第4节　预估–校正算法

§4 预估-校正算法

我们知道线性k步隐式方法虽然具有稳定好、精度高的特点，但是每个隐式方法关于待定的u_{n+k}一般是非线性的，虽然能利用一些迭代法（例如简单迭代法，或Newton迭代法）来求解，但若u_{n+k}的初始近似值选得不好，则其计算量会比利用一次显式公式大得多。 我们既要利用隐式方法的稳定性及精确性，又要利用显式公式的简易性，把两者结合起来，做到取长补短。 办法之一就是先用同阶的显式公式确定较好的迭代初值，然后再按隐式方法迭代一两次达到精度要求。这就是所谓的预估-校正算法（格式）。

解初值问题的线性k步隐式方法形如

$$u_{n+k}+\sum_{j=0}^{k-1}\alpha_j u_{n+j} = h\beta_k f\left(t_{n+k},u_{n+k}\right)+h\sum_{j=0}^{k-1}\beta_j f_{n+j}\,, \qquad \textbf{(4.1)}$$

其中 $f_{n+k}=f(t_{n+k}, u_{n+k})$。

若已求出u_{n+j}, $j=0,1,\cdots,k-1$, 则**(4.1)**关于u_{n+k}为非线性方程，通常用如下迭代法求解：

$$u_{n+k}^{[m+1]}+\sum_{j=0}^{k-1}\alpha_j u_{n+j} = h\beta_k f\left(t_{n+k},u_{n+k}^{[m]}\right)+h\sum_{j=0}^{k-1}\beta_j f_{n+j}\,, \qquad \textbf{(4.2)}$$

$u_{n+k}^{[0]}$ 为给定的迭代初值。显然若 $h<\dfrac{1}{L|\beta_k|}$ 其中**,** L为关于u的**Lipschitz**常数，初值 $u_{n+k}^{[0]}$ 选择适当，则迭代法(4. 2)收敛。

注意：

$$u_{n+k}^{[m+1]}+\sum_{j=0}^{k-1}\alpha_j u_{n+j} = h\beta_k f\left(t_{n+k},u_{n+k}^{[m]}\right)+h\sum_{j=0}^{k-1}\beta_j f_{n+j}\,,$$

$$u_{n+k}^{[m]}+\sum_{j=0}^{k-1}\alpha_j u_{n+j} = h\beta_k f\left(t_{n+k},u_{n+k}^{[m-1]}\right)+h\sum_{j=0}^{k-1}\beta_j f_{n+j}\,,$$

两式相减得：

$$\left|u_{n+k}^{[m+1]}-u_{n+k}^{[m]}\right|\le Lh\left|\beta_k\right|\left|u_{n+k}^{[m]}-u_{n+k}^{[m-1]}\right| \le\cdots\le\; q^m\left|u_{n+k}^{[1]}-u_{n+k}^{[0]}\right|\to 0\quad 0<q<1$$

显然隐式方法(4.1)每一步的计算量取决于迭代法(4.2)的次数，所以选好初值 $u_{n+k}^{[0]}$ 非常重要。最自然的一种方法是用显式多步法计算初值 $u_{n+k}^{[0]}$ ，比如用

$$u_{n+k}^{[0]}+\sum_{j=0}^{k-1}\alpha_j^* u_{n+j} = h\sum_{j=0}^{k-1}\beta_j^* f_{n+j}\,, \tag{4.3}$$

称(4.3)为预估算式（P算式），(4.2)为校正算式（C算式），统称(4.2)，(4.3)为预估-校正算法，简称预-校算法(Predictor-Corrector)。

由于已经用了预估算式，因此，一般来说，校正次数并不很多。
一个极端的情形是允许迭代法(4.2)不断进行，直到不等式成立。

$$\left|u_{n+k}^{[m+1]}-u_{n+k}^{[m]}\right|<\varepsilon$$

其中ε是预先指定的容许误差。

这是一种迭代到收敛的预-校算法。其缺点是，对迭代不加限制，使计算函数值的工作量过大。这种校正次数过多的方法不宜使用。

为此，我们提出另一种限制迭代次数的算法：

$$
\begin{aligned}
&\textbf{P:} \quad u_{n+k}^{[0]}+\sum_{j=0}^{k-1}\alpha_j^* u_{n+j}^{[M]} = h\sum_{j=0}^{k-1}\beta_j^* f_{n+j}^{[M-1]}, \\
&\textbf{E:} \quad f_{n+1}^{[m]} = f\left(t_{n+k}, u_{n+k}^{[m]}\right) \qquad (4.4)\\
&\textbf{C:} \quad u_{n+k}^{[m+1]}+\sum_{j=0}^{k-1}\alpha_j u_{n+j}^{[M]} = h\beta_k f\left(t_{n+k}, u_{n+k}^{[m]}\right)+h\sum_{j=0}^{k-1}\beta_j f_{n+j}^{[M-1]},
\end{aligned}
$$

$$m=0,1,\cdots,M-1$$

首先利用预估算式得到 $u_{n+k}^{[0]}$，然后计算 $f\left(t_{n+k},u_{n+k}^{[0]}\right)$ 接着在使用校正算式，这就完成了一步校正，然后对 $u_{n+k}^{[1]}$ 重复**EC**过程，循环校正**M**次止。则预估一次校正**M**次的算法可即为：

P(EC)^M^=P(EC)…(EC)。

上述预校格式是以最后进行校正结束的。此时已经得到了 $u_{n+k}^{[M]}$，由于未在计算 $f\left(t_{n+k},u_{n+k}^{[M]}\right)$，因此下一步预估算式仍然利用 $f\left(t_{n+k},u_{n+k}^{[M-1]}\right)$ 显然 $u_{n+k}^{[M]}$ 比 $u_{n+k}^{[M-1]}$ 更近似u_{n+k}。因此还可以设计一种算法，每一步算出 $u_{n+k}^{[M]}$ 后，利用它将 $f_{n+1}^{[M]}=f\left(t_{n+k},u_{n+k}^{[M]}\right)$ 算出，供下一步预估算式使用。 这种预估-校正算法记为：**P(EC)^M^E**。

其计算过程如下：

P: $$u_{n+k}^{[0]}+\sum_{j=0}^{k-1}\alpha_j^{*}u_{n+j}^{[M]}=h\sum_{j=0}^{k-1}\beta_j^{*}f_{n+j}^{[M]},$$

E: $$f_{n+k}^{[m]}=f\left(t_{n+k},u_{n+k}^{[m]}\right)$$

C: $$u_{n+k}^{[m+1]}+\sum_{j=0}^{k-1}\alpha_j u_{n+j}^{[M]}=h\beta_k f\left(t_{n+k},u_{n+k}^{[m]}\right)+h\sum_{j=0}^{k-1}\beta_j f_{n+j}^{[M]}, \qquad (4.5)$$

$$m=0,1,\cdots,M-1$$

E: $$f_{n+k}^{[M]}=f\left(t_{n+k},u_{n+k}^{[M]}\right)$$

4.2局部截断误差和局部截断误差主项

现在讨论预估-校正算法的局部截断误差。校正算式(4.2)和预估算式(4.3)的第一和第二特征多项式分别为

$$\rho(\lambda)=\sum_{j=0}^{k}\alpha_j\lambda^j,\alpha_k=1,\quad \sigma(\lambda)=\sum_{j=0}^{k}\beta_j\lambda^j, \tag{4.6}$$

和

$$\rho^*(\lambda)=\sum_{j=0}^{k}\alpha_j^*\lambda^j,\alpha_k^*=1,\quad \sigma^*(\lambda)=\sum_{j=0}^{k}\beta_j^*\lambda^j, \tag{4.7}$$

以L和L^*分别表示由它们确定的校正算式(4.2)和预估算式(4.3)：

$$L[u_{n+k};h]=\sum_{j=0}^{k}\alpha_j u_{n+j}-h\sum_{j=0}^{k}\beta_j f_{n+j}\qquad(\alpha_k=1),$$

$$L^*[u_{n+k};h]=\sum_{j=0}^{k}\alpha_j^* u_{n+j}-h\sum_{j=0}^{k}\beta_j^* f_{n+j}\qquad(\alpha_k^*=1),$$

设它们阶分别为p和$p*$，由(2.37)式知对任意光滑解$u(t)$有

$$L\left[u(t);h\right]= C_{p+1}u^{(p+1)}(t)h^{p+1}+O\left(h^{p+2}\right) \quad \textbf{(4.8)}$$

$$L^*\left[u(t);h\right]= C^*_{p^*+1}u^{(p^*+1)}(t)h^{p^*+1}+O\left(h^{p^*+2}\right) \quad \textbf{(4.9)}$$

设局部化假设成立：$u_{n+j}=u(t_{n+j})$，$j=0, 1, \cdots, k-1$，将(3.8)式用于预估算式得

$$u(t_{n+k})-u_{n+k}^{[0]}=C^*_{p^*+1}u^{(p^*+1)}(t)h^{p^*+1}+O\left(h^{p^*+2}\right) \quad \textbf{(4.10)}$$

$$\sum_{j=0}^{k}\alpha_j u(t_{n+j}) = h\sum_{j=0}^{k}\beta_j f\left(t_{n+j},u(t_{n+j})\right)+L\left[u(t_n);h\right] \quad \textbf{(4.11)}$$

其次，

$$u_{n+k}^{[m+1]}+\sum_{j=0}^{k-1}\alpha_j u_{n+j}^{[M]} = h\beta_k f\left(t_{n+k},u_{n+k}^{[m]}\right)+h\sum_{j=0}^{k-1}\beta_j f_{n+j}^{[M-s]} \quad \textbf{(4.12)}$$

$$m=0,1,\cdots,M-1$$

其中 $s=0$，预估−校正的算法为**P(EC)$^{\mathbf{M}}$E**；

取 $\boldsymbol{s=1}$，预估-校正的算法为**P(EC)$^{\mathbf{M}}$**。

对每一m=0, 1, ···, M−1, 将(4. 11)和(4. 12)相减，并注意到局部化假设得

$$u\left(t_{n+k}\right)-u_{n+k}^{[m+1]}=h\beta_k\left[f\left(t_{n+k},u\left(t_{n+k}\right)\right)-f\left(t_{n+k},u_{n+k}^{[m]}\right)\right]+L\left[u\left(t_n\right);h\right],$$

（4.13）

$$=h\beta_k f_u^{'}\left(t_{n+k},\eta_{n+k,m}\right)\left(u\left(t_{n+k}\right)-u_{n+k}^{[m]}\right)+L\left[u\left(t_n\right);h\right],$$

$$m=0,1,\cdots,M-1$$

其中$h_{n+k,m}$是位于$u(t_{n+k})$，$u_{n+k}^{[m]}$ 之间的某一数。

若p*≥p, 于(4. 13)中，令m=0并将(4. 10)代到右端，得

$$u\left(t_{n+k}\right)-u_{n+k}^{[1]}=C_{p+1}u^{(p+1)}\left(t\right)h^{p+1}+O\left(h^{p+2}\right)+\beta_k f_u^{'}c_{p^*+1}^{*}u^{\left(p^*+2\right)}\left(t_n\right)h^{p^*+2}+O\left(h^{p^*+3}\right)$$

$$=C_{p+1}u^{(p+1)}\left(t\right)h^{p+1}+O\left(h^{p+2}\right)$$

再将上式代入(4. 13) m=1右端，又得关于 $u\left(t_{n+k}\right)-u_{n+k}^{[2]}$ 的同样形式的表达式，

$$u(t_{n+k})-u_{n+k}^{[2]}=h\beta_k f_u' h\left(u(t_{n+k})-u_{n+k}^{[1]}\right)+L\left[u(t_n);h\right]$$
$$=\beta_k f_u'\left(C_{p+1}u^{(p+1)}(t)h^{p+2}+O\left(h^{p+3}\right)\right)+L\left[u(t_n);h\right]$$
$$=\beta_k f_u' C_{p+1}u^{(p+1)}(t)h^{p+2}+O\left(h^{p+3}\right)+L\left[u(t_n);h\right]$$
$$=L\left[u(t_n);h\right]=C_{p+1}u^{(p+1)}(t)h^{p+1}+O\left(h^{p+2}\right)$$

如此下去，可求得一般表达式：

$$u(t_{n+k})-u_{n+k}^{[M]}=C_{p+1}u^{(p+1)}(t)h^{p+1}+O\left(h^{p+2}\right)$$

足见局部截断误差主项和仅用校正算法相同。

若$p^*=p-1$,于**(4.13)**中，令m=**0**并将**(4.10)**代到右端，得

$$u(t_{n+k})-u_{n+k}^{[1]}=L\left[u(t_n);h\right]+\beta_k f_u' C_{p^*+1}^{*}u^{(p^*+1)}(t_n)h^{p^*+2}+O\left(h^{p^*+3}\right)$$
$$=C_{p+1}u^{(p+1)}(t)h^{p+1}+O\left(h^{p+2}\right)+\beta_k f_u' C_p^{*}u^{(p)}(t_n)h^{p+1}+O\left(h^{p+2}\right)$$
$$=\left[\beta_k f_u' C_p^{*}u^{(p)}(t_n)+C_{p+1}u^{(p+1)}(t_n)\right]h^{p+1}+O\left(h^{p+2}\right)$$

所以当M=1时局部截断误差主项的阶和校正算法相同，而主项系数不同。但当 $\boldsymbol{M}\geqslant 2$，

$$u(t_{n+k})-u_{n+k}^{[2]}=h\beta_k f_u' h\left(u(t_{n+k})-u_{n+k}^{[1]}\right)+L\left[u(t_n);h\right]$$

$$=\beta_k f_u' h\left(\left[\beta_k f_u' C_p^* u^{(p)}(t)+C_{p+1}u^{(p+1)}(t)\right]h^{p+1}+O\left(h^{p+2}\right)\right)\ +L\left[u(t_n);h\right]$$

$$=\beta_k f_u'\left(\left[\beta_k f_u' C_p^* u^{(p)}(t)+C_{p+1}u^{(p+1)}(t)\right]h^{p+2}+O\left(h^{p+3}\right)\right)+L\left[u(t_n);h\right]$$

$$=L\left[u(t_n);h\right]=C_{p+1}u^{(p+1)}(t)h^{p+1}+O\left(h^{p+2}\right)$$

预-校算法的局部截断误差主项和校正算法相同。

若$\boldsymbol{p^*}=p-2$，仍于(4.13)中，令m=0并将(4.10)代到右端，得

$$u(t_{n+k})-u_{n+k}^{[1]}=L\left[u(t_n);h\right]+\beta_k f_u' C_{p^*-1}^* u^{(p-1)}(t_n)h^p+O\left(h^{p+1}\right)$$

$$=C_{p+1}u^{(p+1)}(t)h^{p+1}+O\left(h^{p+2}\right)+\beta_k f_u' C_{p-1}^* u^{(p-1)}(t_n)h^p+O\left(h^{p+1}\right)$$

$$=\ \beta_k f_u' c_{p-1}^* u^{(p-1)}(t_n)h^p+O\left(h^{p+1}\right)$$

所以当M=1时，局部截断误差主项的阶比校正算法低一阶，再将它代到**(4.13)(*m*=1)**右端, 则

$$u(t_{n+k})-u_{n+k}^{[2]}=h\beta_k\bar{f}_u'h\left(u(t_{n+k})-u_{n+k}^{[1]}\right)+L\left[u(t_n);h\right]$$
$$=\beta_k\bar{f}_u'h\left(\beta_k f_u' C_{p-1}^* u^{(p-1)}(t)h^p+O\left(h^{p+1}\right)\right)+L\left[u(t_n);h\right]$$
$$=\beta_k^2\bar{f}_u' f_u' C_{p-1}^* u^{(p-1)}(t)h^{p+1}+O\left(h^{p+2}\right)\ +L\left[u(t_n);h\right]$$
$$=\left(C_{p+1}u^{(p+1)}(t)+\beta_k^2\bar{f}_u' f_u' C_{p-1}^* u^{(p-1)}(t)\right)h^{p+1}+O\left(h^{p+2}\right)$$

当M=2时，局部截断误差主项的阶和校正算法相同，而主项系数不同。若继续将所得式子代到**(4.13)**，则当$M\geqslant 3$时，

$$u(t_{n+k})-u_{n+k}^{[3]}=\beta_k\tilde{f}_u'h\left(u(t_{n+k})-u_{n+k}^{[2]}\right)+L\left[u(t_n);h\right]$$
$$=h\beta_k\tilde{f}_u'\left(C_{p+1}u^{(p+1)}(t)+\beta_k^2\bar{f}_u' f_u' C_{p-1}^* u^{(p-1)}(t)\right)h^{p+2}+O\left(h^{p+3}\right)$$
$$+L\left[u(t_n);h\right] \qquad \textbf{(4.14)}$$
$$=L\left[u(t_n);h\right]=C_{p+1}u^{(p+1)}(t)h^{p+1}+O\left(h^{p+2}\right)$$

局部截断误差主项又和校正算法相同。

一般地，设$p^*\geqslant 0, p\geqslant 1, M\geqslant 1$, 当$p^*\geqslant p, p\geqslant 1, M\geqslant 1$,预估与校正算法相同；$p^*=p-q$，$0<p\leqslant p$, 若$M\geqslant q+1$，则预估与校正算法相同

若$M=q$,，则阶相同但系数不相同

若$M\leqslant q$，则具有$Kh^{p-q+M+1}+O(h^{p-q+M+2})$

总之，没有必要取$p^*>p$,而取$p^*=p-M(M\geqslant 1)$是优越的。

若取$p^*=p$是有好处的，可以不需要解的高阶导数就可以估出预估-校正算法的局部截断误差主项。

事实上，由于

$$u(t_{n+k})-u_{n+k}^{[0]}=C_{p^*+1}^{*}u^{(p^*+1)}(t)h^{p^*+1}+O\left(h^{p^*+2}\right)$$

$$u(t_{n+k})-u_{n+k}^{[M]}=C_{p+1}u^{(p+1)}(t)h^{p+1}+O\left(h^{p+2}\right)$$

两式相减，

$$\left(C_{p+1}^{*}-C_{p+1}\right)u^{(p+1)}(t)h^{p+1}=u_{n+k}^{[M]}-u_{n+k}^{[0]}+O\left(h^{p+2}\right)$$

舍去$0(h^{p+2})$，便给出局部截断误差主项的近似表达式：

$$C_{p+1}u^{(p+1)}(t)h^{p+1}\approx\frac{C_{p+1}}{C_{p+1}^{*}-C_{p+1}}\left(u_{n+k}^{[M]}-u_{n+k}^{[0]}\right) \qquad (4.15)$$

4. 3选步长和改善精度

误差公式(4.15)有两个重要的主要用途：

（1）主要用于控制步长的选取。

根据不等式

$$\left| \frac{C_{p+1}}{C_{p+1}^{*} - C_{p+1}} \left(u_{n+k}^{[M]} - u_{n+k}^{[0]} \right) \right| \le \varepsilon$$

其中ε是误差允许值。

若上述不等式成立，应将步长h缩小重新计算，直至不等式成立。若不等式左边比ε小的很多，则可放大h，使不等式左边既不比ε小的很多，又使不等式成立。实际计算时可编一段程序选步长，按选好的步长算若干步后再修改步长，不必每步都修改。

值得注意的是，步长h不只按误差要求确定，还应满足绝对或相对稳定性条件。确定预估校正算法的绝对或相对稳定域的方法与§3相同。

（2）改善解的精度

由误差公式(4.15)不难推出

$$C_{p+1}^{*}u^{(p+1)}(t)h^{p+1} = C_{p+1}^{*}u^{(p+1)}(t_{n-1})h^{p+1} + O(h^{p+2})$$

$$= \frac{C_{p+1}^{*}}{C_{p+1}^{*} - C_{p+1}}\left(u_{n+k-1}^{[M]} - u_{n+k-1}^{[0]}\right) + O(h^{p+2})$$

取修改值

$$\hat{u}_{n+k}^{[0]} = u_{n+k}^{[0]} + \frac{C_{p+1}^{*}}{C_{p+1}^{*} - C_{p+1}}\left(u_{n+k-1}^{[M]} - u_{n+k-1}^{[0]}\right) \qquad (4.16)$$

代替 $u_{n+k}^{[0]}$，同理由误差公式(4. 15)取修改值

$$\hat{u}_{n+k}^{[M]} = u_{n+k}^{[M]} + \frac{C_{p+1}^{*}}{C_{p+1}^{*} - C_{p+1}}\left(u_{n+k-1}^{[M]} - u_{n+k-1}^{[0]}\right) \qquad (4.17)$$

代替 $u_{n+k}^{[M]}$。

预估-校正算法举例

一个好的预估-校正计算方案应该计算稳定，具有所需精度并且节约计算量。 下面列举几个常见的预估-校正算法。

例1 **Adams**四阶预估-校正算法。取四阶**Adams**外插法为预估算法，四阶**Adams**内插法为校正算法，即

$$\textbf{P:}\quad u_{n+4}-u_{n+3}+\frac{h}{24}\left(55f_{n+3}-59f_{n+2}+37f_{n+1}-9f_{n}\right)$$

$$\textbf{C:}\quad u_{n+4}-u_{n+3}+\frac{h}{24}\left(9f_{n+4}+19f_{n+3}-5f_{n+2}+f_{n+1}\right)$$

误差主项系数为：$C_5=\dfrac{6802}{24\times 60}$，$C_5^*=\dfrac{527}{24\times 60}$

误差主项：

$$C_5u^{(5)}(t)h^5\approx\frac{C_5}{C_5^*-C_5}\left(u_{n+4}^{[M]}-u_{n+4}^{[0]}\right)=-\frac{19}{270}\left(u_{n+4}^{[M]}-u_{n+4}^{[0]}\right)$$

最常用的方案是PECE，其绝对稳定域是(−1.25,0).

例2 **Milne**方法。显然线性多步法由其第一、第二特征多项式唯一确定。考虑预估运算**P**和校正运算**C**，设第一、第二特征多项式：

P: $\rho^*(\lambda)=\lambda^4-1 \qquad \sigma^*(\lambda)=\frac{4}{3}\left(2\lambda^3-\lambda^2+2\lambda\right)$

C: $\rho(\lambda)=\lambda^2-1 \qquad \sigma(\lambda)=\frac{1}{3}\left(\lambda^2+4\lambda+1\right)$

其算法**P**是四步四阶，**C**是二步四阶为，为了方便也可写成：

C: $u_{n+4}-u_{n+2}+\frac{h}{3}\left(f_{n+4}+4f_{n+3}+f_{n+2}\right)$

误差主项系数为：$C_5=-\frac{1}{90}, C_5^*=\frac{14}{45}$。

由于$P*=P(=4)$，故任一种预估-校正计算方案的误差主项：

$$C_5u^{(5)}\left(t_{n+k}\right)h^5\approx-\frac{1}{29}\left(u_{n+4}^{[M]}-u_{n+4}^{[0]}\right)$$

其**PECE**方案称为**Milne**算法，计算公式为：

P: $$u_{n+4}^{[0]} - u_{n}^{[1]} + \frac{4h}{3}\left(2f_{n+3}^{[1]} - f_{n+2}^{[1]} + 2f_{n+1}^{[1]}\right),$$

E: $$f_{n+4}^{[0]} = f\left(t_{n+4}, u_{n+4}^{[0]}\right),$$

C: $$u_{n+4}^{[1]} - u_{n+2}^{[1]} + \frac{h}{3}\left(f_{n+4}^{[0]} + 4f_{n+3}^{[1]} + f_{n+2}^{[1]}\right),$$

E: $$f_{n+4}^{[1]} = f\left(t_{n+4}, u_{n+4}^{[1]}\right),$$

其绝对稳定域是(−0.8，−0.3)。

例3 **Hamming**方法。

预估运算取例2的**P**和校正运算**C**的设第一、第二特征多项式为

C: $\rho(\lambda)=\lambda^3-\frac{8}{9}\lambda^2+\frac{1}{8}$ $\sigma(\lambda)=\frac{8}{3}\left(\lambda^3+2\lambda^2-\lambda\right)$

方法的阶P=4，误差主项系数为： $C_5=-\frac{1}{40}$

预估-校正计算方案的误差主项：

$$C_5^* u^{(5)}\left(t_{n+k}\right)h^5 \approx \frac{112}{121}\left(u_{n+k-1}^{[M]}-u_{n+k-1}^{[0]}\right)$$

$$C_5 u^{(5)}\left(t_{n+k}\right)h^5 \approx -\frac{9}{121}\left(u_{n+k}^{[M]}-u_{n+k}^{[0]}\right)$$

用M表示(4.16)或(4.17)定义的修正（***modifier***），则**Hamming**算法的计算过程为**PMPCME**：

P: $$u_{n+4}^{[0]}-\hat{u}_{n}^{[1]}+\frac{4h}{3}\left(2f_{n+3}^{[1]}-\hat{f}_{n+2}^{[1]}+2\hat{f}_{n+1}^{[1]}\right),$$

M: $$\hat{u}_{n+4}^{[0]}=u_{n}^{[0]}+\frac{112}{121}\left(u_{n+3}^{[1]}-u_{n+3}^{[0]}\right)$$

E: $$\hat{f}_{n+4}^{[0]}=f\left(t_{n+4},\hat{u}_{n+4}^{[0]}\right),$$

C: $$u_{n+4}^{[1]}-\frac{9}{8}\hat{u}_{n+3}^{[1]}+\frac{1}{8}\hat{u}_{n+1}^{[1]}=\frac{3h}{8}\left(\hat{f}_{n+4}^{[0]}+2\hat{f}_{n+3}^{[1]}-\hat{f}_{n+2}^{[1]}\right),$$

M: $$\hat{u}_{n+4}^{[1]}=u_{n}^{[1]}-\frac{9}{121}\left(u_{n+4}^{[1]}-u_{n+4}^{[0]}\right)$$

E: $$\hat{f}_{n+4}^{[1]}=f\left(t_{n+4},\hat{u}_{n+4}^{[1]}\right),$$

完 了