

# 新息预报程序的进一步探讨

陈 兴 钩

(数 学 系)

## 摘 要

本文在文〔1〕对新息预报程序设计分析的基础上, 进一步得到了一套一维存贮的公式。这些公式意义明了, 对新息预报程序设计颇有帮助。当然也大大节约存贮单元。同时, 本文还修正文〔1〕中有关 $\gamma_{kj}(y)$ 公式的一个错误。

## (一) 新息预报公式及其总的计算步骤<sup>〔2〕</sup>

假设已知  $w_1, \dots, w_N$  的数学模型为 ARMA  $(p, q)$ ,  $p$  是自回归阶数,  $q$  是滑动平均阶数,  $\varphi_1 \dots \varphi_p$  是自回归参数。令  $Q = \max(p, q)$

今欲求  $w_{N+l}$  的严格线性最小方差预报:  $w_{N+l}, l = 1, 2, \dots, NN$ 。NN 为预报总的长度。首先作线性变换:

$$y_k = \begin{cases} w_k & k \leq Q \\ \varphi(B)w_k = w_k - \sum_{i=1}^p \varphi_i w_{k-i} & k > Q \end{cases}$$

则求新息序列的公式是:

$$\epsilon_k = \begin{cases} y_k & k=1 \\ y_k - \sum_{j=1}^{K-1} C_{Kj} \epsilon_j & 2 \leq k \leq Q \\ y_k - \sum_{j=k-Q}^{K-1} C_{Kj} \epsilon_j & k > Q \end{cases}$$

其中

$$C_{kj} = \begin{cases} \left[ r_{kj}(y) - \sum_{i=1}^{j-1} C_{ji} C_{ki} r_{ii}(\epsilon) \right] [r_{jj}(\epsilon)]^{-1} & 2 \leq k \leq Q \\ & 1 \leq j \leq k-1 \\ \left[ r_{kj}(y) - \sum_{i=k-q}^{j-1} C_{ji} C_{ki} r_{ii}(\epsilon) \right] [r_{jj}(\epsilon)]^{-1} & k > Q \\ & k-q \leq j \leq k-1 \end{cases}$$

$$r_{kk}(\epsilon) = \begin{cases} r_{kk}(y) & k=1 \\ r_{kk}(y) - \sum_{j=1}^{k-1} C_{kj}^2 r_{jj}(\epsilon) & 2 \leq k \leq Q \\ r_{kk}(y) - \sum_{j=k-q}^{k-1} C_{kj}^2 r_{jj}(\epsilon) & k > Q \end{cases}$$

$$r_{kj}(y) \triangleq E y_k y_j, \quad r_{jj}(\epsilon) \triangleq E \epsilon_j^2$$

且上列各式中, 当求和号上限小于下限时, 约定该和式为零.

新息预报公式是:

$$w_{N+l, N} = \begin{cases} \sum_{j=1}^N C_{N+l, j} \epsilon_j & N+l \leq Q \\ \sum_{j=1}^p \varphi_j w_{N+l-j, N} + \sum_{l=N+l-q}^N C_{N+l, j} \epsilon_j & N+l > Q \end{cases}$$

其中  $w_{N+l-j, N} = w_{N+l-j}$ , 当  $l-j \leq 0$  时. 且当求和号上限小于下限时该和约定为零.

新息预报公式应用于实际时, 应考虑以下四种情况, 且一般样本长度  $N > Q$ .

当  $p > 0, q > 0$  时

$$w_{N+l, N} = \begin{cases} \sum_{j=1}^p \varphi_j w_{N+l-j, N} + \sum_{j=N+l-q}^N C_{N+l, j} \epsilon_j & 0 < l \leq q \\ \sum_{j=1}^p \varphi_j w_{N+l-j, N} & l > q \end{cases}$$

当  $p = 0, q > 0$  时

$$w_{N+l, N} = \begin{cases} \sum_{j=N+l-q}^N C_{N+l, j} \epsilon_j & 0 < l \leq q \\ 0 & l > q \end{cases}$$

当  $p > 0, q = 0$  时

$$w_{N+l, N} = \sum_{j=1}^p \varphi_j w_{N+l-j, N} \quad l > 0$$

当  $p = 0, q = 0$  时

$$w_{N+l, N} \equiv 0 \quad l > 0$$

可见,若已知  $\{w_t\}$  是 ARMA( $p, q$ ) 序列,欲对未来作预报,首先作线性变换得序列  $\{y_t\}$ , 并求出  $\gamma_{kj}(y)$ , 然后按以下顺序计算  $C_{kj}, \gamma_{kk}(\epsilon), \epsilon_k$ .

$$\gamma_{11}(\epsilon) \rightarrow \epsilon_1; \rightarrow C_{21}, \rightarrow \gamma_{22}(\epsilon) \rightarrow \epsilon_2; \rightarrow \begin{matrix} C_{31} \\ C_{33} \end{matrix} \rightarrow \gamma_{33}(\epsilon) \rightarrow \epsilon_3; \rightarrow \begin{matrix} C_{41} \\ C_{43} \end{matrix} \rightarrow \dots$$

## (二) 关于 $\gamma_{kj}(y)$ 的计算公式及计算程序

$\gamma_{kj}(y)$  可以适时计算,也可利用  $y_k$  和  $w_k$  的线性关系及  $\{w_k\}$  的平稳性,由以下公式\*计算:

$$\gamma_{kj}(y) \triangleq E y_k y_j = \begin{cases} E w_k w_j = \gamma_{k-j}(w) & j \leq k \leq Q \quad (1) \\ E y_k w_j = \varphi(B) \gamma_{k-j}(w) & j \leq Q < k \quad (2) \\ E y_k y_j = E y_k w_j - \sum_{i=1}^p \varphi_i E y_k w_{j-i} & Q < j \leq k \quad (3) \\ = \varphi(B) \varphi(B^{-1}) \gamma_{k-j}(w) \end{cases}$$

其中  $\varphi(B) = 1 - \varphi_1 B - \varphi_2 B^2 - \dots - \varphi_p B^p$

$$\varphi(B^{-1}) = 1 - \varphi_1 B^{-1} - \varphi_2 B^{-2} - \dots - \varphi_p B^{-p}$$

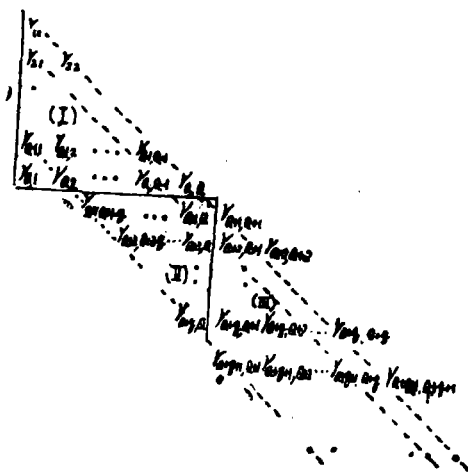
$$B^i w_k = w_{k-i}, \quad i = \pm 1, \pm 2, \dots$$

显然, (1) 和 (2) 成立. (3) 式证明如下:

$$\begin{aligned} \gamma_{ki}(y) &= E y_k y_i \\ &= E y_k \varphi(B) w_j \\ &= E(y_k w_j) - \sum_{i=1}^p \varphi_i E y_k w_{j-i} \\ &= E\left(w_k - \sum_{l=1}^p \varphi_l w_{k-l}\right) w_j - \sum_{i=1}^p \varphi_i E\left(w_k - \sum_{l=1}^p \varphi_l w_{k-l}\right) w_{j-i} \\ &= \gamma_{k-j}(w) - \sum_{l=1}^p \varphi_l \gamma_{k-j-l}(w) - \sum_{i=1}^p \varphi_i \gamma_{k-j+i}(w) + \sum_{i=1}^p \sum_{l=1}^p \varphi_i \varphi_l \gamma_{k-l-j+i}(w) \\ &= \left(1 - \sum_{l=1}^p \varphi_l B^l\right) \gamma_{k-j}(w) - \sum_{i=1}^p \varphi_i B^{-i} \gamma_{k-j}(w) + \sum_{i=1}^p \sum_{l=1}^p \varphi_i \varphi_l B^{-i} B^l \gamma_{k-j}(w) \\ &= \left(1 - \sum_{l=1}^p \varphi_l B^l\right) \gamma_{k-j}(w) - \sum_{i=1}^p \varphi_i B^{-i} \left(1 - \sum_{l=1}^p \varphi_l B^l\right) \gamma_{k-j}(w) \\ &= \left(1 - \sum_{l=1}^p \varphi_l B^l\right) \left(1 - \sum_{i=1}^p \varphi_i B^{-i}\right) \gamma_{k-j}(w) \\ &= \varphi(B) \varphi(B^{-1}) \gamma_{k-j}(w) \end{aligned}$$

\* 文 [1] 中此公式有错

由(一)的公式可以看出,我们在计算  $C_{kj}$ ,  $\gamma_{kh}(\epsilon)$  时所需要用到的  $\gamma_{ki}$  分布如下图:



当  $j \leq k < Q$  时,由(1)式计算,所需计算的个数随  $k$  的增加而增加,组成三角形(I);当  $k > Q$  时,所需计算的个数不变,恒为  $q+1$ . 并且,当  $j \leq Q < k$  时,要用(2)式计算,也组成三角形(II);当  $Q < j \leq k$  时,要用(3)式计算,其结果组成一个台形(III).

因为  $\gamma_{ki}(y)$  是按行计算,且每行仅被引用一次,所以可用一个一维组  $Ry(I) I = 0, 1, \dots, Q$  来存贮就足够了.

又注意到  $\{w_i\}$  是平稳的,对每一种情况,  $\gamma_{ki}(y)$  都只是  $k-j$  的函数,与  $k, j$  的取值无关. 这说明在每一种情况中,同一斜虚线上的  $\gamma_{ki}(y)$  的值是相同的. 所以在三角形(I)中,只须计算第一列的值,即

$$\begin{aligned} \gamma_l(y) &= \gamma_l(w) \\ l &= 0, 1, 2, \dots, Q-1, \end{aligned}$$

在三角形(II)中,只须计算第一行的值,即

$$\gamma_{Q+1,j}(y) = \varphi(B) \gamma_{Q+1-j}(w) = \gamma_{Q+1-j}(w) - \sum_{i=1}^p \varphi_i \gamma_{|Q+1-j-i|}(w)$$

$$j = Q+1-q, \dots, Q.$$

在台形(III)中,只须计算第一列的值,即

$$\gamma_{k,Q+1}(y) = E y_k w_{Q+1} - \sum_{i=1}^p \varphi_i E y_k w_{Q+1-i}$$

$$k = Q+1, \dots, Q+q+1.$$

综上所述,计算  $C_{kj}$ ,  $\gamma_{kh}(\epsilon)$  所需用到的  $\gamma_{ki}(y)$  可分二步来完成.

第一步,先按以下顺序计算

$$\gamma_l(w), l=0, 1, 2, \dots, p+q$$

$$S(l) = \gamma_l(w) - \sum_{i=1}^p \varphi_i \gamma_{|l-i|}(w), \quad l=0, 1, \dots, p+q$$

$$SS(l) = S(l) - \sum_{i=1}^p \varphi_i \cdot S(l+i) \quad l=0, 1, \dots, q$$

第二步, 在总的计算过程中, 引用以上结果, 随时求出所要用的  $\gamma_l(y)$ . 即

当  $1 \leq k \leq Q$  时,  $\gamma_l(y) = \gamma_l(w)$ ,  $l=0, 1, \dots, Q-1$ ,

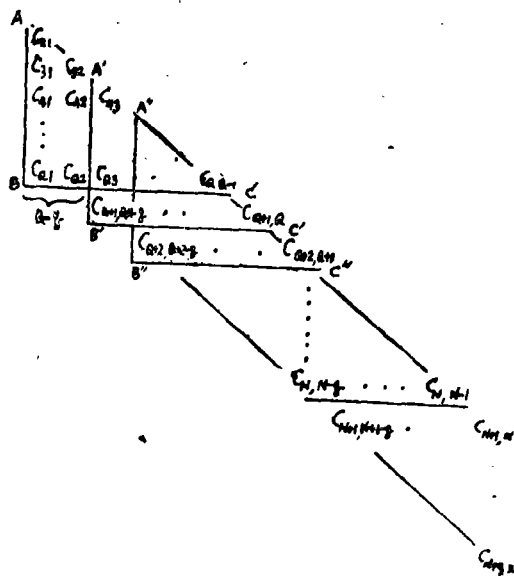
当  $k=Q+1$  时,  $\gamma_l(y) = S(q-l)$ ,  $l=0, 1, \dots, q-1$ ,

当  $k=Q+1+l$  时,  $\gamma_{q-l}(y) = SS(l)$   $l=0, 1, \dots, q$ ,

这样就实现了只用一维数组  $RY(I)$   $I=0, 1, \dots, Q$  来存贮  $\gamma_{kj}(y)$ , 并且对每一  $k$ , 所要用的  $\gamma_{kj}(y)$  的存贮都是从头开始的, 如  $k=Q+1$ ,  $\gamma_{k,k-q}(y)$  存于  $RY(0)$ ,  $\gamma_{k,k-q+1}(y)$  存于  $RY(1)$  等等.

### (三) 关于 $C_{kj}$ 一维存贮的几个对应公式

在(一)的公式中,  $C_{kj}$  的计算是最复杂的, 把所要求的  $C_{kj}$  都排起来是一个变带宽的稀疏矩阵, 如下图.



如果在程序设计时, 按此图形采用二维存贮, 需占用  $(N+q) \times N$  个单元, 那将是很浪费, 以致没有多少实用价值. 注意到  $C_{kj}$  是按行的次序参加运算, 每一元素只被引用一次, 这使我们可考虑用一个一维数组来存贮. 下面分三种情况讨论之:

(1) 当  $1 < k \leq Q$  时, 对每一个  $k$ , 需计算  $C_{k,1} \dots C_{k,k-1}$ , 而每一个  $C_{kj}(j > 1)$  的计算, 又

要用到  $C_{j1}, \dots, C_{j,j-1}$  与  $C_{k,1}, \dots, C_{k,j-1}$ . 计算量是随  $k$  的增加而增加的, 并且计算  $C_{k,1} \dots C_{k,k-1}$  时, 要用到  $k$  行以前按三角形排列的所有元素. 如为计算  $k=Q$  行的  $C_{kj}$ , 要用到第  $Q$  行以前已求得的所有元素, 并把刚求的  $C_{Qj}, j=\dots, Q-1$ , 存于第  $Q$  行. 这时三角形  $ABC$  上元素共有  $Q(Q-1)/2$  个.

为把三角形  $ABC$  上的  $C_{kj}$  按行的次序贮存于一个一维数组  $G(I)$  中, 其对应公式为:

$$G\left(-\frac{(k-1)(k-2)}{2} + j\right) \Leftarrow C_{kj} \quad (4)$$

$$k=2, \dots, Q, \quad j=1, \dots, k-1.$$

可见在此情况下, 新息公式可改写为:

$$\left. \begin{aligned} G\left(-\frac{(k-1)(k-2)}{2} + j\right) &= \left[ \gamma_{k-j}(w) - \sum_{i=1}^j G\left(-\frac{(j-1)(j-2)}{2} + i\right) \right. \\ &\quad \left. G\left(-\frac{(k-1)(k-2)}{2} + i\right) \gamma_{ii}(\epsilon) \right] \left[ \gamma_{ii}(\epsilon) \right]^{-1} \\ &\quad j=1, 2, \dots, k-1. \\ \gamma_{kk}(\epsilon) &= \gamma_0(w) - \sum_{j=1}^{k-1} \left[ G\left(-\frac{(k-1)(k-2)}{2} + j\right) \right]^2 \gamma_{jj}(\epsilon) \\ G_k &= w_k - \sum_{j=1}^{k-1} G\left(-\frac{(k-1)(k-2)}{2} + j\right) \epsilon_j \end{aligned} \right\} \quad (5)$$

(2) 当  $Q < k \leq N$  时, 对每一个  $k$ , 仅需计算  $C_{k,k-q}, \dots, C_{k,k-1}$  共  $q$  个元素. 计算量不再随  $k$  的增加而增加了, 并且计算每一个  $C_{kj}$  时, 要依次引用  $C_{j,k-q}, \dots, C_{j,j-1}$  与  $C_{k,1} \dots C_{k,j-1}$ . 将计算第  $k$  行所有元素时所引用的  $C_{ji}$  全部排列起来, 也是一个三角形. 如  $A'B'C'$ ,  $A''B''C''$  等. 其元素个数不变, 恒为  $\frac{q(q+1)}{2}$ .

因此, 当  $k=Q+1$  时, 为计算  $C_{kj} \quad k-q \leq j \leq k-1$ , 应该先把三角形  $ABC$  中前  $Q-q$  列的元素去掉, 而把留下的元素按行的次序重新编号. 因为此时  $C_{kj} \quad k=2, \dots, Q, j=1, \dots, k-1$ , 已存放于  $G(I)$  中了, 要在  $G(I)$  中实现上述要求, 其对应公式为:

$$G\left(-\frac{j(j-1)}{2} + i\right) \Leftarrow G\left(-\frac{(Q-q)(Q-q+1)}{2} + j(Q-q) + \frac{j(j-1)}{2} + i\right) \quad (6)$$

$$j=1, 2, \dots, q-1, \quad i=1, 2, \dots, j$$

而当  $k$  从  $Q+1$  开始, 每增加 1 时, 三角形要相应下滑一行. 如由  $\triangle A'B'C' \rightarrow \triangle A''B''C''$  等等. 这相当于要在已有一维数组  $G(I)$  中取消  $\triangle A'B'C'$  的第一列元素, 并增添新的一行底边  $B''C''$ , 其对应公式, 只须在 (6) 式中令  $Q-q=1$  即得:

$$G\left(-\frac{j(j-1)}{2} + i\right) \Leftarrow G\left(1+j + \frac{j(j-1)}{2} + i\right) \quad (7)$$

$$j=1, 2, \dots, q-1, \quad i=1, 2, \dots, j$$

(3) 当  $k > N$  时, 由预报公式  $\omega_{k,N}$  知, 这时还需要计算  $C_{k,k-q} \cdots C_{k,N}$ . 显然计算方法同 (2), 只是当  $k$  增加时, 所需计算的  $C_{ki}$  的个数递减了. 当  $k-q > N$  时, 即预报步长  $l > q$  时,  $C_{ki}$  就不必计算了.

总之, 由以上分析可见, 对  $C_{kj}$  作一维存贮处理, 至多只需占用  $Q(Q+1)/2$  个单元. 由于上述三个对应公式, 意义明了, 对求  $C_{kj}$  的程序设计是颇有帮助的. 事实上, 当完成计算  $1 < k \leq Q$  之  $C_{kj}$ ,  $\gamma_{kk}(\epsilon)$  及  $\epsilon_k$  之后, 对  $k > Q$  时, 每计算一个行  $k$  的  $C_{kj}$ ,  $j = 1, \cdots, k-1$  之前, 必须先利用对应的 (6) 或 (7) 式, 对原有的一维数组  $G(I)$  进行“下滑”处理, 使  $G(I)$  变成按行次序存贮三角形  $A'B'C'$  (或三角形  $A''B''C''$ ) 中前  $q-1$  行的元素. 然后遵循  $1 < k \leq Q$  情况下计算  $C_{kj}$  的规律进行运算, 这时计算出的  $C_{ki}$  ( $k > Q$ ) 都应存贮在下滑后新三角形的最底行 (即第  $q$  行), 这相当于应存贮于一维数组  $G(I)$  中的  $G\left(\frac{q(q-1)}{2} + j\right)$   $j = 1, 2, \cdots, q$ .

可见, 当  $k > Q$  时,  $C_{kj}$ ,  $\gamma_{kk}(\epsilon)$  及  $\epsilon_k$  的公式可改写为:

$$\left. \begin{aligned} G\left(\frac{q(q-1)}{2} + j\right) &= \left[ \gamma_{j-1}(y) - \sum_{i=1}^{j-1} G\left(\frac{(j-1)(j-2)}{2} + i\right) G\left(\frac{q(q-1)}{2} + i\right) \gamma_{ji}(\epsilon) \right] \\ &\quad \left[ \gamma_{ji}(\epsilon) \right]^{-1} \quad j = 1, 2, \cdots, q \\ \gamma_{kk}(\epsilon) &= \gamma_q(y) - \sum_{j=1}^q \left[ G\left(\frac{q(q-1)}{2} + j\right) \right]^2 \gamma_{ji}(\epsilon) \\ \epsilon_k &= y_k - \sum_{j=1}^q G\left(\frac{q(q-1)}{2} + j\right) \epsilon_j \end{aligned} \right\} \quad (8)$$

但应强调指出, (8) 式与 (一) 中相应公式所引用到的  $\gamma_{kk}(\epsilon)$  及  $\epsilon_k$  是不相同的, 为了使两者一致起来, 必须讨论  $\gamma_{kk}(\epsilon)$  及  $\epsilon_k$  的存贮和“左移”处理.

#### (四) 关于 $\gamma_{kk}(\epsilon)$ 与 $\epsilon_k$ 的存贮及左移处理

$\gamma_{kk}(\epsilon)$  与  $\epsilon_k$  显然可用一维存贮, 由 (一) 的公式可见, 当  $k \leq Q$  时, 计算  $\epsilon_k$  要用到  $\epsilon_1, \cdots, \epsilon_{k-1}$ . 所以  $\epsilon_k$  的贮量随  $k$  增加而增加, 最多需占用  $Q$  个单元. 而当  $k > Q$  时, 计算  $\epsilon_k$  只要用到  $\epsilon_{k-q} \cdots \epsilon_{k-1}$  恒为  $q$  个值, 不随  $k$  的增加而增加了. 因此, 当  $k > Q$  时,  $\epsilon_k$  贮于一个长度为  $q+1$  单元的一维数组就够了. 可见,  $\epsilon_k$  只须存于长度为  $Q+1$  的一维数组  $E(k)$ . 完全同样的分析知,  $\gamma_{kk}(\epsilon)$  也只须存于一个长为  $Q+1$  的一维数组  $RE(k)$ .

为了使 (8) 式与 (一) 相应公式一致, 我们必须对  $RE(k)$ ,  $E(k)$  作左移处理如下:

当要计算  $k = Q+1$  的  $C_{kj}$ ,  $\gamma_{kk}(\epsilon)$  及  $\epsilon_k$  之前, 必须先把  $RE(k)$ ,  $E(k)$  左移  $Q-q$  步. 即

$$\left. \begin{aligned} RE(k) &\Leftarrow RE(Q-q+k) \quad k = 1 \cdots q \\ E(k) &\Leftarrow E(Q-q+k) \end{aligned} \right\} \quad (9)$$

此后,  $k$  每增加 1 时, 都必须先把  $RE(k)$ ,  $E(k)$  左移一步. 即

$$\left. \begin{aligned} RE(k) &\Leftarrow RE(k+1) \\ E(k) &\Leftarrow E(k+1) \quad k = 1, 2, \cdots, q \end{aligned} \right\} \quad (10)$$

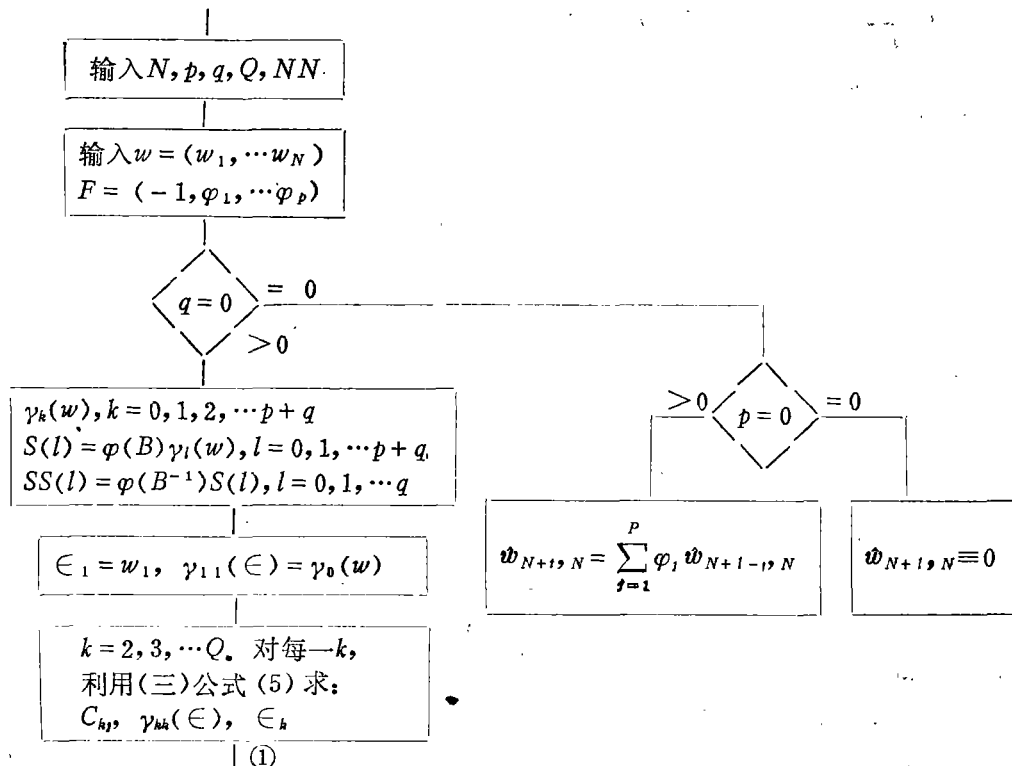
总之, 当  $k > Q$  时, 只要在每一次计算  $C_{kj}$ ,  $\gamma_{kk}(\epsilon)$ ,  $\epsilon_k$  之前, 作下滑和左移处理, 则它们的计算公式可改写为:

$$\left. \begin{aligned} G\left(\frac{q(q-1)}{2} + j\right) &= \left[ RY(j-1) - \sum_{i=1}^{j-1} G\left(\frac{(j-1)(j-2)}{2} + i\right) G\left(\frac{q(q-1)}{2} + i\right) RE(i) \right] \\ &\quad \left[ RE(j) \right]^{-1} \\ &\quad j=1, 2, \dots, q \\ RE(q+1) &= RY(q) - \sum_{j=1}^q \left[ G\left(\frac{q(q-1)}{2} + j\right) \right]^2 RE(j) \\ E(q+1) &= Y_k - \sum_{j=1}^q G\left(\frac{q(q-1)}{2} + j\right) E(j) \end{aligned} \right\} (11)$$

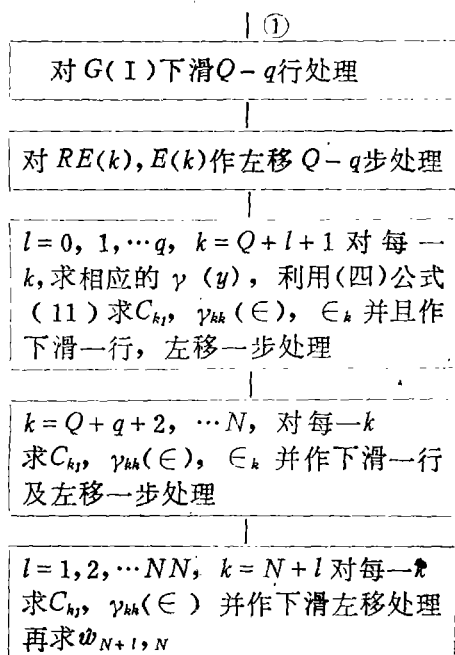
### (五)新息预报程序设计框图

我们已按上述一维存贮的公式, 用 FORTRAN IV-PLUS 语言, 编写新息预报程序。并在 PDP-11/34 机上进行验证, 证明所涉及的公式均正确无误。同时, 我们还在气象预测和经济方面作了一些初步应用。程序设计框图如下:

设  $w_1, \dots, w_N$  是  $ARMA(p, q)$  序列的一个样本,  $p, q$  已知, 令  $Q = \max(p, q)$ , 而且自回归参数  $\varphi_1, \varphi_2, \dots, \varphi_p$  也已求得。欲求  $w_{N+l, N}$   $l=1, 2, \dots, NN$ 。NN 为预报总的步长。







## 参 考 文 献

- [1] 中国科学院应用数学研究所编, 时间序列及其应用, (1982)。
- [2] 陈兴钩, 线性随机序列分析, (1981)。
- [3] Box, G. E. P and Jenkins, G.M, Time Series Analysis Forecasting and Contral, Holden-Day, San Francisco, (1970)。
- [4] Anderson, O.D., Time Series Analysis and Forecasting, The Box-Jenkins Approach, Butterworths, London and Bosto, (1979)。