

Inverse sequential procedures for the monitoring of time series  
Project NAGW-2706  
Principal investigator: Uwe Radok  
CIRES, Campus Box 449, University of Colorado  
Boulder, CO 80309  
303-492-5562 or 303-497-5511  
FAX 303-492-2468  
Progress report #3, 4/30/93

GRANT  
NAGW-2706  
157290  
1993

by Uwe Radok and Timothy Brown

Table of Contents

Summary ..... 1  
1. Introduction ..... 2  
2. Theory ..... 3  
3. Formulae and examples ..... 6  
3.1 Poisson mean (= variance) ..... 6  
3.2 Gaussian means and variances ..... 9  
3.3 Trends - linear or exponential regression ..... 11  
3.4 Chi-square ..... 13  
4. Implementation ..... 16  
References ..... 17  
Appendix: Inverse sequential Fortran program "Sequitur" ..... 19

Summary

Climate changes traditionally have been detected from long series of observations and long after they happened. The "inverse sequential" monitoring procedure is designed to detect changes as soon as they occur. Frequency distribution parameters(s) are estimated both from the most recent existing set of observations and from the same set augmented by 1,2, ... j new observations. Individual-value probability products ("likelihoods") are then calculated which yield probabilities for erroneously accepting the existing parameter(s) as valid for the augmented data set and vice versa. A parameter change is signaled when these probabilities (or a more convenient and robust compound "no change" probability) show a progressive decrease. New parameters are then estimated from the new observations alone to restart the procedure. The detailed algebra is developed and tested for Gaussian means and variances, Poisson and chi-square means, and linear or exponential trends; a comprehensive and interactive Fortran program is provided in the appendix.

(NASA-CR-192959) INVERSE  
SEQUENTIAL PROCEDURES FOR THE  
MONITORING OF TIME SERIES Progress  
Report No. 3 (Colorado Univ.)  
56 p

N93-25666

Unclass



## 1. Introduction

The detection of changes in a developing time series requires some idea of what form the changes are likely to take. When the nature of the forcing is known, filters can be designed that will show their effects most clearly (Kim and North, 1991), but that knowledge is often not available in the geophysical sciences. However, many time series are made up of irregular-length sections each of which differs from its neighbors in one or more of the parameters that define its signal and noise characteristics. As long as its parameters remain unchanged, an individual section can then be said to be in "statistical control" (Shewhart, 1939).

There exists considerable evidence that this concept is realistic in many geophysical contexts. Examples are variables exhibiting the "Hurst phenomenon" much discussed in hydrology (e.g., Klemes, 1974), and atmospheric circulation patterns (Toth, 1992). With its minimum of arbitrary assumptions, the concept of statistical control suggests a general monitoring approach that registers the length and end of each controlled "regime", together with the new parameter values. The magnitude of changes in geophysical parameters cannot be anticipated, but their surveillance might use a probability for regarding the parameters established from existing observations as significantly changed by the addition of one or more new observations.

Such a "sequential" use of accruing information was pioneered by Wald (1947) and has developed into a large special field of statistics (c.f. e.g., Gosh, 1988) which includes a range of procedures utilizing cumulative sums ("cusum" techniques; e.g., Goel, 1982). The typical outcome in the simplest situation is a decision, with *prescribed* error probabilities, to accept one of two specified parameter values, or to continue sampling.

The "inverse" sequential approach here presented instead progressively *determines* "no change" probabilities for parameter estimates based, respectively, on the accrued data and on the same data augmented by one or several new observations. A parameter change is then signaled when these probabilities begin decreasing to small values.

The basic approach is developed in the next section and formulated in Section 3 for the parameters of Gaussian, Poisson, and chi-square distributions, as well as for linear and exponential trends. The technique is illustrated there with constructed examples, and is described as a systematic and comprehensive procedure in Section 4 with reference to an interactive Fortran program, reproduced as code in the appendix.

## 2. Theory

Consider a series of  $m$  observations  $x_i$ ,  $i = 1, 2, \dots, m$ , to which further  $j$  observations are added ( $j = 1, 2, \dots$ ). For a parameter  $\theta$  (such as mean, variance, trend, etc.) the first  $m$  values yield an optimum estimate  $\theta_m$  which the augmented set of observations changes to  $\theta_{m+j}$ . Writing the corresponding probabilities of individual  $x_i$  as  $p(x_i; \theta)$ , the likelihood function of  $n$  observations is

$$L(n; \theta) = p(x_1; \theta) \cdot p(x_2; \theta) \cdots p(x_n; \theta). \quad (1)$$

With  $n = m$  or  $m + j$  and  $\theta = \theta_m$  or  $\theta_{m+j}$  we have four different likelihoods:

$$L_1 = L(m; \theta_m); L_2 = L(m + j; \theta_{m+j}); L_3 = L(m + j; \theta_m); L_4 = L(m; \theta_{m+j}). \quad (2)$$

Now the form of  $L$  shows that the likelihoods decrease systematically with increasing sample size. Those for the initial data can be made comparable to those for the augmented data by multiplying the  $L(m; \theta)$  by some factor  $c(m)$  and the  $L(m + j; \theta)$  by  $c(m + j)$ . Furthermore the sum of the adjusted likelihoods,  $c(m)L_1 + c(m + j)L_3$  represents the probability that  $\theta_m$  is valid for either the initial  $m$  data or the augmented set of  $m + j$ . Since these are taken to be the only choices, that probability is one; the same applies to the sum  $c(m + j)L_2 + c(m)L_4$ . Denoting  $c(m + j)L_3$  by  $\alpha$  and  $c(m)L_4$  by  $\beta$ , the other adjusted probabilities become  $c(m)L_1 = 1 - \alpha$  and  $c(m + j)L_2 = 1 - \beta$ .

(In the terminology of the theory of hypotheses (e.g. Hoel, 1966),  $\alpha$  represents the "type I error" probability for not accepting  $\theta_m$ , even though true, for the augmented sample. The probability  $\beta$  is that of rejecting  $\theta_{m+j}$  though true for the initial data; alternatively, it is the "type II" error probability of preferring  $\theta_m$ , though false, for the initial sample).

The probabilities  $\alpha$  and  $\beta$  are calculated from the factor-free likelihood *ratios*:

$$q(m) = \frac{L_4}{L_1} = \frac{\beta}{1 - \alpha}; \quad (3a)$$

$$q(m + j) = \frac{L_2}{L_3} = \frac{1 - \beta}{\alpha}. \quad (3b)$$

Solving for  $\alpha$  and  $\beta$  yields

$$\alpha = \frac{(1 - q_m)}{(q_{m+j} - q_m)}; \quad (4a)$$

$$\beta = \frac{(q_{m+j}q_m - q_m)}{(q_{m+j} - q_m)}. \quad (4b)$$

With a definite change of control from  $\theta_m$  to  $\theta_{m+j}$ , both probabilities in due course decrease to small values. While the existing regime continues, the variability of the likelihood function and rounding errors can raise the likelihood ratio  $q(m)$  to unrealistic values larger than unity and similarly reduce  $q(m+j)$  to values below one. To avoid probabilities that are negative or larger than 1, such  $q$  values must be replaced by 1, implying equality of the likelihoods involved.

For the inverse sequential monitoring operation, several combined quantities suggest themselves as more stable than  $\alpha$  and  $\beta$ :

- i)  $\alpha + \beta$ , the probability that *either*  $\theta(m)$  is valid for the sample of  $m + j$  *or*  $\theta(m + j)$  for the sample of  $m$ ;
- ii)  $\alpha \cdot \beta$ , the probability that both these statements are true;
- iii) a compound "no-change" probability  $\gamma$  which will be used for the illustrations in Section 3, and is defined by writing

$$\frac{q(m + j)}{q(m)} = Q = \frac{L_1L_2}{L_3L_4} = \frac{(1 - \alpha)(1 - \beta)}{\alpha\beta} \equiv \frac{(1 - \gamma)^2}{\gamma^2}, \quad (5)$$

so that

$$\gamma = (1 + \sqrt{Q})^{-1} . \quad (6)$$

The probability  $\gamma$  falls between 0 and 0.5 as long as  $q(m+j) > q(m)$  and lies between the arithmetic and geometric averages of  $\alpha$  and  $\beta$ , as can be shown by alternatively substituting these averages for  $\alpha$  and  $\beta$  in the original form of (5), i.e. in

$$Q = \frac{[1 - (\alpha + \beta) + \alpha\beta]}{\alpha\beta} . \quad (7)$$

When  $\alpha = \beta = (\alpha + \beta)/2$ , equation (7) becomes

$$Q_1 = \frac{\left[ 1 - (\alpha + \beta) + \alpha\beta + \frac{\alpha^2 + \beta^2}{2} \right]}{\left[ \alpha\beta + \frac{\alpha^2 + \beta^2}{2} \right]} . \quad (8a)$$

This shows that the numerator  $N$  and the denominator  $D$  of  $Q$  both have been increased by  $\epsilon = (\alpha^2 + \beta^2)/2 > 0$ . Since  $Q > 1$  (i.e.,  $N > D$ ), then  $Q = N/D > Q_1 = (N + \epsilon)/(D + \epsilon)$  since  $ND + N\epsilon > ND + D\epsilon$ , or  $N > D$ , the initial condition.

Again with  $\alpha = \beta = (\alpha\beta)^{1/2}$ , equation (7) becomes

$$Q_2 = \frac{[1 - 2(\alpha\beta)^{1/2} + \alpha\beta]}{\alpha\beta} , \quad (8b)$$

so that  $Q_2 - Q = -2(\alpha\beta)^{1/2} + (\alpha + \beta) > 0$  since  $\alpha + \beta > 2(\alpha\beta)^{1/2}$ ; this can be seen by squaring both sides giving

$$(\alpha + \beta)^2 + \alpha^2 + \beta^2 + 2\alpha\beta > 4\alpha\beta , \text{ or } (\alpha - \beta)^2 > 0 .$$

Finally, with  $Q_2 > Q > Q_1$ ,

$$\left[ 1 + \sqrt{Q_2} \right]^{-1} = \gamma_{\text{geometric}} < \gamma = \left[ 1 + \sqrt{Q} \right]^{-1} < \gamma_{\text{arithmetic}} = \left[ 1 + \sqrt{Q_1} \right]^{-1} . \quad (9)$$

(Equations (3a) and (3b) have the form of the decision limits of Wald's (1947) "sequential probability ratio test (SPRT)".  $\log_e Q$  can then be interpreted as the logarithmic width of the indecision region of a SPRT in which  $\log_e q(m+j)$  defines the upper decision limit, and  $\log_e q(m)$  the lower decision limit,

respectively).

A change in parameter(s) lowers  $\gamma$  at a rate that increases with the change magnitude but decreases with an increase in the number of observations before the change. This is further discussed and illustrated in Section 3.1; it suggests using a moving base period, or restarting the procedure with new base values after some interval in which  $\gamma$  shows no clear descending trend.

### 3. Formulae and examples

The formulae give the basic probability  $p$  in the likelihood functions for  $m$  and  $m+j$  observations, and the likelihood ratios  $q(m)$  and  $q(m+j)$  used to calculate the probabilities  $\alpha$ ,  $\beta$ , and  $\gamma$  from equations (4) and (6) in Section 2. For simplicity, subscripts will be used to indicate the number of values used for parameter estimates, and bracketed symbols for the numbers used to calculate the likelihoods and their ratios. Thus  $L(m; \theta_m) = L_1$  becomes  $L_m(m)$ ,  $L(m+j, \theta_m) = L_3 = L_m(m+j)$ , etc.

The examples in this section use constructed data with known properties and illustrate how the detailed properties of the inverse sequential procedure will be established by more extensive calculations using different base lengths and magnitudes of parameter changes.

#### 3.1 Poisson mean (= variance)

This case is rather simple because the basic probability

$$p = \frac{\bar{x}^x}{x! \exp(\bar{x})}, \quad (3.1.1)$$

has only a single parameter, the mean number  $\bar{x}$  of occurrences. The logarithmic likelihood functions are

$$\log_e L_m(m) = m\bar{x}_m \log_e \bar{x}_m - \sum_1^m \log_e x! - m\bar{x}_m; \quad (3.1.2a)$$

$$\log_e L_{m+j}(m) = m\bar{x}_m \log_e \bar{x}_{m+j} - \sum_1^m \log_e x! - m\bar{x}_{m+j}; \quad (3.1.2b)$$

Table 1: Detection of change in Poisson mean (= variance).

The 5 base values are drawn at random from a Poisson population with mean 5. These and another 10 values from the same population are used in test III. Test I and test II each use the same base values as in test III and 10 values from Poisson populations with means 3 and 7, respectively. Base values used are 6, 1, 6, 6, 4 and have a sample mean of 4.6.

Test I		Test II		Test III	
	no-change probability		no-change probability		no-change probability
2	0.47	5	0.50	3	0.49
1	0.36	6	0.49	2	0.44
6	0.43	9	0.40	4	0.43
1	0.33	4	0.43	6	0.47
2	0.25	9	0.32	5	0.48
2	0.18	1	0.43	6	0.50
5	0.22	6	0.41	7	0.50
2	0.16	10	0.28	1	0.49
1	0.09	5	0.29	8	0.50
0	0.04	13	0.11	5	0.50



$$\log_e L_{m+j}(m+j) = (m+j)\bar{x}_{m+j} \log_e \bar{x}_{m+j} - \sum_1^{m+j} \log_e x! - (m+j)\bar{x}_{m+j} ; \quad (3.1.2c)$$

$$\log_e L_m(m+j) = (m+j)\bar{x}_{m+j} \log_e \bar{x}_m - \sum_1^{m+j} \log_e x! - (m+j)\bar{x}_m , \quad (3.1.2d)$$

resulting in the likelihood ratios

$$q(m) = \exp \left[ m \left\{ \bar{x}_m \log_e \frac{\bar{x}_{m+j}}{\bar{x}_m} - (\bar{x}_{m+j} - \bar{x}_m) \right\} \right] ; \quad (3.1.3)$$

$$q(m+j) = \exp \left[ (m+j) \left\{ \bar{x}_{m+j} \log_e \frac{\bar{x}_{m+j}}{\bar{x}_m} - (\bar{x}_{m+j} - \bar{x}_m) \right\} \right] , \quad (3.1.4)$$

so that

$$Q = \exp \left[ \left\{ (m+j)\bar{x}_{m+j} - m\bar{x}_m \right\} \log_e \frac{\bar{x}_{m+j}}{\bar{x}_m} - j(\bar{x}_{m+j} - \bar{x}_m) \right] . \quad (3.1.5)$$

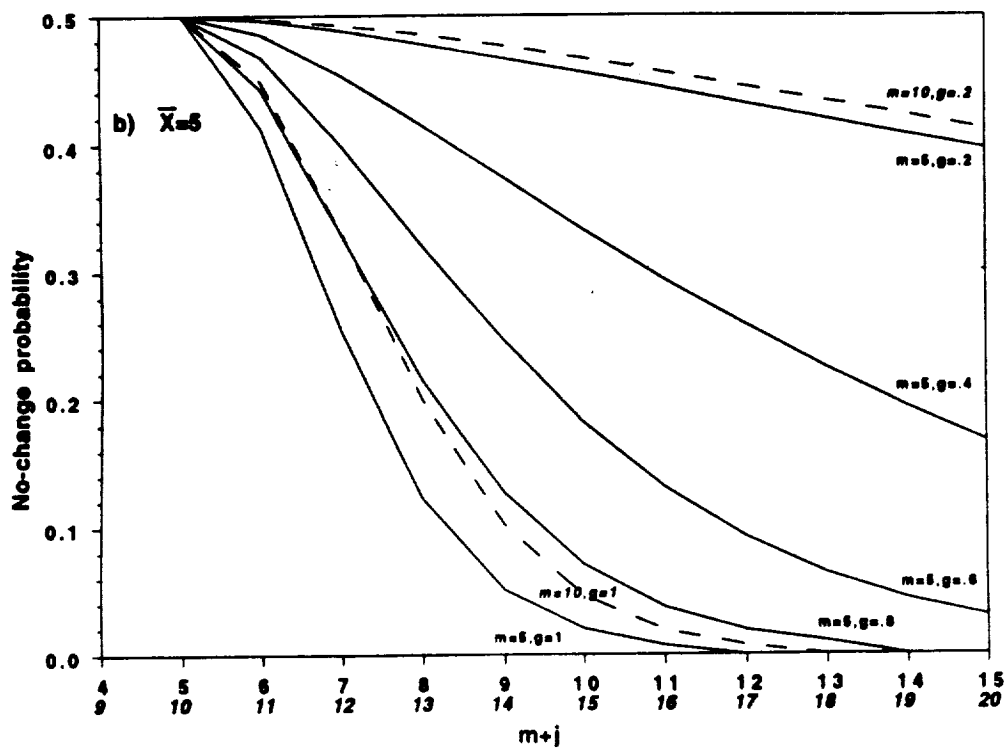
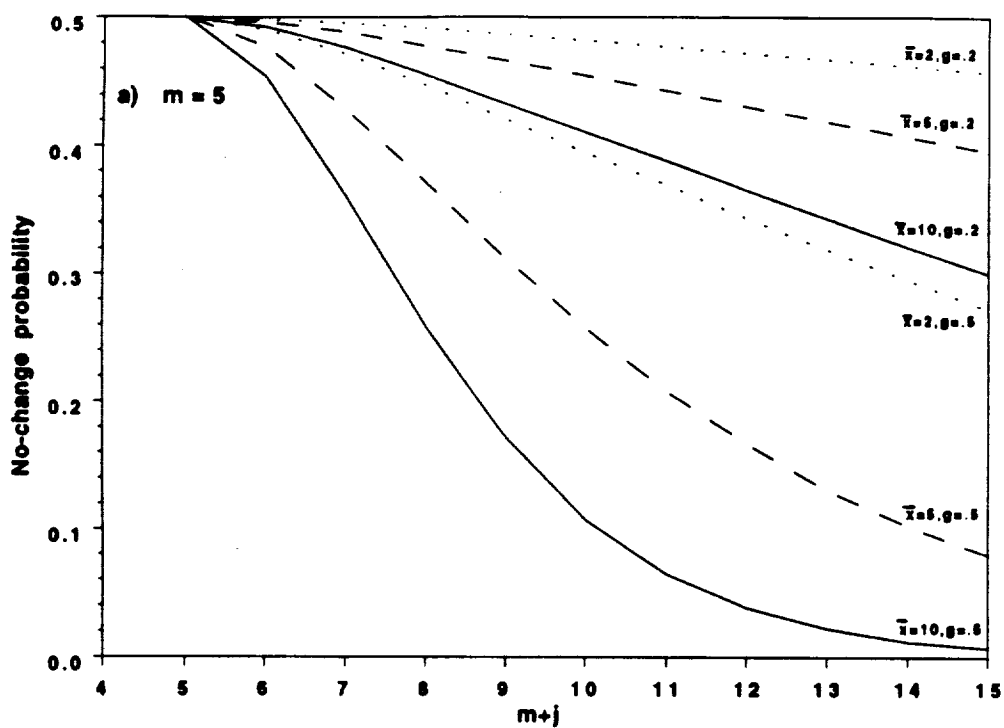
### Examples

Three tests were conducted to demonstrate how the no-change probability is used to detect changes in Poisson mean. Table 1 lists three series each containing ten values drawn at random from Poisson populations with means of 3, 7, and 5 respectively. All three tests use the same five base values, which have a sample mean of 4.6. In tests I and II, progressive decreases occur in the no-change probability suggesting a change in the Poisson mean. This is expected since the values in each series were drawn from different population means than those of the base period. In test III, the no-change probability does not steadily decrease indicating that the series mean is not significantly different from the base period mean. This too is expected since the series values and the base values were drawn from the same population.

The relatively simple form of equation (3.1.5) makes it possible to explore the dependence of the no-change probability on change magnitude and base length for an average Poisson sample. Let  $x_i$ ,

Figure 1. No-change probabilities for change in Poisson mean at  $m=5$  or 10.

Initial mean  $\bar{x}$ ; new mean  $\bar{x}' = \bar{x} + \Delta\bar{x}$ ;  $g = \Delta\bar{x} / \bar{x}$



$i = 1, 2 \dots m$  now represent the average of many samples drawn from a Poisson distribution with mean  $\bar{x}$ , so that each  $x_i$  equals  $\bar{x}$ , and add further  $j$  values each equal to  $\bar{x}'$  from a Poisson distribution with mean  $\bar{x}'$ . With  $\Delta\bar{x} = \bar{x}' - \bar{x}$ , Equation (3.1.5) can be modified to read

$$\log_e Q_P = \bar{x}_m \left[ \left\{ j + (m + j) \frac{\Delta\bar{x}}{\bar{x}_m} \right\} \log_e \left[ 1 + \frac{\Delta\bar{x}}{\bar{x}_m} \right] - j \frac{\Delta\bar{x}}{\bar{x}_m} \right]. \quad (3.1.6)$$

Now the mean  $\bar{x}_m$  is simply equal to  $\bar{x}$ , while

$$\bar{x}_{m+j} = \frac{m\bar{x} + j\bar{x}'}{m+j} = \frac{(m+j)\bar{x}_m + j\bar{x}_m(\bar{x}' - \bar{x})/\bar{x}}{m+j}. \quad (3.1.7)$$

With  $(\bar{x}' - \bar{x})/\bar{x} = g$  and  $\Delta\bar{x}/\bar{x}_m = jg/(m+j) = v$ , say, (3.1.6) becomes

$$\log_e Q_P = \bar{x}_m \left[ \left\{ j + (m + j)v \right\} \log_e (1 + v) - jv \right], \quad (3.1.8)$$

which together with equation (6) leads to the no-change probability

$$\gamma = \left[ 1 + \sqrt{Q_P} \right]^{-1}, \quad (3.1.9)$$

as function of  $x$ ,  $m$ ,  $j$ , and  $g = \Delta\bar{x}/\bar{x}_m$

Equations (3.1.8) and (3.1.9) have been evaluated for three different  $\bar{x} = \bar{x}_m$  and five different values of  $g$ . The results are shown in Figure 1a for three different  $\bar{x}'$  and in Figure 1b for a single value  $\bar{x} = 5$ , for  $j = 1$  through 10.

The two broken lines in Figure 1b show that a longer base (larger  $m$ ) will slow the response of the no-change probability to the change in mean, but leaves the curves essentially unaltered in shape. The decline in the no-change probability starts at the change of mean and accelerates down to values of the order of 0.3–0.2 before becoming more and more gradual as small probabilities are approached. Beyond this general feature, the simple argument here used reveals little about the rate of decline except that it changes markedly with both the base mean and the means ratio  $g$ . More specific features remain to be established by numerical experimentation.

### 3.2 Inverse sequential formulae for Gaussian means and variances

The basic probability for the Gaussian distribution,

$$p = (2\pi\sigma^2)^{-1/2} \exp\left[\frac{-(x - \mu)^2}{2\sigma^2}\right]. \quad (3.2.1)$$

involves two parameters,  $\mu$  and  $\sigma^2$ , which can only be tested jointly since in the present context neither is known a priori. We use the sample mean as an estimate for the population mean  $\mu$ , and  $\sigma^2 = [(n/(n-1))s^2]$ , where  $s^2$  is the sample variance and  $n (= m \text{ or } m + j)$  is the number of values used.

The likelihood products (equation 1) are converted to sums by taking logarithms; thus,

$$\log_e L_m(m) = -\frac{m}{2} \log_e 2\pi - \frac{m}{2} \log_e \sigma_m^2 - \sum_1^m \frac{(x - \mu_m)^2}{2\sigma_m^2}, \quad (3.2.2)$$

with  $\sum_1^m (x - \mu)^2 = (m-1)\sigma_m^2$ , the last term reduces to  $[-(m-1)/2]$ .

Proceeding in the same way for  $L_A = L_{m+j}(m)$  leads to

$$\log_e L_{m+j}(m) = -\frac{m}{2} \log_e 2\pi - \frac{m}{2} \log_e \sigma_{m+j}^2 - \sum_1^m \frac{(x - \mu_{m+j})^2}{2\sigma_{m+j}^2}, \quad (3.2.3a)$$

with  $\mu_{m+j} - \mu_m = \Delta\mu$ , the numerator of the last term can be written as

$$\begin{aligned} -\sum_1^m (x - (\mu_m + \Delta\mu))^2 &= -\sum_1^m (x - \mu_m)^2 - \sum_1^m (-2x\Delta\mu + 2\mu_m\Delta\mu + \Delta\mu^2) \\ &= -(m-1)\sigma_m^2 - 2m\mu_m\Delta\mu + 2m\mu_m\Delta\mu - m(\Delta\mu)^2, \end{aligned}$$

so that equation (3.2.3a) becomes

$$\log_e L_{m+j}(m) = -\frac{m}{2} \log_e 2\pi - \frac{m}{2} \log_e \sigma_{m+j}^2 - \frac{(m-1)\sigma_m^2}{2\sigma_{m+j}^2} - \frac{m(\Delta\mu)^2}{2\sigma_{m+j}^2}. \quad (3.2.3b)$$

Finally, subtracting equation (3.2.2) from equation (3.2.3b) gives the log likelihood ratio

$$\log_e q(m) = \log_e \frac{L_{m+j}(m)}{L_m(m)} = \frac{m}{2} \log_e \frac{\sigma_m^2}{\sigma_{m+j}^2} + \frac{m-1}{2} \left[ 1 - \frac{\sigma_m^2}{\sigma_{m+j}^2} \right] - \frac{m(\Delta\mu)^2}{2\sigma_{m+j}^2}. \quad (3.2.4)$$

Proceeding the same way for the augmented set of  $m+j$  values yields first

$$\log_e L_{m+j}(m+j) = -\frac{m+j}{2} \log_e 2\pi - \frac{m+j}{2} \log_e \sigma_{m+j}^2 - \sum_1^{m+j} \frac{(x - \mu_{m+j})^2}{2\sigma_{m+j}^2}, \quad (3.2.5)$$

where the last term, with  $\sum_1^{m+j} (x - \mu_{m+j})^2 = (m+j-1)\sigma_{m+j}^2$ , reduces to  $-(m+j-1)/2$ . Next,

$$\log_e L_m(m+j) = -\frac{m+j}{2} \log_e 2\pi - \frac{m+j}{2} \log_e \sigma_m^2 - \sum_1^{m+j} \frac{(x - (\mu_{m+j} - \Delta\mu))^2}{2\sigma_m^2}. \quad (3.2.6a)$$

Expanding the last term as before yields

$$\log_e L_m(m+j) = -\frac{m+j}{2} \log_e 2\pi - \frac{m+j}{2} \log_e \sigma_m^2 - \frac{(m+j-1)\sigma_{m+j}^2}{\sigma_m^2} - \frac{(m+j)\Delta\mu^2}{2\sigma_m^2}. \quad (3.2.6b)$$

Subtracting equation (3.2.6b) from equation (3.2.5) we obtain the second log likelihood ratio

$$\log_e q(m+j) = \frac{m+j}{2} \log_e \frac{\sigma_m^2}{\sigma_{m+j}^2} + \frac{m+j-1}{2} \left[ \frac{\sigma_{m+j}^2}{\sigma_m^2} - 1 \right] + \frac{(m+j)(\Delta\mu)^2}{2\sigma_m^2}. \quad (3.2.7)$$

The final formulae therefore are

$$q(m) = \exp \left[ m \log_e \frac{\sigma_m}{\sigma_{m+j}} + \frac{m-1}{2} \left[ 1 - \frac{\sigma_m^2}{\sigma_{m+j}^2} \right] - \frac{m}{2\sigma_{m+j}^2} (\mu_{m+j} - \mu_m)^2 \right], \quad (3.2.8a)$$

and

$$q(m+j) = \exp \left[ (m+j) \log_e \frac{\sigma_m}{\sigma_{m+1}} + \frac{m+j-1}{2} \left[ \frac{\sigma_{m+j}^2}{\sigma_m^2} - 1 \right] + \frac{m+j}{2\sigma_m^2} (\mu_{m+j} - \mu_m)^2 \right]. \quad (3.2.8b)$$

The first two exponents in each formula reflect solely changes in variance, while the third exponent depends primarily on changes in the mean.

Table 2: Detection of change in Gaussian mean and variance.

The 5 base values for tests I and II are drawn from a Gaussian population with mean 5 (sample mean: 3.64) and variance 25 (sample variance: 71.74). The 5 base values for test III come from a Gaussian population with mean 7.5 (sample mean: 8.02) and variance 6.25 (sample variance: 4.20). Test I uses 3 values from the first series followed by 7 values from the second. The 10 values for test II all come from the first series, and those for test III from the second series.

Test I		Test II		Test III	
-5.0	base values	-5.0	base values	10.2	base values
-1.3		-1.3		9.2	
10.2		10.2		6.8	
14.9		14.9		8.8	
-0.6		-0.6		5.1	
	no-change probability		no-change probability		no-change probability
2.6	0.49	2.6	0.49	6.7	0.49
6.6	0.46	6.6	0.46	6.9	0.45
2.5	0.41	2.5	0.41	5.1	0.40
10.2	0.39	12.4	0.41	9.7	0.46
9.2	0.35	10.5	0.37	14.0	0.35
6.8	0.29	8.9	0.32	10.4	0.33
8.8	0.24	8.7	0.27	6.1	0.36
5.1	0.19	7.7	0.21	6.5	0.39
6.7	0.14	8.6	0.17	8.6	0.42
6.9	0.10	8.8	0.12	6.8	0.44

### Examples

Table 2 uses values drawn at random from two Gaussian populations with different means and variances ( $\mu = 5$ ,  $\sigma^2 = .25$ , and  $\mu = 7.5$ ,  $\sigma^2 = 6.25$ , respectively). Test I uses the first 5 values of the first series as base and continues with the next three values of the same series before introducing those of the second series. A steady decrease in the no-change probabilities starts with the third value after the change. By contrast in test III, the second series without major change provides a definite no-change signal.

In test II a regime change is suggested even though the series used was designed without such a change. The mean of the initial 5 "base" values (3.64) is considerably smaller than the population mean (5.0) while the entire series of 15 values has a mean (6.37) that is considerably larger than the population mean. Acting on the sample information alone (all the inverse sequential procedure is designed to do), the test therefore quite properly suggested a significant change from the base parameter.

### 3.3 Trends (least-square regression)

Observations made at equally spaced times  $t = 1, 2, \dots, n$  ( $n = m$  or  $m + j$ ) are represented by

$$y = A + Bt + e \quad (3.3.1)$$

This also covers the case of exponential regression when  $y = \log x$ . The residuals  $e$  are assumed to be normally distributed with zero mean and variance  $\sigma^2$ . Sample estimates of the regression coefficients  $A$  and  $B$  satisfying least-square requirements are

$$a = \bar{y} + b\bar{t}; b = \frac{\sum_1^n (y - \bar{y})(t - \bar{t})}{\sum_1^n (t - \bar{t})^2} \quad (3.3.2)$$

where

$$\bar{t} = \frac{(n+1)}{2}; \sum_1^n (t - \bar{t})^2 = \frac{n(n^2-1)}{12}.$$

The regression estimate  $\hat{y}'$  for a given  $t'$  is then  $\hat{y}' = \bar{y} + b(t' - \bar{t})$ , and the corresponding residual  $e_t = y'_t - \hat{y}'_t$  has a Gaussian distribution with zero mean and variance (see e.g., Anderson and Bancroft,

1952, Section 12.2)

$$s_{t_n}^2 = \frac{1}{n-2} \left[ \sum_{i=1}^n (y_i - \bar{y}_i)^2 - b_n^2 \frac{n(n^2-1)}{12} \right] \left[ \frac{n+1}{n} + \frac{\left[ t - \frac{(n+1)}{2} \right]^2}{\frac{n(n^2-1)}{12}} \right] \quad (3.3.3)$$

The general form of the likelihood functions defined by the  $n$  residuals  $e_t$ ,  $t = 1, 2, \dots, n (= m \text{ or } m + j)$  is

$$L = \exp \sum_{t=1}^n \left[ -\frac{(y_t - \hat{y}_t)^2}{2s_{t_n}^2} \right] \quad (3.3.4)$$

For the inverse sequential detection of trend changes equation (3.3.4) takes the following forms:

$$\log_e L_m = \log_e L_1 = - \left[ \frac{m-2}{2[SS(m) - b_m^2 d(m)]} \right] \cdot \sum_{t=1}^m \frac{(y - \hat{y}_m)_t^2}{\left[ \frac{m+1}{m} + \frac{\left[ t - \frac{m+1}{2} \right]^2}{d(m)} \right]} \quad (3.3.5)$$

$$\log_e L_{m+j}(m+j) = \log_e L_2 = - \left[ \frac{m+j-2}{2[SS(m+j) - b_{m+j}^2 d(m+j)]} \right] \cdot \sum_{t=1}^{m+j} \frac{(y - \hat{y}_{m+j})_t^2}{\left[ \frac{m+j+1}{m+j} + \frac{\left[ t - \frac{m+j+1}{2} \right]^2}{d(m+j)} \right]} \quad (3.3.6)$$

$$\log_e L_m(m+j) = \log_e L_3 = - \left[ \frac{m+j-2}{2[SS(m+j) - b_m^2 d(m+j)]} \right] \cdot \sum_{t=1}^{m+j} \frac{(y - \hat{y}_m)_t^2}{\left[ \frac{m+j+1}{m+j} + \frac{\left[ t - \frac{m+j+1}{2} \right]^2}{d(m+j)} \right]} \quad (3.3.7)$$



$$\log_e L_{m+j}(m) = \log_e L_4 = - \left[ \frac{m-2}{2[SS(m) - b_{m+j}^2 d(m)]} \right] \cdot \sum_{t=1}^m \frac{(y - \hat{y}_{m+j})_t^2}{\left[ \frac{m+1}{m} - \frac{\left[ t - \frac{m+1}{2} \right]^2}{d(m)} \right]} ; \quad (3.3.8)$$

with

$$\left. \begin{aligned} SS(n) &= \sum_{t=1}^n (y - \bar{y})^2 ; & d(n) &= \frac{n(n^2-1)}{12} ; \\ \hat{y}_n &= a_n + b_n t & n &= m \text{ or } m+j \end{aligned} \right\} . \quad (3.3.9)$$

Finally as before

$$q(m) = \exp(\log_e L_4 - \log_e L_1) ,$$

and

$$(3.3.10)$$

$$q(m+j) = \exp(\log_e L_2 - \log_e L_3) .$$

### Examples

Table 3 gives the results of four tests to demonstrate the inverse sequential detection of changes in linear trend (regression coefficient  $b$  in  $y = a + bt$ ). The test III series from Table 2 is used. This series starts with declining values but settles down to a negligible trend for the entire sample 15 values ( $b = -0.02$ ; variance  $s^2 = 5.88$ ). Positive and negative trends of  $b = \pm s/6$  and  $b = \pm s/3$  are then imposed on the last ten  $y$ -values in the series.

The no-change probability starts decreasing in each case as soon as the imposed trend produces a distinct change from that of the base series. By contrast, when the last 10 values of the base series are tested unchanged all the no-change probabilities (not shown) remain above 0.45.

### 3.4 Chi-square

The variances  $s^2$  of samples from a Gaussian population with variance  $\sigma^2$  define a chi-square variate

$$\chi^2 = \frac{hs^2}{\sigma^2} , \quad (3.4.1)$$

Table 3: Detection of changes in linear trend

Base series used is that of test III in table 2, with the following sample parameters: mean 8.10; variance = 5.88; regression coefficients  $a = 8.27$ ,  $b = -0.021$

1) trend  $b = \pm 0.4$  imposed on the 10 y-values from test III, table 2

positive trend			negative trend		
y	b (progressive)	no-change probability	y	b (progressive)	no-change probability
7.1	-0.04	0.50	6.3	-0.06	0.49
7.7	-0.04	0.50	6.1	-0.08	0.48
6.6	-0.06	0.49	4.2	-0.14	0.42
11.3	0.00	0.50	8.1	-0.11	0.45
16.3	0.12	0.38	12.3	-0.03	0.50
12.8	0.16	0.29	8.0	-0.02	0.50
8.9	0.14	0.30	3.3	-0.07	0.48
9.7	0.13	0.29	3.3	-0.11	0.44
12.2	0.15	0.23	5.0	-0.13	0.41
10.8	0.15	0.20	2.8	-0.15	0.35

2) trend  $b = \pm 0.8$  imposed on the 10 y-values from test III, table 2

positive trend			negative trend		
y	b (progressive)	no-change probability	y	b (progressive)	no-change probability
7.5	-0.03	0.50	5.9	-0.07	0.49
8.5	-0.02	0.50	5.3	-0.11	0.45
7.8	-0.02	0.50	3.0	-0.17	0.34
12.9	0.06	0.45	6.5	-0.17	0.35
18.3	0.19	0.23	10.3	-0.10	0.44
15.2	0.25	0.10	5.6	-0.11	0.42
11.7	0.25	0.09	1.0	-0.18	0.30
12.0	0.25	0.08	0.1	-0.23	0.17
15.8	0.28	0.03	1.4	-0.26	0.10
14.8	0.30	0.02	-1.2	-0.30	0.04

where  $h$  is the number of values in each sample. If these values are independent of one another,  $\chi^2$  has  $h-1$  degrees of freedom (d.f.). For "coherent" (autocorrelated) series the d.f. number (which also represents the mean of the chi-square distribution as well as one half its variance) is reduced to

$$\bar{\chi}^2 = \nu = h - \varepsilon(h), \quad (3.4.2)$$

(Radok, 1992) where

$$\varepsilon(h) = 1 + \frac{2(h-1)}{h} r_1 + \frac{2(h-2)}{h} r_2 + \dots + \frac{2}{h} r_{h-1}. \quad (3.4.3)$$

Here the  $r_i$  are the autocorrelations of observations  $i$  values apart, and  $h-\varepsilon(h)$  represents the number of independent observations in each section, which equals  $h-1$  when all autocorrelations are zero.

The basic probability for the chi-square distribution is

$$p = \left[ 2^{\nu/2} \Gamma \left( \frac{\nu}{2} \right) \right]^{-1} (\chi^2)^{(\nu/2)-1} \exp \left[ -\frac{\chi^2}{2} \right]. \quad (3.4.4)$$

Here  $\nu$  is the number of degrees of freedom which equals the mean as well as half the variance of the distribution. Then the logarithmic likelihood functions take the form

$$\log L_m(m) = -\frac{m}{2} \nu_m \log_e 2 - m \log_e \Gamma \left( \frac{\nu_m}{2} \right) + \left( \frac{\nu_m}{2} - 1 \right) \sum_1^m \log_e \chi^2 - \frac{1}{2} \sum_1^m \chi^2; \quad (3.4.5a)$$

$$\log_e L_{m+j}(m) = -\frac{m}{2} \nu_{m+j} \log_e 2 - m \log_e \Gamma \left( \frac{\nu_{m+j}}{2} \right) + \left( \frac{\nu_{m+j}}{2} - 1 \right) \sum_1^m \log_e \chi^2 - \frac{1}{2} \sum_1^m \chi^2; \quad (3.4.5b)$$

$$\begin{aligned} \log_e L_{m+j}(m+j) = & -\frac{m+j}{2} \nu_{m+j} \log_e 2 - (m+j) \log_e \Gamma \left( \frac{\nu_{m+j}}{2} \right) \\ & + \left( \frac{\nu_{m+j}}{2} - 1 \right) \sum_1^{m+j} \log_e \chi^2 - \frac{1}{2} \sum_1^{m+j} \chi^2; \end{aligned} \quad (3.4.5c)$$

$$\begin{aligned} \log_e L_m(m+j) = & -\frac{m+j}{2} v_m \log_e 2 - (m+j) \log_e \Gamma \left[ \frac{v_m}{2} \right] \\ & + \left[ \frac{v_m}{2} - 1 \right] \sum_1^{m+j} \log_e \chi^2 - \frac{1}{2} \sum_1^{m+j} \chi^2 . \end{aligned} \quad (3.4.5d)$$

The likelihood ratios become

$$q(m) = 2^{\frac{m}{2}(v_m - v_{m+j})} \left[ \frac{\Gamma \left[ \frac{v_m}{2} \right]}{\Gamma \left[ \frac{v_{m+j}}{2} \right]} \right]^m \exp \left[ \frac{1}{2} (v_{m+j} - v_m) \sum_1^m \log \chi^2 \right] , \quad (3.4.6)$$

and

$$q(m+j) = 2^{m + \frac{j}{2}(v_m - v_{m+j})} \left[ \frac{\Gamma \left[ \frac{v_m}{2} \right]}{\Gamma \left[ \frac{v_{m+j}}{2} \right]} \right]^{m+j} \exp \left[ \frac{1}{2} (v_{m+j} - v_m) \sum_1^m \log \chi^2 \right] , \quad (3.4.7)$$

so that

$$\log_e Q = \exp \left[ \frac{j}{2} (v_m - v_{m+j}) \log_e 2 + j \log_e \frac{\Gamma \left[ \frac{v_m}{2} \right]}{\Gamma \left[ \frac{v_{m+j}}{2} \right]} + \frac{v_{m+j} - v_m}{2} \sum_{m+1}^{m+j} \log_e \chi^2 \right] . \quad (3.4.8)$$

The gamma functions are evaluated with the Euler relation

$$\Gamma(z)^{-1} = (z) e^{cz} \prod_{i=1}^{\infty} \left[ 1 + \frac{z}{i} \right] e^{-z/i} , \quad (3.4.9)$$

where

$$c = \lim_{i \rightarrow \infty} \left[ 1 + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{i} \log_e i \right] = 0.5772156649 \dots \quad (3.4.10)$$

## Examples

Tests of the inverse sequential procedure for detecting changes in coherence use three series of values drawn from a random independent Gaussian data set. As described by Radok (1992), these data were rendered coherent by applying 3-term and 7-term moving averages that produced for different lags  $\lambda$  autocorrelations of magnitude  $r_\lambda = (M - \lambda)/M$ , where  $M$  is the length of the moving average. According to equations (3.4.2) and (3.4.3), the chi-square distributions of sample variances computed from sets of 5 successive values should have 2.53 and 1.14 degrees of freedom (d.f.), respectively, compared with 4 for the 5-term sample variances from the original series.

Table 4 shows the chi-square values used for testing the ability of the inverse sequential procedure to detect changes in coherence. Test I uses the first 5 values of the series with 2.53 d.f. as base and continues with the chi-square values of the non-coherent series; this implies a nominal change from 2.53 to 4 degrees of freedom. Test II after the same 5 initial values, continues with the chi-square values derived from the series of 7-term moving averages, implying a nominal change from 2.53 to 1.14 degrees of freedom. Finally test III uses the chi-square values of the 2.53 d.f. series throughout.

In test I the loss of coherence (increase in degrees of freedom) is shown by a slow but ultimately clear decrease in the no-change probability. In test II the increase in coherence does not lower most of the no-change probabilities below 0.40; a larger base period might have rendered the test more sensitive for this small signal. The probabilities of test II are similar to those of test III in which only one series of chi-square values was used to simulate absence of change.

## 4. Implementation

The inverse sequential procedure is designed to detect changes in statistical control from a few new observations added progressively to a representative sample drawn from the most recent controlled data regime. As an initialization (which also establishes the presence or absence of such regimes in the existing data) the procedure is applied to the full available data set. For this, progressive means, variances,

Table 4: Detection of a change in chi-square degrees of freedom  
(d.f. = mean = one half variance)

All tests use as base the following 5 values drawn from a chi-square series constructed with 2.53 degrees of freedom (for details cf. section 3.4):

2.72, 3.88, 2.12, 0.18, 1.27 sample mean (d.f.) 2.034

Another 15 values drawn from the same series are used for test III. Test I uses 15 values from a chi-square series constructed to have 4 d.f., while the data for Test II come from a series constructed to have 1.14 d.f.

Test I		Test II		Test III	
no-change probability		no-change probability		no-change probability	
4.31	0.47	0.40	0.48	0.79	0.50
5.33	0.39	5.00	0.50	4.04	0.50
1.07	0.42	0.10	0.49	1.36	0.50
5.03	0.34	0.69	0.47	4.67	0.42
3.21	0.30	0.08	0.40	2.03	0.47
0.57	0.37	1.11	0.39	2.14	0.46
3.08	0.34	0.84	0.38	3.16	0.42
1.54	0.35	3.03	0.42	0.59	0.46
3.19	0.32	1.02	0.42	2.26	0.45
0.35	0.38	1.20	0.43	2.96	0.43
1.70	0.39	2.35	0.45	3.86	0.39
2.35	0.37	1.80	0.47	1.27	0.40
3.85	0.33	0.50	0.45	1.49	0.41
14.1	0.18	0.44	0.44	8.74	0.29
12.3	0.10	0.42	0.42	3.37	0.26

and both linear and exponential regression coefficients are calculated as basic information. In addition, the data are divided into successive small subsamples; their means and variances, apart from being used for determining changes in coherence (c.f. Section 3.4) provide indications of the underlying probability distribution. For Gaussian data these means and variances are independent of one another; a linear dependence of sample variance on sample mean can be removed (and hence normality created) by taking the square root of each observed value, and a dependence of both mean and variance on the sample size is eliminated by a logarithmic transformation (for details see Kendall and Stuart, 1966, Chapter 37).

The Fortran program in the appendix has been designed interactive to allow for the fact that in practice some parameters may not have to be considered. For instance, no stationary mean can exist in the presence of a clear linear or exponential growth; for discrete rate events a Poisson rather than a normal mean is alone relevant, and only a chi-square variable is involved in the test of variance and coherence changes described in Section 3.4.

In order to establish base values of the parameters that do describe the current data and their variation, the inverse sequential procedure is applied backward from the most recent observation to find the most recent time interval within which the no-change probability remains high. Questions concerning the optimum length of such a "base period", and the efficiency of the inverse sequential procedure in detecting a given parameter change magnitude, will be addressed in systematic experiments with constructed sample series during the remainder of the project, together with real-time tests of some of the GEDEX data (Olsen and Warnock, 1992; Schiffer and Unimayar, 1992).

When several parameters are tested they will in general not lose control at or even near the same time. The successive no-change probabilities of a single parameter, as well as concurrent probabilities for several different parameters and/or separate series, can be combined, following Fisher (1941, Section 21.1) as a "fingerprint" of change in the form of a chi-square variate with  $2k$  degrees of freedom

$$\chi^2_{2kd.f.} = \sum_1^k (-2 \log_e \gamma_k)$$

where  $k$  is the number of probabilities thus combined.

Finally, it must be emphasized that the inverse sequential algorithm is intended for the *exploration*, rather than a *confirmation*, of parameter changes. Flueck and Brown (1993) have shown that an exploration can be carried out without the panoply of rigorous statistical procedures needed for a confirmation. Even so the full properties of exploratory parameters such as the no-change probability  $\gamma$  deserve to be clarified with numerical experiments planned for the remainder of this project.

**Acknowledgement:** Support for this work has been provided by NASA Grant NAGW-2706. Partial support for the second author was provided by NOAA's Climate and Global Change Program.

## References

- Flueck, J.A., and T.J. Brown, 1993: Criteria and methods for performing and evaluating solar weather studies. *J. Climate*, **6**, 373-385.
- Goel, A.L., 1982: Cumulative sum control charts. *Encyclopedia of Statistical Sciences*, **2**, John Wiley and Sons, New York, pp. 233-241.
- Gosh, B.K., 1988: Sequential analysis. *Encyclopedia of Statistical Sciences*, **8**, John Wiley and Sons, New York, pp. 378-387.
- Hoel, P.G., 1966: *Introduction to Mathematical Statistics*. John Wiley and Sons, New York, 427 pp.
- Kim, K.-Y. and G.R. North, 1991: Surface temperature fluctuations in a stochastic climate model, *J. Geophys. Res.*, **96**(D10), pp. 18,573-18,580.
- Klimes, V., 1974: The Hurst phenomenon - a puzzle?, *Water Resour. Res.*, **10**(4), pp. 675-688.
- Olsen, L.M. and A. Warnock, 1992: *Greenhouse Effects Detection Experiment: GEDEX Selected Data*



*Sets*. NASA Climate Data System Staff, Goddard Distributed Active Archive Center, NASA - Goddard Space Flight Center, Greenbelt, MD, 14 pp.

Radok, U., 1992: Assessing climate persistence from climate "noise". *Climatic Change*, **20**, pp. 155-167.

Schiffer, R.A. and S. Unimayar, 1992: *The Detection of Climate Change Due to the Enhanced Greenhouse Effect*, NASA GEDEX Atmospheric Temperature Workshop, Columbia, MD, 9-11 July 1991, 52 pp.

Shewhart, W.A., 1939: *Statistical method from the viewpoint of quality control*, U.S. Dept. of Agriculture, Washington, D.C., 155 pp.

Toth, Z., 1992: Quasi-stationary and transient periods in the northern hemisphere circulation series, *J. Climate*, **5**, 1235-1247.

Wald, A., 1947: *Sequential analysis*. John Wiley and Sons, New York, 212 pp.



## Appendix

Program "SEQUITOR" is designed to be an interactive program for analysis of Gaussian mean and variance, Poisson mean, chi-square (coherence), and linear (or exponential) trend changes in a sequential time series. The user typically will receive FORTRAN source code, providing an opportunity to make code changes as desired. For example, in the original code, data input is assumed to be free format. However, the user may desire to change this to a specific format. It may also be desirable for the user to add write statements that exclude headings, such that the results can then be easily imported into a graphics package.

An input/output flowchart is included in this appendix. Each square box represents an input step by the user, and an oval represents results output. A brief description of the input steps follows:

Enter input filename: This is the input data filename up to 80 characters.

Enter descriptive title: This a descriptive header of the data and/or the analysis up to 80 characters.

Enter number of values in series: This is the total number of rows in the input data file. It is assumed that the input data file contains a column of  $x$ -values (column 1) which represent an index or year for example, followed by  $n$  columns of  $y$ -values containing the actual series for analysis.

Enter missing value: This program allows for missing values. Enter a unique number (e.g., -999.) to represent missing values.

Enter column number: Input data files may contain multiple  $y$ -value columns. This entry should be from 1 to  $n$  depending upon which  $y$ -value column is desired for analysis. The very first column in the input data file is considered column 0.

Enter 0=continue, 1=reverse data input order: Often it is desirable to do the sequential monitoring analysis beginning with the most recent value and working backwards. This helps identify "regimes" in the time series. Enter either a 0 or 1.

Enter beginning and ending  $x$ -range values: Enter values separated by a comma. This range corresponds to the  $x$ -values in the very first column of the input data file. Since the  $x$ -values might represent an index or year, examples would be 10,18 or 1985,1992. Note that these values can represent a sub-set of the input data file.

Enter window size for sub-samples: In determining a regime, it is useful to examine smaller sub-sets of values. A typical sub-sample might contain 5 values. If the total number of cases in the series is not evenly divisible by the window size, the remaining values will be ignored in only the sub-sample analysis.

Enter analysis type: Here there are several options. Entering 1 through 4 places the user in the desired sequential analysis routine. Other options include changing the sub-sample size, changing

the column number, changing the data range, reversing the data order, or simply quitting the program.

Enter number of base period values: Within each analysis routine, the user is prompted for the number of base period values. These should typically be small, say 5 to 15 or so. Base period results is output at this point.

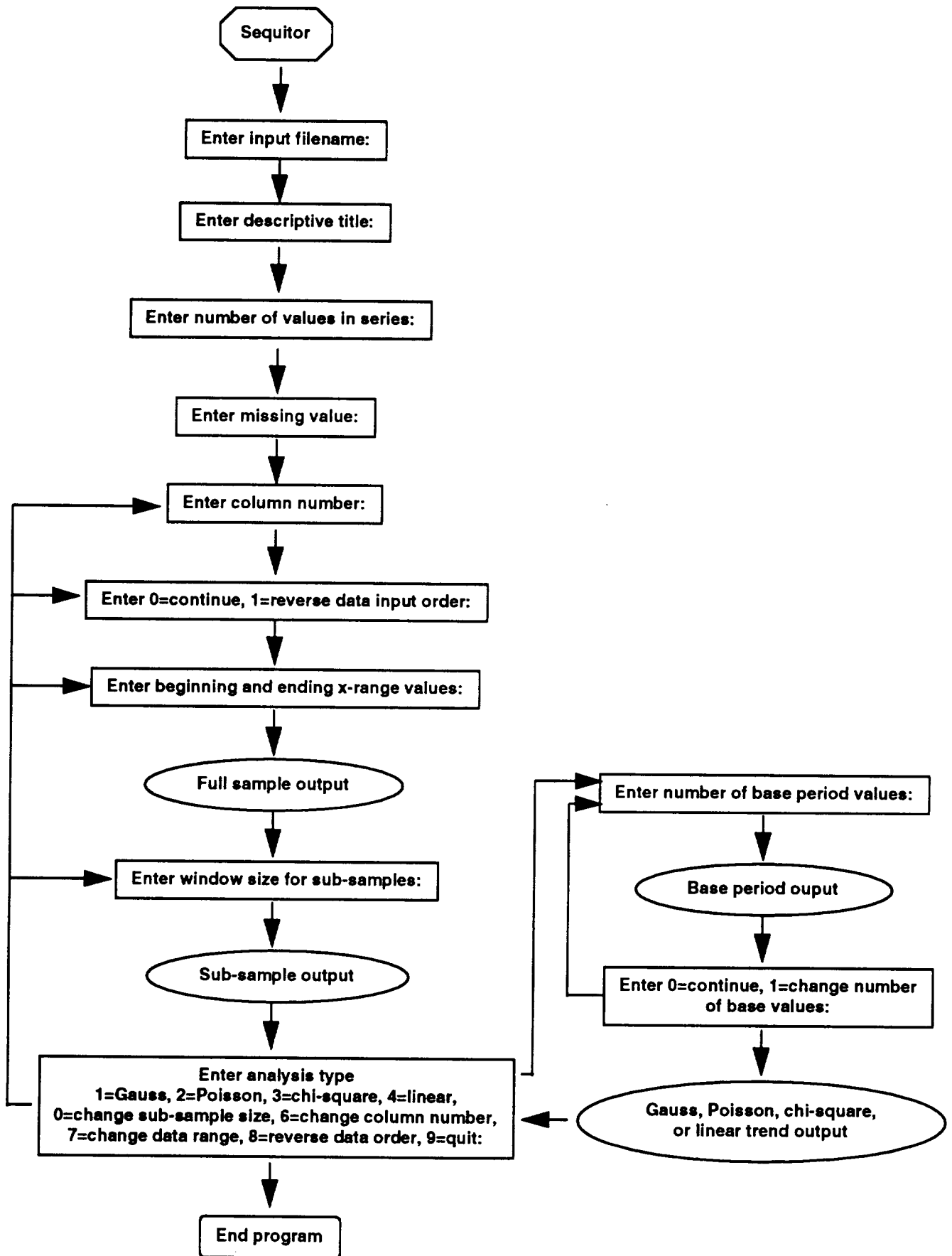
Enter 0=continue, 1=change number of base values: Upon examining the base period results, the user is given the option to change the base period size, or continue with the final analysis.

This program was written interactively because it is intended to be exploratory in nature. An attempt was made to allow the user to make changes during the analysis, instead of having to restart the program several times. Results are output to the screen and to a file named "sequitor.out", which is replaced each time the program is run.

The program contains minimal comments, but variables are defined at the beginning of each subroutine to help the user understand the program. It is intended to use the program in conjunction with this progress report, and some attempt at consistency of variable names in relation to the formulae has been made. However, it is possible that updates or changes to program will occur after this initial release. Hence, users may want to contact the authors for additional information.

Output from a sample analysis is included in this appendix. It is test I from Section 3.2 in the progress report. The associated input data are also included.

# Flowchart of input/output for program SEQUITOR



1978.	-5.0	-5.0	10.2
1979.	-1.3	-1.3	9.2
1980.	10.2	10.2	6.8
1981.	14.9	14.9	8.8
1982.	-0.6	-0.6	5.1
1983.	2.6	2.6	6.7
1984.	6.6	6.6	6.9
1985.	2.5	2.5	5.1
1986.	20.2	12.4	9.7
1987.	9.2	10.5	14.0
1988.	6.8	8.9	10.4
1989.	8.8	8.7	6.1
1990.	5.1	7.7	6.5
1991.	6.7	8.6	8.6
1992.	6.9	8.8	6.8

ENTER INPUT FILE NAME:  
 gauss.dat  
 ENTER DESCRIPTIVE TITLE:  
 Detection of change in Gaussian mean and variance, Test I  
 INVERSE SEQUENTIAL MONITORING (PROGRAM <SEQUITOR>)  
 ENTER NUMBER OF VALUES IN SERIES:  
 15  
 ENTER MISSING VALUE:  
 -999  
 ENTER COLUMN NUMBER:  
 1  
 ENTER 0=CONTINUE, 1=REVERSE DATA INPUT ORDER:  
 0  
 ENTER BEGINNING AND ENDING X-RANGE VALUES:  
 1978,1992

FULL SAMPLE UNIVARIATE STATISTICS:  
 X-VALUE RANGE = 1978.-1992.  
 NUMBER OF VALUES = 15  
 NUMBER OF MISSING VALUES = 0  
 SAMPLE MEAN = 6.240  
 SAMPLE VARIANCE = 40.034  
 SAMPLE SLOPE = 0.466

ENTER WINDOW SIZE FOR SUB-SAMPLES:  
 5

SUB-SAMPLE PARAMETERS:

INDEX RANGE	X-VALUE RANGE	NUMBER OF VALUES	MISSING VALUES	MEAN	VARIANCE	CHI-SQUARE
1- 5	1978.-1982.	5	0	3.640	71.713	7.165
6- 10	1983.-1987.	5	0	8.220	52.852	5.281
11- 15	1988.-1992.	5	0	6.860	1.723	0.172

ENTER ANALYSIS TYPE  
 1=GAUSS, 2=POISSON, 3=CHI-SQUARE, 4=LINEAR,  
 0=CHANGE SUB-SAMPLE SIZE, 6=CHANGE COLUMN NUMBER,  
 7=CHANGE DATA RANGE, 8=REVERSE DATA ORDER, 9=QUIT:  
 1

```

+-----+
| TEST FOR CHANGE IN GAUSSIAN MEAN AND VARIANCE |
+-----+

```

ENTER NUMBER OF VALUES IN BASE PERIOD:  
 5

BASE PERIOD PARAMETERS:  
 X-VALUE RANGE = 1978.-1982.  
 NUMBER OF VALUES = 5  
 NUMBER OF MISSING VALUES = 0  
 BASE MEAN = 3.640  
 BASE VARIANCE = 71.713

ENTER 0=CONTINUE, 1=CHANGE NUMBER OF BASE VALUES:  
 0

PROGRESSIVE PARAMETERS:

INDEX	X OBSERVATIONS	Y	MEAN	VARIANCE	GAMMA	DELTA GAMMA
-------	----------------	---	------	----------	-------	-------------

6	1983.000	2.60	3.467	57.551	0.486	
7	1984.000	6.60	3.914	49.361	0.456	0.030
8	1985.000	2.50	3.737	42.560	0.410	0.047
9	1986.000	20.20	5.567	67.353	0.453	-0.044
10	1987.000	9.20	5.930	61.189	0.422	0.031
11	1988.000	6.80	6.009	55.139	0.394	0.028
12	1989.000	8.80	6.242	50.775	0.351	0.043
13	1990.000	5.10	6.154	46.644	0.322	0.029
14	1991.000	6.70	6.193	43.078	0.280	0.042
15	1992.000	6.90	6.240	40.034	0.236	0.044

ENTER 0=CONTINUE, 1=CHANGE NUMBER OF BASE VALUES:

0

ENTER ANALYSIS TYPE

1=GAUSS, 2=POISSON, 3=CHI-SQUARE, 4=LINEAR,

0=CHANGE SUB-SAMPLE SIZE, 6=CHANGE COLUMN NUMBER,

7=CHANGE DATA RANGE, 8=REVERSE DATA ORDER, 9=QUIT:

9

END OF SEQUITOR RUN



PROGRAM SEQUITOR

```
C-----]
C          AUTHOR: TIMOTHY J. BROWN
C
C INVERSE SEQUENTIAL PROGRAM
C
C-----REVISION HISTORY-----]
C LEVEL AUTHOR DATE      DESCRIPTION
C =====
C .01A. TJB      93/04/28. ORIGINAL VERSION.
C-----]
C
C SIX SUBROUTINES ARE ATTACHED TO THE MAIN PROGRAM:
C 'SGAUSS' COMPUTES CHANGE IN GAUSSIAN MEAN AND VARIANCE.
C 'SPOISS' COMPUTES CHANGE IN POISSON MEAN.
C 'SCHI' COMPUTES CHANGE IN CHI-SQUARE DEGREES OF FREEDOM.
C 'SLINEAR' COMPUTES CHANGE IN LINEAR TREND.
C 'UNIVAR' COMPUTES UNIVARIATE STATISTICS MEAN, VARIANCE, AND SUM.
C 'RCOEFF' COMPUTES LLS REGRESSION B0 AND B1 COEFFICIENTS.
C
C INPUT IS ASSUMED TO BE FREE-FORMAT, BUT USER CAN CHANGE AS DESIRED.
C
C THE PARAMETER STATEMENT AND COMMON BLOCK IS LOCATED IN ALL SUBROUTINES.
C THE USER SHOULD CHANGE 'NDIM' AS REQUIRED.
C
C THE FOLLOWING ARRAYS AND VARIABLES ARE USED IN THE COMMON BLOCK:
C 'FCHI2' CHI-SQUARE VALUE FOR EACH SUB-SAMPLE.
C 'FINDEX' INDEX VALUE (1, 2,...N) ASSOCIATED WITH EACH Y-VALUE.
C 'FXVAL' INPUT X-VALUES.
C 'FYVAL' INPUT Y-VALUES.
C 'XDATA' WORK ARRAY FOR X-VALUES.
C 'YDATA' WORK ARRAY FOR Y-VALUES.
C
C 'FMISS' NUMBER REPRESENTING MISSING VALUES.
C 'NDIM' DIMENSION SIZE FOR DATA AND WORK ARRAYS.
C 'NCASE' NUMBER OF FULL SAMPLE VALUES WITHIN INDEX RANGE.
C 'OUNIT' OUTPUT UNIT NUMBER.
C
C THE FOLLOWING ARRAYS AND VARIABLES ARE USED IN THE MAIN PROGRAM:
C 'FDATA' HOLDS THE Y-VALUES WHEN THEY ARE INPUT; SHOULD BE
C DIMENSIONED >= NUMBER OF COLUMNS IN INPUT FILE.
C 'FXDATA' HOLDS THE ORIGINAL X-VALUES OR REVERSED ORDER VALUES.
C 'FYDATA' HOLDS THE ORIGINAL Y-VALUES OR REVERSED ORDER VALUES.
C 'XWORK' WORK ARRAY FOR X-VALUES.
C 'YWORK' WORK ARRAY FOR Y-VALUES.
C
C 'B0' INTERCEPT FROM LLS REGRESSION.
C 'B1' SLOPE FROM LLS REGRESSION.
C 'CFILE' INPUT DATA FILE NAME.
C 'CTITLE' DESCRIPTIVE TITLE.
C 'H' NUMBER OF VALUES WITHIN EACH SUB-SAMPLE.
C 'I' DO LOOP COUNTER.
C 'II' INDEX COUNTER.
C 'IBEG' INDEX COUNTER.
C 'IEND' INDEX COUNTER.
```

C 'IDIR' DATA DIRECTION FLAG (1=REVERSE DATA ORDER, 0=CONTINUE).  
C 'ITYPE' ANALYSIS TYPE.  
C 'IUNIT' INPUT UNIT NUMBER.  
C 'J' DO LOOP COUNTER.  
C 'K' COUNTER.  
C 'N' DO LOOP COUNTER.  
C 'NBEG' BEGINNING INDEX NUMBER FOR INDEX RANGE.  
C 'NEND' ENDING INDEX RANGE FOR INDEX RANGE.  
C 'NN' COUNTER.  
C 'NCOL' COLUMN NUMBER OF Y-VALUES TO BE ANALYZED.  
C THIS IS USEFUL FOR FILES CONTAINING MULTIPLE COLUMNS OF DATA.  
C X-VALUES ARE ASSUMED TO BE IN COLUMN ONE.  
C 'NMISS' NUMBER OF MISSING VALUES.  
C 'NPOP' NUMBER OF POPULATION VALUES.  
C 'NVALS' NUMBER OF NON-MISSING VALUES.  
C 'POPVAR' POPULATION VARIANCE FROM FULL SAMPLE.  
C 'S2' SUM OF SQUARES IN CHI-SQUARE CALCULATION.  
C 'XBEG' BEGINNING VALUE OF X-RANGE.  
C 'XEND' ENDING VALUE OF X-RANGE.  
C 'YBAR' MEAN OF Y-VALUES.  
C 'YSUM' SUM OF Y-VALUES.  
C 'YVAR' VARIANCE OF Y-VALUES.

C  
C PARAMETER (NDIM=150)  
C COMMON /WORK/ FXVAL (NDIM), FYVAL (NDIM), XDATA (NDIM),  
+ YDATA (NDIM), FINDEX (NDIM), FCHI2 (NDIM), NCASE,  
+ OUNIT, FMISS  
C REAL FDATA (15)  
C REAL FXDATA (NDIM), FYDATA (NDIM), XWORK (NDIM), YWORK (NDIM)  
C INTEGER H, OUNIT  
C CHARACTER\*80 CTITLE, CFILE

DATA IUNIT, OUNIT / 1, 2 /

C-----  
C  
C THIS SECTION REQUESTS THE INPUT INFORMATION, OPENS FILES, INPUTS  
C THE DATA, AND COMPUTES FULL SAMPLE UNIVARIATE STATISTICS.  
C

WRITE (\*, 801)  
READ (\*, 101) CFILE

OPEN (IUNIT, FILE=CFILE, STATUS='OLD')  
OPEN (OUNIT, FILE='sequitor.out')

WRITE (\*, 802)  
READ (\*, 101) CTITLE

WRITE (OUNIT, 900)  
WRITE (\*, 900)  
WRITE (OUNIT, 901) CTITLE

WRITE (\*, 803)  
READ (\*, \*) NPOP  
WRITE (\*, 804)  
READ (\*, \*) FMISS

```
3      CONTINUE

      WRITE(*,805)
      READ(*,*) NCOL
      REWIND IUNIT
C
C INPUT THE DATA AND FILL WORK ARRAYS. REVERSE DATA ORDER IF REQUESTED.
C
C-----
      DO 13 I = 1, NPOP

          READ(IUNIT,*) FXDATA(I), (FDATA(J),J=1,NCOL)

          FINDEX(I) = FLOAT(I)
          FYDATA(I) = FDATA(NCOL)

13      CONTINUE
C-----
5      CONTINUE

      WRITE(*,806)
      READ(*,*) IDIR
C-----
      IF( IDIR .EQ. 1 ) THEN

          K = 0
C-----
          DO 14 I = NPOP, 1, -1

              K = K + 1
              XWORK(K) = FXDATA(I)
              YWORK(K) = FYDATA(I)

14      CONTINUE
C-----
C-----
          DO 15 I = 1, NPOP

              FXDATA(I) = XWORK(I)
              FYDATA(I) = YWORK(I)

15      CONTINUE
C-----
      END IF
C-----
C
C FILL WORK ARRAYS WITH DATA WITHIN SELECTED INDEX RANGE AND COLUMN.
C
4      CONTINUE

      WRITE(*,807)
      READ(*,*) XBEG, XEND
C-----
      DO 18 I = 1, NPOP

          IF( FXDATA(I) .EQ. XBEG ) NBEG = I
```

```

      IF( FXDATA(I) .EQ. XEND ) NEND = I

18      CONTINUE
C-----
C-----
      IF( NBEG .LT. 1 ) THEN

          WRITE(*,810)
          GOTO 4

      END IF
C-----
C-----
      IF( NEND .GT. NPOP ) THEN

          WRITE(*,810)
          GOTO 4

      END IF
C-----
      NCASE = 0
C-----
      DO 16 I = 1, NPOP
C-----
          IF( I .GE. NBEG .AND. I .LE. NEND ) THEN

              NCASE = NCASE + 1

              FXVAL(NCASE) = FXDATA(I)
              FYVAL(NCASE) = FYDATA(I)
              XDATA(NCASE) = FINDEX(I)
              YDATA(NCASE) = FYVAL(NCASE)

          END IF

      END DO
C-----
16      CONTINUE
C-----
C
C COMPUTE FULL SAMPLE STATISTICS AND OUTPUT RESULTS
C
      CALL UNIVAR( NCASE, NVALS, NMISS, YBAR, YVAR, YSUM )
      CALL RCOEFF( NCASE, B0, B1 )

      IRANGE = NEND - NBEG + 1

      WRITE(OUNIT,902) FXVAL(1), FXVAL(IRANGE), NVALS, NMISS, YBAR,
+                   YVAR, B1
      WRITE(*,902) FXVAL(1), FXVAL(IRANGE), NVALS, NMISS, YBAR,
+               YVAR, B1

1      CONTINUE

      WRITE(*,808)
      READ(*,*) H

      WRITE(OUNIT,903)
```

```

        WRITE(*,903)
C
C  COMPUTE SUB-SAMPLE STATISTICS.
C
        POPVAR = YVAR
        NN = 0
        K = 0
        N = 0
        IBEG = -(H) + 1
C-----
        DO 17 I = NBEG, NEND

            K = K + 1
            IBEG = IBEG + 1
            N = N + 1
C-----
            IF( FYVAL(N) .GT. FMISS ) THEN

                NN = NN + 1
                YDATA(NN) = FYVAL(N)

            END IF
C-----
C-----
            IF( K .EQ. H ) THEN

                CALL UNIVAR( NN, NVALS, NMISS, YBAR, YVAR, YSUM )
C
C  COMPUTE CHI-SQUARE VALUES.
C
C-----
                IF( NVALS .GT. 0 ) THEN

                    S2 = 0.
C-----
                    DO 19 M = 1, NN

                        IF( YDATA(M) .NE. FMISS )
                            S2 = S2 + (YDATA(M) - YBAR)**2
                    +
                    CONTINUE
                    19
C-----
                    FCHI2(N) = S2 / POPVAR

                ELSE

                    FCHI2(N) = FMISS

                END IF
C-----
C
C  OUTPUT SUB-SAMPLE STATISTICS.
C
        II = (I-H) + 1
        IEND = IBEG + H - 1

```

```
      WRITE(OUNIT,904) II, I, FXVAL(IBEG), FXVAL(IEND), NVALS,  
+           NMIS, YBAR, YVAR, FCHI2(N)  
      WRITE(*,904) II, I, FXVAL(IBEG), FXVAL(IEND), NVALS, NMIS,  
+           YBAR, YVAR, FCHI2(N)
```

```
      NN = 0  
      K = 0
```

```
      END IF
```

```
C-----  
17      CONTINUE
```

```
C-----  
C  
C BRANCH OFF TO APPROPRIATE SUBROUTINE, CHANGE SUB-SAMPLE SIZE,  
C CHANGE COLUMN NUMBER, OR STOP PROGRAM.  
C IF CHOOSING CHI-SQUARE, THE NUMBER OF CASES BECOMES THE NUMBER  
C OF SUB-SAMPLE INTERVALS.
```

```
C  
2      CONTINUE
```

```
      WRITE(*,809)  
      READ(*,*) ITYPE
```

```
C-----  
      IF( ITYPE .EQ. 0 ) THEN
```

```
          GOTO 1
```

```
      ELSE IF( ITYPE .EQ. 1 ) THEN
```

```
          WRITE(OUNIT,1001)  
          WRITE(*,1001)
```

```
          CALL SGAUSS
```

```
      ELSE IF( ITYPE .EQ. 2 ) THEN
```

```
          WRITE(OUNIT,1002)  
          WRITE(*,1002)
```

```
          CALL SPOISS
```

```
      ELSE IF( ITYPE .EQ. 3 ) THEN
```

```
          WRITE(OUNIT,1003)  
          WRITE(*,1003)
```

```
          NCASE = INTERVL
```

```
          CALL SCHI
```

```
      ELSE IF( ITYPE .EQ. 4 ) THEN
```

```
          WRITE(OUNIT,1004)  
          WRITE(*,1004)
```

```
          CALL SLINEAR
```

```

ELSE IF( ITYPE .EQ. 6 ) THEN
    GOTO 3
ELSE IF( ITYPE .EQ. 7 ) THEN
    GOTO 4
ELSE IF( ITYPE .EQ. 8 ) THEN
    GOTO 5
ELSE IF( ITYPE .EQ. 9 ) THEN
    WRITE(OUNIT,907)
    WRITE(*,907)
    GOTO 999
ELSE
    WRITE(*,905)
    GOTO 2
END IF
C-----
GOTO 2
101  FORMAT(A)
801  FORMAT(' ENTER INPUT FILE NAME:')
802  FORMAT(' ENTER DESCRIPTIVE TITLE:')
803  FORMAT(' ENTER NUMBER OF VALUES IN SERIES:')
804  FORMAT(' ENTER MISSING VALUE:')
805  FORMAT(' ENTER COLUMN NUMBER:')
806  FORMAT(' ENTER 0=CONTINUE, 1=REVERSE DATA INPUT ORDER:')
807  FORMAT(' ENTER BEGINNING AND ENDING X-RANGE VALUES:')
808  FORMAT(/,' ENTER WINDOW SIZE FOR SUB-SAMPLES:')
809  FORMAT(/' ENTER ANALYSIS TYPE',/,
+      ' 1=GAUSS, 2=POISSON, 3=CHI-SQUARE, 4=LINEAR',/,/,
+      ' 0=CHANGE SUB-SAMPLE SIZE, 6=CHANGE COLUMN NUMBER',/,/,
+      ' 7=CHANGE DATA RANGE, 8=REVERSE DATA ORDER, 9=QUIT:')
810  FORMAT(/' RANGE EXCEEDS TOTAL NUMBER OF CASES')
900  FORMAT(' INVERSE SEQUENTIAL MONITORING (PROGRAM <SEQUITOR>)' )
901  FORMAT(/,A)
902  FORMAT(/,' FULL SAMPLE UNIVARIATE STATISTICS:',/,
+      ' X-VALUE RANGE = ',F5.0,'-',F5.0,/,
+      ' NUMBER OF VALUES = ',I3,/,
+      ' NUMBER OF MISSING VALUES = ',I3,/,
+      ' SAMPLE MEAN = ',F8.3,/,
+      ' SAMPLE VARIANCE = ',F8.3,/,
+      ' SAMPLE SLOPE = ',F8.3)

```





```

C 'SDMJ'    PROGRESSIVE STANDARD DEVIATION.
C 'SDMJ2'   PROGRESSIVE VARIANCE.
C 'T1'     WORK VARIABLE; FIRST TERM IN EITHER Q(M) OR Q(M+J) EQUATIONS.
C 'T2'     WORK VARIABLE; SECOND TERM IN EITHER Q(M) OR Q(M+J) EQUATIONS.
C 'T3'     WORK VARIABLE; THIRD TERM IN EITHER Q(M) OR Q(M+J) EQUATIONS.
C

```

```

    PARAMETER (NDIM=150)
    COMMON /WORK/ FXVAL(NDIM), FYVAL(NDIM), XDATA(NDIM),
+             YDATA(NDIM), FINDEX(NDIM), FCHI2(NDIM), NCASE,
+             OUNIT, FMISS
    REAL J, M
    INTEGER OUNIT

```

```

C
C FILL WORK ARRAYS WITH BASE PERIOD VALUES, COMPUTE BASE PERIOD
C STATISTICS, AND OUTPUT RESULTS.
C

```

```

1    CONTINUE

```

```

    WRITE(*,801)
    READ(*,*) MBASE

```

```

C-----
    DO 20 I = 1, MBASE

```

```

        XDATA(I) = FINDEX(I)
        YDATA(I) = FYVAL(I)

```

```

20   CONTINUE

```

```

C-----
    CALL UNIVAR( MBASE, NVALS, NMISS, YBAR, YVAR, YSUM )

```

```

C-----
    IF( NVALS .GT. 0 ) THEN

```

```

        WRITE(OUNIT,1001) FXVAL(1), FXVAL(MBASE), NVALS, NMISS, YBAR,
+             YVAR
        WRITE(*,1001) FXVAL(1), FXVAL(MBASE), NVALS, NMISS, YBAR,
+             YVAR

```

```

    ELSE

```

```

        WRITE(*,1000)
        RETURN

```

```

    END IF

```

```

C-----
    WRITE(*,802)
    READ(*,*) ITYPE
    IF( ITYPE .EQ. 1 ) GOTO 1

```

```

    WRITE(OUNIT,1002)
    WRITE(*,1002)

```

```

C
C INITIALIZE AND COMPUTE FIXED VALUES; BEGIN PROGRESSIVE VALUE LOOP.
C

```

```

    BARM = YBAR
    SDM = SQRT(YVAR)
    SDM2 = YVAR

```

```

M = FLOAT(MBASE)
GAMOLD = FMISS
J1 = MBASE + 1
J2 = NCASE

C
C  FILL PROGRESSIVE WORK ARRAYS, COMPUTE PROGRESSIVE STATISTICS, AND
C  OUTPUT RESULTS.
C
C-----
      DO 22 JJ = J1, J2

          XDATA(JJ) = FINDEX(JJ)
          YDATA(JJ) = FYVAL(JJ)

          CALL UNIVAR( JJ, NVALS, NMISS, YBAR, YVAR, YSUM )

C
C  COMPUTE GAMMA.  SEE TEXT FOR EQUATION DETAILS.
C
C-----
      IF( NVALS .GT. 0 ) THEN

          BARMJ = YBAR
          SDMJ = SQRT(YVAR)
          SDMJ2 = YVAR
          J = FLOAT(JJ) - M

          T1 = M * ALOG(SDM / SDMJ )
          T2 = ((M - 1.) / 2.) * (1. - (SDM2 / SDMJ2))
          T3 = (M / (2. * SDM2)) * (BARMJ - BARM)**2
          QM = EXP(T1 + T2 - T3)

          T1 = (M + J) * ALOG(SDM / SDMJ )
          T2 = ((M + J - 1.) / 2.) * ((SDMJ2 / SDM2) - 1.)
          T3 = ((M + J) / (2. * SDM2)) * (BARMJ - BARM)**2
          QMJ = EXP(T1 + T2 + T3)

          Q = QMJ / QM
          GAMMA = 1. / (1. + SQRT(Q))
          DGAMMA = GAMOLD - GAMMA

C-----
      IF( GAMOLD .EQ. FMISS ) THEN

          WRITE(OUNIT,1003) JJ, FXVAL(JJ), YDATA(JJ), YBAR, YVAR,
+                               GAMMA, CFLAG
          WRITE(*,1003) JJ, FXVAL(JJ), YDATA(JJ), YBAR, YVAR,
+                               GAMMA, CFLAG

      ELSE

          WRITE(OUNIT,1003) JJ, FXVAL(JJ), YDATA(JJ), YBAR, YVAR,
+                               GAMMA, CFLAG, DGAMMA
          WRITE(*,1003) JJ, FXVAL(JJ), YDATA(JJ), YBAR, YVAR,
+                               GAMMA, CFLAG, DGAMMA

      END IF

C-----

```

GAMOLD = GAMMA

ELSE

WRITE(OUNIT,1004) JJ, FXVAL(JJ), YDATA(JJ)  
 WRITE(\*,1004) JJ, FXVAL(JJ), YDATA(JJ)  
 GAMOLD = FMISS

END IF

C-----  
 22 CONTINUE  
 C-----

WRITE(\*,802)  
 READ(\*,\*) ITYPE  
 IF( ITYPE .EQ. 1 ) GOTO 1

801 FORMAT(/,' ENTER NUMBER OF VALUES IN BASE PERIOD:')  
 802 FORMAT(/,' ENTER 0=CONTINUE, 1=CHANGE NUMBER OF BASE VALUES:')

1000 FORMAT(' ALL VALUES IN BASE PERIOD MISSING')

1001 FORMAT(/,'BASE PERIOD PARAMETERS:',/,  
 + 'X-VALUE RANGE = ',F5.0,'-',F5.0,/,  
 + 'NUMBER OF VALUES = ',I3,/,  
 + 'NUMBER OF MISSING VALUES = ',I3,/,  
 + 'BASE MEAN = ',F8.3,/,  
 + 'BASE VARIANCE = ',F8.3,/) )

1002 FORMAT(/,'PROGRESSIVE PARAMETERS:',/,  
 + ' X Y DELTA',/,  
 + 'INDEX OBSERVATIONS MEAN VARIANCE GAMMA GAMMA',/,  
 + '=====') )

1003 FORMAT(I5,2X,F8.3,1X,F7.2,2(2X,F8.3),2X,F6.3,2X,F6.3)

1004 FORMAT(I5,2X,F8.3,1X,F7.2)

RETURN  
 END

C-----  
 SUBROUTINE SPOISS

C  
 C SUBROUTINE TO COMPUTE CHANGE IN POISSON MEAN.  
 C  
 C VARIABLES USED IN SUBROUTINE NOT DESCRIBED IN MAIN PROGRAM:  
 C 'BARM' MEAN OF BASE PERIOD VALUES.  
 C 'BARMJ' PROGRESSIVE MEAN.  
 C 'DGAMMA' CHANGE IN GAMMA FROM PREVIOUS VALUE.  
 C 'GAMMA' GAMMA VALUE.  
 C 'GAMOLD' PREVIOUS VALUE OF GAMMA.  
 C 'J' PROGRESSIVE VALUE INDEX.  
 C 'JJ' DO LOOP COUNTER.  
 C 'J1' STARTING VALUE FOR DO LOOP COMPUTING PROGRESSIVE VALUES.  
 C 'J2' ENDING VALUE FOR DO LOOP COMPUTING PROGRESSIVE VALUES.  
 C 'M' REAL VALUE OF NUMBER OF BASE VALUES.  
 C 'MBASE' NUMBER OF BASE PERIOD VALUES.

```
C 'Q'      COMBINED LIKELIHOOD RATIOS.
C 'T1'     WORK VARIABLE; FIRST TERM IN EITHER Q EQUATION.
C 'T2'     WORK VARIABLE; SECOND TERM IN EITHER Q EQUATION.
C 'T3'     WORK VARIABLE; THIRD TERM IN EITHER Q EQUATION.
C
      PARAMETER (NDIM=150)
      COMMON /WORK/ FXVAL(NDIM), FYVAL(NDIM), XDATA(NDIM),
+             YDATA(NDIM), FINDEX(NDIM), FCHI2(NDIM), NCASE,
+             OUNIT, FMISS
      REAL J, M
      INTEGER OUNIT
C
C FILL WORK ARRAYS WITH BASE PERIOD VALUES, COMPUTE BASE PERIOD
C STATISTICS, AND OUTPUT RESULTS.
C
1     CONTINUE

      WRITE(*,801)
      READ(*,*) MBASE
C-----
      DO 20 I = 1, MBASE

          XDATA(I) = FINDEX(I)
          YDATA(I) = FYVAL(I)

20    CONTINUE
C-----
      CALL UNIVAR( MBASE, NVALS, NMISS, YBAR, YVAR, YSUM )
C-----
      IF( NVALS .GT. 0 ) THEN

          WRITE(OUNIT,2001) FXVAL(1), FXVAL(MBASE), NVALS, NMISS,
+             YBAR, YVAR, YSUM
          WRITE(*,2001) FXVAL(1), FXVAL(MBASE), NVALS, NMISS, YBAR,
+             YVAR, YSUM

      ELSE

          WRITE(*,2000)
          RETURN

      END IF
C-----
      WRITE(*,802)
      READ(*,*) ITYPE
      IF( ITYPE .EQ. 1 ) GOTO 1

      WRITE(OUNIT,2002)
      WRITE(*,2002)
C
C INITIALIZE AND COMPUTE FIXED VALUES; BEGIN PROGRESSIVE VALUE LOOP.
C
      BARM = YBAR
      M = FLOAT(MBASE)
      GAMOLD = FMISS
      J1 = MBASE + 1
```

```

      J2 = NCASE
C
C  FILL PROGRESSIVE WORK ARRAYS, COMPUTE PROGRESSIVE STATISTICS, AND
C  OUTPUT RESULTS.
C
C-----
      DO 22 JJ = J1, J2

          XDATA(JJ) = FINDEX(JJ)
          YDATA(JJ) = FYVAL(JJ)

          CALL UNIVAR( JJ, NVALS, NMISS, YBAR, YVAR, YSUM )
C
C  COMPUTE GAMMA.  SEE TEXT FOR EQUATION DETAILS.
C
C-----
          IF( NVALS .GT. 0 ) THEN

              BARMJ = YBAR
              J = FLOAT(JJ) - M

              T1 = (BARMJ * (M + J)) - (BARM * M)
              T2 = ALOG(BARMJ / BARM)
              T3 = (BARMJ - BARM) * J
              Q = EXP(T1 * T2 - T3)
              GAMMA = 1. / (1. + SQRT(Q))
              DGAMMA = GAMOLD - GAMMA
C-----
              IF( GAMOLD .EQ. FMISS ) THEN

                  WRITE(OUNIT,2003) JJ, FXVAL(JJ), YDATA(JJ), BARMJ, YVAR,
+                               YSUM, GAMMA, CFLAG
                  WRITE(*,2003) JJ, FXVAL(JJ), YDATA(JJ), BARMJ, YVAR,
+                               YSUM, GAMMA, CFLAG

                  ELSE

                  WRITE(OUNIT,2003) JJ, FXVAL(JJ), YDATA(JJ), BARMJ, YVAR,
+                               YSUM, GAMMA, CFLAG, DGAMMA
                  WRITE(*,2003) JJ, FXVAL(JJ), YDATA(JJ), BARMJ, YVAR,
+                               YSUM, GAMMA, CFLAG, DGAMMA

                  END IF
C-----
              GAMOLD = GAMMA

          ELSE

              WRITE(OUNIT,2004) JJ, FXVAL(JJ), YDATA(JJ)
              WRITE(*,2004) JJ, FXVAL(JJ), YDATA(JJ)
              GAMOLD = FMISS

          END IF
C-----
      22  CONTINUE
C-----

```

```

WRITE(*,802)
READ(*,*) ITYPE
IF( ITYPE .EQ. 1 ) GOTO 1

801  FORMAT(/,' ENTER NUMBER OF VALUES IN BASE PERIOD:')
802  FORMAT(/,' ENTER 0=CONTINUE, 1=CHANGE NUMBER OF BASE VALUES:')

2000  FORMAT(' ALL VALUES IN BASE PERIOD MISSING')

2001  FORMAT(/,'BASE PERIOD PARAMETERS:',/,
+      'X-VALUE RANGE = ',F5.0,'-',F5.0,/,
+      'NUMBER OF VALUES = ',I3,/,
+      'NUMBER OF MISSING VALUES = ',I3,/,
+      'BASE MEAN = ',F8.3,/,
+      'BASE VARIANCE = ',F8.3,/,
+      'BASE SUM = ',F6.0,/)

2002  FORMAT(/,'PROGRESSIVE PARAMETERS:',/,
+      '      X          Y          ',
+      ' DELTA',/,
+      ' INDEX    OBSERVATIONS    MEAN    VARIANCE    SUM    GAMMA',
+      ' GAMMA',/,
+      ' =====',
+      ' =====')

2003  FORMAT(I5,2X,F8.3,1X,F7.0,2(2X,F8.3),2X,f6.0,2X,F6.3,2X,F6.3)

2004  FORMAT(I5,2X,F8.3,1X,F7.0)

RETURN
END

```

```

C-----
C      SUBROUTINE SCHI
C
C SUBROUTINE TO COMPUTE CHANGE IN CHI-SQUARE DEGREES OF FREEDOM.
C
C VARIABLES USED IN SUBROUTINE NOT DESCRIBED IN MAIN PROGRAM:
C 'CHANGE' PERCENT CHANGE OF 'OLDPROD' TO 'PRODZ'.
C 'D' GAMMA FUNCTION CONSTANT.
C 'DGAMMA' CHANGE IN GAMMA FROM PREVIOUS VALUE.
C 'GAMMA' GAMMA VALUE.
C 'GAMNUM' VALUE OF THE GAMMA FUNCTION.
C 'GAMOLD' PREVIOUS VALUE OF GAMMA.
C 'J' PROGRESSIVE VALUE INDEX.
C 'JJ' DO LOOP COUNTER.
C 'J1' STARTING VALUE FOR DO LOOP COMPUTING PROGRESSIVE VALUES.
C 'J2' ENDING VALUE FOR DO LOOP COMPUTING PROGRESSIVE VALUES.
C 'M' REAL VALUE OF NUMBER OF BASE VALUES.
C 'MBASE' NUMBER OF BASE PERIOD VALUES.
C 'NUM' BASE PERIOD MEAN.
C 'NUMJ' PROGRESSIVE MEAN.
C 'OLDPROD' PREVIOUS VALUE OF 'PRODZ'.
C 'PRODZ' PRODUCT OF GAMMA FUNCTION EULER RELATION.
C 'Q' COMBINED LIKELIHOOD RATIOS.
C 'SUMCHI2' PROGRESSIVE SUM OF CHI-SQUARE VALUES.
C 'T1' WORK VARIABLE; FIRST TERM IN EITHER Q EQUATION.

```

```

C          FIRST TERM IN GAMMA FUNCTION EULER RELATION.
C 'T2'     WORK VARIABLE; SECOND TERM IN EITHER Q EQUATION.
C          SECOND TERM IN GAMMA FUNCTION EULER RELATION.
C 'T3'     WORK VARIABLE; THIRD TERM IN EITHER Q EQUATION.
C 'Z'      VALUE USED IN GAMMA FUNCTION EULER RELATION.
C

```

```

PARAMETER (NDIM=150)
COMMON /WORK/ FXVAL(NDIM), FYVAL(NDIM), XDATA(NDIM),
+            YDATA(NDIM), FINDEX(NDIM), FCHI2(NDIM), NCASE,
+            OUNIT, FMISS
REAL J, M, NUM, NUMJ
INTEGER OUNIT

```

```

D = .5772156649

```

```

C
C FILL WORK ARRAYS WITH BASE PERIOD VALUES, COMPUTE BASE PERIOD
C STATISTICS, AND OUTPUT RESULTS.
C

```

```

1      CONTINUE

```

```

      WRITE(*,801)
      READ(*,*) MBASE

```

```

C-----
      DO 20 I = 1, MBASE

```

```

          XDATA(I) = FLOAT(I)
          YDATA(I) = FCHI2(I)

```

```

20     CONTINUE

```

```

C-----
      CALL UNIVAR( MBASE, NVALS, NMISS, YBAR, YVAR, YSUM )

```

```

C-----
      IF( NVALS .GT. 0 ) THEN

```

```

          WRITE(OUNIT,3001) FXVAL(1), FXVAL(MBASE), NVALS, NMISS, YBAR
          WRITE(*,3001) FXVAL(1), FXVAL(MBASE), NVALS, NMISS, YBAR

```

```

      ELSE

```

```

          WRITE(*,3000)
          RETURN

```

```

      END IF

```

```

C-----
      WRITE(*,802)
      READ(*,*) ITYPE
      IF( ITYPE .EQ. 1 ) GOTO 1

```

```

      WRITE(OUNIT,3002)
      WRITE(*,3002)

```

```

C
C COMPUTE GAMMA FUNCTION FOR THE BASE PERIOD.
C 10,000 ITERATIONS OF THE LOOP IS ARBITRARY, BUT DOES SEEM TO ALLOW
C FOR REASONABLE CONVERGENCE OF THE FUNCTION. 'CHANGE' IS USED TO
C COMPUTE THE PERCENT CHANGE FROM THE PREVIOUS FUNCTION VALUE.

```

C THUS, IT CAN BE USED TO EXIT FROM A LARGE ITERATION LOOP.  
 C HOWEVER, WE HAVE NOT USED THIS CRITERIA CONSISTENTLY THUS FAR, BUT  
 C WILL LEAVE IT BUILT INTO THE CODE FOR NOW.  
 C

```

NUM = YBAR
PRODZ = 1.
Z = NUM / 2.
    
```

C-----

```

DO 26 I = 1, 10000
    
```

```

T1 = 1. + Z / FLOAT(I)
T2 = EXP(-Z / FLOAT(I))
PRODZ = PRODZ * T1 * T2
    
```

C IF( I .GT. 1 ) CHANGE = ABS((OLDPROD - PRODZ) / OLDPROD)

C IF( CHANGE .LE. 1.E-5 ) GOTO 27

C OLDPROD = PRODZ

26 CONTINUE

C-----

27 CONTINUE

```

GAMNUM = 1. / (Z * EXP(D * Z) * PRODZ)
    
```

C  
 C INITIALIZE AND COMPUTE FIXED VALUES; BEGIN PROGRESSIVE VALUE LOOP.  
 C

```

J1 = MBASE + 1
J2 = NCASE
M = FLOAT(MBASE)
GAMOLD = FMISS
SUMCHI2 = 0.
    
```

C  
 C FILL PROGRESSIVE WORK ARRAYS, COMPUTE PROGRESSIVE STATISTICS, AND  
 C OUTPUT RESULTS.  
 C

C-----

```

DO 22 JJ = J1, J2
    
```

```

SUMCHI2 = SUMCHI2 + ALOG(FCHI2(JJ))
XDATA(JJ) = FLOAT(JJ)
YDATA(JJ) = FCHI2(JJ)
    
```

```

CALL UNIVAR( JJ, NVALS, NMISS, YBAR, YVAR, YSUM )
    
```

C-----

```

IF( NVALS .GT. 0 ) THEN
    
```

C  
 C COMPUTE GAMMA FUNCTION FOR PROGRESSIVE VALUES.  
 C SEE FURTHER DESCRIPTION ABOVE.  
 C

```

NUMJ = YBAR
J = FLOAT(JJ) - M
PRODZ = 1.
Z = NUMJ / 2.
    
```

C-----



```

DO 28 I = 1, 10000

    T1 = 1. + Z / FLOAT(I)
    T2 = EXP(-Z / FLOAT(I))
    PRODZ = PRODZ * T1 * T2

C          IF( I .GT. 1 ) CHANGE = ABS((OLDPROD - PRODZ)
C      +          / OLDPROD)

C          IF( CHANGE .LE. 1.E-5 ) GOTO 29

C          OLDPROD = PRODZ

28          CONTINUE
C-----
29          CONTINUE

    GAMNUMJ = 1. / (Z * EXP(D * Z) * PRODZ)

C
C COMPUTE GAMMA.  SEE TEXT FOR EQUATION DETAILS.
C
    T1 = (J * (NUM - NUMJ) * ALOG(2.)) / 2.
    T2 = J * ALOG(GAMNUM / GAMNUMJ)
    T3 = ((NUMJ - NUM) / 2.) * SUMCHI2
    Q = EXP(T1 + T2 + T3)

    GAMMA = 1. / (1. + SQRT(Q))
    DGAMMA = GAMOLD - GAMMA
C-----
    IF( GAMOLD .EQ. FMISS ) THEN

        WRITE(OUNIT,3003) JJ, YDATA(JJ), NUMJ, GAMMA, CFLAG

    ELSE

        WRITE(OUNIT,3003) JJ, YDATA(JJ), NUMJ, GAMMA, CFLAG,
C      +          DGAMMA

    END IF
C-----
    GAMOLD = GAMMA

ELSE

    WRITE(OUNIT,3004) JJ, YDATA(JJ)
    WRITE(*,3004) JJ, YDATA(JJ)
    GAMOLD = FMISS

END IF
C-----
22          CONTINUE
C-----
    WRITE(*,802)
    READ(*,*) ITYPE
    IF( ITYPE .EQ. 1 ) GOTO 1

```

```

801  FORMAT(//, ' ENTER NUMBER OF VALUES IN BASE PERIOD:')
802  FORMAT(//, ' ENTER 0=CONTINUE, 1=CHANGE NUMBER OF BASE VALUES:')

3000  FORMAT(' ALL VALUES IN BASE PERIOD MISSING')

3001  FORMAT(//, 'BASE PERIOD PARAMETERS:', //,
+      ' X-VALUE RANGE = ', F5.0, '-', F5.0, //,
+      ' NUMBER OF VALUES = ', I3, //,
+      ' NUMBER OF MISSING VALUES = ', I3, //,
+      ' BASE DEGREES OF FREEDOM = ', F8.3, //)

3002  FORMAT(//, 'PROGRESSIVE PARAMETERS:', //,
+      ' CHI-SQUARE          DELTA', //,
+      ' INDEX  OBSERVATIONS    MEAN      GAMMA    GAMMA', //,
+      ' =====  =====  =====  =====  =====')

3003  FORMAT(I5, 2X, F10.3, 2X, F8.3, 2X, F6.3, 2X, F6.3)

3004  FORMAT(I5, 2X, F10.3)

      RETURN
      END

```

C-----  
SUBROUTINE SLINEAR

```

C
C SUBROUTINE TO COMPUTE CHANGE IN LINEAR TREND.
C
C VARIABLES USED IN SUBROUTINE NOT DESCRIBED IN MAIN PROGRAM:
C 'AM'      BASE PERIOD INTERCEPT FROM LLS REGRESSION.
C 'BM'      BASE PERIOD SLOPE FROM LLS REGRESSION.
C 'DM'      CONSTANT USED IN Q(M) LIKELIHOOD RATIO.
C 'DMJ'     CONSTANT USED IN Q(M+J) LIKELIHOOD RATIO.
C 'DENOM'   DENOMINATOR IN 'T21' AND 'T22' TERMS.
C 'DGAMMA'  CHANGE IN GAMMA FROM PREVIOUS VALUE.
C 'GAMMA'   GAMMA VALUE.
C 'GAMOLD'  PREVIOUS VALUE OF GAMMA.
C 'I'       DO LOOP COUNTER.
C 'J'       PROGRESSIVE VALUE INDEX.
C 'JJ'      DO LOOP COUNTER.
C 'J1'      STARTING VALUE FOR DO LOOP COMPUTING PROGRESSIVE VALUES.
C 'J2'      ENDING VALUE FOR DO LOOP COMPUTING PROGRESSIVE VALUES.
C 'LNL1'    NATURAL LOG LIKELIHOOD 1.
C 'LNL2'    NATURAL LOG LIKELIHOOD 2.
C 'LNL3'    NATURAL LOG LIKELIHOOD 3.
C 'LNL4'    NATURAL LOG LIKELIHOOD 4.
C 'M'       REAL VALUE OF NUMBER OF BASE VALUES.
C 'MBASE'   NUMBER OF BASE PERIOD VALUES.
C 'PREDM'   PREDICTED VALUES FROM REGRESSION EQUATION USING 'M' VALUES.
C 'PREDMJ'  PREDICTED VALUES FROM REGRESSION EQUATION USING 'M+J' VALUES.
C 'Q'       RATIO OF QMJ/QM.
C 'QM'      BASE PERIOD LIKELIHOOD RATIO.
C 'QMJ'     PROGRESSIVE PERIOD LIKELIHOOD RATIO.
C 'RESIDM'  ARRAY OF BASE PERIOD RESIDUALS FROM LLS REGRESSION.
C 'RESIDMJ' ARRAY OF PROGRESSIVE VALUE RESIDUALS FROM LLS REGRESSION.
C 'SSM'     SUM OF SQUARES FOR BASE PERIOD.
C 'SSMJ'    SUM OF SQUARES FOR 'M+J' VALUES.

```

```

C 'SUMT2'  SUM OF 'T2' TERM IN 'LNL1' AND 'LNL4' FORMULAE.
C 'SUMT21' SUM OF 'T2' TERM IN 'LNL2' FORMULAE.
C 'SUMT22' SUM OF 'T2' TERM IN 'LNL3' FORMULAE.
C 'T'     REAL VALUE OF BASE PERIOD LOOP INDEX.
C 'T1'    WORK VARIABLE FOR FIRST TERM IN 'LNL1' AND 'LNL4' FORMULAE.
C 'T2'    WORK VARIABLE FOR SECOND TERM IN 'LNL1' AND 'LNL4' FORMULAE.
C 'T11'   WORK VARIABLE FOR FIRST TERM IN 'LNL2' FORMULAE.
C 'T12'   WORK VARIABLE FOR FIRST TERM IN 'LNL3' FORMULAE.
C 'T21'   WORK VARIABLE FOR SECOND TERM IN 'LNL2' FORMULAE.
C 'T22'   WORK VARIABLE FOR SECOND TERM IN 'LNL3' FORMULAE.
C

```

```

PARAMETER (NDIM=150)
COMMON /WORK/ FXVAL(NDIM), FYVAL(NDIM), XDATA(NDIM),
+           YDATA(NDIM), FINDEX(NDIM), FCHI2(NDIM), NCASE,
+           OUNIT, FMISS
REAL RESIDM(NDIM), RESIDMJ(NDIM), PREDM(NDIM), PREDMJ(NDIM)
REAL M, J, LNL1, LNL2, LNL3, LNL4
INTEGER OUNIT

```

```

C
C  FILL WORK ARRAYS WITH BASE PERIOD VALUES, COMPUTE BASE PERIOD
C  STATISTICS, AND OUTPUT RESULTS.
C

```

```

C
1  CONTINUE

```

```

WRITE(*,801)
READ(*,*) MBASE

```

```

C-----
DO 20 I = 1, NCASE

```

```

XDATA(I) = FINDEX(I)
YDATA(I) = FYVAL(I)

```

```

20 CONTINUE

```

```

C-----
CALL UNIVAR( MBASE, NVALS, NMISS, YBAR, YVAR, YSUM )

```

```

C-----
IF( NVALS .GT. 0 ) THEN

```

```

CALL RCOEFF( MBASE, AM, BM )

```

```

WRITE(OUNIT,4001) FXVAL(1), FXVAL(MBASE), NVALS, NMISS,
+           YBAR, YVAR, BM
WRITE(*,4001) FXVAL(1), FXVAL(MBASE), NVALS, NMISS, YBAR,
+           YVAR, BM

```

```

ELSE

```

```

WRITE(*,4000)
RETURN

```

```

END IF

```

```

C-----
WRITE(*,802)
READ(*,*) ITYPE
IF( ITYPE .EQ. 1 ) GOTO 1

```

```

WRITE (OUNIT,4002)
WRITE (*,4002)

M = FLOAT(MBASE)
DM = (M * (M**2 - 1.)) / 12.
SSM = 0.
C-----
DO 22 I = 1, MBASE
C-----
    IF( YDATA(I) .NE. FMISS ) THEN

        SSM = SSM + (YDATA(I) - YBAR)**2

    END IF
C-----
22    CONTINUE
C-----
C-----
DO 23 I = 1, NCASE
C-----
    IF( YDATA(I) .NE. FMISS ) THEN

        PREDM(I) = (XDATA(I) * BM + AM)
        RESIDM(I) = (YDATA(I) - PREDM(I))

    ELSE

        PREDM(I) = FMISS
        RESIDM(I) = FMISS

    END IF
C-----
23    CONTINUE
C-----
    SUMT2 = 0.
    T1 = (-1.) * (M - 2.) / (2. * (SSM - (BM**2 * DM)))
C-----
DO 24 I = 1, MBASE
C-----
    IF( YDATA(I) .NE. FMISS ) THEN

        T = FLOAT(I)

        T2 = RESIDM(I)**2 / ((M + 1.) / M)
        + (T - ((M + 1.) / 2.))**2 / DM

        SUMT2 = SUMT2 + T2

    END IF
C-----
24    CONTINUE

    LNLI = T1 * SUMT2
C-----
C
C INITIALIZE AND COMPUTE FIXED VALUES; BEGIN PROGRESSIVE VALUE LOOP.

```

```

C
      GAMOLD = FMISS
      J1 = MBASE + 1
      J2 = NCASE
C
C  FILL PROGRESSIVE WORK ARRAYS, COMPUTE PROGRESSIVE STATISTICS, AND
C  OUTPUT RESULTS.
C
C-----
      DO 26 JJ = J1, J2

      XDATA(JJ) = FINDEX(JJ)
      YDATA(JJ) = FYVAL(JJ)

      CALL UNIVAR( JJ, NVALS, NMISS, YBAR, YVAR, YSUM )

      IF( YDATA(JJ) .EQ. FMISS ) NVALS = 0
C
C  COMPUTE EQUATION TERMS AND GAMMA.  SEE TEXT FOR EQUATION DETAILS.
C
C-----
      IF( NVALS .GT. 0 ) THEN

        CALL RCOEFF( JJ, AMJ, BMJ )

        SSMJ = 0.
        J = FLOAT(JJ) - M
        DMJ = ((M + J) * ((M + J)**2 - 1.)) / 12.
C-----
        DO 30 I = 1, JJ

          SSMJ = SSMJ + (YDATA(I) - YBAR)**2
          PREDMJ(I) = (XDATA(I) * BMJ + AMJ)
          RESIDMJ(I) = (YDATA(I) - PREDMJ(I))

          30      CONTINUE
C-----
          SUMT2 = 0.
          T1 = (-1.) * (M - 2.) / (2. * (SSM - (BMJ**2 * DM)))
C-----
          DO 32 I = 1, MBASE

            T = FLOAT(I)

            T2 = RESIDMJ(I)**2 / (((M + 1) / M) +
+              (T - ((M + 1.) / 2.))**2 / DM)

            SUMT2 = SUMT2 + T2

          32      CONTINUE
C-----
          LNL4 = T1 * SUMT2

          SUMT21 = 0.
          SUMT22 = 0.

```

```

+      T11 = (-1.) * (M + J - 2.)
+      / (2. * (SSMJ - (BMJ**2 * DMJ)))
+      T12 = (-1.) * (M + J - 2.)
+      / (2. * (SSMJ - (BM**2 * DMJ)))
C-----
      DO 34 I = 1, JJ

          T = FLOAT(I)

          DENOM = ((M + J + 1.) / (M + J))
+              + ((T - ((M + J + 1.) / 2.))**2 / DMJ)

          T21 = RESIDMJ(I)**2 / DENOM
          T22 = RESIDM(I)**2 / DENOM

          SUMT21 = SUMT21 + T21
          SUMT22 = SUMT22 + T22

34      CONTINUE
C-----
      LNL2 = T11 * SUMT21
      LNL3 = T12 * SUMT22

      QM = EXP(LNL4 - LNL1)
      QMJ = EXP(LNL2 - LNL3)
      Q = QMJ / QM
      GAMMA = 1. / (1. + SQRT(Q))
      DGAMMA = GAMOLD - GAMMA
C-----
      IF( GAMOLD .EQ. FMISS ) THEN

          WRITE(OUNIT,4003) JJ, FXVAL(JJ), YDATA(JJ), YBAR, YVAR,
+              BMJ, GAMMA
+          WRITE(*,4003) JJ, FXVAL(JJ), YDATA(JJ), YBAR, YVAR,
+              BMJ, GAMMA

          ELSE

          WRITE(OUNIT,4003) JJ, FXVAL(JJ), YDATA(JJ), YBAR, YVAR,
+              BMJ, GAMMA, DGAMMA
+          WRITE(*,4003) JJ, FXVAL(JJ), YDATA(JJ), YBAR, YVAR,
+              BMJ, GAMMA, DGAMMA

          END IF
C-----
      GAMOLD = GAMMA

      ELSE

          WRITE(OUNIT,4004) JJ, FXVAL(JJ), YDATA(JJ)
          WRITE(*,4004) JJ, FXVAL(JJ), YDATA(JJ)

          END IF
C-----
26      CONTINUE
C-----

```

```

WRITE(*,802)
READ(*,*) ITYPE
IF( ITYPE .EQ. 1 ) GOTO 1

801  FORMAT(/,' ENTER NUMBER OF VALUES IN BASE PERIOD:')
802  FORMAT(/,' ENTER 0=CONTINUE, 1=CHANGE NUMBER OF BASE VALUES:')

4000  FORMAT(' ALL VALUES IN BASE PERIOD MISSING')

4001  FORMAT(/,'BASE PERIOD PARAMETERS:',/,
+      'X-VALUE RANGE = ',F5.0,'-',F5.0,/,
+      'NUMBER OF VALUES = ',I3,/,
+      'NUMBER OF MISSING VALUES = ',I3,/,
+      'BASE MEAN = ',F8.3,/,
+      'BASE VARIANCE = ',F8.3,/,
+      'BASE SLOPE = ',F6.3,/)

4002  FORMAT(/,'PROGRESSIVE PARAMETERS:',/,
+      '          X          Y          ',
+      ' DELTA',/,
+      ' INDEX  OBSERVATIONS      MEAN    VARIANCE    SLOPE    GAMMA',
+      ' GAMMA',/,
+      '=====',
+      ' =====')

4003  FORMAT(I5,2X,F8.3,1X,F7.2,2(2X,F8.3),2(2X,F6.3),2X,F6.3)

4004  FORMAT(I5,2X,F8.3,1X,F7.2)

RETURN
END

```

C-----  
SUBROUTINE RCOEFF( N, B0, B1 )

```

C
C SUBROUTINE TO COMPUTE LINEAR LEAST SQUARES (LLS) INTERCEPT AND SLOPE.
C
C VARIABLES USED IN SUBROUTINE NOT DESCRIBED IN MAIN PROGRAM:
C 'I'      DO LOOP COUNTER.
C 'N'      NUMBER OF VALUES.
C 'NN'     COUNTER FOR NON-MISSING VALUES.
C 'SX2'    SUMS OF SQUARED DEVIATIONS FROM THE MEAN X-VALUE.
C 'SXY'    SUM OF THE CROSS PRODUCTS OF DEVIATIONS.
C 'SUMX'   SUM OF NON-MISSING X-VALUES.
C 'SUMY'   SUM OF NON-MISSING Y-VALUES.
C 'SUMSX2' SUM OF X-VALUES SQUARED.
C 'SUMSXY' SUM OF X TIMES Y SQUARED.
C 'XBAR'   MEAN OF NON-MISSING X-VALUES.
C 'XBAR2'  NUMBER OF NON-MISSING VALUES TIMES 'XBAR' SQUARED.
C 'XWORK'  WORK ARRAY FOR X-VALUES.
C 'XYN'    'XBAR' TIMES 'YBAR' TIMES NUMBER OF NON-MISSING VALUES.
C 'YBAR'   MEAN OF NON-MISSING Y-VALUES.
C 'YWORK'  WORK ARRAY FOR Y-VALUES.
C

```

```

C      PARAMETER (NDIM=150)
C      COMMON /WORK/ FXVAL(NDIM), FYVAL(NDIM), XDATA(NDIM),
+      YDATA(NDIM), FINDEX(NDIM), FCHI2(NDIM), NCASE,

```

```
+          OUNIT, FMISS
REAL XWORK (NDIM), YWORK (NDIM)
INTEGER OUNIT

SUMX = 0.
SUMY = 0.
NN = 0

C
C  FILL WORK ARRAYS WITH NON-MISSING VALUES.
C
C-----
      DO 15 I = 1, N
C-----
          IF ( YDATA(I) .GT. FMISS ) THEN

              NN = NN + 1
              XWORK(NN) = XDATA(I)
              YWORK(NN) = YDATA(I)

          END IF

C-----
15      CONTINUE
C
C  COMPUTE SUMS OF X- AND Y-VALUES.
C
C-----
      DO 20 I = 1, NN

          SUMX = SUMX + XWORK(I)
          SUMY = SUMY + YWORK(I)

20      CONTINUE
C
C  COMPUTE MEANS OF X- AND Y-VALUES.
C
      XBAR = SUMX / FLOAT(NN)
      YBAR = SUMY / FLOAT(NN)

      SUMSX2 = 0.
      SUMSXY = 0.

C
C  COMPUTE SUMS OF SQUARED DEVIATIONS AND CROSS PRODUCTS.
C
C-----
      DO 22 I = 1, NN

          SUMSX2 = SUMSX2 + XWORK(I)**2
          SUMSXY = SUMSXY + (XWORK(I) * YWORK(I))

22      CONTINUE
C-----
      XBAR2 = XBAR**2 * FLOAT(NN)
      XYN = XBAR * YBAR * FLOAT(NN)
```



```

      SX2 = SUMSX2 - XBAR2
      SXY = SUMSXY - XYN
C
C COMPUTE COEFFICIENTS.
C
      B1 = SXY / SX2
      B0 = YBAR - B1 * XBAR

      RETURN
      END
C-----
      SUBROUTINE UNIVAR( N, NVALS, NMISS, YBAR, YVAR, YSUM )
C
C SUBROUTINE TO COMPUTE LINEAR LEAST SQUARES (LLS) INTERCEPT AND SLOPE.
C
C VARIABLES USED IN SUBROUTINE NOT DESCRIBED IN MAIN PROGRAM:
C 'I'      DO LOOP COUNTER.
C 'N'      NUMBER OF VALUES.
C 'SUMS2'  SUM OF SQUARED DIFFERENCES.
C 'YWORK'  WORK ARRAY FOR NON-MISSING Y-VALUES.
C
      PARAMETER (NDIM=150)
      COMMON /WORK/ FXVAL(NDIM), FYVAL(NDIM), XDATA(NDIM),
+                YDATA(NDIM), FINDEX(NDIM), FCHI2(NDIM), NCASE,
+                OUNIT, FMISS
      REAL YWORK(NDIM)
      INTEGER OUNIT

      YSUM = 0.
      NVALS = 0
      NMISS = 0
C
C FILL WORK ARRAY WITH NON-MISSING VALUES.
C
C-----
      DO 15 I = 1, N
C-----
          IF ( YDATA(I) .GT. FMISS ) THEN

              NVALS = NVALS + 1
              YWORK(NVALS) = YDATA(I)

          ELSE

              NMISS = NMISS + 1

          END IF
C-----
      15 CONTINUE
C-----
C COMPUTE SUM OF Y-VALUES.
C
C-----
      DO 20 I = 1, NVALS

```

YSUM = YSUM + YWORK(I)

20 CONTINUE

C-----

C

C COMPUTE MEAN OF Y-VALUES.

C

YBAR = YSUM / FLOAT(NVALS)

C

C COMPUTE SUM OF SQUARED DIFFERENCES.

C

SUMS2 = 0.

C-----

DO 22 I = 1, NVALS

SUMS2 = SUMS2 + (YWORK(I) - YBAR)\*\*2

22 CONTINUE

C-----

C

C COMPUTE VARIANCE OF Y-VALUES.

C

YVAR = SUMS2 / (FLOAT(NVALS) - 1)

RETURN

END

C-----