他分布随机变量的抽样都是借助于随机数来实现的。因此，随机数是随机抽样的基本工具，能否高效地产生随机数是MC方法应用好坏的一个重要标志。

产生随机数的方法，可分为物理方法和数学方法两种类型。有一些并行处理文献，为了解决产生随机数的问题，设计了相应的硬件结构；还有一些并行处理文献仍然使用串行的数学方法产生随机数。

用数学方法产生的随机数并不是严格意义上的随机数，一般称之为伪随机数。伪随机数序列通常是利用递推公式产生，这样到一定长度之后，该序列就会出现退化为零或重复发生的周期现象。产生伪随机数的主要方法有：平方取中法、移位指令加法、同余法。从随机性、在计算机上实现的难易、时间开销和伪随机数的周期等几点来考虑，乘同余法具有比较好的特性。这种方法及其并行计算将在第二节予以介绍。

尽管产生随机数就是在均匀分布总体中产生简单子样，也就是说产生随机数是属于随机变量抽样的一个特殊情况。但就产生方法而言，随机数的产生与一般意义上的随机变量的抽样有原则上的差别。产生伪随机数的方法是半经验性的，近似的，而随机变量抽样一般是有严格的理论根据，只要所用的随机数序列满足均匀且相互独立的要求，那么由它所产生的已知分布的简单子样，也具有相同的总体分布且相互独立。

在计算机上实现随机变量抽样，需要涉及抽样的数学方法及其抽样费用。一般而言，对于某种抽样方法，只要它的费用小，不管它的实现如何复杂，都认为它是一种好的方法。能否设计出高效的并行抽样方法，也是MC方法在并行机上应用好坏的重要标志。有关抽样方法及其并行实现将在第三节至第六节予以讨论。

13.2 乘同余法及并行计算

13.2.1 乘同余法

乘同余法的递推公式为

$$x_{n+1}\equiv\lambda x_n(\mathrm{mod}M)$$

$$n=0,\ 1,\ 2,\ \cdots$$

$$r_n=x_n/M$$

当选取适当的M，λ和x_0时，就可得到随机性好、周期长的伪随机数序列$\{r_n\}$。注意，在实际产生$\{x_n\}$时，x_{n+1}取λx_n被M除后的余数。在今后遇到需要产生具体数值的同余运算，均作类似的理解，不再另行说明。

雷曼（Lehmer）曾取$M=10^8+1$，$\lambda=23$,$x_0=47594118$，得到8位数的伪随机序列，周期为5，882，352。经对5000个这样的数所作的随机性检验，认为结果是好的。

徐钟济等人曾考虑，$M=2^t$，x_0随机地选取一个$4q+1$型的数，λ取成5^{2k+1}型的正整数，其中q为任意整数，k为使5^{2k+1}

在计算机上所能容纳的最大奇数，s 为计算机的字长。一般地有

$$\lambda=5^{2k+1},\begin{cases}x_0=1，得\ x_n=1，5，9，\cdots，4q+1，\cdots，2^r-3\\ x_0=3，得\ x_n=3，7，\cdots，4q+3，\cdots，2^r-1\end{cases}$$

$$\lambda=3^{2k+1},\begin{cases}x_0=1，得x_n=1，3，9，\cdots，8q+1，\\ \qquad\qquad 8q+3，\cdots，2^r-7，2^r-5\\ x_0=3，得x_n=5，7，13，\cdots，8q+5，\\ \qquad\qquad 8q+7，\cdots，2^r-3，2^r-1\end{cases}$$

依此曾得到二百万个伪随机数。经几种方法检验，认为在计算机上采用同余法是适宜的。

关于乘同余法的周期，考虑（λ，M）$=1$。由递推公式

$$x_{n+1}\equiv\lambda x_n(\mathrm{mod}M)$$

可知，当$x_{n+l}\equiv\lambda^n x_i=x_i(\mathrm{mod}M)$，即$\lambda^n\equiv1(\mathrm{mod}M)$ 时，伪随机数序列发生重复。因此，对于任意初值x_i，周期P不可能大于满足同余式$\lambda^n\equiv1(\mathrm{mod}M)$ 的最小正整数n，即λ^n-1能被M整除的最小正整数n。

由周期的定义，知1，λ，λ^2，$\cdots$，λ^{p-1}对于模M两两互不同余。事实上，若$\lambda^l\equiv\lambda^k(\mathrm{mod}M)$，$0<l$，$k<n$，即$\lambda^{l-k}\equiv1(\mathrm{mod}M)$，$0<l-k<n$，矛盾。

由欧拉定理，设M为任一自然数，λ为与M互素的任意整数，则$\lambda^{\varphi(M)}-1$可为M整除，即$\lambda^{\varphi(M)}\equiv1(\mathrm{mod}M)$。在计算机上，一般取$M=2^s$，$\varphi(M)=\varphi(2^s)=\frac{1}{2}\cdot2^s$，即不大于$M$且与$M$互素的正整数个数为$2^{s-1}$（$2^{s-1}$个奇数）。因此，必有周期$P=2^r$，$0<r\leqslant s-1$。

例如若取$\lambda=5^{2k+1}$，$M=2^s$，则周期为2^{s-2}。因为$5^2\equiv1(\mathrm{mod}8)$，$5^3\equiv5(\mathrm{mod}8)$，于是$5^{2k+1}\equiv5(\mathrm{mod}8)$为$8q+5$型的数。

又因$\lambda^{2r}-1=(1+8q+4)^{2r}-1$

$$=(1+4(2q+1))^{2r}-1$$

$$
\begin{aligned}
&= 2^r 4(2q+1) + \frac{2^r(2^r-1)}{2!}4^2(2q+1)^2 + \cdots \\
&\quad + \frac{2^r(2^r-1)(2^r-2)\cdots(2^r-m+1)}{m!} \\
&\quad \cdot 4^m(2q+1)^m + \cdots + 4^{2r}(2q+1)^{2r} \\
&= 2^{r+2}\left[(2q+1) + \frac{2r-1}{2!}4(2q+1)^2\right. \\
&\quad + \cdots + \frac{(2^r-1)\cdots(2^r-m+1)}{m!} \\
&\quad \cdot 4^{m-1}(2q+1)^m + \cdots \\
&\quad \left. + \frac{(2^r-1)\cdots(2^r-2^r+1)}{(2^r)!}4^{2r-1}(2q+1)^{2r}\right]
\end{aligned}
$$

若要$M/(\lambda^{2r}-1)$，则应有$r+2=s$，所以$r=s-2$。即当$\lambda = 5^{2k+1}$时，最大周期为2^{r-2}。由此可得乘同余法的伪随机数序列的周期与λ的取值有关，而与初值选取M的哪一个互素数无关。表13-1中列出了产生伪随机数的常用参数及周期。

表13-1　产生伪随机数的常用参数及周期

M	λ	M	λ	x_0	周期
2^{30}	5^{11}	2^{30}	3^{17}		2^{r-2}
$2^{31}\sim 2^{34}$	5^{13}	$2^{31}\sim 2^{33}$	3^{19}		（当$M=2^r$
$2^{35}\sim 2^{39}$	5^{15}	$2^{34}\sim 2^{36}$	3^{21}		时，周期即为
$2^{41}\sim 2^{44}$	5^{17}	$2^{37}\sim 2^{39}$	3^{23}	1或任意奇数	2^{r-2}）
$2^{45}\sim 2^{48}$	5^{19}	$2^{40}\sim 2^{42}$	3^{25}		
		$2^{43}\sim 2^{45}$	3^{27}		
		$2^{46}\sim 2^{48}$	3^{29}		
10^8+1	23			47594118	5882352
10^{11}	7^5			1	10^8
10^{11}	$7^{4k-1}(k\neq 1)$			1	5.10^8
$2^{31}-1$	16807			任意整数	$2^{31}-1$

13.2.2 乘同余法的并行计算

根据同余运算性质

（1）若 $p \equiv q \pmod{M}$，则 $pc \equiv qc \pmod{M}$，其中 c 为任一整数；

（2）若 $p \equiv q \pmod{M}$，$q \equiv c \pmod{M}$，则 $p \equiv c \pmod{M}$。可设计求解递推公式

$$x_{n+1} \equiv \lambda x_n \pmod{M} \tag{13-2}$$

的并行计算格式。

定理13-1 假设序列$\{x_n\}$是由递推公式（13-2）决定的。对任意正整数N，令

$$a \equiv \lambda^N \pmod{M} \tag{13-3}$$

$$\mathbf{X}_1 = (x_1, x_2, \cdots, x_N)^T$$

$$\mathbf{X}_2 = (x_{N+1}, x_{N+2}, \cdots, x_{2N})^T$$

则成立关系式

$$\mathbf{X}_2 \equiv a * \mathbf{X}_1 \pmod{M} \tag{13-4}$$

现就$N=2$的情形给出证明。至于一般情形可递推得出或用数学归纳法加以证明。

由式（13-2），知

$$x_2 \equiv \lambda x_1 \pmod{M} \tag{13-5}$$

$$x_3 \equiv \lambda x_2 \pmod{M} \tag{13-6}$$

$$x_4 \equiv \lambda x_3 \pmod{M}$$

由同余运算性质（1），式（13-5）可化为

$$\lambda x_2 \equiv \lambda^2 x_1 \pmod{M} \tag{13-7}$$

再由同余运算性质（2）和式（13-7），式（13-6）可化为

$$x_3 \equiv \lambda^2 x_1 \pmod{M}$$

同理可得

$$x_4 \equiv \lambda^2 x_2 \pmod{M}$$

下面可由式（13-4）推出与此一致的结论。

就$N=2$的情形，展开式（13-4）

$$\begin{pmatrix} x_3 \\ x_4 \end{pmatrix} \equiv a * \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} (\mathrm{mod}M)$$

或者

$$x_3 \equiv a x_1 (\mathrm{mod}M) \tag{13-8}$$

$$x_4 \equiv a x_2 (\mathrm{mod}M) \tag{13-9}$$

由同余运算性质（1），式（13-3）可化为：

$$a x_1 \equiv \lambda^2 x_1 (\mathrm{mod}M) \tag{13-10}$$

再由同余运算性质（2）和式（13-10），式（13-8）可化为

$$x_3 \equiv \lambda^2 x_1 (\mathrm{mod}M)$$

同理可得

$$x_4 \equiv \lambda^2 x_2 (\mathrm{mod}M)$$

于是定理得证。

利用定理13-1并行计算$\{x_n\}$，可分为两步进行。

第一步，计算x_1，x_2，…，x_P，其中$P=2^m$为并行台数或等效并行台数。取

$\boldsymbol{X}_1=(x_1,\ x_2)^T$，$a_1\equiv\lambda^2(\mathrm{mod}M)$，$\boldsymbol{Y}_1=(x_3,\ x_4)^T$，

$\boldsymbol{X}_2=(x_1,\ x_2,\ x_3,\ x_4)^T$，$a_2\equiv\lambda^4(\mathrm{mod}M)$，

$\boldsymbol{Y}_2=(x_5,\ x_6,\ x_7,\ x_8)^T$

…

$\boldsymbol{X}_{m-1}=(x_1,\ x_2,\ \cdots,\ x_{P/2})^T$，$a_{m-1}\equiv\lambda^{P/2}(\mathrm{mod}M)$，

$\boldsymbol{Y}_{m-1}=(x_{P/2+1},\ \cdots,\ x_P)^T$，

$\boldsymbol{X}_m=(x_1,\ x_2,\ \cdots,\ x_P)^T$，$a_m\equiv\lambda^P(\mathrm{mod}M)$

则

$$\boldsymbol{Y}_i \equiv a_i * X_i (\mathrm{mod}M)\ i=1,\ 2,\ \cdots,\ m-1 \tag{13-11}$$

式（13-11）给出了计算x_1，x_2，…，x_P的并行递推格式。

第二步，计算x_{P+1}，x_{P+2}，…。这时取

$\boldsymbol{X}_{m+1}=(x_{P+1},\ x_{P+2},\ \cdots,\ x_P)^T$

$\boldsymbol{X}_{m+2}=(x_{2P+1},\ x_{2P+2},\ \cdots,\ x_{3P})^T$

…

$$X_{m+k}=(x_{kP+1},\ x_{kP+2},\ \cdots,\ x_{(k+1)P})^T$$

$\cdots$

则

$$X_{i+1}\equiv a_m * X_i(\mathrm{mod}M)\ i=m,\ m+1,\ \cdots \qquad (13\text{-}12)$$

式（13-12）给出了计算x_{P+1}，x_{P+2}，$\cdots$的并行递推格式。

考虑到这一节所讨论的有关λ的取值，λ^{2^k}有可能发生溢出，于是需要讨论

$$a_k\equiv\lambda^{2^k}(\mathrm{mod}M) \qquad (13\text{-}13)$$

的计算。

若λ^{2^k}不会发生溢出，a_k的计算当然可以通过$\lambda^{2^{k-1}}$自乘之后，再除以M得出。

若λ^{2^k}发生溢出，为计算此时a_k的值，先证明一个同余式

$$\lambda^n(\mathrm{mod}M)\equiv(\lambda^{n-1}(\mathrm{mod}M)\cdot\lambda)(\mathrm{mod}M) \qquad (13\text{-}14)$$

事实上，由同余运算的反身性$p\equiv p(\mathrm{mod}M)$，知

$$\lambda^{n-1}\equiv\lambda^{n-1}(\mathrm{mod}M)\equiv(\lambda^{n-1}(\mathrm{mod}M))(\mathrm{mod}M)$$

再由同余运算性质（1）和（2），即得式（13-14）。

从式（13-14）出发可推出

$$\begin{aligned}a_k&\equiv\lambda^{2^k}(\mathrm{mod}M)\\&\equiv(\cdots(\lambda^{2^{k-1}}(\mathrm{mod}M)\cdot\underbrace{\lambda)(\mathrm{mod}M)\cdots\lambda}_{2^{k-1}\text{重}})(\mathrm{mod}M)\\&\equiv(\cdots(a_{k-1}\cdot\lambda)(\mathrm{mod}M)\cdots\lambda)(\mathrm{mod}M)\end{aligned} \qquad (13\text{-}15)$$

因此，当λ^{2^k}发生溢出时，可利用式（13-15）递推计算a_k的值。

考察上述第一步的计算，由于$a_k(k=1,\ 2,\ \cdots,\ m)$的计算要在这一步进行，因此这一步的计算并没有比从递推公式（13-2）直接计算的方式改进多少。但考虑到第二步的计算需要用到a_m，因此一般而言$a_k(k=1,\ 2,\ \cdots,\ m)$的计算是不可避免的。在这种情况下，使用第一步的并行计算格式显然要比从递推公式（13-2）直接计算为好。

MC方法往往需要产生数以百万计的随机数，而第一步的计

算量相对于第二步的计算量来说是非常小的，因此整个并行计算效率是很高的。

将所求得的X_m，X_{m+1}，…，X_{m+k}，…，分别作一次向量除法运算，即可得伪随机数序列$\{r_n\}$。

13.3 离散型随机变量抽样

13.3.1 抽样方法的一般描述

离散型随机变量一般使用直接抽样方法。对于任意给定的分布函数$F(x)$，直接抽样方法的一般形式如下：

$$\xi_n = \inf_{F(t)\geqslant r_n} t, \quad n=1, 2, \cdots, N \tag{13-16}$$

其中r_1，r_2，…，r_N为随机序列。

为了证明由等式（13-16）所确定的随机变量序列ξ_1，ξ_2，…，ξ_N是由总体分布$F(x)$中产生的简单子样，只须证明它们具有相同分布$F(x)$，且相互独立。事实上，由于如下明显等式

$$P(\xi_n<x)=P(\inf_{F(t)\geqslant r_n} t<x)=P(r_n<F(x))$$

$$=F(x)$$

对于任意n成立。因此，由等式（13-16）所确定的随机变量序列具有相同分布$F(x)$。另外，由于r_1，r_2，…，r_N是相互独立的，由等式（13-16）所确定的函数是波雷尔可测的，因此由它所确定的ξ_1，ξ_2，…，ξ_N也是相互独立的。

考虑离散型分布的一般形式

$$F(x)=\sum_{x_i<x} P_i \tag{13-17}$$

其中x_1，x_2，…，x_k，…为离散型随机变量的跳跃点；P_1，P_2，…，P_k，…为相应的概率。

根据直接抽样方法的一般公式（13-16），可得离散型分布的直接抽样方法如下：

$$\xi_n = x_k, \quad 当 \sum_{i=1}^{k-1} P_i < r_n \leqslant \sum_{i=1}^{k} P_i \tag{13-18}$$

这一结果表明，为了实现任意的离散型分布（13-17）的抽样，直接抽样方法是非常理想的。直接抽样方法是离散型分布随机变量抽样的基本方法。其串行程序框图如图13-1所示。

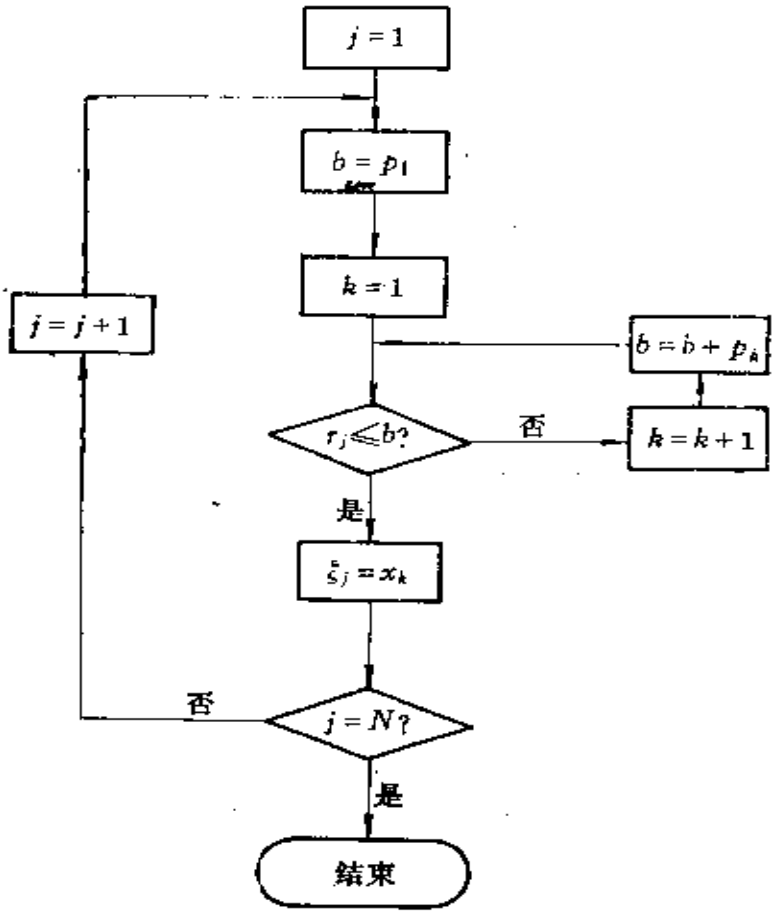


图13-1　离散型随机变量串行抽样框图

13.3.2　抽样方法的并行实现

这里只考虑$N = p$的情形，p为并行台数或等效并行台数，至于$N > p$的情形可分段进行。并行抽样的基本想法是：让p个随机数r_1，r_2，…，r_p，以及要抽样产生的p个简单子样ξ_1，ξ_2，…，ξ_p分别形成向量

$$\xi=\begin{pmatrix}\xi_1\\ \xi_2\\ \vdots\\ \xi_p\end{pmatrix},\quad \boldsymbol{R}=\begin{pmatrix}r_1\\ r_2\\ \vdots\\ r_p\end{pmatrix}$$

然后将$\boldsymbol{R}$与$\sum_{i=1}^{k-1}P_i$和$\sum_{i=1}^{k}P_i$比较。比较方法是：$\boldsymbol{R}$的分量r_1，r_2，…，r_p同时各自与$\sum_{i=1}^{k-1}P_i$和$\sum_{i=1}^{k}P_i$比较。若$\sum_{i=1}^{k-1}P_i<r_j\leqslant\sum_{i=1}^{k}P_i$，则令$\xi_j=x_k$；否则，$\xi_j$保持原值不变，$j=1$，2，…，$P$。这一条件赋值也是以向量的方法并行进行的。条件赋值完成后，再将$\boldsymbol{R}$与$\sum_{i=1}^{k}P_i$和$\sum_{i=1}^{k+1}P_i$比较。若$\sum_{i=1}^{k}P_i<r_j\leqslant\sum_{i=1}^{k+1}P_i$，则令$\xi_j=x_{k+1}$；否则，$\xi_j$保持原值不变，$j=1$，2，…，$P$。继续这一过程，直至向量$\xi$的所有分量$\xi_1$，$\xi_2$，…，$\xi_p$均被赋值。这时，$\xi_1$，$\xi_2$，…，$\xi_p$获得的都是由式(13-18)所确定的抽样值。

这一想法与串行方式不同的是：比较和赋值是以向量形式进行的，用比较的结果来控制向量各分量的赋值，这种赋值方式一般称之为条件向量赋值。关于YH机的条件向量赋值，请参阅《YHFT使用手册》。另外，为了简洁地描述并行计算格式，对向量与向量之间的运算以及向量与标量之间的运算，采用了一些YHFT所允许的表示方法，同时也采用了一些临时约定的表示方法，这将与常规的表示方法有所不同。上述想法的程序框图如图13-2所示。

比较串行和并行的程序框图，不难发现

（1）求出P个抽样值ξ_1，ξ_2，…，ξ_p，并行所需的求和运算次数等于串行求出ξ_{n_0}所需的求和运算次数k_{n_0}；

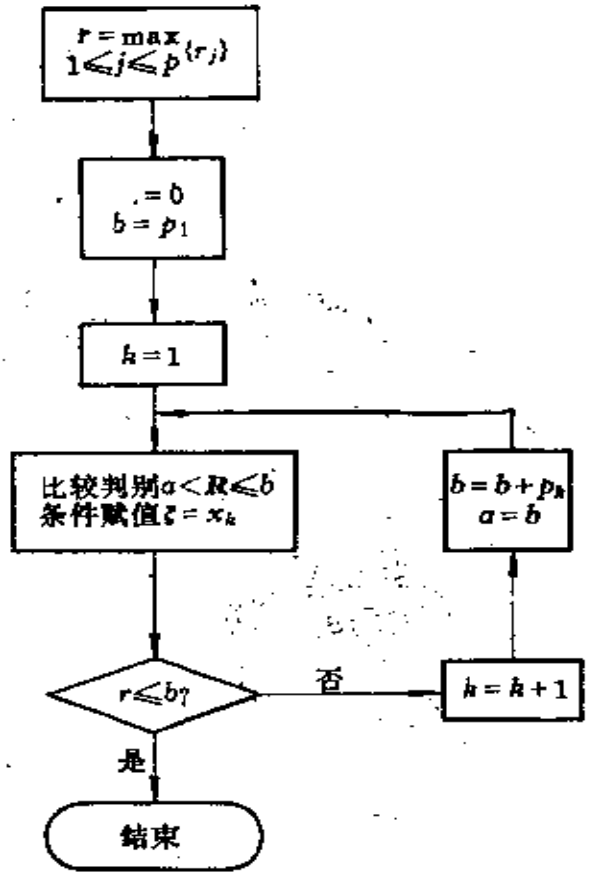


图13-2　离散型随机变量并行抽样框图

$k_{n_0}=\max\limits_{1\leqslant j\leqslant p}\{$串行求出$\xi_j$所需的求和运算次数$\}$

（2）并行条件判断运算次数一般要比串行求出ξ_{n_0}所需的判断运算次数多$2k_{n_0}$次。其中k_{n_0}次出现在循环判别$a<R\leqslant b$中，串行只需判别$r_{n_0}\leqslant b$；k_{n_0}次出现在循环判别$r\leqslant b$中。但不一定多于串行求出ξ_1，ξ_2，…，ξ_p所需的总的判别运算次数。

（3）并行方式要求出$\max\limits_{1\leqslant j\leqslant p}\{r_j\}$以便判别条件赋值完毕与否。这在串行方式是不需要的。用并行方法求出$\max\limits_{1\leqslant j\leqslant p}\{r_j\}$，一般需要作$\log_2 p$次比较交换。关于判别条件赋值完毕与否，也可以采用另外的方法。例如在YHFT中就可以通过向量压缩操作语句来实现。对于跳跃点为有限个的情形，还可以用跳跃点的个

数M来判别，显然当$k=M$时，ξ的分量已全部被赋值，这是因为此时$b=\sum_{k=1}^{M}p_k=1$。在一类事件模拟问题中，判别方法是具有普遍性意义的问题，它直接决定并行模拟效率的好坏。

上述分析表明，并行抽样效率取决于所考虑的随机数序列$\{r_n\}$及概率分布序列$\{p_n\}$的性态，因此分析算法的时间复杂性有一定的困难。一般而言，只要随机数的均匀性较好，串行求出ξ_i（$i=1,2,\cdots,p$）所需的求和运算次数以及条件判断运算次数等具有一种均匀性，不会大起大落，那么并行抽样运算次数将大大地少于串行抽样相应运算的总次数。当然，这只是并行抽样效率要比串行抽样效率好的一种情形。应该指出，因为随机数实际上就是〔0，1〕上均匀分布总体所产生的简单子样，即使是用数学方法产生的伪随机数在经过均匀性检验之后，它的均匀性也是比较好的。换句话说序列$\{r_n\}$的性态是比较确定的，只要相应的序列$\{p_n\}$的性态不出现某些特殊的情形，并行抽样效率就应该是相当高的。

13.3.3 快速并行抽样

既然使用数学方法产生伪随机数，那么考察一下所产生的伪随机数序列的特点，是最自然不过的了。在第二节中，介绍了乘同余法

$$x_{n+1}\equiv\lambda x_n(\mathrm{mod}M)$$
$$r_n=x_n/M$$

一般地有

$$\lambda=5^{2k+1},\begin{cases}x_0=1,\text{得 }x_n=1,5,9,\cdots,4q+1,\cdots,2^r-3\\ x_0=3,\text{得}x_n=3,7,\cdots,4q+3,\cdots,2^r-1\end{cases}$$

$$\lambda=3^{2k+1},\begin{cases}x_0=1,\text{得}x_n=1,3,9,\cdots,8q+1,\\ \qquad\qquad 8q+3,\cdots,2^r-7,2^r-5\\ x_0=3,\text{得}x_n=5,7,13,\cdots,8q+5,\\ \qquad\qquad 8q+7,\cdots,2^r-3,2^r-1\end{cases}$$

最明显的一点是序列$\{x_n\}$是不减的，因而序列$\{r_n\}$也是不减的。根据序列$\{r_n\}$是不减的这一点，对并行抽样方法加以改进，即可得快速并行抽样方法

（1）直接取图13-2中的$r=\max\limits_{1\leqslant j\leqslant p}\{r_j\}=r_p$

（2）当$(\xi_1,\ \xi_2,\ \cdots,\ \xi_p)^T$求出之后，且若此时$\xi_p=x_m$，那么保持此时的$a$与$b$的值，并将它们作为求$\xi'=(\xi_{p-1},\ \xi_{p+2},\cdots,\ \xi_{2p})^T$时的相应的初值，此时$k=m$作为循环变量的初值，比较判别和条件赋值分别为$a<R'\leqslant b$，$\xi'=x_k$，其中$R'=(r_{p+1},r_{p+2},\cdots,\ r_{2p})^T$。当然此时$r=\max\limits_{p+1\leqslant j\leqslant 2p}\{r_j\}=r_{2p}$。如此类推即可得到一系列抽样值$\xi_1,\ \xi_2,\cdots,\xi_p;\ \xi_{p+1},\cdots,\ \xi_{2p};\ \xi_{2p+1},\cdots,\xi_{3p};\cdots$。

根据$\{r_n\}$是不减的，同样可得到快速串行抽样方法。快速并行抽样方法和快速串行抽样方法，两者的求和运算次数是一样的，但前者的比较判别次数以及赋值次数一般要比后者少。

在一类事件模拟问题中，若能充分利用$\{r_n\}$的不减性，将使并行模拟效率得到极大的提高。

13.4 连续型随机变量抽样

连续型随机变量抽样所采用的方法，往往根据分布的特点而定，这一节将在介绍各种不同抽样方法的基础上，讨论抽样的并行实现。

13.4.1 直接抽样方法

连续型分布的一般形式如下：

$$F(x)=\int_{-\infty}^{x}f(t)dt \tag{13-19}$$

其中$f(x)$为密度函数。

根据上一节所介绍的直接抽样方法的一般公式(13-16)，如果分布函数的反函数存在，则有连续型分布的直接抽样方法

$$\xi_n=F^{-1}(r_n)\qquad n=1,\ 2,\ \cdots,\ N \tag{13-20}$$

用向量的形式可表为

$$\begin{pmatrix}\xi_1\\ \xi_2\\ \vdots\\ \xi_N\end{pmatrix}=\begin{pmatrix}F^{-1}(r_1)\\ F^{-1}(r_2)\\ \vdots\\ F^{-1}(r_N)\end{pmatrix} \tag{13-21}$$

式（13-21）是适宜于并行计算的。

〔例13-1〕 指数分布

指数分布的一般形式如下：

$$f(x)=\lambda e^{-\lambda x} \quad x\geqslant 0 \tag{13-22}$$

其中 $\lambda>0$。根据直接抽样方法的一般结果（13-20），对此分布的直接抽样如下：

$$\xi_n=-\frac{1}{\lambda}\ln r_n \tag{13-23}$$

13.4.2 舍选抽样方法

在实际问题中，连续型分布往往是很复杂的，有的只能给出分布函数的解析表达式，但给不出其反函数的解析表达式，如 β-分布；有的则连分布函数的解析表达式都给不出，如正态分布；有的尽管能够给出分布函数的反函数，但反函数的计算量较大。因此，对于某些连续型分布采用直接抽样方法有一定的困难。舍选抽样方法能够克服直接抽样方法的某些不足。

一、舍选抽样方法的一般形式

设随机变量 η 的密度函数 $f(x)$ 可表为

$$f(x)=L\int_{-\infty}^{h(x)} g(x,\ y)dy \tag{13-24}$$

其中 $g(x,\ y)$ 是随机变量 $(X,\ Y)$ 的联合密度函数，$h(x)$ 在 y 的定义域上取值，L 为规格化常量。舍选抽样方法的一般形式如图13-3。

现给出舍选抽样方法的证明。对任意的 x

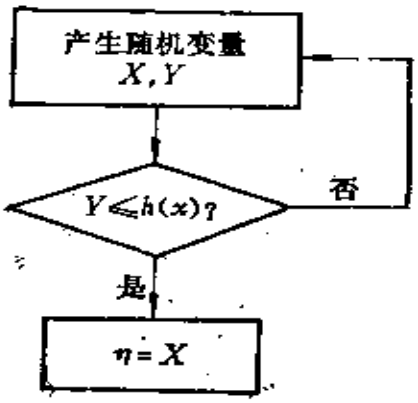


图 13-3

$$
\begin{aligned}
P(\eta<x)&=P\{X<x \mid y\leqslant h(X)\}\\
&=P\{X<x,\ Y<h(X)\}/P\{Y<h(X)\}\\
&=\int_{-\infty}^{x}dt_1\int_{-\infty}^{h(t_1)}g(t_1,\ t_2)dt_2\Big/\\
&\quad\int_{-\infty}^{\infty}\int_{-\infty}^{h(t_1)}g(t_1,\ t_2)dt_1dt_2\\
&=\int_{-\infty}^{x}\left[L\int_{-\infty}^{h(t_1)}g(t_1,\ t_2)dt_2\right]dt_1\\
&=\int_{-\infty}^{x}f(t_1)dt_1
\end{aligned}
$$

舍选抽样方法的抽样效率为

$$E=\frac{1}{L}=\int_{-\infty}^{\infty}\int_{-\infty}^{h(x)}g(x,\ y)dxdy$$

设X_1, X_2, …, X_N, Y_1, Y_2, …, Y_N分别为随机变量X, Y的抽样值。η_1, η_2, …, η_N是随机变量η的待求抽样值。令

$$\boldsymbol{X}=\begin{pmatrix}X_1\\X_2\\\vdots\\X_N\end{pmatrix},\quad \boldsymbol{Y}=\begin{pmatrix}Y_1\\Y_2\\\vdots\\Y_N\end{pmatrix},\quad \boldsymbol{\eta}=\begin{pmatrix}\eta_1\\\eta_2\\\vdots\\\eta_N\end{pmatrix},\quad \boldsymbol{h}=\begin{pmatrix}h(X_1)\\h(X_2)\\\vdots\\h(X_N)\end{pmatrix}$$

可得舍选抽样方法的并行格式如图13-4。

显然，并不是$\boldsymbol{Y}$中的每个分量都一定满足

$$Y_i\leqslant h(X_i)\qquad i=1,\ 2,\ \cdots,\ N$$

因此并不是$\boldsymbol{\eta}$中的每个分量都被赋值。为了记录被赋值的分量及

其个数，这里采用了向量压缩操作，这一操作也由条件 $Y \leqslant h$

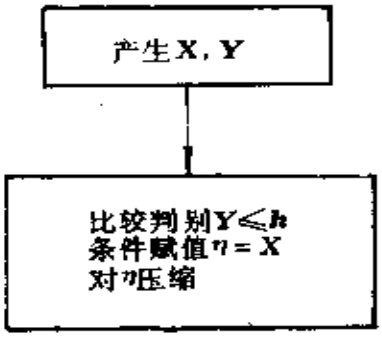


图 13-4

控制。

〔例13-2〕 各向同性散射方向角余弦分布

在核物理模拟中，质点各向同性散射方向角 φ 均匀分布在〔0，2π〕上，需要产生方向角余弦的抽样。

产生随机数 r，用直接抽样方法有 $\varphi = 2\pi r$，方向余弦 $\eta = \cos\varphi$。

直接抽样计算余弦的运算量比较大。如改用舍选抽样方法可以更快。很明显，η 有密度函数

$$f(x)=\begin{cases} \dfrac{1}{\pi\sqrt{1-x^2}}, & |x|<1 \\ 0, & \text{其它} \end{cases}$$

定义随机变量

$$\begin{cases} X=\dfrac{r_1^2-r_2^2}{r_1^2+r_2^2} \\ Y=r_1^2+r_2^2 \end{cases}$$

它们有联合分布密度

$$g(x,\ y)=\begin{cases} \dfrac{1}{4\sqrt{1-x^2}}, & |x|<1,\ 0<y<\dfrac{2}{1+|x|} \\ 0, & \text{其它} \end{cases}$$

这时

$$f(x)=\frac{4}{\pi}\int_{-\infty}^{1} g(x,\ y)dy$$

即 $L=\frac{4}{\pi}$，$h(x)=1$。由舍选抽样方法得 $\cos\varphi$ 的抽样方法如图13-5。

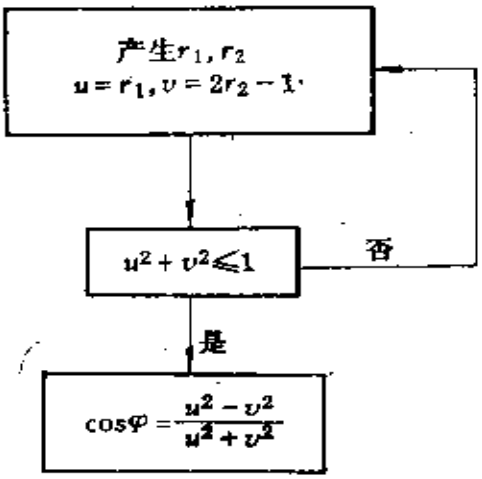


图 13-5

抽样效率 $E=\frac{\pi}{4}\approx 0.7854$。

其并行格式如图13-6。

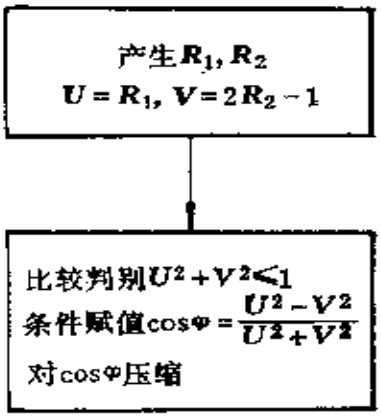


图 13-6

框图中

$$\boldsymbol{R}_1=\begin{pmatrix} r_{11} \\ r_{12} \\ \vdots \\ r_{1N} \end{pmatrix}, \quad \boldsymbol{R}_2=\begin{pmatrix} r_{21} \\ r_{22} \\ \vdots \\ r_{2N} \end{pmatrix}, \ r_{ij}\text{为随机数}$$

$$\boldsymbol{V}=2\boldsymbol{R}_2-1 \triangleq \begin{pmatrix} 2r_{21}-1 \\ 2r_{22}-1 \\ \vdots \\ 2r_{2N}-1 \end{pmatrix}$$

$$\boldsymbol{U}^2 \triangleq \boldsymbol{U}*\boldsymbol{U}=\boldsymbol{R}_1*\boldsymbol{R}_1 \triangleq \begin{pmatrix} r_{11}^2 \\ r_{12}^2 \\ \vdots \\ r_{1N}^2 \end{pmatrix}$$

$$\frac{\boldsymbol{U}^2-\boldsymbol{V}^2}{\boldsymbol{U}^2+\boldsymbol{V}^2} \triangleq (\boldsymbol{U}*\boldsymbol{U}-\boldsymbol{V}*\boldsymbol{V}) \div (\boldsymbol{U}*\boldsymbol{U}+\boldsymbol{V}*\boldsymbol{V})$$

$$\cos\boldsymbol{\varphi} \triangleq \begin{pmatrix} \cos\varphi_1 \\ \cos\varphi_2 \\ \vdots \\ \cos\varphi_N \end{pmatrix}, \ \cos\varphi_i\text{为方向角}\varphi_i\text{的余弦}$$

二、简单分布的舍选抽样方法

设随机变量η的密度函数$f(x)$定义在有限区间$[a, b]$上，而且是有界的。记

$$M=\sup_{a\leqslant x\leqslant b} f(x)$$

根据式（13-24），可选取

$$g(x, y)=\frac{1}{M(b-a)} \qquad a\leqslant x\leqslant b, \ 0\leqslant y\leqslant M$$

$$h(x)=f(x), \ L=M(b-a)$$

于是，根据舍选抽样方法的一般形式（13-25），有对$f(x)$的舍选抽样方法如图13-7。

其并行格式如图13-8。抽样效率 $E=\frac{1}{L}=1/M(b-a)$。

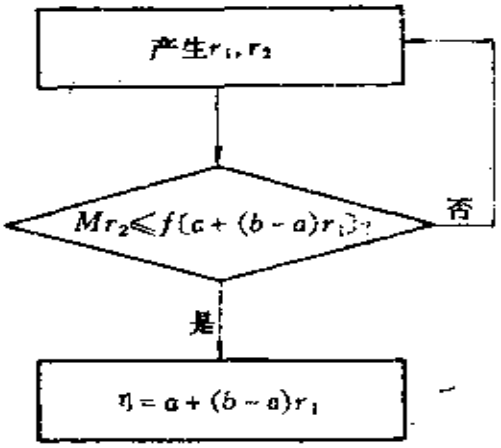


图 13-7

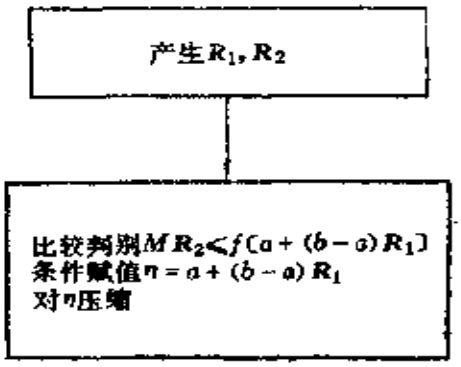


图 13-8

框图中

$$f[a+(b-a)\boldsymbol{R}_1] \triangleq \begin{pmatrix} f[a+(b-a)r_{11}] \\ f[a+(b-a)r_{12}] \\ \vdots \\ f[a+(b-a)r_{1N}] \end{pmatrix}$$

三、乘分布的舍选抽样方法

设随机变量η的密度函数$f(x)$可表为

$$f(x)=Lh(x)g(x)$$

其中$L>1$，$0 \leqslant h(x) \leqslant 1$，$g(x)$是随机变量$\xi$的密度函数。根据式(13-24)，可选取

$$g(x, y)=g(x) \quad 0 \leqslant y \leqslant 1$$

于是，根据舍选抽样方法的一般形式(13-25)，有乘分布的舍选抽样方法如图13-9。

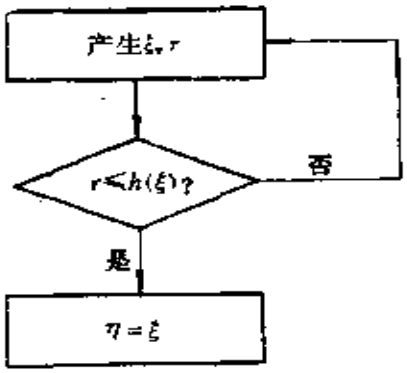


图 13-9

抽样效率$E=\frac{1}{L}$。

其并行格式如图13-10。

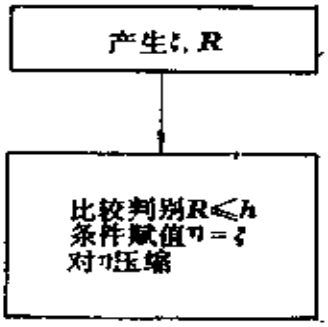


图 13-10

框图中

$\boldsymbol{R}=(r_1,\ r_2,\ \cdots,\ r_N)^T$，$r_i$ 为随机数

$\boldsymbol{\xi}=(\xi_1,\ \xi_2,\ \cdots,\ \xi_N)^T$，$\xi_i$ 为随机变量 ξ 的抽样值

$\boldsymbol{h}=(h(\xi_1),\ h(\xi_2),\ \cdots,\ h(\xi_N))^T$

$\boldsymbol{\eta}=(\eta_1,\ \eta_2,\ \cdots,\ \eta_N)^T$，$\eta_i$ 为 η 的待求抽样值

〔例13-3〕 正态分布

正态分布的一般形式如下：

$$f(x)=\sqrt{\frac{1}{2\pi}}\,e^{-\frac{x^2}{2}}$$

根据乘分布的舍选抽样方法，分解 $f(x)$ 为

$$f(x)=\sqrt{\frac{2e}{\pi}}\cdot\frac{1}{2}e^{-1/2}\cdot e^{-\frac{x^2}{2}}=\sqrt{\frac{2e}{\pi}}\cdot\frac{1}{2}e^{-\frac{1}{2}(x-1)^2}\cdot e^{-x}$$

这里 $L=\sqrt{\frac{2e}{\pi}}$，$h(x)=\frac{1}{2}e^{-\frac{1}{2}(x-1)^2}$，$g(x)=e^{-x}$。由此可得正态分布的舍选抽样方法如图 13-11 所示。

抽样效率 $E=\sqrt{\frac{2e}{\pi}}\approx 0.760$。

其并行格式如图13-12。

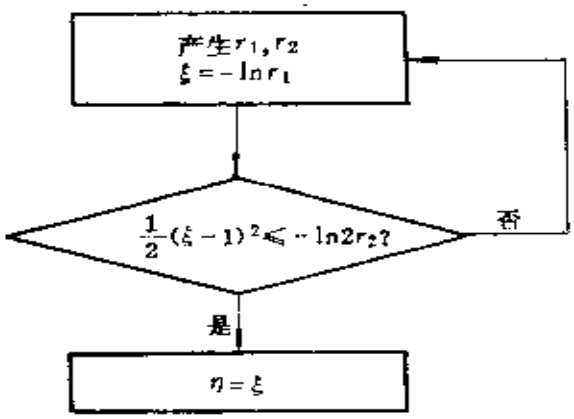


图 13-11

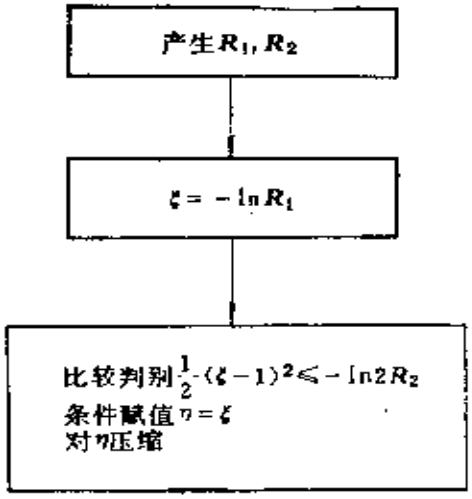


图 13-12

框图中

$$-\ln \boldsymbol{R}_1 \triangleq (-\ln r_{11},\ -\ln r_{12},\ \cdots,\ -\ln r_{1N})^T$$

$$-\ln 2\boldsymbol{R}_2 \triangleq (-\ln 2r_{21},\ -\ln 2r_{22},\ \cdots,\ -\ln 2r_{2N})^T$$

在采用一般舍选抽样方法抽样时，如能考虑到分布的特殊性，可以得到抽样效率更高的抽样方法。现举例说明这一点。

〔例13-4〕 对称分布

对称分布的一般形式如下：

$$f(x)=g(x)+h(x)$$

其中 $g(x)$ 为某一随机变量 ξ 的密度函数，满足偶函数条件；$h(x)$ 为任意奇函数。$f(x)$ 为随机变量 η 的密度函数，其抽样方法如图 13-13 所示。抽样效率 $E=1$。

其并行格式如图13-14。

框图中

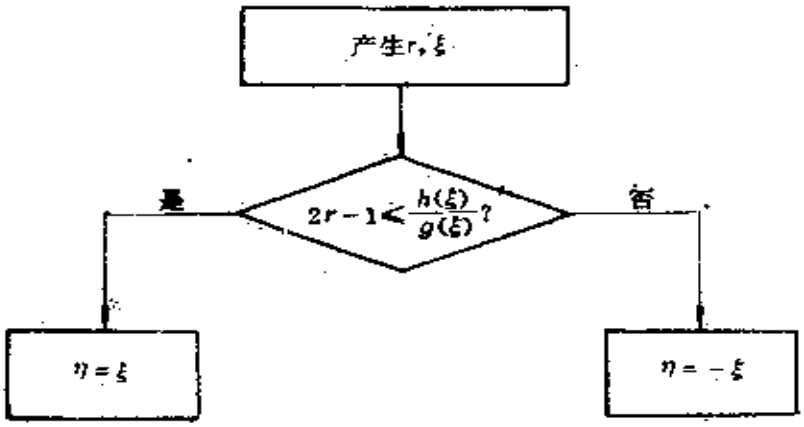


图 13-13

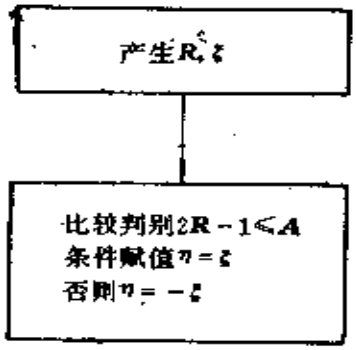


图 13-14

$$A \triangleq \begin{pmatrix} h(\xi_1) \\ h(\xi_2) \\ \vdots \\ h(\xi_N) \end{pmatrix} \div \begin{pmatrix} g(\xi_1) \\ g(\xi_2) \\ \vdots \\ g(\xi_N) \end{pmatrix} \triangleq \begin{pmatrix} h(\xi_1)/g(\xi_1) \\ h(\xi_2)/g(\xi_2) \\ \vdots \\ h(\xi_N)/g(\xi_N) \end{pmatrix}$$

13.4.3 复合抽样方法

考虑如下复合分布

$$f(x)=\int_{-\infty}^{\infty} g(x|y)dF_1(y)$$

其中 $g(x|y)$ 表示与参数 y 有关的条件分布密度函数，$F_1(y)$ 表示分布函数。复合方法一般形式如下：

$$\xi_i = X_g(x|Y_{F_1})$$

其中 Y_{F_1} 是在分布 $F_1(y)$ 中抽样确定，$X_g(x|Y_{F_1})$ 是在分布

$g(x|Y_{F_1})$ 中抽样确定；ξ_i 即是满足分布 $f(x)$ 的抽样值。

作为复合抽样方法的直接应用是对加分布所给出的复合抽样方法。加分布的一般形式如下：

$$f(x)=\sum_k p_k f_k(x)$$

其中 $0<p_k<1$，$\sum_k p_k=1$。以概率 p_k 从密度函数 $f_k(x)$ 中抽样随机变量 η_k，它们的复合密度函数就是随机变量 ξ 的密度函数 $f(x)$。根据复合抽样方法的一般结果，选取

$$F_1(x)=\sum_{k<x} p_k$$

通过对 $F_1(x)$ 直接抽样，有对加分布的复合抽样方法如下：

$$\xi_n=\eta_k,\quad 当\ \sum_{i=1}^{k-1} p_i<r_n\leqslant\sum_{i=1}^{k} p_i$$

加分布的复合抽样很容易并行实现。设

$$\boldsymbol{\eta}_k=\begin{pmatrix}\eta_{k1}\\ \eta_{k2}\\ \vdots\\ \eta_{kN}\end{pmatrix},$$

$$\boldsymbol{R}=\begin{pmatrix}r_1\\ r_2\\ \vdots\\ r_N\end{pmatrix},\quad \boldsymbol{\xi}=\begin{pmatrix}\xi_1\\ \xi_2\\ \vdots\\ \xi_N\end{pmatrix}$$

其中，η_{kj} 为在 $f_k(x)$ 中产生的抽样值。$j=1,2,\cdots,N$ 由此可得如图 13-15 所示的并行抽样方法。

从框图中，可以看出并行抽样需要产生有关的 $\boldsymbol{\eta}_k(k=1,2,\cdots)$。但不一定 $\boldsymbol{\eta}_k$ 的每个分量都赋给了 $\boldsymbol{\xi}$，甚至有可能出现 $\boldsymbol{\eta}_k$ 的分量一个都没有赋给 $\boldsymbol{\xi}$ 的情况，因此必须综合考虑产生 $\boldsymbol{\eta}_k$ 以及 $\boldsymbol{\xi}$ 的抽样费用。

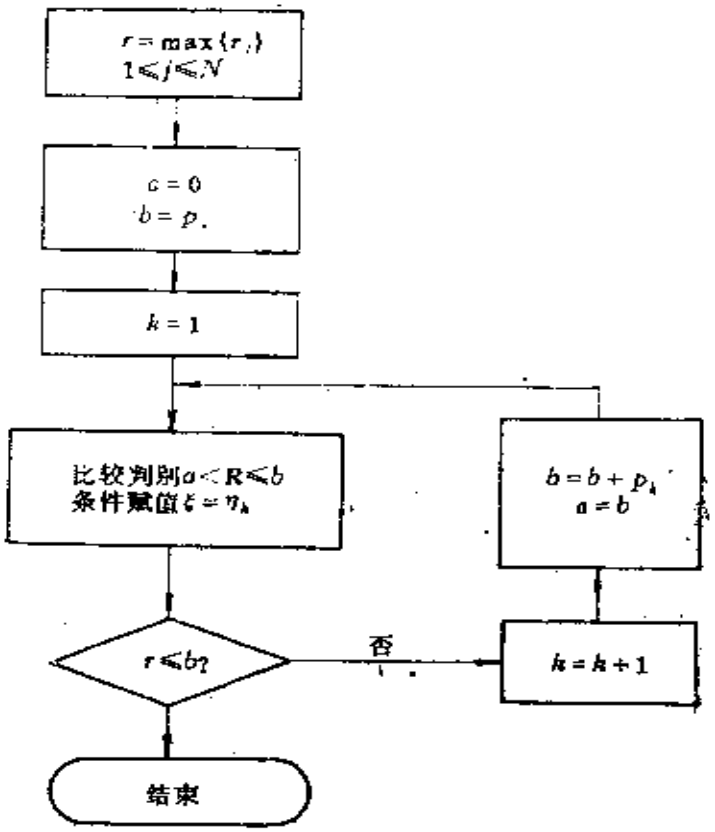


图13-15　加分布随机变量并行抽样框图

将舍选抽样和复合抽样结合起来，可得复合舍选抽样，其并行计算可仿照前述的方式加以讨论，这里不再赘述。

13.4.4　近似抽样方法

舍选抽样方法、复合抽样方法、复合舍选抽样方法以及某些变换抽样方法都是为了解决直接抽样方法的困难而提出来的。然而，由于随机变量所服从的分布是千差万别的，用这些方法实现随机变量抽样还是有困难。一是有的分布用上述方法虽然可以实现，但抽样效率很低；二是有的分布用上述方法虽然抽样效率不低，但运算量太大。由于上述原因，采用某种近似抽样方法是很必要的。

近似抽样方法主要有三种形式

第一种方法是根据连续型分布的直接抽样方法，对分布函数

的反函数 $F_\xi^{-1}(r)$ 给出近似计算方法。

第二种方法是根据概率统计学中的理论，在渐近分布 $f_n(x)$ 的极限分布 $f(x)$ 为已知分布的情况下，当 n 足够大时，用渐近分布 $f_n(x)$ 的随机变量 ξ_n 替代 $f(x)$ 的随机变量 ξ。这种方法不要求给出 $f_n(x)$ 的具体形式，而只要求 ξ_n 所服从分布的极限分布为已知分布 $f(x)$。

第三种方法是对已知密度函数 $f(x)$（或分布函数 $F(x)$）进行近似，确定某一近似的分布 $f_a(x)=f(x)$，然后对近似分布 $f_a(x)$ 进行抽样，用近似分布的随机变量 ξ_{f_a} 替代随机变量 ξ_f。在这种方法中最常见的有阶梯近似

$$f_a(x)=\int_{x_{i-1}}^{x_i} f(x)dx \quad x_{i-1}<x\leqslant x_i$$

和线性近似

$$f_a(x)=c\left\{f_{i-1}+\frac{x-x_{i-1}}{x_i-x_{i-1}}(f_i-f_{i-1})\right\} \quad x_{i-1}<x\leqslant x_i$$

其中 x_0，x_1，…，x_n 为任意分点；f_0，f_1，…，f_n 为相应的密度函数值；c 为归一化因子。此二种近似分布的抽样方法分别为

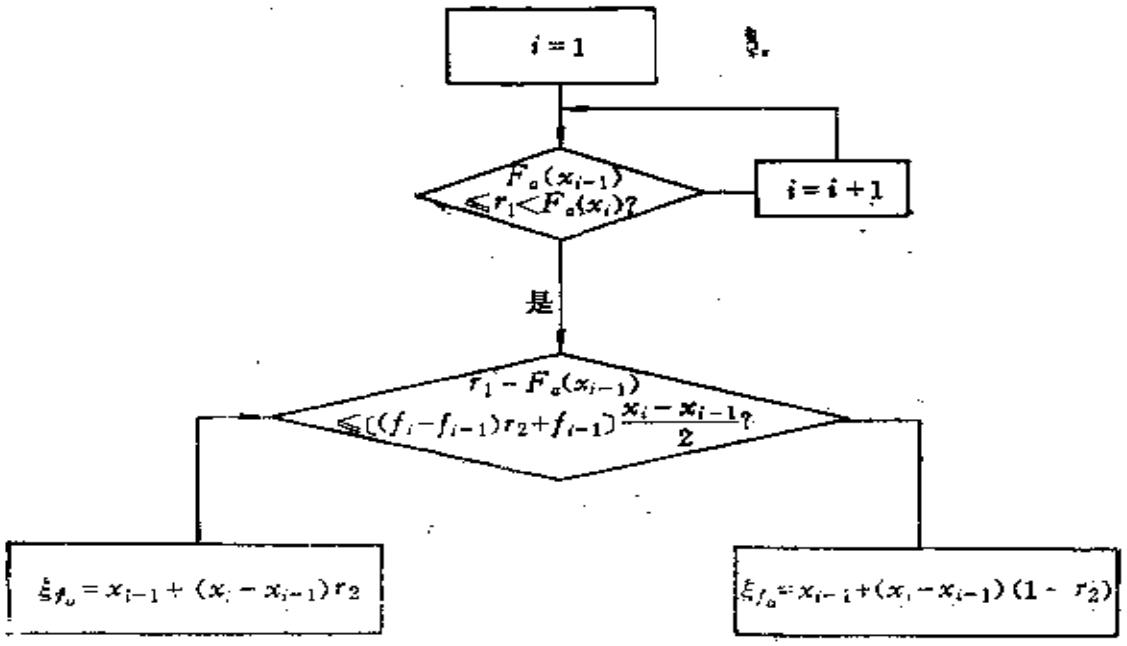


图 13-16

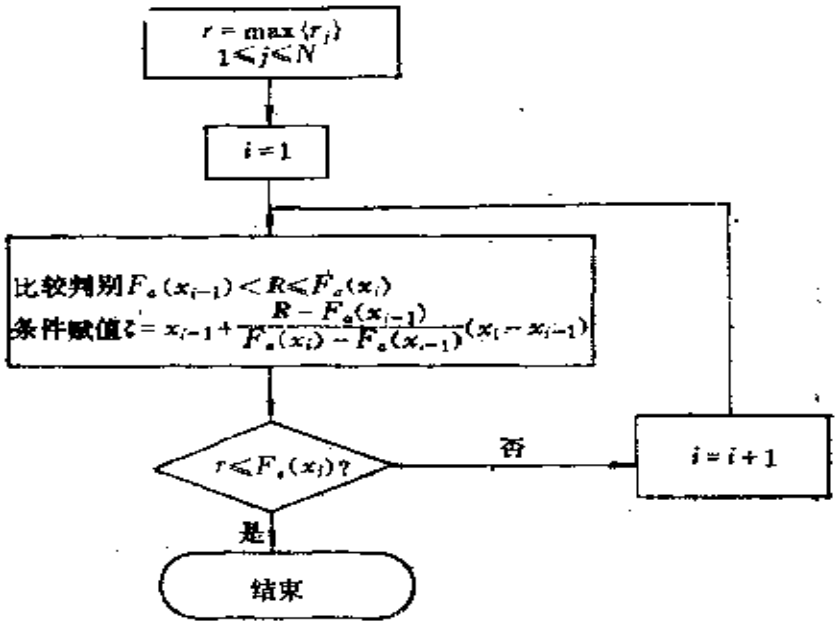


图13-17　阶梯近似并行抽样框图

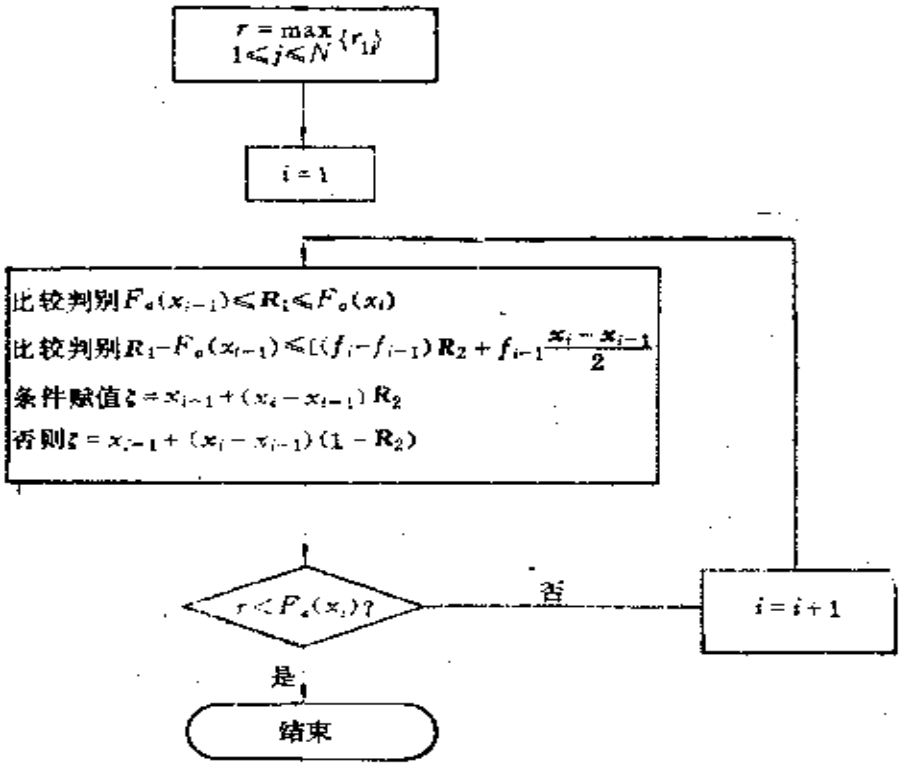


图13-18　线性近似并行抽样框图

$$\xi_{i'}=x_{i-1}+\frac{r-F_a(x_{i-1})}{F_a(x_i)-F_a(x_{i-1})}(x_i-x_{i-1}),$$

当$F_a(x_{i-1})<r\leqslant F_a(x_i)$;

其中$F_a(x)$为各自密度函数$f_a(x)$所对应的分布函数。

此二种近似分布的并行抽样方法分别如图 13-17 及 图 13-18 所示。

图 13-17 框图中 $\boldsymbol{R}=(r_1, r_2, \cdots, r_N)^T$；另外，请注意 $F_a(x)$ 为分布函数，因此 $F_a(x)$ 是不减的且 $F_a(x_n)=1$。

图 13-18 框图中 $\boldsymbol{R}_1=(r_{11}, r_{12}, \cdots, r_{1N})^T$；$\boldsymbol{R}_2=(r_{21}, r_{22}, \cdots, r_{2N})^T$。向量 ξ 的赋值受双重条件控制。

13.5 随机向量抽样

在用 MC 方法求解某些应用问题时，经常需要产生多维随机向量的样本。如果随机向量的各分量彼此独立，则可应用前两节的方法，分别对各分量独立地进行抽样。如果分量是相关的，可以分别求出它们的边缘分布和条件分布，再化成几个单变量的分布，然后依次进行抽样。这里仅以二维随机向量为例，讨论相关随机向量的抽样，向更高维的推广完全是类似的。

设二维随机向量(ξ, η)的联合分布密度函数为$f(x, y)$，随机变量ξ的边缘分布密度函数为

$$f_1(x)=\int_{-\infty}^{\infty} f(x, y)dy$$

依次选取随机数集合$\{r_{2i-1}\}$，抽样产生边缘分布$f_1(x)$的随机变量$\xi=x_i$。此时随机变量η的条件分布密度函数为

$$f_2(y|x_i)=f(x_i, y)/f_1(x)$$

由随机数集合$\{r_{2i}\}$，产生η的抽样值y_i。

上述的分析表明，随机向量的抽样最终都归结为单个随机变量的抽样，因而前两节所讨论的并行抽样方法完全适用于随机向量的抽样。

正态随机向量简单、常用，而且在变换抽样中也占有很重要

的位置。下面我们讨论多维正态随机向量的抽样，其抽样过程与上面所表述的随机向量抽样的一般过程将有所差异。

设 n 维正态随机向量 ξ 的分量为 ξ_1，ξ_2，…，ξ_n，均值为 $E\xi_k=0$，协方差为 $E\xi_k\xi_l=m_{kl}$，密度函数为：

$$f(x_1,x_2,\cdots,x_n)=(2\pi)^{-\frac{n}{2}}|M|^{-1/2}e^{-x^TM^{-1}x/2}$$

其中 $x=(x_1,\ x_2,\ \cdots,\ x_n)^T$，$|M|$ 为协方差矩阵 M 的行列式。

抽样方法是：先产生期望值为 0，方差为 1 的 n 个独立正态随机变量组成的向量 $\eta=(\eta_1,\ \eta_2,\ \cdots,\ \eta_n)^T$，然后经过线性变换 A，使所得随机向量 $\xi=(\xi_1,\ \xi_2,\ \cdots,\ \xi_n)^T$ 具有预先给定的协方差矩阵 $M=(m_{kl})$。注意，经过线性变换之后，分布的正态性是不变的。现在说明如何选择线性变换 A。

设线性变换矩阵 A 为

$$A=\begin{pmatrix} a_{11} & & 0 \\ a_{21} & a_{22} & \\ \cdots\cdots & \cdots\cdots & \\ a_{n1} & a_{n2}\cdots & a_{nn} \end{pmatrix}$$

作变换 $\xi=A\eta$，展开得到

$$\begin{aligned} &\xi_1=a_{11}\eta_1 \\ &\xi_2=a_{21}\eta_1+a_{22}\eta_2 \\ &\cdots \\ &\xi_n=a_{n1}\eta_1+a_{n2}\eta_2+\cdots+a_{nn}\eta_n \end{aligned} \tag{13-26}$$

很明显，均值 $E\xi_k=0$（$k=1,2,\cdots,n$）。至于协方差，由于

$$E\eta_k\eta_l=\delta_{kl}=\begin{cases} 1, & k=l \\ 0, & k\neq l \end{cases}$$

于是

$$E\xi_k\xi_l=E\left(\sum_{i=1}^{k}a_{ki}\eta_i\right)\left(\sum_{j=1}^{l}a_{lj}\eta_j\right)=\sum_{j=1}^{k}a_{kj}a_{lj}$$

因此

$$M=AA^T \tag{13-27}$$

比较两端的对应元素，得到元素 a_{kl} 按列计算公式

$$a_{kl}=\frac{m_{kl}-\sum_{j=1}^{l-1}a_{kj}a_{lj}}{\sqrt{m_{ll}-\sum_{j=1}^{l-1}a_{lj}^2}}\quad(1\leqslant l\leqslant k\leqslant n)$$

(13-28)

考察上述抽样过程，不难看出，n 维正态分布随机向量的抽样是能够并行实现的

（1）期望值为 0，方差为 1 的正态随机向量的抽样可由舍选抽样的并行方式完成。

（2）变换（13-26）可并行实现。事实上，令

$$\boldsymbol{A}_1=\begin{pmatrix}a_{11}\\a_{21}\\\vdots\\a_{n1}\end{pmatrix},\quad \boldsymbol{A}_2=\begin{pmatrix}0\\a_{22}\\\vdots\\a_{n2}\end{pmatrix},\quad \cdots,\quad \boldsymbol{A}_n=\begin{pmatrix}0\\0\\\vdots\\a_{nn}\end{pmatrix}$$

则式（13-26）变为

$$\boldsymbol{\xi}=\eta_1\boldsymbol{A}_1+\eta_2\boldsymbol{A}_2+\cdots+\eta_n\boldsymbol{A}_n$$

（3）公式（13-28）的计算也可以并行。实际上，式（13-27）可理解为将一个对称矩阵作三角分解，这完全类似于正定矩阵的乔列斯基分解，这种分解是能够并行实现的。

13.6 马尔科夫链模拟

用MC方法模拟某些实际问题时，经常要用到随机过程 $\eta(t)$ 在离散时刻 $t_1<t_2<\cdots$ 上取值的随机序列 $\eta(t_1)$，$\eta(t_2)$，$\cdots$。显然，一个给定的随机过程 $\eta(t)$，经截断和数字离散化处理后，得到一组随机变量 $\eta(t_1),\eta(t_2),\cdots,\eta(t_N)$，可以看作一个 N 维随机向量，因而可以利用随机向量的抽样方法给出 $(\eta(t_1),\eta(t_2),\cdots,\eta(t_N))^T$ 的模拟值。但由于 N 的取值很大，利用这种方法模拟随机过程显然比较困难。为此往往针对具体的随机过程，寻找具

体的有效方法。鉴于马尔科夫链在MC方法应用中的重要性，这一节先介绍马尔科夫链的模拟，然后讨论它的并行实现。

13.6.1 基本概念

考虑仅在离散时间（用整数1，2，3，…，k表示）发生转移的马尔科夫链。设系统的状态为有限个，例如s_1，s_2，…，s_k，并设$s(t)$是在时间t内的状态，那么$s(t)$是一个随机变量。系统s各个状态之间的转移概率由条件概率P_{ij}给出。这里条件概率

$$P_{ij}=p\{s(t)=s_j|\,s(t-1)=s_i\}$$

表示系统在时刻$t-1$处于状态s_i，经一步转移在时刻t处于状态s_j的概率，通常称为转移概率。由于转移概率$P_{ij}(i,j=1,2,3,\cdots,k)$与转移发生的时间$t$无关，它只依赖于系统$s$在前一时刻所处的状态$s_i$而不依赖于系统过去的历史，因此它组成一个平稳马尔科夫链。

转移概率P_{ij}的全体组成一个k阶转移矩阵(P_{ij})，它完全确定了系统s的统计性质。根据转移矩阵(P_{ij})的性质，可把系统s的各个状态加以分类。

系统中任一状态s_i若有

$$P_{ij}=\begin{cases}1, & \text{当 } i=j\\ 0, & \text{当 } i\neq j\end{cases}$$

则称状态s_i为吸收态，如果一个系统经多次转移，随机转移到一个吸收态上，则这个系统以概率1终止在吸收态上。

系统中任意两个状态s_i及s_j，若由s_i经有限步τ转移到状态s_j的概率$P_{ij}(\tau)$不为零，则称由状态s_i可转移到状态s_j，简记为$s_i\to s_j$。显然，状态的可转移性是不对称的，但可以递推，即由$s_i\to s_j$，$s_j\to s_l$可以得到$s_i\to s_l$。

一个马尔科夫链，如果它的所有状态都可转移到吸收态，则系统从任意状态开始，经有限步转移总可以概率1进入到一个吸收

态，这种链常称为吸收性马尔科夫链。在数字计算机上进行统计模拟的多数模型中，如在求逆矩阵、解线性代数方程组和解积分方程的模拟中；在微分方程边界条件求解的随机游动模型中；在中子物理、稀薄气体等质点随机游动模拟中，都遇到了吸收性马尔科夫链的模型。

13.6.2 马尔科夫链的模拟

设马尔科夫链的状态是 $s_1, s_2, \cdots, s_k$，转移矩阵 $P=(P_{ij})$ 采用事件模拟的法则。设系统从某初始状态 s_{i_0} 出发，任取一随机数 r_1，如果

$$\sum_{j=1}^{i_1-1} P_{i_0 j} < r_1 \leqslant \sum_{j=1}^{i_1} P_{i_0 j}$$

则系统就从 s_{i_0} 出发到 s_{i_1}，即 $s_{i_0} \to s_{i_1}$。游动依次继续下去，直至第 τ 步（τ 也是随机的）游动终止在吸收态 s_{i_τ} 上。此时，随机游动的路线 ξ 记为

$$\xi = x \quad x: \ s_{i_0} \to s_{i_1} \to \cdots \to s_{i_\tau}$$

亦即 ξ 是初始状态 s_{i_0}，终止状态 s_{i_τ}，路线 x 的随机函数。显然，定义在状态集合 $\{s_i\}$ 上的任意函数 $f(x)$ 也是一个随机变量。求得 $f(x)$ 的一个观测值之后，系统 s 返回初始状态 s_{i_0}，然后开始下一次随机游动。设从 s_{i_0} 开始，进行了 N 次试验，得到 $f(x)$ 的 N 个观测值，其平均值为

$$\bar{f} = \sum_{n=1}^{N} f(x_n)$$

当 N 充分大时，$\bar{f}$ 即可作为解 $E\{f(x)\}$ 的一个近似值。

13.6.3 模拟的并行实现

并行模拟的设想是同时从 m 个初始状态出发进行随机游动，每一步游动都以并行的方式进行，直至所有 m 个游动都终止在吸收态上，此时得到分别对应于相应初始态的 m 条路线。对于随机

函数 $f(x)$ 而言，可望获得分别对应于相应初始态的 m 个观测值；但由于函数 $f(x)$ 的自变量 x 与通常意义下的自变量有很大的差异，所以关于 $f(x)$ 的并行计算要视具体情况而定。

为了便于算法描述，在确定意义下涉及到状态的表示时，将用下标 i 表示 s_i，即有对应关系

$$s_i \longleftrightarrow i \qquad i = 1, 2, \cdots, k$$

现考虑 m 个初始状态 $s_{i_{10}}$，$s_{i_{20}}$，$\cdots$，$s_{i_{m0}}$，建立如下的对应关系

$$\begin{pmatrix} s_{i_{10}} \\ s_{i_{20}} \\ \vdots \\ s_{i_{m0}} \end{pmatrix} \longleftrightarrow \begin{pmatrix} i_{10} \\ i_{20} \\ \vdots \\ i_{m0} \end{pmatrix} \triangleq \boldsymbol{I}_0$$

即 $\boldsymbol{I}_0$ 是由 m 个初始态下标组成的初始向量。当每个初始状态都随机游动了一步时，即

$$\begin{pmatrix} s_{i_{10}} \\ s_{i_{20}} \\ \vdots \\ s_{i_{m0}} \end{pmatrix} \longrightarrow \begin{pmatrix} s_{i_{11}} \\ s_{i_{21}} \\ \vdots \\ s_{i_{m1}} \end{pmatrix}$$

同样有

$$\begin{pmatrix} s_{i_{11}} \\ s_{i_{21}} \\ \vdots \\ s_{i_{m1}} \end{pmatrix} \longleftrightarrow \begin{pmatrix} i_{11} \\ i_{21} \\ \vdots \\ i_{m1} \end{pmatrix} \triangleq \boldsymbol{I}_1$$

依此类推可获得 $\boldsymbol{I}_2$，$\boldsymbol{I}_3$，$\cdots$，$\boldsymbol{I}_\tau$。随机游动路线也组成一个向量，可记为

$$\xi = \boldsymbol{X} \quad \boldsymbol{X}: \ \boldsymbol{I}_1 \rightarrow \boldsymbol{I}_2 \rightarrow \cdots \rightarrow \boldsymbol{I}_\tau$$

显然，就一般而言，不是所有 m 个游动在某一步同时都终止在吸收态上，上述表示的含义可理解为直到第 τ 步所有 m 个游动都已终止在吸收态上。因此，规定对于

$$\boldsymbol{I}_j=\begin{pmatrix} i_{1j} \\ \vdots \\ i_{lj} \\ \vdots \\ i_{mj} \end{pmatrix} \quad j=0,1,\cdots,\tau-1;\quad 1\leqslant l\leqslant m$$

若 i_{lj} 的对应状态是吸收态，则

$$\boldsymbol{I}_n=\begin{pmatrix} i_{1n} \\ \vdots \\ i_{lj} \\ \vdots \\ i_{mn} \end{pmatrix} \quad \text{即} i_{ln}=i_{lj},\ n=j+1,j+2,\cdots,\tau$$

将转移矩阵 $P=(P_{ij})$ 表为

$$\begin{array}{c} \\ s_1 \\ s_2 \\ \\ s_k \end{array}\begin{array}{c} \begin{array}{cccc} s_1 & s_2 & & s_k \end{array} \\ \begin{pmatrix} P_{11} & P_{12} & \cdots & P_{1k} \\ P_{21} & P_{22} & \cdots & P_{2k} \\ \cdots\cdots & \cdots\cdots & \cdots\cdots & \cdots\cdots \\ P_{k1} & P_{k2} & \cdots & P_{kk} \end{pmatrix} \end{array}$$

令

$$\begin{aligned} \boldsymbol{P}_1&=(P_{11},\ P_{12},\ \cdots,\ P_{1k}) \\ \boldsymbol{P}_2&=(P_{21},\ P_{22},\ \cdots,\ P_{2k}) \\ &\cdots\cdots \\ \boldsymbol{P}_k&=(P_{k1},\ P_{k2},\ \cdots,\ P_{kk}) \end{aligned}$$

对应于 I_j，存在矩阵

$$A_j=\begin{pmatrix} P_{i_{1j},1} & P_{i_{1j},2} & \cdots & P_{i_{1j},k} \\ P_{i_{2j},1} & P_{i_{2j},2} & \cdots & P_{i_{2j},k} \\ \cdots\cdots & \cdots\cdots & \cdots\cdots & \cdots\cdots \\ P_{i_{mj},1} & P_{i_{mj},2} & \cdots & P_{i_{mj},k} \end{pmatrix}$$

令 $\boldsymbol{A}_{j1},\ \boldsymbol{A}_{j2},\ \cdots,\ \boldsymbol{A}_{jm}$ 为矩阵 A_j 的行向量，则

$$\boldsymbol{A}_{jq}=\boldsymbol{P}_{i_{qj}} \quad q=1,2,\cdots,m$$

令 $\boldsymbol{B}_{j1},\ \boldsymbol{B}_{j2},\ \cdots,\ \boldsymbol{B}_{jk}$ 为矩阵 A_j 的列向量，则

$$\boldsymbol{B}_{jl}=\begin{pmatrix} P_{i_{1j},l} \\ P_{i_{2j},l} \\ \vdots \\ P_{i_{mj},l} \end{pmatrix} \quad l=1,2,\cdots,k$$

又设已产生随机数向量

$$\boldsymbol{R}_j=\begin{pmatrix} r_{1j} \\ r_{2j} \\ \vdots \\ r_{mj} \end{pmatrix} \quad j=1,2,\cdots$$

在作了上述准备之后，可给出如图 13-19 所示的用 $\boldsymbol{R}_j$ 控制随机游动的并行模拟方法。

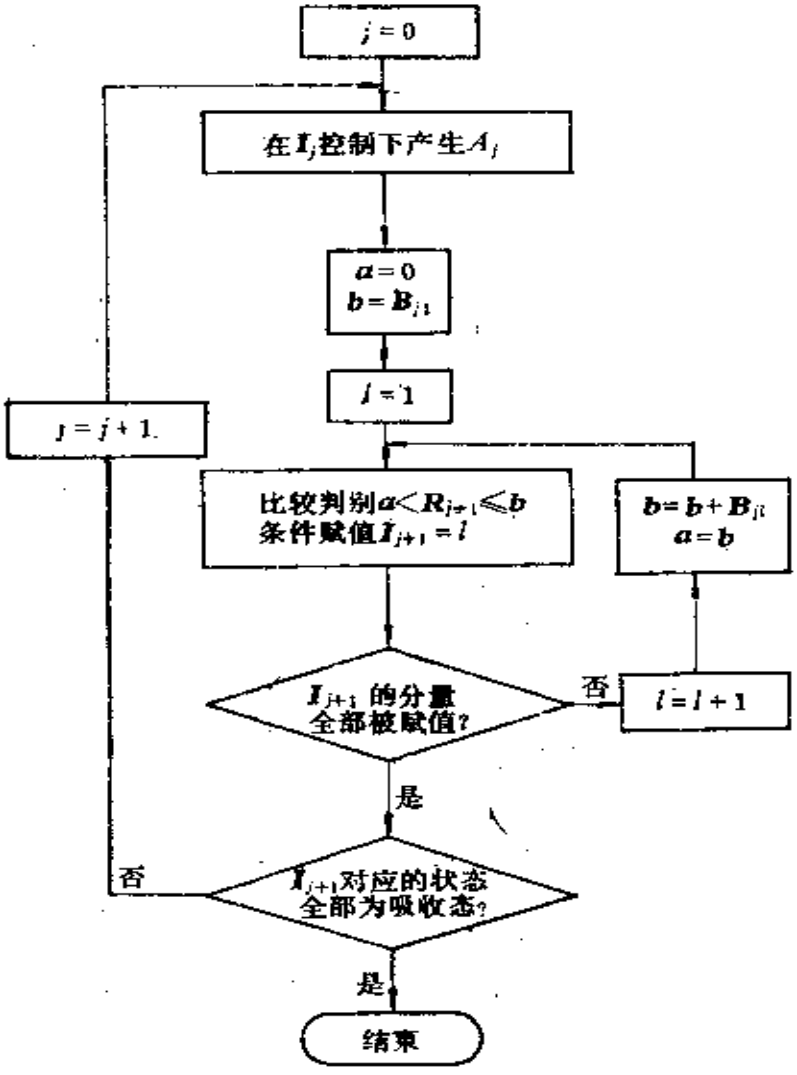


图13-19　马尔科夫链并行模拟框图

（1）在 $\boldsymbol{I}_j$ 控制下产生 A_j，这可以通过 m 次向量赋值实现。注意，用 $\boldsymbol{I}_j(q)$ 表示 $\boldsymbol{I}_j$ 的第 q 个分量，见图13-20。

（2）判断 I_{j+1} 的分量全部赋值与否，可以用两种方式

① 直接判 $l=k$。因为 $\sum_{l=1}^{k} P_{jl}=1$，于是当 $l=k$ 时，必有 $R_{j+1}\leqslant b$，从而 I_{j+1} 的分量一定全部被赋值；

② 事先进行赋值 $I_{j+1}=0$，然后才开始执行关于 l 的循环。在每次循环执行完毕后，判 I_{j+1} 的分量是否为0，若分量全不为0，则分量已全部被赋值，见图13-21。

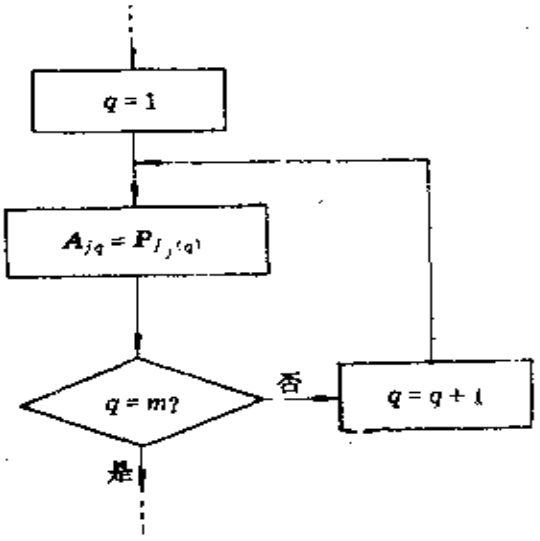


图 13-20

显然，如果每次循环都采用第②种方式进行判别，将极大地影响模拟的并行性；但如果采用第①种方式进行判别，有可能进行一些不必要的运算。比如若当 $l=l_0(l_0<k)$ 时，I_{j+1} 的分量已经全部被赋值，这时就没有必要再循环下去。因此，判 I_{j+1} 的分量全部被赋值与否将是影响并行模拟效率的关键所在。除了直接用上述两种方式判别外，还可以把两种方式结合起来进行判别。关于 j 的每一次循环，确定一个适当的 l_j，若 $l<l_j$，则意味着 I_{j+1} 的分量还有未赋值的；若 $l>l_j$，则采用第②种方式判别。当然这实现起来并不困难，但最佳的 l_j 却是难于确定的。

（3）判 I_{j+1} 对应的状态是否全部为吸收态，见图13-22。

至此，马尔科夫链并行模拟的讨论可以告一段落。其时间复杂性分析这里暂不考虑。很显然，不仅并行模拟效率要视具体情况而定，甚至模拟过程也可以根据具体情况作相应的修改。作为这一节的结束，并考虑到产生随机数的费用，在此指出并行模拟的一个很有意义的改进：比较判别 $a<R_{j+1}\leqslant b$ 可用比较判别 $a<r_{j+1}\leqslant b$ 代替，r_{j+1} 是随机数，$j=0,1,2,\cdots$。

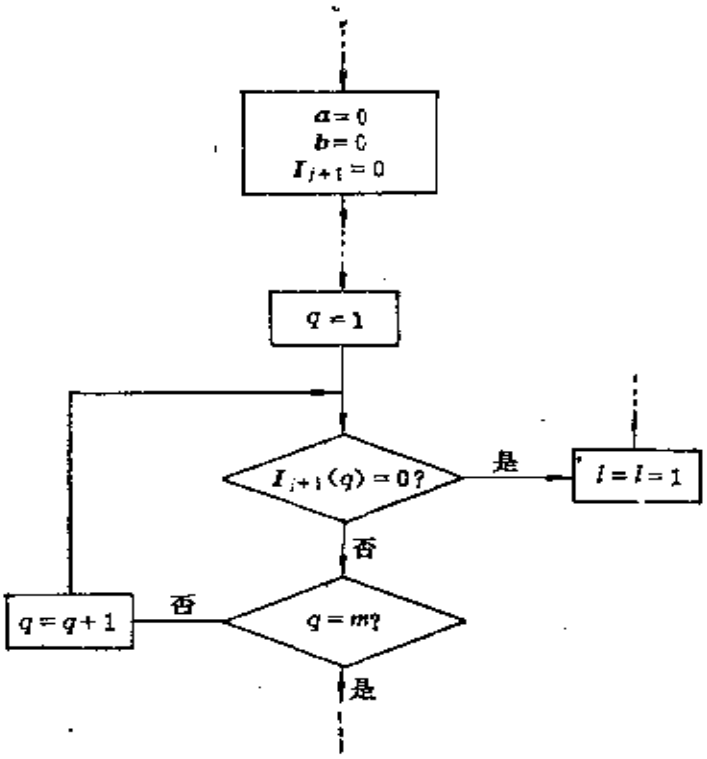


图 13-21

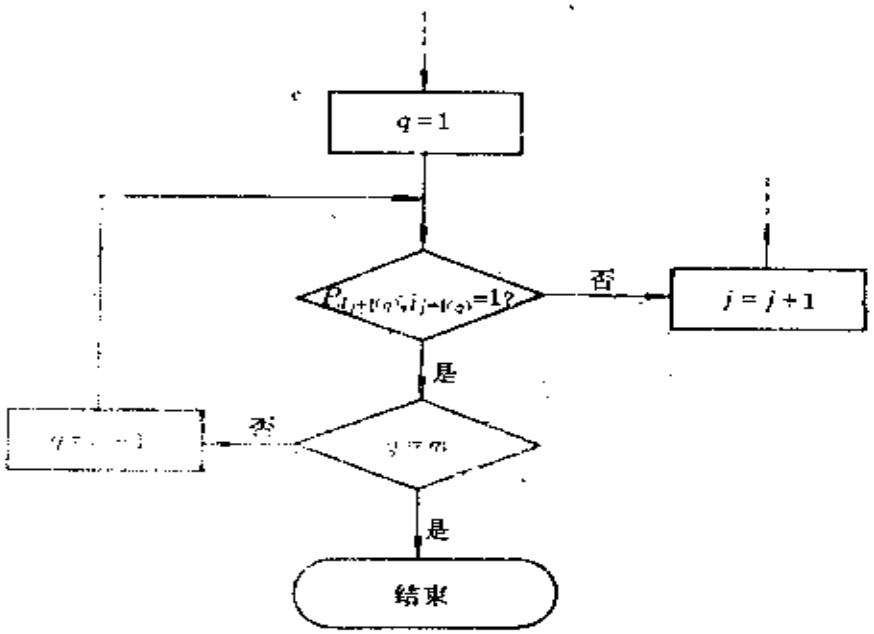


图 13-22

13.7 泊松方程第一边值问题 MC 解法

MC 方法的应用领域是非常广泛的。这一节仅以泊松方程第一边值问题为例，研究差分方程边值问题与随机过程之间的联系，并在此基础上讨论有关并行计算问题。

13.7.1 MC 解法的一般描述

考虑泊松方程第一边值问题：

$$\begin{cases} \Delta u(p) = q(p) & p \in D \qquad (13\text{-}29) \\ u(Q) = f(Q) & Q \in \gamma \qquad (13\text{-}30) \end{cases}$$

其中 D 是一个二维域，γ 是 D 的边界，p 是 D 内的点，Q 是 γ 上的点。

给定步长 h，并作正方形网格把 D 覆盖。落于区域内部的网格点可分为两类：一类是它的四个相邻网格点位于区域内部或边界上，这类点称为内部网格点；另一类是它的相邻网格点的个数少于四个，这类点称为边界网格点。联结边界网格点的折线构成新的边界 Γ'，其内部区域记为 D'。Γ' 上的函数值可按某种规则确定（如图 13-23 所示）。

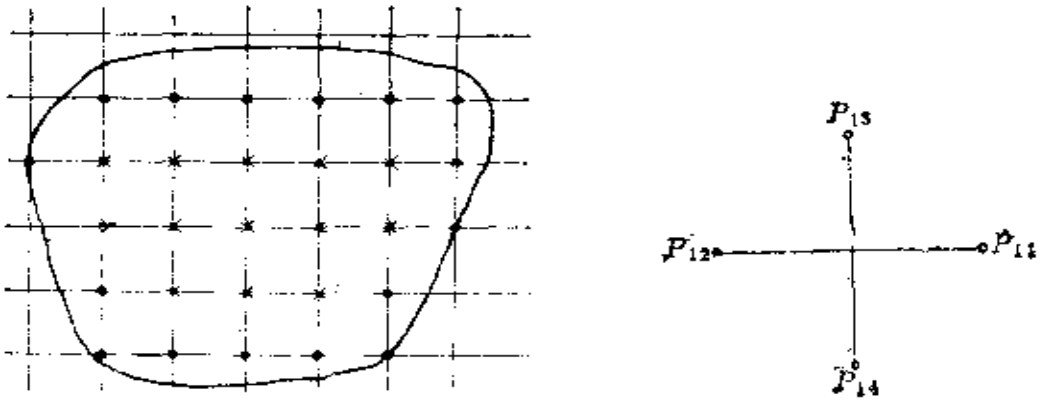


图 13-23

与方程（13-29）对应的差分方程是

$$u(p)=\frac{1}{4}\sum_{j=1}^{4}u(p_{1j})-\frac{h^2}{4}q(p)\quad p\in D' \tag{13-31}$$

其中 p 是 D' 内的节点，$p_{1j}(j=1,2,3,4)$ 是 p 的四个相邻的节点。

与条件（13-30）对应的边界条件是

$$u(Q)=f(Q)\quad Q\in\Gamma' \tag{13-32}$$

其中 Q 是 Γ' 上的节点。

下面介绍满足边界条件（13-32）的差分方程(13-31)的 MC 解法。这里假定差分方程有唯一解。

为了求函数 u 在 $p\in D'$ 处的值 $u(p)$，构造随机游动模型如下：

设有一质点自 p 点出发，以等概率 1/4 向与 p 相邻的四个节点 p_{11}、p_{12}、p_{13}、p_{14} 处随机游动一步。然后再按同样的方式自新的位置向与之邻接的四节点处随机游动一步。如此继续下去，直至质点第一次到达边界 Γ' 上的一个节点处，游动即告终止。

现定义随机变量 ξ。设一条游动路线是

$$\rho_p:\quad p\to p_1\to p_2\to\cdots\to p_{k-1}\to Q\in\Gamma'$$

定义

$$\xi=v(\rho_p)=-\frac{h^2}{4}[q(p)+q(p_1)+\cdots+q(p_{k-1})]+f(Q) \tag{13-33}$$

若质点游动的出发点是边界节点 Q，那么它就停留在点 Q 处，这时定义

$$\xi=v(\rho_Q)=f(Q) \tag{13-34}$$

若能证明期望 $E(\xi)=u(P)$，那么通过计算 $E(\xi)$ 的估值就能获得 $u(P)$ 的近似值。现在证明

$$E(\xi)=E(v(\rho_p))=u(p)$$

由于已假设解是唯一的，所以只需证明

$$E(v(\rho_p))=\frac{1}{4}\sum_{j=1}^{4}E(v(\rho_{p_{1j}}))-\frac{h^2}{4}q(p)\quad p\in D'$$

和

$$E(v(\rho_Q))=f(Q)\quad Q\in\Gamma'$$

假设沿路线 ρ_p 游动的概率为 $p_r(\rho_p)$，那么按照数学期望的定义

$$E(v(\rho_p))=\sum_{\{\rho_p\}}v(\rho_p)\cdot p_r(\rho_p)$$

上式右端是对从点 p 出发的一切可能的游动路线的集合 $\{\rho_p\}$ 取和。

首先要注意到，沿路线 ρ_p 前进的事件可分为以下四个互不相容事件之和

（1）由点 p 出发第一步到达点 p_{11}，而后沿路线 $\rho_{p_{11}}$ 游动的事件；

（2）由点 p 出发第一步到达点 p_{12}，而后沿路线 $\rho_{p_{12}}$ 游动的事件；

（3）由点 p 出发第一步到达点 p_{13}，而后沿路线 $\rho_{p_{13}}$ 游动的事件；

（4）由点 p 出发第一步到达点 p_{14}，而后沿路线 $\rho_{p_{14}}$ 游动的事件。

若将 ρ_p 看成是由 $p\to p_{1j}$ 和 $\rho_{p_{1j}}$ 两条路线组成的，那么对于 $p\to p_{1j}$ 这一步来讲，随机变量 ξ 应取的值是 $-\frac{h^2}{4}q(p)$，而对于 $\rho_{p_{1j}}$ 这条路线来讲，ξ 应取值为 $v(\rho_{p_{1j}})$，又

$$p_r(p\to p_{1j})=\frac{1}{4}\text{及}\sum_{\{\rho_{p_{1j}}\}}p_r(\rho_{p_{1j}})=1,\ j=1,2,3,4$$

于是

$$E(v(\rho_p)) = \sum_{\{\rho_p\}} v(\rho_p)\cdot p_r(\rho_p)$$

$$= \sum_{j=1}^{4} \sum_{\{\rho_{p1j}\}} \left[-\frac{h^2}{4} q(p) + v(\rho_{p1j}) \right] \cdot p_r(p \to p_{1j}) \cdot p_r(\rho_{p1j})$$

$$= \frac{1}{4} \sum_{j=1}^{4} \sum_{\{\rho_{p1j}\}} \left[-\frac{h^2}{4} q(p) + v(\rho_{p1j}) \right] \cdot p_r(\rho_{p1j})$$

$$= -\frac{1}{4} \sum_{j=1}^{4} \left[-\frac{h^2}{4} q(p) \right] \sum_{\{\rho_{p1j}\}} p_r(\rho_{p1j}) + \frac{1}{4} \sum_{j=1}^{4} \sum_{\{\rho_{p1j}\}} v(\rho_{p1j}) p_r(\rho_{p1j})$$

考虑到$E(v(\rho_p))$的定义，则有

$$E(v(\rho_p)) = -\frac{h^2}{4} q(p) + \frac{1}{4} \sum_{j=1}^{4} E(v(\rho_{p1j}))$$

对于每一个节点$Q \in \Gamma'$，按照数学期望的定义

$$E(v(\rho_Q)) = \sum_{\{\rho_Q\}} v(\rho_Q)\cdot p_r(\rho_Q)$$

ρ_Q表示自点Q出发而终止在边界Γ'上的游动路线，这样的游动路线只有一条，即质点从一开始就停止在点Q处，没有任何移动。所以

$$E(v(\rho_Q)) = \sum_{\{\rho_Q\}} v(\rho_Q) = v(\rho_Q) = f(Q)$$

这就是要证明的结论。

这样，对应于由P点出发而在边界Γ'上终止的一次随机游动，可以获得随机变量ξ的一个观察值，经过N次随机游动，即可获得N个观察值ξ_1，ξ_2，…，ξ_N，其平均值

$$\bar{\xi}=\frac{1}{N}\sum_{j=1}^{N}\xi_j$$

即可作为解 $u(p)$ 的近似值。

13.7.2 MC 解法的并行实现

若将区域内部的节点看成是某一系统的状态，那么全部节点构成一个马尔科夫链，边界节点为其吸收态。因此，可用上一节的并行方法模拟所构造的随机游动模型。但由于节点数目往往很大，而且此处的随机游动有其特殊性（每一步游动只有四种可能性），所以用上一节的并行模拟方法，效果是不好的，甚至有可能比用串行方法直接模拟的效果还差。为此，应根据随机游动模型的特殊性，构造新的并行模拟方法。

这里仅就定解区域是凸的情形，给予详细的讨论。让网格点构成一个网格矩阵 $S=(s_{ij})$，区域内的节点均为 S 的元素；让同样的网格点的横坐标构成一个网格横坐标矩阵 X，其纵坐标构成一个纵坐标矩阵 Y。同时定义

$$p_{ij}=\begin{cases}1, & s_{ij}\text{ 为边界节点}\\ 0, & s_{ij}\text{ 为非边界节点}\end{cases}$$

例如对于如图 13-24 所示的区域。

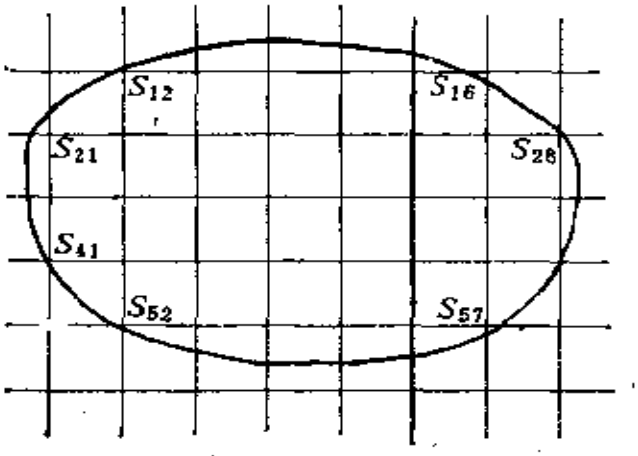


图 13-24

有

$$S=\begin{pmatrix} & s_{12} & s_{13} & s_{14} & s_{15} & s_{16} & & \\ s_{21} & s_{22} & s_{23} & s_{24} & s_{25} & s_{26} & s_{27} & s_{28} \\ s_{31} & s_{32} & s_{33} & s_{34} & s_{35} & s_{36} & s_{37} & s_{38} \\ s_{41} & s_{42} & s_{43} & s_{44} & s_{45} & s_{46} & s_{47} & s_{48} \\ & s_{52} & s_{53} & s_{54} & s_{55} & s_{56} & s_{57} & \end{pmatrix}$$

$$X=\begin{pmatrix} & x_{12} & x_{13} & x_{14} & x_{15} & x_{16} & & \\ x_{21} & x_{22} & x_{23} & x_{24} & x_{25} & x_{26} & x_{27} & x_{28} \\ x_{31} & x_{32} & x_{33} & x_{34} & x_{35} & x_{36} & x_{37} & x_{38} \\ x_{41} & x_{42} & x_{43} & x_{44} & x_{45} & x_{46} & x_{47} & x_{48} \\ & x_{52} & x_{53} & x_{54} & x_{55} & x_{56} & x_{57} & \end{pmatrix}$$

$$Y=\begin{pmatrix} & y_{12} & y_{13} & y_{14} & y_{15} & y_{16} & & \\ y_{21} & y_{22} & y_{23} & y_{24} & y_{25} & y_{26} & y_{27} & y_{28} \\ y_{31} & y_{32} & y_{33} & y_{34} & y_{35} & y_{36} & y_{37} & y_{38} \\ y_{41} & y_{42} & y_{43} & y_{44} & y_{45} & y_{46} & y_{47} & y_{48} \\ & y_{52} & y_{53} & y_{54} & y_{55} & y_{56} & y_{57} & \end{pmatrix}$$

$$p_{12}=p_{13}=p_{14}=p_{15}=p_{16}=p_{21}=p_{28}=p_{31}=p_{38}=p_{41}=p_{48}$$
$$=p_{52}=p_{53}=p_{54}=p_{55}=p_{56}=p_{57}=1$$

其余的 $p_{ij}=0$

要使MC解法能够并行执行，其关键是随机游动能够并行实现。这里仍然利用上一节马尔科夫链并行模拟的基本思想，给出一种按列并行模拟的方法。矩阵S的列向量记为 $\boldsymbol{S}_1$，$\boldsymbol{S}_2$，…，$\boldsymbol{S}_M$；矩阵X的列向量记为 $\boldsymbol{X}_1$，$\boldsymbol{X}_2$，…，$\boldsymbol{X}_M$；矩阵Y的列向量记为 $\boldsymbol{Y}_1$，$\boldsymbol{Y}_2$，…，$\boldsymbol{Y}_M$。

任选矩阵S中某列（如第j列）的m个行连续元素

$$I_0\triangleq\begin{pmatrix} s_{i,j} \\ s_{(i+1),j} \\ \vdots \\ s_{(i+m-1),j} \end{pmatrix}$$

s_{kj} 为边界节点或内部节点

$$k=i,\ i+1,\cdots i+m-1$$

对于任取的随机数r，若I_0所对应的节点均为内部节点，则有

（1）当$0<r\leqslant\frac{1}{4}$，令$\boldsymbol{I}_1=\begin{pmatrix}s_{i,(j-1)}\\ s_{(i+1),(j-1)}\\ \vdots\\ s_{(i+m-1),(j-1)}\end{pmatrix}$（向左游动）

（2）当$\frac{1}{4}<r\leqslant\frac{2}{4}$，令$\boldsymbol{I}_1=\begin{pmatrix}s_{i,(j+1)}\\ s_{(i+1),(j+1)}\\ \vdots\\ s_{(i+m-1),(j+1)}\end{pmatrix}$（向右游动）

（3）当$\frac{2}{4}<r\leqslant\frac{3}{4}$，令$\boldsymbol{I}_1=\begin{pmatrix}s_{(i-1),j}\\ s_{i,j}\\ \vdots\\ s_{(i+m-2),j}\end{pmatrix}$（向上游动）

（4）当$\frac{3}{4}<r\leqslant\frac{4}{4}$，令$\boldsymbol{I}_1=\begin{pmatrix}s_{(i+1),j}\\ s_{(i+2),j}\\ \vdots\\ s_{(i+m),j}\end{pmatrix}$（向下游动）

若$\boldsymbol{I}_0$的第n个分量$\boldsymbol{I}_0(n)$为边界节点，则令$\boldsymbol{I}_1(n)=\boldsymbol{I}_0(n)$，$1\leqslant n\leqslant m$；对于$\boldsymbol{I}_0(n)$中是内部节点的分量，仍由上述方式确定。此时游动可记为$\boldsymbol{I}_0\rightarrow\boldsymbol{I}_1$。根据这一思想，游动依次继续下去，直至第$\tau$步得到$\boldsymbol{I}_\tau$，$\boldsymbol{I}_\tau$所对应的节点均为边界节点，游动即告终止。

不难看出，如果定解区域是凸的，那么随着k的递增，向量$\boldsymbol{I}_k$所对应的节点将以从向量的两端到中间的方式变为边界节点。因此，在确定游动$\boldsymbol{I}_k\rightarrow\boldsymbol{I}_{k+1}$时，可先令$\boldsymbol{I}_{k+1}=\boldsymbol{I}_k$，然后对于$\boldsymbol{I}_{k+1}$中是内部节点的分量（这些分量是行连续的），根据所取的随机数的大小，顺序赋给矩阵S的某一列向量中的一段（这一赋值能够以向量的方式进行）。当然，至于选取矩阵S的哪一列以及哪一段，还需要受$\boldsymbol{I}_k$的控制。对照游动$\boldsymbol{I}_0\rightarrow\boldsymbol{I}_1$，这一点是不难理解的。

至此，对于定解区域是凸的情形，随机游动并行模拟的基本思想便比较清楚了。现在需要解决一些细节问题。

显然，判别 I_k 中某一分量 $I_k(n_0)=s_{i_k,j_k}$ 是否为边界节点，只需判别是否 $p_{i_k,j_k}=1$。这表明所关心的只是下标 (i_k, j_k)。因此在模拟中，若直接用 I_k 作控制变量，将给算法增添一些不必要的麻烦。为此，将用下标 (i, j) 表示节点 s_{ij}，这实际上就是常用的节点编号的方法。这样，存在对应关系

$$I_0=\begin{pmatrix} s_{i,j} \\ s_{(i+1),j} \\ \vdots \\ s_{(i+m-1),j} \end{pmatrix}\longleftrightarrow\begin{pmatrix} i, j \\ i+1, j \\ \vdots \\ i+m-1, j \end{pmatrix}\longleftrightarrow$$

$$A_0\triangleq\begin{pmatrix} i \\ i+1 \\ \vdots \\ i+m-1 \end{pmatrix},\quad B_0\triangleq\begin{pmatrix} j \\ j \\ \vdots \\ j \end{pmatrix}$$

显然 $(i+n, j)=(A_0(n), B_0(n))$，$n=0, 1, \cdots, m-1$。

对于 I_k，存在类似的对应关系

$$I_k\longleftrightarrow(A_k, B_k)\quad k=0, 1, 2, \cdots$$

令 $C=(1, 2, \cdots, N)^T$，N 为矩阵 S 的最大行指数。

在作了上述准备之后，可给出如图 13-7 所示的 MC 解法的并行框图。

在框图中，假设编号属于 (A_0, B_0) 的节点均为内部节点；l_1 与 l_2 分别标记下标数组 (A, B) 两端的边界节点数；向量 ξ 标记对应相应的初始节点及游动路线所产生的 m 个随机向量 $\xi_1, \xi_2, \cdots, \xi_m$。

很明显，对于凸区域同样可构造按行求解的并行 MC 解法。

实际上，定解区域也不一定要求是凸的。对于按列求解的方式，只需一列中不要出现边界节点处于内部节点之间即可；对于按行求解的方式，只要求一行中不要出现类似的情形。当然，有时这一要求仍然得不到满足，那么在求解过程中可以分段进行，其基本思想和求解步骤完全与上述所讨论的一样，只是算法的有关部分要作相应的修改。

第十四章　多处理机并行算法

14.1 前　　言

多处理机的并行算法可归结为同步和异步并行算法。根据这两类并行算法的设计与分析，给出一些重要的概念，并分析它们的一些重要性质。

关于多处理机并行算法的思想，早在1962年，斯克特(Schechter)就曾提出“自由转向”松弛方法，由于当时条件的限制没有深入研究。1969年查占（Chazan）和麦连科(Miranker）在研究线性代数解法时，提出了“混乱松弛”迭代法，后来E_1—塔拉基（E_1—Tarazi）等人将这一方法推广到非线性迭代上。八十年代初期，由于多处理机系统不断问世，美、英、苏、日和波兰等国计算机和数值工作者在这方面做了许多开创性工作。其中英国的依文思（Evans）和巴罗（Barlow）在异步并行算法的并行潜力和计算复杂性方面做了许多基础性研究工作。并提出一类特殊矩阵特征值问题的异步乘幂法，弟贝思(Dubais)和布里吉思（Briggs）论述了多处理机同步迭代算法的性能。黎德(Reed）和帕特利克（Patrich）给出了一种解大型稀疏线性方程组的异步迭代法。与此同时，多处理机非数值并行算法的研究也得到进一步发展。

八十年代初期，我国武汉大学康立山教授等在分布式多处理机Wupp-80上，做了很多研究工作。给出了解数学物理问题的许瓦兹（Schwarz）区域分裂法并做了较系统的数值试验和算法分析。由于篇幅所限，本文仅对异步并行算法进行简要地综合介绍。

14.2 基本概念

14.2.1 多处理机并行算法定义

定义 多处理机的并行算法是一个并发进程的集合。

引入该定义之后，就可以将并行算法和串行算法统一起来。如果一个并行算法是由 k 个并发进程构成的集合，我们称该并行算法具有 k 个进程。当 $k=1$ 时，则称为串行算法。

几点说明：

（1）定义中的并发进程即为同时发生的进程。

（2）区分进程概念和处理机概念是很重要的。一个进程指的是多处理机操作系统中一段程序的执行。因此一个进程就是由一段程序和至多一个处理机来控制，该处理机由操作系统分配，并且在任何给定的时间内执行该段程序。而处理机则是一个物理实体，它实现进程的执行。

（3）一个进程的自始至终不必仅由一台处理机执行，根据操作系统的决策，在一个进程的有效时间内，对于不同的时间间隔，可以把不同的处理机分配给它。因此，一个进程是由若干个子进程组成，在子进程的衔接点上，进程之间进行通讯，这些衔接点称为临界点。

（4）一个进程的执行时间通常不是一个常数，而是一个随机变量，即以一种不可预测的方式波动。波动的原因是

① 一个进程的运行方式是不确定的。因为多处理机系统可能由不同速度的处理机组成，各处理机又是异步的运行。

② 一个进程的运行可能被推迟，推迟的时间是不确定的。因为当一台处理机正在执行一个进程的某个子进程时，该处理机可能被 OS 中断去执行其他事情。在这种情况下，要花费前后交换的执行时间，该子进程可能由另一台处理机去执行，也可能被挂起。

③ 一个进程所分配的资源数是不确定的。这是因为一个进

程的速度是受整个用户体影响。在多用户环境下，在给定时间内，一个进程所分配的资源数，取决于用户所创建的进程数和用户题目的优先级。

④ 一个进程的某个子进程的输入性质，一般说来是不可预测的。

例如计算

$$P=\begin{cases} P_G(e,\ \rho)=(r-1)e\rho+\left[(a+1-r)+\dfrac{b}{\dfrac{e}{e_0\eta^2}+1}\right] \\ \qquad e\rho e^{-a\left(\frac{1}{\eta}-1\right)}+Aue^{-2\left(\frac{1}{\eta}-1\right)} \qquad e>e'' \\ [(e-e')P_G(e'',\ \rho)+(e''-e)P_r(e',\rho)]/(e''-e') \\ \qquad e'\leqslant e\leqslant e'' \\ P_r(e,\ \rho)=\left(a+\dfrac{b}{\dfrac{e}{e_0\eta^2}+1}\right)\rho+Au-Bu^2 \qquad e<e' \end{cases} \tag{14-1}$$

由于 e 所在区间不同，计算 P 的表达式也不同。这表明计算一个函数在一点上的值所需工作量，依赖于这个点的位置。

③ 不同进程可以同时访问存储器，由于存储器访问冲突，一个进程可以被延迟。

14.2.2 并行算法所需时间

将多处理机上并行算法所需时间定义为算法最终完成时，它的进程所用的时间。进程所用时间是下列三部分时间之和

（1）进程的基本处理时间。一个进程的各子进程所需时间之和作为一个进程的基本处理时间。

（2）进程的封锁时间。对于同步并行算法来说，进程的子进程要等待输入，对于异步并行算法来说，进程的子进程要进入临界点执行。在这种情况下，进程要被封锁。

（3）同步原语执行时间。对于同步并行算法来说，要使用同步原语达到进程的同步；对于异步并行算法来说，进程的子进程要进入临界点执行，这都需要使用同步原语。一般情况下，一个原语执行时间相当于一组上百个数对的加法执行时间。

14.3 并行算法的基本概念

众所周知，多处理机上的并行算法分为同步并行算法和异步并行算法两大类。本节介绍这两类算法的定义和它们的优缺点。

14.3.1 同步并行算法

定义 由“协作进程”所组成的并行算法称为同步并行算法。

由定义可知，同步并行算法是由具有下述特征的进程组成。即存在这样一个进程，该进程的某一子进程完成之后，该进程在重新执行前必须等待其他进程的结果。也就是说该进程的一些子进程要等待另一进程完成它的程序的确定部分后才能启动执行，同时利用同步原语达到必要的同步。

例如计算

$$Z = A * B + (C * D) * (I + G) \qquad (14\text{-}2)$$

计算Z的一个同步并行算法是：它由三个进程组成。进程P_1，P_2和P_3分别由二个子进程，一个子进程和二个子进程组成。参见图14-1。

很清楚，进程P_3的第二个子进程的启动要等待P_2的完成。同样，进程P_1的第二个子进程的启动要等待进程P_3的第二个子进程完成。由此可看出，同步并行算法有如下一些缺点：

（1）要使用各种同步原语达到各进程的同步，而同步原语的执行是很费时间的。

（2）由于同步，要引起某些处理机空闲。

（3）要得到一个比较有效的同步并行算法，就要求把问题分解成相同的子任务，这些子任务要按某种有规则的方式进行通

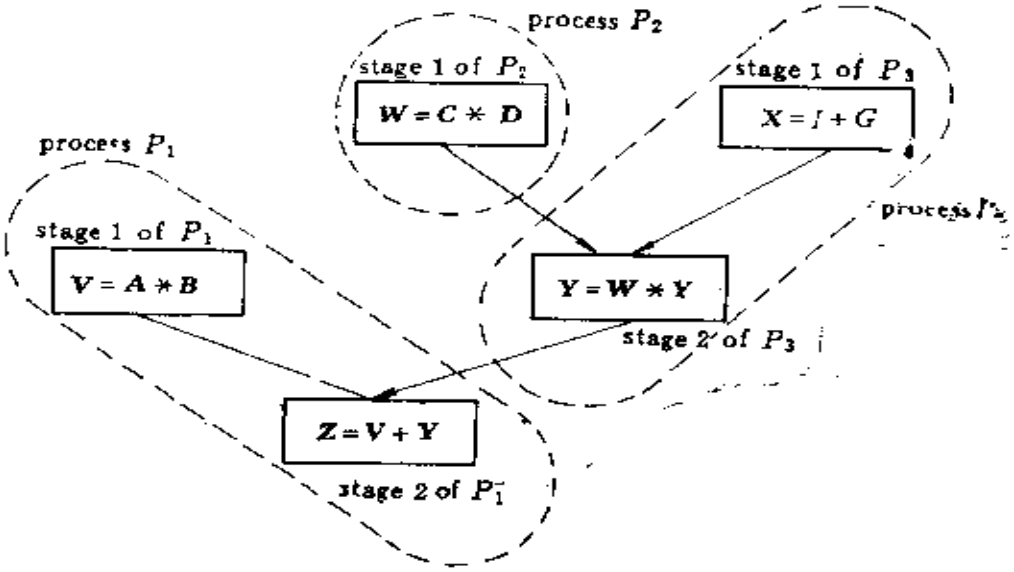


图 14-1

讯。因此，同步并行算法的应用范围是有限的。

（4）由于进程的某一子进程所用时间满足某种分布函数的随机变量，因此在给定的时间，某些进程可能被封锁，这样算法完成的时间就会被推迟。

14.3.2 加速因子和罚因子

（1）加速因子

为了考虑进程封锁对算法执行时间的影响，我们考虑一个由 n 个进程组成的同步并行算法。假定算法在 n 台多机系统上执行，算法所需总时间为 T_n。算法的 i 个进程执行，$n-i$ 个进程被封锁，所需时间为 t_i，则有

$$T_n = \sum_{i=1}^{n} t_i \tag{14-3}$$

假定该算法在单机上运行的总时间为 T_1，有

$$T_1 \leqslant \sum_{i=1}^{n} it_i \tag{14-4}$$

于是在 n 台处理机系统下，算法加速因子为

$$S_n=\frac{T_1}{T_n}\leqslant\frac{\sum_{i=1}^{n} it_i}{\sum_{i=1}^{n} t_i}<n \tag{14-5}$$

（2）罚因子

考虑一类其进程的相同子进程是同步的并行算法。如果希望 n 个相同子进程是同步的，第 i 个子进程所需时间是随机变量 t_i。由于这些子进程是相同的，所以 t_1，t_2，…，t_n 是具有均值 $\bar{t}$ 的相同分布随机变量。因此任一进程的同步子进程所需时间是随机变量

$$T=\max(t_1,\ t_2,\ \cdots,\ t_n)$$

的平均值 $\overline{T}$。一般 $\overline{T}>\bar{t}$。称比率

$$\lambda_n=\frac{\overline{T}}{\bar{t}} \tag{14-6}$$

为 n 个相同子进程的同步罚因子。若 t_i 是相同的独立的按指数分布的随机变量，则有

$$\lambda_n=\int_0^{\infty}\left[1-(1-e^{-t})^n\right]dt=\int_0^1\frac{1-u^n}{1-u}du$$

$$=\int_0^1\left(\sum_{i=1}^{n}u^{i-1}\right)du=\sum_{i=1}^{n}\frac{1}{i}=H_n \tag{14-7}$$

很清楚，如果罚因子 λ_n 是大的，则同步并行算法的性能就会严重地降低。式（14-6）的罚因子 λ_n 就是第 n 次谐波数 H_n。注意 H_n 象 l_m 一样，随 n 增加而增大。

加速限 S_n 和罚因子 λ_n 是评价同步并行算法性能的两个重要因素。但是，在最坏的情况下，同步并行算法所需时间之长，可以达到令人不能接受的程度。当一个进程为等待一个信号而被封锁时（而这个信号又是一个由失败的进程流出），就可能发生上述情况。

14.3.3 异步并行算法

异步并行算法是具有下列性质的并行算法。

第一，有一个可为所有进程接触的整体变量集合。

第二，当进程的一个子进程完成后，进程首先读一些整体变量，然后根据这些变量之值及上一子进程刚得到的结果，进程修改某些整体变量，接着启动下一子进程或结束其本身。

例：解线性代数方程组

$$Ax = B \tag{14-8}$$

其中 A 为 $n \times n$ 阶矩阵，x 和 B 为 n 维向量。(14-8) 式的迭代形式

$$x = A_1 x + B \tag{14-9}$$

式 (14-8) 的异步雅可比方法是：

第一，算法由 k 个进程 P_1，P_2，…，P_k 组成，每个进程执行 $q = n / k$ 个元素。

第二；一个进程一旦完成其 q 个元素计算，立即修改对应的全局向量 x 的元素，为其他进程释放新值。

第三，该进程又立即对其 q 个元素子集开始重新计算，使用的元素值为重新求值开始时的已知值。

第四，假定某个变量为 $\{x, EPS, E\}$，局部变量为 i，s。其中

$$\begin{cases} x = (x_1, x_2, \cdots, x_n)^T \\ EPS = (\varepsilon_1, \varepsilon_2, \cdots, \varepsilon_n)^T \\ E \end{cases} \tag{14-10}$$

并规定 E 由 P_k 更新。E 的初值为 1，其余整体变量初值为零。在这种定义之下，异步雅可比算法程序如下：

```
process  P_j(j = 0, 1, …, K − 1)
begin
 while  E > ε  do
   begin
```

```
while  i ≤q  do
 begin
     execute  task  J_{jq+i}
   i → i + 1
 e_{jq+i} → |x_{jq+i}^{(s+1)} − x_{jq+i}^{(s)}|
 (||EPS|| ⟶ E,  while  J = K
 end
end
end
```

异步雅可比方法的特点是：一个过程仅使用在循环开始时已知的分量值，且该进程在循环终止时释放所有新值。

从这个例子可看出，异步并行算法的优点是

（1）进程之间的通讯是通过整体变量或共享数据来实现的，进程之间没有明显的从属性。而同步并行算法中所创建的进程之间有明显的从属性；

（2）进程在任何时候都不等待输入，而只根据整体变量里的最新信息来决定继续或结束。在不同时刻有对所有进程都适用的专门输入，而不需要同步化。在任何时刻进程都不被封锁，但执行临界点时除外；

（3）可以避免同步原语开销。

异步并行算法的困难是：由异步迭代算法所产生的序列性能分析是困难的，这也正是我们感兴趣的问题。

14.4 并行算法的正确性与效率

14.4.1 并行算法正确性

正确性对任何算法来说都是需要的。对于串行算法来说，其正确性证明已作过许多研究，也给出各种证明方法，但这些方法大都适用于简单结构的串行程序证明上。要把这些方法推广到并行算法（特别是异步并行程序）上，正确性证明都是很有限的。

所谓正确性是：给定有效输入之后，算法经过有限时间运算并产生出正确答案，那么这个算法（即程序）是正确的。

通常情况下，如果一个算法比较简单，使用指令也很少，在这样情况下，经常使用一些不正规方法（难以描述方法），使我们确信算法各部分确实做了我们期望要做的工作。比如仔细检查某些细节（如循环的初值和终值），并且使用少量小型的实例，人工模拟该算法。虽然这样并不能证明算法是正确的，但对小型程序，这些技术就足够了。

对于大型程序的正确性证明：一般是将大型程序分解成一些较小程序段，如果较小程序段都正确地完成了他们的工作，则整个程序就是正确的，然后回头证明每个程序段的正确性。而对每一个程序段进行严格证明的最有用的技术之一是数学归纳法。

对于复杂的，大型并行算法的正确性证明，目前一般采用如下方法：

（1）通过检验程序里的全局变量在整个程序的执行过程中所满足的一些性质，给出一个并行程序正确性证明。

这种证明手段，主要是通过检查进程间相互作用点（即临界点）前后的全局变量可能转变而获得的。

（2）通过数值分析技术引伸出来的收敛性理论，得到正确性证明。对于数值分析方法来说，大部分均采用这种技术。

例　考虑二点边值问题的许瓦兹交替算法，假设$\Omega=\{x\mid 0<x<1\}$，在Ω内解下面二阶常微分方程：

$$\begin{cases} -y''+q^2y=f(x), & x\in\Omega \qquad (14\text{-}11)\\ y(0)=a \qquad (14\text{-}12)\\ y(1)=b \qquad (14\text{-}13)\end{cases}$$

其中q，a，b为已知常数，$f(x)$为已知函数。

上述问题的一个异步并行算法如下：

第一，将区域〔0，1〕分解成二个子区域：$\Omega_1=\{x\mid 0<x<x_1\}$，$\Omega_2=\{x\mid x_1<x<1\}$。

第二，在Ω_1和Ω_2上建立二个二点边值问题：

$$\begin{cases} -y''+q^2y=f(x), & x\in\Omega_1 \\ y(0)=a \\ y(x_k)=z(x_k) \end{cases} \tag{14-14}$$

$$\begin{cases} -z''+q^2z=f(x), & x\in\Omega_2 \\ z(x_l)=y(x_l) \\ z(1)=b \end{cases} \tag{14-15}$$

其中 Ω_1，Ω_2，x_l，x_k 如图 14-2 所示，且 x_l 和 x_k 称为局部拟边界点。

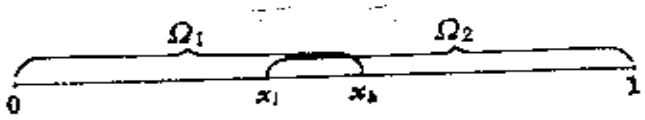


图 14-2

第三，许瓦兹交替算法过程如下：给定初始猜测 $z^{(0)}(x_k)$，则有

$$\begin{cases} -y^{(i+1)''}+q^2y^{(i+1)}=f(x), & x\in\Omega_1 \\ y(0)^{(i+1)}=a \\ y^{(i+1)}(x_k)=z^{(i)}(x_k) \end{cases} \tag{14-16}$$

$$\begin{cases} -z^{(i+1)''}+q^2z^{(i+1)}=f(x) & x\in\Omega_2 \\ z^{(i+1)}(x_l)=y^{(i+1)}(x_l) \\ z^{(i+1)}(1)=b & i=0,\ 1,\ 2,\ \cdots \end{cases} \tag{14-17}$$

利用拟边界点，分别在重叠子区域 Ω_1 和 Ω_2 上解式（14-16）和（14-17）。这样得出的解与原二点边值问题精确解存在多大误差呢？回答如下：

假定 $y^*(x)$ 表示式（14-11）～（14-13）的精确解，则误差满足方程

$$l^{(i)}(x)=y^*(x)-y^{(i)}(x),\quad E^{(i)}(x)=y^*(x)-z^{(i)}(x) \tag{14-18}$$

且有

$$\begin{cases} -e^{(i+1)''}+q^2e^{(i+1)}=0 \quad \text{在}\ \Omega_1\ \text{上} \\ e^{(i+1)}(0)=0 \\ e^{(i+1)}(x_k)=E^{(i)}(x_k) \end{cases} \tag{14-19}$$

及

$$\begin{cases} -E^{(i+1)''}+q^2E^{(i+1)}=0\text{，在}\ \Omega_2\ \text{上} \\ E^{(i+1)}(x_l)=e^{(i+1)}(x_l) \\ E^{(i+1)}(1)=0 \end{cases} \tag{14-20}$$

式（14-19）和（14-20）的第一个方程的特征方程均为 $-\lambda^2+q^2=0$，所以它们的通解分别为

$$e(x)=C_1e^{qx}+C_2e^{-qx} \tag{14-21}$$

$$E(x)=C_1^*e^{qx}+C_2^*e^{-qx} \tag{14-22}$$

将式（14-19），（14-20）中第二，三式分别代入式（14-21）和（14-22），通过解方程得

$$\begin{cases} C_1=\dfrac{-E(x_k)}{e^{-qx_k}-e^{qx_k}} \\ C_2=\dfrac{E(x_k)}{e^{-qx_k}-e^{qx_k}} \end{cases} \tag{14-23}$$

$$\begin{cases} C_1^*=\dfrac{-e(x_l)e^{-q}}{e^{q(1-x_l)}-e^{-q(1-x_l)}} \\ C_2^*=\dfrac{e(x_l)e^{q}}{e^{q(1-x_l)}-e^{-q(1-x_l)}} \end{cases} \tag{14-24}$$

将式（14-23），（14-24）分别代入式（14-21），（14-22），则在 Ω_1 和 Ω_2 内得到

$$e(x)=\frac{e^{-qx}-e^{qx}}{e^{-qx_k}-e^{qx_k}}E(x_k)=\frac{shqx}{shqx_k}E(x_k) \tag{14-25}$$

$$E(x)=\frac{-e^{q(1-x)}+e^{q(1-x)}}{e^{q(1-x_l)}-e^{-q(1-x_l)}}e(x_l)=\frac{shq(1-x)}{shq(1-x_l)}e(x_l) \tag{14-26}$$

按式（14-19），（14-20）边界条件有

$$e^{(i+1)}(x_l)=\frac{shqx_l}{shqx_k}E^{(i)}(x_k)=\frac{shqx_l}{shqx_k}\cdot\frac{shq(1-x_k)}{shq(1-x_l)}e^{(i)}(x) \tag{14-27}$$

$$E^{(i+1)}(x_k)=\frac{shq(1-x_k)}{shq(1-x_l)}e^{(i+1)}(x_l)=\frac{shq(1-x_k)}{shq(1-x_l)}\cdot\frac{shqx_l}{shqx_k}E^{(i)}(x_k) \tag{14-28}$$

由式（14-27）和（14-28）及最大模原理知，在Ω_1和Ω_2内下式成立

$$\begin{cases}|e^{(i+1)}(x)|\leqslant\rho_q(x_l,\ x_k)|E^{(i-1)}(x_k)| & (14\text{-}29)\\ |E^{(i+1)}(x)|\leqslant\rho_q(x_l,\ x_k)|e^{(i)}(x_l)| & (14\text{-}30)\end{cases}$$

其中

$$\rho_q(x_l,\ x_k)=\frac{shqx_l}{shqx_k}\cdot\frac{shq(1-x_k)}{shq(1-x_l)}$$

继续利用式（14-29），（14-30）得到

$$|e^{(i)}(x)|\leqslant[\rho_q(x_l,\ x_k)]^{(i-1)}|E^{(0)}(x_k)|$$

$$|E^{(i)}(x)|\leqslant[\rho_q(x_l,\ x_k)]^{(i-1)}\cdot|E^{(0)}(x_k)|$$

假定区域重叠部分为d时，〔87〕中证明了如下事实。当$x_l=\dfrac{1-d}{2}$时，$\rho_q(x_l,\ x_{l+d})$取最大值$\left[\dfrac{shq\frac{1-d}{2}}{shq\frac{1+d}{2}}\right]^2$且上述方程收敛率为

$$k=-\ln\rho_q\left(\frac{1-d}{2},\ \frac{1+d}{2}\right) \tag{14-31}$$

这说明解二点边值问题（14-11）～（14-13）的许瓦兹交替算法是正确的，即解二点边值问题的异步并行算法是正确的。

14.4.2 并行算法的效率

效率是算法设计中的一个极重要问题，并行算法设计的重要目标之一就是达到比串行算法具有更高的效率。

一般情况下，一个同步并行算法若不加以改造就直接搬到多处理机上运行，将会产生很糟的后果。正如前面第三部分所说，这是因为同步原语本身的执行很费时间，同步化会引起在一个任务中，进程并发执行的堵塞；同时同步化还会引起某些处理机空间和转换。因此，要根据给定问题，针对具体多机系统设计异步并行算法或将同步并行算法改造成异步并行算法。

下面针对式（14-9），设计一个同步并行算法来说明这一问题。式（14-9）迭代式为

$$x^{(l+1)}=A_1x^{(l)}+B \tag{14-32}$$

展开（14-32）有

$$\begin{cases} x_1^{(l+1)}=a'_{11}x_1^{(l)}+a'_{12}x_2^{(l)}+\cdots+a'_{1n}x_n^{(l)} \\ x_2^{(l+1)}=a'_{21}x_1^{(l)}+a'_{22}x_2^{(l)}+\cdots+a'_{2n}x_n^{(l)} \\ \cdots \\ x_n^{(l+1)}=a'_{n1}x_1^{(l)}+a'_{n2}x_2^{(l)}+\cdots+a'_{nn}x_n^{(l)} \end{cases} \tag{14-33}$$

(14-33) 的同步雅可比算法是

（1）算法由 n 个进程组成，每一进程计算一个分量。

（2）所有分量计算完后释放新值，同时开始新迭代步计算。

假定一个迭代步作为一个周期，则同步雅可比方法的一个周期模型为



第一个等待段是由于所有进程在求一次迭代值结束和求下一次迭代值之前的等待。

第二个等待段是由于临界段存在时，为修改和读取当前迭代元素的值而使用的（当一个进程在使用临界段时，如果其他进程也要使用临界段，就必须等待）。

假定使用 n 个进程执行一个完全周期的平均时间为 a_n+b_n。其中 a_n 和 b_n 分别为周期第一部分和第三部分各自平均执行时间。同步雅可比算法由 n 个进程组成，假定第 i 个进程在第一部分执

行时间为随机变量 t_i，且 t_1，t_2，…，t_n 为具有均值 τ 的相同分布随机变量。记

$$T=\max(t_1,\ t_2,\ \cdots,\ t_n)$$

T 的平均值为 $\overline{T}$，则有

$$a_n=\overline{T}$$

如果 t_i 是按同样概率分布函数 F 而分布的随机变量，则

$$a_n=\overline{T}=\int_0^\infty [1-F^n(t)]dt$$

即当 $F(t)=1-e^{-t/\tau}$ 时，F 为常参数 $\frac{1}{\tau}$ 的指数分布，这时有

$$a_n=\int_0^\infty [1-(1-e^{-t/\tau})^n]dt=\tau\int_0^1\frac{1-u^n}{1-u}du$$

$$=\tau\sum_{i=1}^{n}\frac{1}{i}=H_n\tau \tag{14-34}$$

其中 H_n 是调和级数前 n 项，它是该算法的同步罚因子。这时进程由于空载而等待最后完成所耗费的时间由下式给出

$$\frac{1}{n}=(H_1+H_2+\cdots+H_{n-1})/H_n$$

$$=1-\frac{1}{H_n}\approx 1-\frac{1}{\ln n}\to 1,\quad n\to\infty \tag{14-35}$$

下面我们分析未知数 b_n。若 n 个进程执行第二部分的平均执行时间用 C_n 表示。由于在同一时间内，所有进程都有可能同时访问临界段，因此 b_n 可表为

$$b_n=\frac{1}{n}(C_n+2C_1+\cdots+nC_n)=\frac{n+1}{2}C_n \tag{14-36}$$

式（14-35）表明：进程几乎空载而互相等待。这也说明，尽管同步雅可比方法的每一进程所含工作量相同，但由于我们在本文第三部分所阐述的各种原因，使得它们实际计算时间差别较大，即在每一迭代步末尾，进程同步罚因子比较大，从而使得同步雅可比方法效率是低的。

式（14-36）也表明，n 个进程花在临界段执行和等待时间

是比较大的，且随着进程数增加而增加。

如果将同步雅可比方法改造成异步雅可比方法、异步 高斯-塞德尔方法或纯异步方法，将会看到，后三种方法的性能大大优于第一种方法。

综上所述可清楚看出：要在多机系统上设计一个有效的并行算法，必须针对具体问题和系统，对问题进行恰当分解，建立合适的进程数。并且要尽量做到

（1）很小心地使用同步化；

（2）一个任务的并发进程间的通讯应严格地减少到最少；

（3）要尽量减少使用临近段，在使用时必须精选同步 化 方法，使之在临界段所花时间尽可能少。

此外，如果能迅速中断那些不做有效计算的进程，并且由中断而引起的额外开销不是太过分的话，异步并行算法的速度还可以改进。注意，一个进程在给定的时间内是否做有效的计算，可以通过检验某些全局变量现时的内容来确定。因此一旦认识到某些变量的值可以通过一个进程来修改，就可以让进程检查这些变量的新的状况，从而实现中断。一个进程被中断的能力当然会引起此进程在时间上的开销。但是，对于一个给定的异步并行算法和中断方案，开销通常是能够估计出来的。因此，在这种情况下，就有可能决定在算法中是否允许进程中断。

14.5 创建最优的进程数

为了在多处理机上执行给定的任务，首先遇到的问题是到底创建多少个进程为好，这一节作选择最优进程数方面的一些考虑。

首先，研究同步并行算法。注意：执行原语同步化通常要消耗时间，并且罚因子往往随着同步进程数目的增加而增大。因此，根据给定的最大分解作出的同步并行算法，可能是不合乎需要的。例如，有三个任务 J_1，J_2，J_3。其中 J_1，J_2 在 J_3 开始计算之前就完成了计算，并且假设 J_1 和 J_2 所需的时间可用具有平均值 $\bar{t}$ 和标准方差 σ 的正态分布的随机变量来近似表示，而 J_3 所 需 要

的时间是用另外一个具有平均值 $\bar{t}'$ 的随机变量来表示。又假设，两个进程同步所需的同步原语的执行时间是 s，试考虑执行下面三个任务的两种方法。

第一种方法基于最大分解原理，J_1 和 J_2 并行的执行。在二者完成之后，开始 J_3。因此，这个方法所需要的期望时间是

$$\bar{t}+\frac{\sigma}{\sqrt{\pi}}+s+\bar{t}'$$

第二种方法是明显的串行执行 J_1，J_2，J_3 的方法。这种方法所需要的时间期望值是

$$2\bar{t}+\bar{t}'$$

因此，当 $s>\bar{t}-\frac{\sigma}{\sqrt{\pi}}$ 时，既使在多于一个的处理机情况下，使用串行算法也是合算的。这个例子表明，问题的最大分解未必导致解决问题时创建最优的进程数。

现在研究异步并行算法。如同前面所述，许多异步并行算法需要有临界部分。在这些算法中，从进入临界部分开始，进程可能被封锁。所以算法的性能要降低。降低的总量估计如下。分析一个具有 k 个进程 P_1，P_2，…，P_k 的异步并行算法。令 T_i 是进程 P_i 所需的总时间，而 C_i 是这个进程在临界部分所花费的时间。在进程不被封锁的情况下，定义 $\alpha_i=C_i/T_i$，而 α 是 α_1，α_2，…，α_k 的下限。一般，α 的估计值可以用检验进程的程序获得。α 可以是也可以不是 k 的函数。如果 α 是 k 的函数，则随着同时发生的进程数 k 的增加，可能的状态数也增加。这表明，修正全局变量的复杂性随着 k 的增加而增长。因此 α 是 k 的递增函数。我们把 α 写成 $\alpha(k)$。注意，在异步并行算法中，临时部分的执行不能重叠。因此，算法的加速因子至多是

$$\frac{T_1+T_2+\cdots+T_k}{C_1+C_2+\cdots+C_k}\leqslant \max\left(\frac{1}{\alpha_1},\ \frac{1}{\alpha_2},\ \cdots,\ \frac{1}{\alpha_k}\right)$$

$$=\frac{1}{\alpha(k)}$$

加速因子也由 k 限制，这是明显的。所以存在 k 的最优选择。事实上，它被方程式

$$k=\frac{1}{\alpha(k)} \tag{14-37}$$

的最小正数解所限制。上述算法也表明，在异步并行算法中，较大的进程数目不一定有用，除非 α 能保持最小。实际上，最重要的是使用小的临界部分来设计算法，并精选同步化方法，使所花的时间尽可能的小。

在多处理机程序设计中，对于给定的任务，如何去发现同步和异步并行算法的最优进程数，是一个实际的问题。而且这个问题不会容易地得到解决。正如我们所看到的，一个好的选取需要进行大量的分析。这正是困难所在。

参 考 文 献

〔1〕 高庆狮，张祥，王嘉谟。一类递归计算的另一种有效并行算法。计算机应用与应用数学，1974(8)：11～15。

〔2〕 齐东旭，田自贤，张玉心等。曲线拟合的数值磨光方法。数学学报，1975，18(3)：173～184。

〔3〕 冯康等编。数值计算方法。北京：国防工业出版社，1978。

〔4〕 李岳生，齐东旭著。样条函数方法。北京：科学出版社，1979。

〔5〕 慈云桂，胡守仁。亿次级巨型计算机技术发展近况。计算机工程与科学，1980 (2)：1～14。

〔6〕 苏步青，刘鼎元著。计算几何。上海：科学技术出版社，1981。

〔7〕 孙家昶著。样条函数与计算几何。北京：科学出版社，1982。

〔8〕 Heller D. A Survey of Porallel Algorithms in Numerieal Linear Algebra. SIAM Review, 1978, 20(4): 740～777.

〔9〕 Carl DB. A Practical Guide to Splines. New York: Springer-Verlag, 1978.

〔10〕 Greville f. et al. Theory and Applications of Spline Functions. New York: Ap, 1969.

〔11〕 Shedler. G S Parallel Numerical Methods for the Solution of Equation. Comm. of the ACM, 1967, 10: 286.

〔12〕 王德人。非线性方程组迭代解法。上海：科学技术出版社，1979。

〔13〕 奥特加，莱因博尔特。多元非线性方程组迭代解法。朱季纳译。北京：科学出版社，1983。

〔14〕 White R E Parallel Algorithms for Solving Systems of Nonlinear Equations. Comp. Maths. With Appls, 1981, 7: 235～250.

〔15〕 Sherman A. On Newton-Iterative Methods for the Solution of Systems of Nonlinear Equations. SIAM J. Numer Anal, 1978, 15, 755～771.

〔16〕 White R E. Parallel Algorithms for Nonlinear Problems. SIAM J. Alg. Disc. Meth, 1986, 3(1).

〔17〕 White R E. A Nonlinear Parallel Algorithm with Application to the Stefan Problem. SIAM. Numer. Anal, 1986, 23(3).

〔18〕 Oleary D P. White. R E. Multi-Splings of matrices and parallel solution of linear systems. SIAM J, Alg. Disc. Meth, 1985, 6: 630～640.

〔19〕 Henrici P. Discrete Variable Methods in Ordinary Differential Equations. New York, Wiley, 1962.

〔20〕 Gear C W. Numerical Initial Value Problems in Ordinary Differential Equations. Prentice-Hall, 1971.

〔21〕 Miranker W L, Liniger W. Parallel Methods for the Numerical Integ-

ration of Ordinary Differential Equations. Math. Comput, 1967, 21: 303～320.

〔22〕 费景高。线性多步公式的并行计算。计算机工程与科学，1984，4：14～19。

〔23〕 费景高。解常微分方程初值问题的线性多步公式的并行算法。计算数学，1986，8(2)：114～120。

〔24〕 费景高，刘德贵。Stiff常微分方程初值问题的数值解法。计算机应用与应用数学，1978，1(2)：1～40。

〔25〕 Liniger W. A Criteria for A Stability of Linear Multistep Integration Formulae. Computing, 1968, 3: 280～285.

〔26〕 Liniger W, Odeh F. A Stable Accurate Averaging of Multistep Methods for Stiff Differential Equations. IBM J. Res. Develop, 1972, 16: 335～348.

〔27〕 费景高。一类线性隐式多步并行算法。计算机工程与科学，1980(4)：43～58。

〔28〕 祝楚恒，费景高。联合应用 Runge— Kutta 公式与 Adams 公式的积分方法。电子计算机动态，1963(2)：31～35。

〔29〕 费景高。应用插值的数值求解常微分方程组初值问题的分解算法的收敛性和误差阶。数值计算与计算机应用，1984，5(4)：209～218。

〔30〕 费景高。应用Richardson 外插的梯形公式的并行计算。计算机工程与设计，1984，4：37～45。

〔31〕 Steffer H J. Analysis of Discretization Methods for Ordinary Differential Equation。Berlin: Springer-Verlag, 1973.

〔32〕 何爱芳。关于梯形法的局部外插。计算数学，1983，5(3)：27～31

〔33〕 Kung H T. 多处理机同步和异步并行算法。吴功广译。计算机工程与科学，1984(3)：34～42。

〔34〕 费景高。常微分方程初值问题异步并行迭代数值积分法。计算机工程与设计，1983，3：20～25。

〔35〕 Shampine L F, Watts H A. Block Implicit Onestep Methods. Math. Comput, 1969, 23: 731～740.

〔36〕 Watts H A, Shampine L F. A Stable Block Implicit Onestep Methods. BIT, 1972, 12: 252～266.

〔37〕 武汉大学，山东大学。计算方法。北京：人民教育出版社，1979。

〔38〕 Rosser J B. A Runge-Kutta for all Seasons. SIAM Rev, 1967, 9: 417～452.

〔39〕 Worland P B. Parallel Methods for the Numerical Solution of Ordinary Differential Equations. IEEE Trans. Comput, 1976, C (25): 1045～1048.

〔40〕 Birta L G, Abou-Rabia O. Parallel Block Predictor-Conector Methods foc ODE'S. IEEE Trans. Comput, 1987, C(36): 299～311.

[41] Chu M T, Hamilton H. Parallel solution of ODE's by Multi-Block Methods. J. Sci. Stat. Comput, 1987, 8: 342～353.

[42] Nievergelt J. Pacallel Methods for Integrating ODE'S. Comm. ACM, 1964, 7: 731～733.

[43] Hamming R W. Numerical Methods for Scientists and Engineers. 2 nd Ed. Now York: Mc Graw-Hill, 1973.

[44] Morrison D D, Riley J D, Iancanaro J F. Multiple Shooting Methods for Two-Point Boundary Value Problems. Comm. ACM, 1962, 5: 613-614.

[45] Osborne M R. On Shooting Methods for Boundary Value Problems. J. Math. Anal. Apple, 1969, 27: 417～433.

[46] 刘智良。并行差分把靶法。计算机工程与科学，1982，3：21～26。

[47] Keller H B. Nuvnerical Solution of Baundary Value Problems for ODE'S; Survey and Some Recent Result on Difference Methods, Numerical Solution of Boundary Value Problems for ODE'S. New York: Academic Press, 1975.

[48] Travassos R, Kaufman H. Parallel Algorithms for Solving Nanlinear Two-Point Baundary Value Problems Which Arise in Optimal Control. Journal of Optimization Theory and Application, 1980, 30, 53～71.

[49] Lentini M, Pereyra V. A Variable Order Finite Difference Method for Nanlinear Multipaint Baundary Value Problems. Math. Comput, 1974. 28: 981～1004.

[50] Lentini M, Pereyra M. An Adaptive Finite Difference Soluer for Nanlinear Two-Point Boundary Value Problems With Mild Baundary Layers. SIAM J. Number. Anal, 1977, 14: 91～111.

[51] Daniel J W, Martin A J. Numerov's Method, With Deferred Corrections for Two-Point Boundary-Value Problems. SIAM J. Numer. Anal, 1977, 14: 1033～1050.

[52] Cash J R, Singhal A. High Order Methods for the Numerical Solution of Two-Point Boundary Value Problems. BIT, 1982, 22: 184～199.

[53] Skeel R D. A Theoretical Framework for Proving Accuracy Results for Deferred Corrections. SIAM J. Numer. Anal, 1982, 19, 171～196.

[54] Keller H B. Accurate Difference Methods for Nonlinear Two-paint Boundary Value Problems. SIAM J. Numer. Anal, 1974, 11: 305～320.

[55] Cash J R. A Variable Order Deferred Correction Algorithm for the Numerical solution of Nanlinear Two-Point Boundary Vaiue Problems. Comp. Maths. Appls, 1983, 9: 257～265.

[56] Cash J R. Adaptive Runge-Kutta Methods for Nonlinear Two-Point

Boundary Value Problems With mild Baundary Layers。Comp. Maths. Appls, 1985, 11, 605～619.

〔57〕 Evans D J. Alternating Group Explicit Method for the Diffusion Equation. Applied Mathematical Momelling, 1984.

〔58〕 Ortega J. Voigt R G. Solution of Partial Differential Equations on Vector and Parallel Compaters. 1985 Society for Industrial and Applied Mathematics, 1985, 27(2).

〔59〕 苏煜城，吴启光。偏微分方程数值解法。北京：科学出版社，1979。

〔60〕 矢嶋信男，野木达夫。发展方程数值分析。北京：人民教育出版社，1982。

〔61〕 王嘉谟。向量机的使用与并行算法初步。并行算法会议资料，1981。

〔62〕 徐国荣等。多物质可压缩流体力学的欧拉数值方法。数值计算与计算机应用，1980，1(3)：17～24。

〔63〕 北京应用物理与计算数学研究所五室。质点网格法及其应用。中国数学，1978，2：37～45。

〔64〕 徐钟济。蒙特卡罗方法。上海：科学技术出版社，1985。

〔65〕 中国科学院计算中心概率统计组。概率统计计算。北京：科学出版社，1979。

〔66〕 华罗庚。数论导引。北京：科学出版社，1979。

〔67〕 Bhavsar V C, Isaeu J R. Design and Analysis of Parallel Monte Carlo Algorithms. SIAM J. Sci. Stat. Comput, 1987, 8(1)。

〔68〕 国防科技大学计算机研究所。YHFT使用手册，长沙：国防科技大学出版社，1983。

〔69〕 Reed D A, Patrick M L. NASA ICASE Report. NASA Langley Research Center, 1984(6).

〔70〕 Wegner L M. Proc. 1982 Conf. Information Sci. Systems Princeton. New Jersey, 1982(3): 505～509.

〔71〕 Rotem D, Santoro N, Sidney J B. Tech. Rep. No. SCSTR #34. Canada: School of Computer Science Canleton University, 1983.

〔72〕 AKL S G. Notes and Reports in Computer Science and Applied Mathematics. Academic Press. 1985.

〔73〕 武汉大学分布式系统研究组。分布式并行处理系统探索。武汉：武汉大学出版社，1984。

〔74〕 康立山，孙乐林，陈毓屏。解数学物理问题的异步并行算法。北京：科学出版社，1985。

〔75〕 Kang Li-Shan. Parallel Algorithms and Domain Decomprition. WuHan Univercity Press, 1987.

〔76〕 Hwang. K, Briggs F A. Computer Architecture and Parallel Processing. Megrawhill Series in Computer Organization and Architectute, 1986.