STOCHASTIC PROJECTIONS FOR WATER RESOURCES PLANNING

A Report by

Leo R. Beard, Technical Director

and

Shin Chang, Research Associate

Center for Research in Water Resources

The University of Texas at Austin

for

The Office of Water Research and Technology

November 30, 1975

The work upon which this publication is based was supported in part by funds provided by the United States Department of the Interior as authorized under the Water Resources Research Act of 1964, as amended and the Water Resources Council as authorized by the Water Resources Planning Act of 1962, as amended.

Foreword

This is the final technical report issued under the Office of Water Research and Technology Grant No. 14-31-0001-9088, "Techniques for Projecting Alternative Futures for Water Resources Planning and for Estimating Flood Flow Frequencies". A previous report, "Flood Flow Frequency Techniques", was published on October 1, 1974.

CONTENTS

TABLE

FIGURES

APPENDIX A

APPENDIX B

STOCHASTIC PROJECTIONS FOR WATER RESOURCES PLANNING

1. PROJECT OBJECTIVES

Evaluation of the future benefits and costs of water resources projections have traditionally been based on unique projections of key variables such as population and interest rates. Recognizing that the actual changes or trends that will occur in the future will be somewhat different from the theoretical projections, there is a fundamental need to evaluate the differences in benefits and costs that will occur if different trends occur. Also, the more comprehensive nature of water resources planning that is required today involves projections of many variables and, in projecting these variables, the interrelationships must be taken into consideration.

The objective of the research project reported on herein is to develop a technique for evaluating a range of projections of several types of variables that are involved in water resources planning and to develop techniques for using these projections effectively and efficiently. Variables used for illustration are economic, demographic and hydrologic. Only successive annual values are considered, since the project study is not concerned with seasonal variations. Only mathematical projection techniques are considered, since it is the technique of projecting sequences that is of concern to this project, and not developing cause-effect relationships wherein the projection of one variable can be used in deducing the projection of a second variable. Further, it is not an objective of this study to determine or develop best mathematical functions for projecting specific types of variables, but to suggest procedures, techniques and criteria for assessing the entire range of future potential as contrasted with the single most likely future.

2. BASIC CONSIDERATIONS

Of primary importance in most water resources planning studies, in regard to projections of pertinent variables, is the projection level that will be reached at some specified time in the future. Of secondary importance are variations that occur along the line of projection. In most past studies, single projections of each pertinent variable have been made, and this ordinarily represents a smooth function that arrives at the most likely projection levels at various future times. Considering the uncertainty that available data on past quantities adequately represent future potential, and considering the virtual certainty that the actual future variations will not coincide with the most likely projections, water resources studies should consider alternative projection studies that can occur consistently with past variations of the particular variable being projected and considering our knowledge of how frequently and severely unexpected variations from indicated trends have occurred in the past.

As in the formulation of any mathematical model, the degree of complexity of a projection model must be adequate to represent variations of the phenomenon being studied with a degree of accuracy necessary for the particular applications anticipated. On the other hand, the model must be simple enough so that it can be calibrated satisfactorily with available data. Often it is not possible to meet both of these requirements, in which case the latter requirement would normally control, since there is little point in having a model that cannot be adequately calibrated. When this occurs, extreme caution must be used in interpreting the predictions that are made with the model.

Projection models described below have been restricted to a modest degree of complexity because of the limitations of data for calibration. Nevertheless, they include two types of variation that have not been used in traditional projection studies. These consist of a range of projections that adequately represent future possibilities and their associated probabilities, and variations within each projection that represent differences in trends that can be expected to occur from time to time. A small number of such projections covering the entire range of future expectations can be used separately to evaluate the accomplishments of proposed water resource projects, and expected project benefits can be computed by adding the cross products of the benefits obtained for each projection and the associated probability attached to that projection.

3. SINGLE-VARIABLE PROJECTION TECHNIQUES

Time series definitions. Some terms of special relevance to this report are defined as follows:

1. Time Series. A time series is a set of observations generated sequentially in time.
2. Stochastic Process. A stochastic process is a statistical phenomenon that evolves in time (or space) according to probabilistic laws. The time series to be analyzed may then be thought of as one particular realization, produced by the underlying probability mechanism, of the system under study.
3. Stationary Time Series. A time series is said to be stationary if the imbedded stochastic process of the time series remains in equilibrium about a constant level. Statistically, an nth-order stationary series requires that the first n moments of its distribution function be constant. For example, a second-order stationary series must have constant mean and constant variance.
4. Backward Shift Operator B. The backward shift operator B on a variable Z is defined by the equation: $B^m Z_t = Z_{t-m}$.
5. Backward Difference Operator ∇. The backward difference operator ∇ on a variable Z is defined in terms of B as: $\nabla = (1-B)$; hence $\nabla Z_t = Z_t - Z_{t-1}$.
6. Differences of a Time Series. Differences of a time series are obtained by successively subtracting the previous value from the current one, and this can be repeated to obtain differences of higher order. The degree of differencing is designated d. Using the backward difference operator ∇, the dth difference of a time series Z can be represented as $\nabla^d Z$. For example, $\nabla^2 Z = \nabla(\nabla Z) = Z_t - 2Z_{t-1} + Z_{t-2}$.

Stationary Stochastic Models. Three categories of stationary stochastic models used in the description of stationary time series are autoregressive, moving average, and mixed autoregressive and moving average. These are defined as follows:

1. Autoregressive Models (AR). In these models, the current value

of the series is expressed as a finite, linear aggregate of the previous values plus a random component, a_t. Denoting the values of a time series at equally spaced times t, t+1, ..., by Z_t, Z_{t+1}, ..., and denoting the deviation from μ, the mean value of the time series, by $\bar{Z}_t$, ...$\bar{Z}_{t+1}$. Then the p-th AR model has the form

$$\mu = (\sum_{t=1}^{n} Z_t) / n,$$

$$\bar{Z}_t = \phi_1 \bar{Z}_{t-1} + \phi_2 \bar{Z}_{t-2} + \ldots + \phi_p \bar{Z}_{t-p} + a_t \tag{3.1}$$

where ϕ_1, ϕ_2,....ϕ_p, are p's autoregressive parameters. The model contains p+2 unknown parameters, μ, ϕ_1,...., ϕ_p and σ_a^2, to be identified from the data. The additional parameter σ_a^2 is the variance of the white noise process or random component, a_t. The stochastic AR model can be useful in the representation of certain practical series. For instance, the first order and second order Markov models frequently used in streamflow synthesis are AR models.

2. Moving Average Models (MA). In these models, the current value of the series, $\bar{Z}_t$, is expressed as a finite weighted sum of q previous random terms a_t. Thus the q-th MA model has the form

$$\bar{Z}_t = a_t - \Theta_1 a_{t-1} - \Theta_2 a_{t-2} - \ldots - \theta_q a_{t-q} \tag{3.2}$$

where Θ_1, Θ_2,...., Θ_q are q's moving average parameters. This model contains q + 2 unknown parameters μ, Θ_1, Θ_2,...., Θ_q and σ_a^2 to be estimated from the data. Although this is called a moving average model, it is conceptually an infinite-term autoregressive model of specific form. The MA model is of practical importance in the representation of observed time series.

3. Mixed Autoregressive and Moving Average Models (ARMA). To achieve greater flexibility in fitting of actual time series, it is sometimes advantageous to include both autoregressive and moving average terms in one model. This leads to the mixed ARMA model. An ARMA(p,q) with p autoregressive terms and q moving average terms has the form,

$$\bar{Z}_t = \phi_1 \bar{Z}_{t-1} + \ldots + \phi_p \bar{Z}_{t-p} + a_t - \phi_1 a_{t-1}$$

$$- \ldots - q \, a_{t-q} \quad (3.3)$$

The ARMA(p,q) model requires p+q+2 unknown parameters to be estimated from the data. It should be noted that the ARMA(p,q) model becomes an AR(p) model if q is set to zero and an MA(q) model if p is set to zero.

Nonstationary Stochastic Models. There are an unlimited number of ways in which a time series can be nonstationary. In fact, many time series in industry, business, and economics are better represented as nonstationary. Frequently, however, the nonstationary time series being analyzed exhibit a particular kind of homogenous nonstationary behavior. (i.e. the series exhibits homogeneity in the sense that, apart from the local level, or local level and trend, one segment of the series behaves much like any other segment). Differences of a nonstationary series of this kind can usually be expected to be stationary. Given the condition that the d-th difference of a nonstationary series is stationary, the autoregressive-integrated moving average model (which is a modified form of the ARMA model), can be used to describe the nonstationary series.

Autoregressive-Integrated Moving Average Models (ARIMA). The stochastic, nonstationary ARIMA model (with p autoregressive terms and q moving average terms) which describes a nonstationary series that is stationary in its dth difference has the form,

$$(1-\phi_1 B- \ldots -\phi_p B^p) \quad W_t = \Theta_o + (1-\theta_1 B- \ldots -\theta_q B^q) \, a_t \quad (3.4)$$

with

$$W_t = \nabla^d Z_t, \text{ if } d>0$$

$$= Z_t - \mu, \text{ if } d=0$$

where Z is the series to be modeled,

W is the dth difference of the series Z,

Θ_o is the long term trend factor.

It should be noted that the ARIMA(p,d,q) model becomes an AR(p) model if q and d are zero and an MA(q) model if p and d are zero and an ARMA model if d is zero.

Specific Models Studied. Seven models were selected for consideration under this project. They are

1. Second-Order Linear Markov Model.

$$Z_t = aC_1 Z_{t-1} + (1-a)\ C_2 Z_{t-2} \tag{3.5}$$

where a is the weighting factor, C_1 and C_2 are random variables; $C_1 = Z_t/Z_{t-1} + b_1 t$ and $C_2 = Z_t/Z_{t-2} + b_2 t$ and b_1 and b_2 are stochastic linear regression coefficients, and t is time. The values of b_1 and b_2 for forecasting are estimated stochastically as a function of Student's t distribution from the known historical data.

2. Second-Order Nonlinear Markov Model.

$$Z_t = Z_{t-1}^{aC_1} \cdot Z_{t-2}^{(1-a)C_2} \tag{3.6}$$

This model is the non-linear version of model 1. By taking logarithms of the variable, we have

$$\ln Z_t = aC_1\ \ln Z_{t-1} + (1-a)\ C_2\ \ln Z_{t-2}, \tag{3.7}$$

this equation has the same form as that of model 1 except that the logarithms of the time series Z are used instead of the time series itself. As in model 1, C_1 and C_2 are estimated stochastically.

3. Nth-Order Linear Markov Model.

$$Z_t = aC_1Z_{t-1} + bC_2Z_{t-2} + cC_3Z_{t-3} + dC_4Z_{t-4} + eC_5Z_{t-5} \text{ for } N\leq 5. \qquad (3.8)$$

The factors a, b, c, d, and e provide for weighting of the independent variables. The linear coefficients C_1, C_2, C_3, C_4, and C_5 are random variables and are distributed according to certain probability functions. For the purposes of this study, these coefficients were assumed to be normally distributed and defined by the equation

$$C_i = Z_t/Z_{t-i}, \qquad i = 1, 2, 3, 4, 5 \qquad (3.9)$$

When the model is being used for forecasting, the values of the linear coefficients are estimated stochastically from the historical data.

4. Nth-Order Nonlinear Markov Model.

$$Z_t = Z_{t-1}^{aC_1} \cdot Z_{t-2}^{bC_2} \cdot Z_{t-3}^{cC_3} \cdot Z_{t-4}^{dC_4} \cdot Z_{t-5}^{eC_5} \qquad (3.10)$$

This model is the nonlinear version of model 3. By taking logarithms of the variable, the resulting equation has the same form as that of model 3.

5. Decade and Annual Ratio Autocorrelation Model. This model takes advantage of the availability of 10-year census data and the relatively accurate estimates of annual data that have been obtained using census techniques and omitting annual data obtained by mathematical interpolation. This permits the most effective use of early census information when only decade values are available. The model consists of a lag-1 Markov chain of the logarithms of the ratios of successive decade values and an interpolation routine using a similar Markov chain for annual logarithms where annual observations are available. When generating successive sequences, random values of average decade ratios are used for each sequence rather than sample values.

6. Decade Regression and Annual Autocorrelation Model. This model is similar to the preceding model except that a linear regression on the logarithms of the decade ratios is used instead of autocorrelation, and a new random value of the regression coefficient is used for each generated sequence.

7. The ARIMA(p,d,q). The ARIMA model includes three classes of models previously described: the AR, the MA, and the ARMA. It can be applied to both stationary and nonstationary time series. The degree of differencing, d, and the number of parameters, p and q, are identified first, then values of parameters are estimated by the least mean square errors method.

The first six models are essentially special cases of the AR process and they all experienced forecasting problems in some applications, particularly in the nature of forecasting unreasonable quantities. Thus, the ARIMA model was chosen as the standard forecasting model because of its flexibility and adaptability to various types of time series, thus providing means of avoiding unreasonable forecasts.

4. THE ARIMA MODEL

ARIMA(p,d, q) Model Building. To fit a series by an ARIMA(p,d,q) model, the model building process is ordinarily based on available observations. Computationally, the model building includes three steps: identification, estimation, and diagnostic checking. The order of differencing, d, the number of autoregressive parameters, p, and the number of moving average parameters, q, are identified in the identification step; values of parameters are then estimated in the estimation step; and the adequacy of the model in describing and projecting the series is checked in the diagnostic step.

According to Box and Jenkins (1970), no pure mathematical approach can be adopted in the model building process. Graphical methods and judgment are needed throughout because of the trial-and-error nature of the process. Following their basic ideas, a systematic noniterative algorithm is developed for model building, and although the algorithm is conceptually an exhaustive trial-and-error approach, operationally it is automatic. All possible models from the general ARIMA family of limited order are examined, and the best model is selected such that four proposed performance criteria are either satisfied or maximized. These performance criteria are established to measure the goodness of fit and reasonableness of projections.

1. <u>Identification of the Order of Differencing d</u>. The failure of the autocorrelation function to die out quickly is taken as an indication that the underlying stochastic process of the series Z is nonstationary, but the process of its first or higher order differences may be stationary. Therefore, it is assumed that the degree of differencing, d, necessary to achieve stationarity has been reached when the autocorrelation function of $W_t = \nabla^d Z_t$ dies out quickly. In practice, d is generally 0, 1, or 2, and it is usually sufficient to inspect the first 20 (or so) estimated autocorrelations of the original series and of its first and second differences. According to Box and Jenkins (1970), to judge whether or not the autocorrelation function dies out quickly is completely subjective, as no standard judging criterion is available. The following method was found to be an effective way of detecting the speed of decay of the autocorrelation function and identifying the order of differencing, d.

Given a series of n elements and denoting the lag i autocorrelation coefficient by r_i, the Chi-square statistic, χ_d^2, of the first 24 autocorrelation coefficients can be calculated as

$$\chi_d^2 = n \cdot \sum_{i=1}^{24} r_i^2 , \qquad (4.1)$$

Where d can be 0, 1, or 2 to stand for, respectively, the Chi-square statistic of the original series, of its first difference, or of its second difference. Let R_1 and R_2 be the ratios of

$$R_1 = \chi_1^2 / \chi_0^2, \text{ and } R_2 = \chi_2^2 / \chi_1^2 \qquad (4.2)$$

In general, R_1 and R_2 are less then one. These ratios give the relative decay speed of the autocorrelation function of the series and its differences. For example, if $R_1 = 0.2$, then the Chi-square statistic of the first difference is only 0.2 times that of the original series. This indicates that the speed of decay of the autocorrelation function of the first difference is five times faster than that of the original series. Based on this argument, the following rule can be used to select the degree of differencing d for the model:

$$\text{Select } d = 0, \text{ if } R_1 > \varepsilon \text{ and } R_2 > \varepsilon;$$
$$d = 1, \text{ if } R_1 < \varepsilon$$
$$d = 2, \text{ if } R_1 > \varepsilon \text{ and } R_2 < \varepsilon.$$

The criterion, ε, is set from experience and judgment. A value of 0.4 was used in identifying d for all sample series in this study.

Parameter Estimation. Once it has been determined what the degree of differencing should be, values of parameters in all the member models of the general ARIMA family are next estimated. In the general order of decreasing complexity, the fourteen most used members of the ARIMA(p,d,q) family (listed by their (p,d,q) values) are: (2,d,3), (3,d,2), (1,d,3), (2,d,2), (3,d,1), (0,d,3), (1,d,2), (2,d,1), (3,d,0), (0,d,2), (1,d,1), (2,d,0), (0,d,1), (1,d,0). Box and Jenkins suggest that several of the models most likely to best describe the time series can be identified by comparing the shapes of their autocorrelation and partial autocorrelation functions with those of certain established theoretical patterns. The best model is then selected by a detailed study of the residuals. For the following reasons, however, it is believed that the following approach is more attractive and easier to apply than that of Box and Jenkins:

1. From the experience of modeling 30 sample series, it was found that in most cases, the autocorrelation and partial autocorrelation functions of the appropriately differenced series fail to show any pattern to yield a meaningful conclusion in selecting p and q in accordance with the findings of Box and Jenkins.

2. In some cases, models with different values of p and q can be fitted equally well to a given series and yield equally high levels of significance in the diagnostic checking step. This in turn suggests that

correspondence between the selection of p and q of an ARIMA(p,d,q) model and the shape of autocorrelation and partial autocorrelation functions is not unique.

3. There is no definite trend in either direction between the complexity (number of parameters used in a model) of a model and the performance (to be discussed in the model screening step) of the model. In other words, no selective search technique, rather than an exhaustive search of all possible models, can be developed to find the best model.

4. No plot of the autocorrelation and the partial autocorrelation function is required for the selected approach.

5. No intermediate subjective feedback is needed in the course of selecting the best model by the current approach; hence an automatic computer search procedure is possible.

The estimation of autoregressive and moving average parameters for an ARIMA(p,d,q) model is based on the Marquardt algorithm for nonlinear least-square parameter estimation. Computer programs based on this algorithm are available from the IMSL (International Mathematical and Statistical Libraries) program library 3 (for CDC Cyber 70/6000/7000), in which FTARPS and FTMAPS (both in FORTRAN) were used in this study to estimate the autoregressive and moving average parameters, respectively. After estimating the parameters of an ARIMA(p,d,q) model, the residuals, a_t, are computed by the equation

$$a_t = \hat{W}_t - W_t, \tag{4.3}$$

where $\hat{W}_t$ is the fitted value of the time series and W_t the observed value.

<u>Model Screening</u>. Four criteria for selecting the best model to describe a time series are used:

1. <u>Goodness of Fit</u>. The goodness of fit of an ARIMA model to a time series is measured by the level of significance of the Chi-square test for the residual, a_t, in relation to white noise (randomness). The Chi-square statistic of the residual a_t for $t = 1,2, \ldots, n-d$, is calculated using the formula,

$$\chi^2 = (n-d) \sum_{i=1}^{m} r_i^2 \qquad (4.4)$$

where r_i is the ith autocorrelation coefficient of series a_t, and m is the number of autocorrelation coefficients used in the computation of the statistic. With m-p-q degrees of freedom, the significance level associated with χ^2 can be found from a Chi-square test table. The computer program MDCH of IMSL was used to find this level. The resulting residuals, a_t, should not significantly differ from a white noise process, if the time series of interest is fitted by an adequate ARIMA model. Thus, a minimum level of significance can be set to reject inadequate models. A minimum level of 0.01 was used in this study.

2. Model Stability. There exists an admissible region of parameters for every ARIMA(p,d,q) model which will ensure the stationarity and invertibility of the model (see chapter 3 of Box and Jenkins, 1970). Models with parameters outside this region will be dropped from further consideration. The region can be expressed implicitly as:

a. For Autoregressive Parameters. The roots of the polynomial $(1-\phi_1 B-\phi_2 B^2 - \ldots - \phi_p B^p)$ which is formed by taking all autoregressive of the ARIMA(p,d,q) model (i.e. left hand side of equation 3.4) when solved for B, must be greater than one.

b. For Moving Average Parameters. The roots of the polynomial $(1-\Theta_1 B-\Theta_2 B^2 - \ldots - \Theta_q B^q)$, which is formed by taking all moving average terms of the ARIMA(p,d,q) model (i.e. right hand side of equation 3.4), when solved for B, must be greater than one.

3. Reasonableness of Forecasting. A model that can be fitted to a series adequately in the sense of passing the minimum level of significance in the goodness of fit test might nevertheless give unreasonable future projections, such as interest rates of 1000 percent. This is especially true when split-record tests are performed on a U-shaped or a W-shaped series. Therefore, a well-fitted model with a high level of significance is not necessarily a good projection model for the purpose of planning. A projection-reasonableness test is used as a supplementary criterion to the level of significance in selecting a model. A physical range and/or lower and upper bounds for the future are established based on available information and basic characteristics of the series. In this study, the bounds

are established at a distance of 50 times the range of values within the record from each extreme of record in terms of the transformed variable (logarithm, usually). A model is said to have projection reasonableness with respect to a series if projections of the series made by this model can satisfy all the designated range limitations. In this test, five independent projection sets, each with five times the record length, are generated randomly by the model. A model which cannot pass this test will be discarded.

4. Simplicity of Model. According to the principle of parsimony in using parameters, one should not try to overfit a time series with a complicated model when a simpler model can fit the series adequately. Thus, a trade-off between the performance (measured by the level of significance in the goodness of fit test) and the complexity (measured by the total number of parameters used) of models is desirable in selecting the best model for practical uses. The trade-off can be achieved by employing a discount factor, i, to the level of significance, α , such that α is reduced exponentially according to the complexity of the model. Mathematically, let (p+q) be the degree of complexity of an ARIMA (p,d,q), then the discounted level of significance, α', is computed as

$$\alpha' = \alpha(1-i)^{p+q-1} \tag{4.9}$$

In selecting a value of i, personal judgment must be exercised. A value of 0.3 was used in this study, and this represents a high degree of discrimination against the more complex models.

Following the above four model screening steps, the best model can be selected simply by choosing the model which has the highest discounted significance level.

Applications of the ARIMA(p,d,q) Model. The ARIMA(p,d,a) model can be used to fit, to forecast, and to simulate a given time series.

1. Fitting a time series. Let $\hat{W}_t$ be the fitted value of the observed time series W_t and a_t be the residual of fitting $\hat{W}_t$ to W_t at time t. The first fitted value of $\hat{W}_t$ to be computed using the fitted ARIMA model is $\hat{W}_{p+1}$, because the first p values of W_t are not used in the estimation

of autoregressive parameter ϕ_p. Thus for $t \leq p$, we have

$$\hat{W}_t = W_t \tag{4.5}$$

$$a_t = 0 \tag{4.6}$$

and for $t > p$, we have

$$\hat{W}_t = \phi_1 \hat{W}_{t-1} + \ldots + \phi_p \hat{W}_{t-p} + \Theta_o - \Theta_1 a_{t-1} - \ldots - \Theta_q a_{t-q} \tag{4.7}$$

$$a_t = \hat{W}_t - W_t, \tag{4.8}$$

and in equation (4.7), the random noise term, a_j with negative lag j is set to zero. It should be pointed out that W is the dth difference of the original observed series Z. Let $\hat{Z}_t$ be the fitted value of the observed Z_t at time t, then the functional relationship of Z_t can be derived by substituting $W_t = \nabla^d Z_t$ into equation (4.7). Upon substituting and collecting similar terms, we have

$$\tilde{Z}_t = \sum_{i=1}^{p+d} \psi_i \hat{Z}_{t-i} + \Theta_o - \sum_{i=1}^{q} \Theta_i a_{t-i} \tag{4.9}$$

where ψ_i's are coefficients of $\hat{Z}$. For example, if W is the first difference of Z_t (i.e. $W_t = \nabla Z_t = Z_t - Z_{t-1}$), then, by substituting $\hat{W}_t = \hat{Z}_t - \hat{Z}_{t-1}$ into equation (4.7), we have

$$\hat{Z}_t - \hat{Z}_{t-1} = \phi_1(\hat{Z}_{t-1} - \hat{Z}_{t-2}) + \phi_2(\hat{Z}_{t-2} - \hat{Z}_{t-3}) + \ldots + \phi_p(\hat{Z}_{t-p} - \hat{Z}_{t-p-1}) + \Theta_o - \Theta_1 a_{t-1} - \ldots - \Theta_q a_{t-q}, \tag{4.10}$$

collecting similar terms,

$$\hat{Z}_t = (1-\phi_1)\hat{Z}_{t-1} + (\phi_2 - \phi_1)\hat{Z}_{t-2} + \ldots + (-\phi_p)\hat{Z}_{t-p-1} + \Theta_o - \Theta_1 a_{t-1} - \ldots - \Theta_q a_{t-q} \tag{4.11}$$

or, in terms of ψ_i,

$$\hat{Z}_t = \sum_{i=1}^{p+1} \psi_i \hat{Z}_{t-i} + \Theta_o - \sum_{i=1}^{q} \Theta_q a_{t-q} \tag{4.12}$$

2. Forecasting and Simulation. Let $\hat{W}_{t+s}$ be the forecast of the series W_t at lead time s from the origin t. It can be shown (Box and Jenkins, 1970, Chapter 5) that: (a) the forecast for W_{t+s} using minimum mean square error estimation is the conditional expectation of W at time t+s; (b) the expected forecast error of $\hat{W}_{t+s}$ is equal to zero, hence the forecast is unbiased; and (c) the one-step-ahead forecast error is equal to a_t (the residual at time t), hence the Chi-square significance test of a_t can be used as an effective indicator of the goodness of fit of the ARIMA model for the series being modeled. The forecasting equation for $\hat{W}_{t+s}$ is

$$\hat{W}_{t+s} = \phi_1\hat{W}_{t+s-1}+\ldots+\phi_p\hat{W}_{t+s-p}+\Theta_o-\Theta_1 a_{t+s-1}-\ldots-\Theta_q a_{t+s-q} \tag{4.13}$$

where $\hat{W}_t = W_t$, if $t \leq n-d$,
$= \hat{W}_t$, if $t > n-d$,
$a_t = a_t$, if $t \leq n-d$,
$= E(a_t) = 0$, if $t>n-d$ and $E(a_t)$ is the expected value of the white noise a_t.

And similar to the derivation of the fitted value $\hat{Z}_t$ from $\hat{W}_t$, the forecasting function for $\hat{Z}_{t+s}$ can be shown to be

$$\hat{Z}_{t+s} = \psi_1\hat{Z}_{t+s-1}+\ldots+\psi_{p+d}\hat{Z}_{t+s-p-d}+\Theta_o-\Theta_1 a_{t+s-1} -\ldots-\Theta_q a_{t+s-q} \tag{4.14}$$

where $\hat{Z}_t = Z_t$, if $t \leq n$
$= \hat{Z}_t$, if $t > n$, and
$a_t = a_t$, if $t \leq n$
$= 0$, if $t > n$.

The forecast $\hat{Z}_{t+s}$ represents the most likely projection level for lead time s in the future. For reasons mentioned before, alternative projections are needed and can be generated by simulation using the fitted ARIMA model with random components.

Each independent set of simulated projections is generated using a stochastic trend factor β and a white noise term b_t. The stochastic trend factor β is obtained by assuming that the trend factor, Θ_o, used in forecasting, has a Student-t distribution. The white noise term b_t is generated by a random variable generator using the known variance of the white noise a_t. That is

$$b_t = a_t \text{ if } t \leq n$$

$$= q\sqrt{\sigma_a^2}, \text{ if } t>n \qquad (4.15)$$

q is a normally distributed variable with zero mean and unit variance. Then, the simulation function of $\hat{Z}_{t+s}$ is

$$\hat{Z}_{t+s} = \psi_1 Z_{t+s-1} +\ldots.+ \psi_{p+d} Z_{t+s-p-d} +\beta -\Theta_1 b_{t+s-1}$$

$$-\ldots- \Theta_q b_{t+s-q}, \qquad (4.16)$$

where $\hat{Z}_t = Z_t$ if $t \leq n$

$= \hat{Z}_t$ if $t > n$.

5. MULTIVARIATE PROJECTION MODEL

General Approach. For variables that are interrelated, an acceptable model should project every variable simultaneously so as to preserve the interrelationship among projections of all variables. There is no specific multivariate projection technique, but principal-component analysis can be used where data for all variables exist for the same period, in order to form uncorrelated variables (principal components), each of which is expressed as a linear function of the original variables. Then the new variables can be projected independently and transformed to form projections of the original variables. The principal-component technique was selected for convenience in this study as an illustrative procedure. Other multivariate techniques, such as multiple linear regression, can be used if desired.

Principal-Component Analysis. Principal-Component analysis is a striaght-forward method of transforming a given set of variables into a new set of composite variables (or principal components) that are mutually orthogonal (uncorrelated). The first principal component represents the single best summary of linear relationships exhibited in the data. The second component is defined as the second best linear combination of variables that accounts for the most residual variance after the effect of the first component is removed from the data. Subsequent components are defined similarly until all the variance in the data is exhausted. Unless at least one variable is perfectly determined by the remaining variables in the data, the principal-component solution requires as many components as there are variables. Let Z_j, j=1, 2, ..., m be the m original variables; F_k, k=1, 2, ..., m be the m principal components; A_{jk}, j=1, 2, ..., m and k=1, 2, ..., m be the component-pattern coefficients, and b_{jk}, j=1,2,...,m and k=1,2,...,m be the component-estimation coefficients. Then the principal component F_k can be computed from the values of the m original variables Z as

$$F_k = b_{k1} Z_1 + b_{k2} Z_2 +....+ B_{km} Z_m, \; k = 1, 2, ..., m \tag{5.1}$$

and the original variable Z_j can be described by the m principal components F as

$$Z_j = A_{j1} + A_{j2} F_2 +....+ A_{jm} F_m, \; j = 1, 2, ..., m. \tag{5.2}$$

It should be noted that a reduction in the number of principal components used in equation 5.2 is possible if a certain percentage of unexplained variance of the m original variables Z is allowed. For example, if the first m_1 ($m_1 < m$) principal components account for 98 percent of the variance (hence 2 percent unexplained variance) in the m original variables, and if a 2 percent unexplained variance in expressing the m original variables is allowed, then only the first m_1 principal components are needed in describing the m original variables. The value of m_1 is dependent on the value of m and the maximum percentage of unexplained variance of the original variables allowed. The reduction in the number of principal components used from m to m_1 represents a way of reducing the size of the multivariate projection model, since only projections of m_1 principal components are needed to calculate projections of m original variables. In the demonstration examples herein, no reduction in variance was allowed, hence m_1 = m in these cases.

Multivariate Projection Model. Simultaneous projection of a set of original variables can be obtained by the following steps using the principal-component analysis allied with the single-variable projection model, ARIMA(p,d,q):

1. Find the component-pattern coefficients, A_{jk}, and component-estimate coefficients, b_{jk}, by principal-component analysis. (For the purpose of this study, these coefficients were calculated by the computer program BMD01M which is written in FORTRAN and available from Biomedical Computer Programs (BMD) developed at UCLA for IBM system 360/370.)

2. Construct a time series for each of m principal components by substituting values of the m original variables in equation 5.1.

3. Apply the ARIMA(p,d,q) model as described in section 4 to project future values of the most likely projection level and of a number of simulation sets for each of m principal components independently.

4. Compute projections of the most likely projection level and simulations of the m original variables by substituting projections of principal components in equation 5.2.

6. TEST TECHNIQUE

Split-Record Testing. In order to develop some degree of confidence that projections using techniques studied herein would in fact represent the entire range of future possibilities, or at least to evaluate the degree to which they do represent the future range and possibly to develop correctional criteria, the split-record type of testing has been employed in this study. This consists of calibrating the projection models on the basis of one part of the record (usually half) and examining the range of projections in comparison with the actual variations that occured in the remainder of the record. This technique has been particularly favored, because it conforms to the exact manner in which projections are employed in water resources planning. In the test procedure, the first half of the record might represent all of the record that would be available at the time of planning the project, and the second half of the record would represent what occurs during the operation of the project. Thus, in concept, the test would show how the various projections made in project planning would actually compare to events that really occurred during the project life.

The split-record technique should be evaluated on the basis of a substantial number of variables. It is possible that a particular technique might accidentally show exceptional results when applied to one or two variables, but it is not likely that consistently superior results will be accidentally obtained as the number of variables increases. Even when a substantial number of test variables are used, care must be exercised that a great many techniques are not arbitrarily tested and the best selected, because, if sufficient alternatives are tried, there is an increasing probability that one will accidentally show favorable results.

When records are studied in halves for the purpose of testing, it is often possible to make more efficient use of available data by reversing the halves and repeating the test procedure. Thus, a model calibrated on the first half could be used for projecting through the period of the second half, and a model calibrated in reverse on the second half could be used for projecting in reverse through the first half of the record. Depending on the nature of the trends, this double use of the data could actually double the effectiveness of the test, but in cases where the trends are consistently upward or downward, this double use of the data might not contribute greatly to the test results insofar as they would apply to those particular variables. In order to maximize the similarity of the tests to actual application, halves were not reversed in this study.

It should be recognized that, even though the test procedures are limited to calibration of the model using only half of the data, actual projections for water resources planning studies should be based on models calibrated using all available data.

Sample Time Series. To facilitate the test, 31 time series of 17 economic, 10 population, and 4 hydrologic variables were collected and listed in table 1.

Selection of Representative Projections. In each split-record test, 51 stochastic projections were made. These projections are numbered in the order of magnitude of the final value at the end of the projection (end of the second half of the record). In order to illustrate the range of projections and, in fact, to represent the entire set of projections with a small number of projections, projection numbers 2, 8, 26, 44 and 50 were selected. These are illustrated in figures 1 to 37. In each figure, the heavy line represents the observed data, the line connecting the symbol, ⌖, represents the most likely projection level, and the other five plots represent the five selected representative projections. On the basis of the variables used in this study, it appears that variations can occur in the future with a substantial probability (in the order of once in 50 projections) that differ in magnitude from the extreme projections computed in this study by an amount exceeding the range of projections computed and illustrated herein. In order to take these "maverick" possibilities into account properly, it is proposed that the following weights be assigned to the five representative projections:

Projection no.	Representing Projection	Probability
2	1 - 3	.06
8	4 - 12	.18
26	13 - 39	.50
44	40 - 48	.18
50	49 - 51	.06
Total		.98

Two more projections, to be assigned .01 probability each, can be obtained by utilizing projections numbers 2 and 50 to obtain extreme projections A and B such that values for each year are defined as:

$$Z_A = 2Z_2 - Z_{50}$$

$$Z_b = Z_2 - 2Z_{50}$$

This gives 2 new (maverick) projections that differ from projections 2 and 50 by the difference between 2 and 50 values for each respective year in terms of the transformed variable, Z. The maverick projections would each have a weight (probability) of .01, thus completing the range of probability represented in the above tabulation. Figures 38 and 39 are used to demonstrate the use of these extreme projections for the variable Man-Hours in Farming and variable Construction Cost, respectively. In each chart, projection A is represented by a single line connecting the symbol, *, and projection B by a single bold line.

7. STUDY RESULTS

Analysis of Economic Variables. Of the 17 economic variables (Fig 1 to Fig 17), observed quantities at the end of the projection period were outside of the range of projection in 5 cases (Figures 6, 7, 8, 13, and 17). If the projection techniques adequately represented all future probabilities, only one of these would be expected to be outside the range (an expected frequency of 4 out of 52 random projections would lie outside the range of the second-largest and second-smallest of 51 projections).

A careful review of these projections will indicate that, while different types of models would have greatly improved the projections in individual cases, no systematic model will satisfactorily represent what actually happened in all cases. Likewise, once a person sees the outcome of an observed variable, it is often possible to explain the variations, usually in terms of related factors, but it should be remembered that this study is concerned only with statistical means of projecting a stochastic variable or a set of interrelated stochastic variables.

It becomes obvious, therefore, as has been generally recognized, that the probabilistic range of projections should include provision for possible radical changes not indicated by the model calibration data. This is partly due to the fact that no model simple enough to be calibrated satisfactorily can adequately represent the complex factors involved and partly due to the likelihood that factors of potentially major impact are not adequately reflected in a particular sample of data.

Analysis of Demographic Variables. Population projections were made individually for all 10 cases (Fig 18 to Fig 27), and then 6 cases (Fig 28 to Fig 33) were projected jointly. Models selected by the computer program are as follows:

Principal-component No.	ARIMA(p,d,q) model
1	0, 1, 1
2	1, 1, 0
3	1, 1, 0
4	0, 1, 1
5	2, 1, 0
6	0, 1, 1

In the joint projections, selection of extreme projections was based on the weighted total population at the end of the sequence, giving equal weight to each variable, and simultaneous projections for all variables were used. Thus, the second highest of 51 projections for the weighted total population was not necessarily the second highest projection for each variable. In fact, it would tend to be less extreme in most or all cases, thus reducing the range of projections for each variable below that obtained in the independent projections. Whether the projections should be made jointly or independently would depend on the application, and it is just as vital that they be made independently for independent applications as it is that they be made jointly for applications where their simultaneous impact is of primary concern.

Of the 10 independent projections, observed values at the end of the projection peiod lie outside the projection range in 2 cases, which is at least twice the expected frequency. Thus, again it is necessary to account for greater variation in potential projections than a reasonably simple model indicates.

Analysis of Streamflow Variables. In the cases of natural streamflows, projections (Fig 34 to Fig 37) appear to be reasonable in relation to observed variations during the period of projection, except for the Columbia River, where much lower quantities occurred in some years than were projected to any year. There is a strong downward trend through the early years and until 1935, after which a period of 20 years showed a strong resurgence of streamflows. The model used for streamflows is based on steady-state projection, because this gives best results for streamflows in general. If the model using d = 1 (first differences projected were used), it is apparent that the flows from 1935 to 1955 could fall well above all projections. It is difficult to determine whether the selected model is inapplicable or the observed data represent rare occurences. There is a tendency to jump to the conclusion that actual events that conform to a fore-casted projection "prove" the projection to be acceptable and that when actual events do not conform, the projection was in error. This, of course, is an over-simplification.

It is pertinent to recognize that the streamflow model determined to be most suitable is a steady-state model (d=0), and that the more general and extensively tested monthly streamflow simulation model developed in the Hydrologic Engineering Center of the Corps of Engineers is more suited for generation of sequences of this type of variable.

8. APPLICATION

General. In discussing the application of techniques studied under this project, it must be remembered that the techniques are meant to be applied to basic variables that cannot be derived directly from other variables that are more easily predictable. That is, the extrapolation is strictly statistical, as it must be in the case of many variables.

In making such extrapolations, natural or physical bounds must be recognized. The natural lower asymptotic bound of zero is taken into account in this study by transforming the variable to a logarithm, which is then unbounded at the lower end. If a variable is asymptotically bounded only on the upper end, taking the logarithm of the difference between the upper limit and the variable would remove that bound. If there are both lower and upper asymptotic bounds, the following transform can be used:

$$Z = \log(\log \frac{B-A}{X-A})$$

where B is the upper bound, A the lower bound, X is the observed variable, and Z is the transformed variable. Other transforms might be devised and perform better under specific circumstances, but this one will at least assure that impossible values are not predicted. The transformed variable is projected, and resulting values are transformed back to the original variable needed in the contemplated study.

When more than one variable are needed in the study and where these are interdependent, it is necessary to perform a principal-component analysis of the transformed variables and to project these components, then recombine them into the transformed variables before transforming back to the original variables.

All of the above provisions are incorporated into the computer program described herein.

Water Resource System Simulation. Because of the complexity of water resources planning and management studies, the most acceptable general procedure for analysis consists of postulating a system and operation rules, and simulating the operation of the system under anticipated system demands and system inputs. Traditionally, system demands have been projected as a most likely sequence of demands, and system inputs have been assumed similar to past experience, adjusted for future changes in controlling conditions (such as urbanization and reservoir construction). It is the thesis of this study that a range of demands and inputs should be considered and that appropriate weights should be assigned to the results of different projections. The technique described above for computing 51 projections and selecting 5 of these plus 2 "maverick" projections is intended to represent the full range of future probabilities with a minimum of computational effort.

Computer programs such as HEC-3, "Reservoir System Analysis for Conservation", developed in the Hydrologic Engineering Center of the Corps of Engineers, are very useful in performing the detailed simulation studies necessary for evaluating system performance under each combination of projections of the pertinent input and demand variables. In the case of hydrologic variables such as streamflow, rainfall and evaporation, where the process can be treated as stationary and where stochastic variations from month to month are important, they should be projected using a model such as HEC-4, "Monthly Streamflow Simulation".

Integration of simulation outputs. To the extent that any of the input and demand variables are independent of each other, it will be necessary to give special consideration to the manner in which the different sequences of the various variables are combined. In extreme cases, it may be necessary to perform simulations corresponding to all combinations of sequences, but it is hoped that this will not be necessary in ordinary applications. For example, principal variations in physical results of a plan of operation may depend almost entirely on hydrologic sequences, in which case the detailed water resource simulation studies need be made only once for each hydrologic sequence. Evaluation of the results in terms of economic output would require integration of many combinations of these outputs with the various economic projections. The expected system output would then be the sum of the cross products of the output for each simulation times the probability associated with the particular input. The probability associated with any particular input when a combination of independent inputs is used equals the product of the probabilities associated with the independent inputs. It must be assured that probabilities associated with all simulations add to 1.0.

Illustrative Projections. For illustration purposes, projections of sample variables to year 2020 were obtained by using the projection techniques and computer programs developed in this study. These are illustrated in figures 40 to 61. Variables which are considered to be interdependent are projected by use of the multivariate projection model. These are:

1. Per capita GNP, industrial production index, gross nonfarm product, and gross farm product (shown in figures 40, 41, 42, 43, respectively).
2. Output per unit of labor input index, and output per unit of capital input index (shown in figures 44, 45 respectively)
3. Wholesale price index and consumer price index (shown in figures 46 and 47 respectively).
4. Manhattan Island real estate mortgage rates, stock exchange call loan rates and commercial paper rates (shown in figures 48, 49, and 50, respectively).
5. Population of New England, Middle Atlantic, East North Central, West North Central, South Atlantic, and East South Central divisions (shown in figures 51, 52, 53, 54, 55, 56 respectively).

Variables projected by single-variable projection models are man-hours in manufacturing, man-hours in farming, number of operating business, unemployment rate, applications for patents, and construction cost index, and projections are shown in figures 57 to 62, respectively. It should be noted that projections illustrated herein are used only to demonstrate the projection techniques developed in this study and should not be used as a formal projection of those variables.

Computer Programs. Algorithms described herein are combined for convenient use into two computer programs -- one for single-variable projections, described in Appendix A, and one for multivariate projections, described in Appendix B.

9. CONCLUSION

Comparison of actual sequences of demographic and economic variables demonstrates that no projection technique can be depended upon to yield acceptable results if only a single sequence is projected. It is not possible to predict variations accurately, so it is essential that a range of projections be considered if errors of projection are important to a proposed study.

A model used for projecting such variables must be flexible enough to define variations reasonably, yet simple enough to be calibrated satisfactorily with available data. The two requirements are often unobtainable, and it becomes necessary in those cases to conform with the latter requirement. As a consequence, estimated errors of projection are usually smaller than true errors, and it has become necessary to expand the range of possible projections as described herein.

The autoregressive-moving-average family of projection models described herein provides a range of models from which one can be selected on the basis of goodness of fit to observed data and reasonableness of projection as described herein. The selected model can then be calibrated on the basis of all available data, and a set of projections generated to constitute input and demand data for any water resource system simulation study. Computer programs for accomplishing this are described in appendices A and B. The weighted average accomplishments of any proposed plan of development or management can then be computed as described herein, for the purpose of evaluating the plan under the complete range of future possible projections and computing and evaluation that should be far superior to any obtainable by conventional methodology based on single projections of each variable.

10. REFERENCES

A search of recent statistical literature concerning time series analysis especially in the area of forecasting and control was undertaken in order to compare methodologies and to apply newly advanced treatments to the problems addressed by this study. It was found that most of the related material published prior to 1970 was included in the book entitled Time Series Analysis; Forecasting and Control (Box and Jenkins, 1970). Chatfield and Pepper (1971) discussed some practical aspects of time-series analysis, Bhattacharyya (1974) employed the forecasting technique in time-series analysis to predict the demand for telephones in Australia, and Box and Jenkins (1974) presented some of the more recent advances in forecasting and control of a time-series.

1. Bhattacharyya, M.N., "Forecasting the Demand for Telephones in Australia", Applied Statistics, Vol. 23 No. 1, 1974.

2. BMD Biomedical Computer Programs, University of California, Publications in Automatic Computation No. 2, University of California Press, Berkeley and Los Angeles, 1970.

3. Box, G.E.P., and Jenkins, G.M., Time Series Analysis; Forecasting and Control, Holden-Day, Inc. San Francisco, 1970.

4. Box, G.E.P., Jenkins, G.M. and MacGregor, "Some Recent Advances in Forecasting and Control, Part II. Applied Statistics, Vol. 23 No. 2, 1974.

5. Chatfield, C., and Pepper, M.P.G., "Time-series Analysis: An Example from Geophysical Data", Applied Statistics, Vol. 20 No. 3, 1971.

6. IMSL Manual, International Mathematical and Statistical Libraries, Inc., Houston, Texas, 1974.

11. ACKNOWLEDGMENT

The principal financial support for this study was furnished by the U.S. Office of Water Research and Technology. Dr. Tsann Yu and Rajendra Juyal contributed the early development of the study. Mathematical models and computer routines obtained from outside sources as acknowledged within the text came from the International Mathematics and Statistical Library and the Biomedical Computer Programs Library.

RECORDED ECONOMIC, DEMOGRAPHIC AND HYDROLOGIC DATA

A11 PER CAPITA GNP (1958 DOLLARS,1889-1966) 78

805	847	867	932	869	829	910	875	941	944
1013	1022	1119	1107	1141	1105	1164	1275	1271	1144
1259	1246	1266	1305	1331	1206	1224	1377	1328	1449
1446	1403	1344	1404	1562	1578	1593	1674	1667	1666
1748	1565	1442	1234	1208	1301	1413	1591	1664	1573
1690	1818	2086	2348	2679	3041	3126	2516	2408	2429
2367	2519	2655	2679	2771	2706	2855	2856	2847	2778
2899	2913	2917	3052	3125	3243	3393	3555		

A15 INDUSTRIAL PRODUCTION INDEX (1913=100, 1860-1959) 100

7.48	7.49	6.94	7.88	8.35	8.00	9.84	10.30	10.80	11.60
11.70	12.30	14.60	14.40	13.90	13.50	13.40	14.60	15.50	17.50
20.30	22.30	23.90	24.40	23.10	23.20	27.90	29.50	30.60	32.60
35.00	36.00	38.80	34.70	33.70	39.70	36.90	39.70	44.70	49.20
50.60	56.70	63.20	65.40	62.30	73.60	78.90	80.60	68.00	80.20
85.30	82.20	93.70	100.00	94.10	109.30	129.60	129.70	128.80	113.20
124.00	100.10	125.90	144.40	137.70	153.00	163.10	164.50	171.80	188.30
155.60	129.70	100.50	119.90	129.70	149.10	178.30	194.50	152.30	188.00
213.90	275.50	340.30	405.20	398.70	343.60	291.70	320.90	333.90	317.70
366.30	398.70	411.70	447.30	421.40	473.20	489.40	492.70	457.00	517.30

A17 GROSS NONFARM PRODUCT (BILLIONS OF 1958 DOLLARS) 1889-1966) 78

33.4	37.1	38.8	44.4	41.9	39.8	45.5	43.3	47.9	48.3
54.3	55.9	64.3	65.2	68.7	67.1	73.1	82.8	85.0	75.9
87.8	88.3	92.6	94.9	101.8	90.4	92.0	110.3	103.6	110.4
115.5	116.9	114.8	121.4	139.3	144.9	147.4	158.6	159.6	162.0
162.5	142.7	126.7	103.3	100.7	114.3	126.2	148.2	156.3	144.5
160.2	177.1	206.7	225.4	242.8	257.2	254.2	246.0	253.3	264.6
263.8	292.2	313.5	321.4	338.0	332.4	361.7	368.5	374.3	367.8
395.2	404.5	411.4	441.2	460.3	488.4	521.2	557.2		

RECORDED ECONOMIC, DEMOGRAPHIC AND HYDROLOGIC DATA

A21 GROSS FARM PRODUCT (BILLIONS OF 1958 DOLLARS) 1889-1966) 78

11.7	11.5	12.0	11.4	11.1	11.5	12.2	13.1	14.1	14.7
14.7	14.9	14.8	14.6	15.1	15.5	15.7	16.5	15.8	16.1
15.7	16.2	15.3	18.1	15.7	17.5	18.8	16.5	18.2	16.5
16.6	16.4	15.4	16.5	17.6	16.7	17.9	17.8	18.3	17.9
18.4	17.4	19.7	19.2	18.7	15.6	17.6	16.1	19.0	19.0
19.5	18.9	20.4	22.2	21.1	20.8	19.5	20.2	18.6	20.5
19.9	20.9	19.9	20.6	21.6	21.9	22.5	22.4	21.8	22.5
22.7	23.5	23.9	23.8	24.5	24.1	25.7	24.2		

A72 MAN-HOURS IN MANUFACTURING (1958=100,1889-1969) 81

35.7	37.2	38.1	40.8	38.7	35.7	39.8	38.3	39.6	40.8
46.5	48.3	50.7	55.5	57.2	52.6	59.9	63.0	66.1	56.6
64.4	67.8	67.2	70.2	70.4	66.5	69.1	83.5	89.0	88.1
80.8	82.5	58.0	65.1	76.4	68.6	71.8	74.1	73.2	72.4
76.9	65.4	53.3	42.6	45.7	48.1	54.1	62.8	68.0	54.1
62.7	69.1	88.8	109.0	129.9	128.3	109.9	97.7	102.5	102.2
92.9	99.6	107.7	109.7	114.1	104.5	109.7	111.2	109.4	100.0
107.1	107.3	104.3	109.0	110.0	113.5	118.5	126.6	125.7	128.2
130.1									

A74 MAN-HOURS IN FARMING (1958=100, 1889-1969) 81

181.3	183.9	185.5	187.2	189.0	190.7	192.5	194.3	196.0	197.8
199.3	201.1	202.4	203.7	205.0	206.1	207.4	208.7	209.8	211.2
212.5	213.8	218.2	221.0	218.2	225.0	220.4	219.1	225.2	228.1
223.9	227.4	209.8	217.1	218.6	221.0	225.6	226.3	217.5	221.5
219.5	217.2	222.0	214.2	213.7	191.7	199.5	193.7	209.4	195.0
196.0	196.8	190.0	197.6	196.0	194.1	181.1	174.6	166.7	162.2
162.7	150.0	141.9	134.7	125.3	121.4	123.4	117.8	107.9	100.0
99.8	98.0	91.7	89.5	84.6	81.3	79.2	72.0	69.6	68.0
63.6									

TABLE 1

RECORDED ECONOMIC, DEMOGRAPHIC AND HYDROLOGIC DATA

A160 NUMBER OF OPERATING BUSINESSES (THOUSAND, 1870-1970) 101

427	457	500	494	558	603	639	637	661	702
747	782	822	864	905	920	970	994	1047	1051
1111	1143	1173	1193	1114	1209	1152	1058	1106	1148
1174	1219	1253	1281	1320	1357	1393	1418	1448	1486
1515	1525	1564	1617	1655	1675	1708	1733	1708	1711
1821	1927	1983	1996	2047	2113	2158	2172	2199	2213
2183	2125	2077	1961	1974	1983	2010	2057	2102	2116
2156	2171	2152	2023	1855	1909	2142	2405	2550	2679
2687	2608	2637	2667	2632	2633	2629	2652	2675	2708
2708	2641	2589	2544	2524	2527	2520	2519	2481	2444
2442									

A163 OUTPUT PER UNIT OF LABOR INPUT INDEX (1958=100, 1889-1969) 81

26.8	28.0	28.5	30.0	28.9	29.6	30.9	30.1	32.1	32.6
32.6	33.0	34.9	33.1	33.7	34.0	34.3	36.6	36.4	35.1
37.2	36.2	36.9	37.3	38.4	36.3	37.6	39.4	37.2	40.1
43.0	42.6	46.4	45.4	47.2	49.7	49.5	50.5	51.1	51.3
53.5	52.3	54.0	53.2	52.1	56.2	57.8	59.2	59.3	61.5
63.5	64.9	66.3	65.7	65.8	70.1	74.0	71.7	71.0	73.5
76.8	82.3	82.9	83.8	87.1	90.3	94.4	94.4	96.4	100.0
103.0	104.5	107.7	112.4	115.6	119.4	122.9	125.4	127.4	131.1
132.0									

A165 OUTPUT PER UNIT OF CAPITAL INPUT INDEX (1958=100, 1889-1969) 81

58.0	60.3	59.8	61.7	55.7	52.4	57.0	53.6	57.6	56.8
60.4	59.7	64.5	62.5	63.3	60.4	63.3	68.3	66.8	59.1
65.3	63.9	64.3	66.0	66.6	59.4	59.7	67.8	64.0	66.7
67.3	66.2	62.9	66.4	74.3	73.9	74.7	76.9	75.6	74.5
77.5	68.1	63.1	56.2	56.8	62.1	69.8	78.6	81.6	76.4
83.9	89.3	98.2	102.9	108.6	115.7	115.6	109.4	105.6	106.7
102.4	109.1	109.5	106.7	108.0	103.2	108.4	105.6	103.4	100.0
104.5	104.1	103.5	108.0	109.4	111.9	114.5	115.3	112.7	113.7
112.3									

TABLE 1

RECORDED ECONOMIC, DEMOGRAPHIC AND HYDROLOGIC DATA

B1 UNEMPLOYMENT RATE (1890-1971) 82

4.00	5.40	3.00	11.7	18.4	13.7	14.4	14.5	12.4	6.50
5.00	4.00	3.70	3.90	5.40	4.30	1.70	2.80	8.00	5.10
5.90	6.70	4.60	4.30	7.90	8.50	5.10	4.60	1.40	1.40
5.20	11.7	6.70	2.40	5.00	3.20	1.80	3.30	4.20	3.20
8.70	15.9	23.6	24.9	21.7	20.1	16.9	14.3	19.0	17.2
14.6	9.90	4.70	1.90	1.20	1.90	3.90	3.90	3.80	5.90
5.60	3.30	3.00	2.90	5.50	4.40	4.10	4.30	6.80	5.50
5.50	6.70	5.50	5.70	5.20	4.50	3.80	3.80	3.60	3.50
4.90	5.90								

B50 APPLICATIONS FOR PATENTS (1860-1970) 111

7653	4643	5038	6014	6932	10664	15269	21276	20420	19271
19171	19472	18246	20414	21602	21638	21425	20308	20260	20059
22395	25556	31218	34311	35422	35559	35806	35461	35684	40464
40930	40443	30644	38353	38344	40608	43905	47811	35758	41337
41898	46334	49490	50059	51986	54815	56277	58575	61273	65642
64448	68904	70818	70177	70228	69872	70759	70135	59581	80337
86575	93063	88725	80333	80622	84290	85708	91692	92364	94272
93752	83967	71397	60185	61070	63917	69143	72576	75006	71306
69484	59609	49827	48520	59295	75964	91826	83179	75847	74660
74108	64788	69631	77833	82745	83070	79834	79012	82552	83587
84246	87921	90077	90837	92976	100150	93250	92719	98402	103993
109052									

TABLE 1

RECORDED ECONOMIC, DEMOGRAPHIC AND HYDROLOGIC DATA

B68 WHOLE SALE PRICE INDEX (1967=100, 1860-1970) 111

31.8	32.8	38.1	48.5	70.4	67.2	62.9	58.2	56.5	53.6
47.5	45.4	48.4	46.5	44.0	41.7	38.4	37.3	31.9	31.3
35.0	35.5	37.0	34.3	31.9	29.4	28.3	29.4	30.0	28.7
28.9	28.8	26.9	27.5	24.7	25.2	23.9	24.0	25.0	26.9
28.9	28.5	30.4	30.7	30.8	31.0	32.0	33.6	32.4	34.9
36.4	33.5	35.6	36.0	35.2	35.8	44.1	60.6	67.6	71.4
79.6	50.3	49.9	51.9	50.5	53.3	51.6	49.3	50.0	49.1
44.6	37.6	33.6	34.0	38.6	41.3	41.7	44.5	40.5	39.8
40.5	45.1	50.9	53.3	53.6	54.6	62.3	76.5	82.8	78.7
81.8	91.1	88.6	87.4	87.6	87.8	90.7	93.3	94.6	94.8
94.9	94.5	94.8	94.5	94.7	96.6	99.8	100.0	102.5	106.5
110.4									

B69 CONSUMER PRICE INDEX (1967=100, 1860-1970) 111

27.0	27.0	30.0	37.0	48.0	47.0	44.0	43.0	41.0	40.0
38.0	37.0	37.0	37.0	35.0	33.0	32.0	32.0	30.0	29.0
30.0	30.0	30.0	29.0	28.0	28.0	28.0	28.0	28.0	28.0
28.0	28.0	28.0	28.0	26.0	25.0	25.0	25.0	25.0	25.0
25.0	25.0	26.0	27.0	27.0	27.0	28.0	29.0	28.0	28.0
29.0	29.0	30.0	29.7	30.1	30.4	32.7	38.4	45.1	51.8
60.0	53.6	50.2	51.1	51.2	52.5	53.0	52.0	51.3	51.3
50.0	45.6	40.9	38.8	40.1	41.1	41.5	43.0	42.2	41.6
42.0	44.1	48.8	51.8	52.7	53.9	58.5	66.9	72.1	71.4
72.1	77.8	79.5	80.1	80.5	80.2	81.4	84.3	86.6	87.3
88.7	89.6	90.6	91.7	92.9	94.5	97.2	100.0	104.2	109.8
116.3									

RECORDED ECONOMIC, DEMOGRAPHIC AND HYDROLOGIC DATA

B77 MANHATTAN ISLAND REAL ESTATE MORTGAGE RATES (1879-1970) 92

5.92	5.78	5.80	5.65	5.43	5.35	5.20	5.10	5.14	5.20
5.13	5.45	5.38	5.20	5.20	5.16	5.04	5.12	5.04	4.96
5.05	5.17	5.11	5.09	5.18	5.35	5.50	5.68	5.45	5.60
5.35	5.35	5.47	5.46	5.50	5.58	5.60	5.50	5.47	5.55
5.65	5.75	5.97	5.95	5.91	5.92	5.90	5.89	5.88	5.85
5.92	5.95	5.75	5.77	5.60	5.45	5.26	5.09	5.11	5.00
5.05	5.03	4.90	4.98	4.77	4.71	4.70	4.74	4.80	4.91
4.93	4.95	4.93	5.03	5.09	5.15	5.18	5.19	5.42	5.58
5.71	5.85	5.87	5.90	5.89	5.88	5.87	6.04	6.18	6.42
7.06	7.73								

B79 STOCK EXCHANG CALL LOAN RATES (1860-1970) 111

5.99	5.76	5.23	6.19	6.59	6.17	5.23	6.26	7.54	10.29
5.72	5.56	8.38	14.24	3.43	3.11	3.35	3.87	4.22	5.44
4.86	5.76	4.78	3.71	3.03	1.66	4.03	5.01	2.51	4.48
5.84	3.42	3.08	4.57	1.07	1.88	4.28	1.75	2.18	5.08
2.94	4.00	5.15	3.71	1.78	4.44	6.54	7.01	1.92	2.71
2.98	2.57	3.52	3.22	3.43	1.92	2.62	3.43	5.28	6.32
7.74	5.97	4.29	4.86	3.08	4.18	4.50	4.06	6.04	7.61
2.94	1.74	2.05	1.16	1.00	0.56	0.91	1.00	1.00	1.00
1.00	1.00	1.00	1.00	1.00	1.00	1.16	1.38	1.55	1.63
1.63	2.17	2.48	3.06	3.05	3.20	4.03	4.50	3.72	4.22
4.99	4.50	4.50	4.50	4.50	4.69	5.78	5.66	6.33	7.96
7.95									

B80 COMMERCIAL PAPER RATES (1890-1970) 81

6.91	6.48	5.40	7.64	5.22	5.80	7.02	4.72	5.34	5.50
5.71	5.40	5.81	6.16	5.14	5.18	6.25	6.66	5.00	4.67
5.72	4.75	5.41	6.20	5.47	4.01	3.84	5.07	6.02	5.37
7.50	6.62	4.52	5.07	3.98	4.02	4.34	4.11	4.85	5.85
3.59	2.64	2.73	1.73	1.02	0.76	0.75	0.94	0.81	0.59
0.56	0.54	0.66	0.69	0.73	0.75	0.81	1.03	1.44	1.49
1.45	2.16	2.33	2.52	1.58	2.18	3.31	3.81	2.46	3.97
3.85	2.97	3.26	3.55	3.97	4.38	5.55	5.10	5.90	7.83
7.72									

TABLE 1

RECORDED ECONOMIC, DEMOGRAPHIC AND HYDROLOGIC DATA

ENR CONSTRUCTION COSTS INDEX (1913=100,1903-1973) 71

94	87	91	95	101	97	91	96	93	91
100	89	93	130	181	189	198	251	202	174
214	215	207	208	206	207	207	203	181	157
170	198	196	206	235	236	236	242	258	276
290	299	308	346	413	461	477	510	543	569
600	628	660	692	724	759	797	824	847	872
901	936	971	1019	1070	1155	1269	1386	1581	1753
1895									

C71 POPULATION-UNITED STATES (THOUSANDS, 1900-1970) 71

76094	77585	79160	80632	82165	83820	85437	87000	88709	90492
92407	93868	95331	97227	99118	100549	101966	103266	103203	104512
106466	108541	110055	111950	114113	115832	117399	119038	120501	121770
123077	124040	124840	125579	126374	127250	128053	128825	129825	130880
131954	133121	133920	134245	132885	132481	140054	143446	146093	148665
151235	153310	155687	158241	161164	164308	167305	170372	173320	176289
179975	182973	185738	188438	191085	193460	195501	197374	199312	201306
203805									

C72 POPULATION-NEW ENGLAND DIVISION (THOUSANDS, 1900-1970) 71

5579	5624	5719	5809	5906	6009	6131	6254	6364	6472
6565	6629	6729	6883	7034	7137	7198	7267	7180	7304
7454	7574	7668	7756	7852	7946	8018	8076	8095	8130
8175	8193	8220	8254	8296	8361	8391	8409	8427	8438
8450	8556	8572	8404	8335	8321	8839	9039	9233	9379
9315	9288	9358	9627	9832	9871	9929	10030	10218	10436
10532	10666	10800	10986	11185	11329	11431	11562	11638	11736
11873									

RECORDED ECONOMIC, DEMOGRAPHIC AND HYDROLOGIC DATA

C73 POPULATION-MIDDLE ATLANTIC DIVISION (THOUSANDS, 1900-1970) 71

15480	15777	16096	16439	16810	17205	17638	18066	18515	18968
19393	19709	20017	20341	20650	20912	21229	21547	21503	22031
22220	22613	22951	23372	23909	24328	24609	24810	25369	25883
26364	26675	26885	27017	27137	27234	27332	27389	27564	27553
27526	27432	27003	26477	26000	25746	27755	28796	29557	30172
30243	30357	30821	31418	31991	32407	32699	33065	33548	33935
34270	34717	35033	35416	35768	36122	36358	36544	36797	36941
37271									

C74 POPULATION-EAST NORTH CENTRAL DIVISION (THOUSANDS, 1900-1970) 71

16002	16230	16499	16727	16969	17216	17435	17654	17842	18034
18344	18614	18910	19321	19748	20116	20452	20771	20798	21172
21811	22349	22680	23159	23667	24053	24390	24792	24984	25187
25332	25426	25535	25632	25694	25824	25960	26096	26243	26456
26725	27049	27135	26498	26524	26381	28411	29151	29833	30325
30530	30884	31498	32167	33063	33780	34404	34967	35578	35929
36291	36615	36926	37357	37869	38406	38953	39347	39645	39904
40368									

C75 POPULATION-WEST NORTH CENTRAL DIVISION (THOUSANDS, 1900-1970) 71

10357	10492	10627	10719	10861	10987	11089	11214	11345	11496
11675	11771	11868	12006	12166	12316	12460	12474	12168	12396
12562	12656	12734	12812	12904	12986	13071	13145	13214	13260
13335	13446	13518	13567	13593	13630	13601	13544	13492	13505
13498	13285	13101	12786	12481	12427	13165	13447	13604	13851
14102	14169	14175	14269	14453	14731	14905	14978	14994	15195
15424	15570	15657	15716	15786	15819	15887	15942	16047	16203
16367									

TABLE 1

RECORDED ECONOMIC, DEMOGRAPHIC AND HYDROLOGIC DATA

C76 POPULATION-SOUTH ATLANTIC DIVISION (THOUSANDS, 1900-1970) 71

10458	10625	10787	10937	11082	11276	11446	11595	11811	12038
12259	12474	12669	12954	13266	13500	13667	13788	13976	13883
14101	14340	14484	14649	14835	15015	15234	15480	15610	15702
15816	15942	16066	16205	16397	16563	16743	16960	17263	17594
17935	18439	18933	19460	19339	19210	19774	20086	20199	20607
21250	21796	22134	22384	22719	23207	23842	24528	25043	25556
26091	26668	27183	27742	28259	28743	29133	29484	29925	30324
30772									

C77 POPULATION-EAST SOUTH CENTRAL DIVISION (THOUSANDS, 1900-1970) 71

7554	7688	7775	7858	7922	8021	8118	8171	8250	8340
8442	8516	8599	8737	8852	8866	8897	8933	8869	8845
8909	9030	9144	9265	9420	9495	9600	9730	9812	9852
9895	9952	10044	10137	10241	10326	10374	10427	10535	10681
10814	10883	10888	10847	10468	10339	10810	11019	11078	11169
11486	11534	11492	11380	11338	11435	11470	11559	11681	11863
12073	12198	12318	12416	12536	12627	12677	12717	12737	12755
12823									

C78 POPULATION-WEST SOUTH CENTRAL DIVISION (THOUSANDS, 1900-1970) 71

6553	6801	7037	7260	7488	7706	7911	8113	8345	8585
8843	9035	9189	9375	9552	9685	9869	10055	10130	10132
10347	10577	10738	10923	11128	11332	11540	11762	11930	12072
12209	12282	12346	12438	12524	12627	12695	12744	12838	12975
13075	13300	13448	13626	13195	13045	13677	13936	14137	14205
14610	14991	15214	15164	15209	15581	15839	16200	16399	16658
17010	17292	17678	17850	18058	18209	18395	18570	18828	19112
19397									

TABLE 1

RECORDED ECONOMIC, DEMOGRAPHIC AND HYDROLOGIC DATA

C79 POPULATION-MOUNTAIN DIVISION (THOUSANDS, 1900-1970) 71

1684	1784	1892	1982	2049	2133	2236	2333	2435	2541
2654	2725	2793	2868	2951	3024	3112	3211	3209	3301
3344	3403	3435	3478	3538	3567	3592	3633	3655	3650
3714	3751	3775	3797	3835	3881	3939	3996	4042	4101
4155	4082	4098	4393	4272	4212	4439	4568	4727	4896
5102	5179	5330	5515	5670	5900	6139	6345	6487	6684
6916	7183	7381	7541	7662	7740	7802	7878	8008	8171
8345									

C80 POPULATION-PACIFIC DIVISION (THOUSANDS, 1900-1970) 71

2428	2567	2728	2900	3078	3265	3435	3601	3799	4016
4231	4393	4558	4742	4901	4992	5082	5221	5366	5446
5715	5998	6220	6536	6864	7107	7350	7610	7831	8033
8235	8371	8450	8533	8655	8805	9019	9257	9421	9581
9776	10096	10743	11754	12271	12800	13185	13405	13725	14062
14596	15113	15665	16318	16889	17395	18079	18699	19371	20034
21368	22063	22761	23414	23963	24464	24865	25329	25688	26162
26589									

MEAN ANNUAL FLOW-COLUMBIA R.(CFS, 1879-1970) 92

242000	264000	252000	232000	212000	214000	214000	210000	260000	201000
134000	196000	165000	198000	219000	313600	194000	229000	243000	230000
234000	224000	219000	197000	211000	242000	140000	157000	229000	196000
190000	213000	188000	183000	213000	187000	147000	238000	210000	205000
172000	157000	230000	183000	179000	137000	201000	118000	206000	231000
133000	131000	122000	186000	198000	211500	170200	158900	128700	190200
149700	148500	130000	178600	207300	119600	150800	196180	193700	235500
180100	217100	226400	198200	179300	209100	179000	243400	194100	180700
211600	195300	189900	169300	174100	184200	224500	162000	191300	165700
205800	166500								

TABLE 1

RECORDED ECONOMIC, DEMOGRAPHIC AND HYDROLOGIC DATA

MEAN ANNUAL FLOW-ST. LAWRENCE R.(CFS, 1861-1965) 105

275000	284000	273000	266000	266000	246000	272000	242000	249000	279000
259000	224000	237000	261000	231000	263000	252000	252000	259000	244000
232000	256000	251000	270000	253000	278000	269000	241000	246000	262000
259000	229000	242000	242000	216000	213000	218000	228000	228000	226000
227000	228000	242000	244000	238000	241000	244000	262000	241000	231000
219000	229000	258000	241000	221000	240000	241000	247000	251000	226000
234000	229000	217000	221000	214000	208000	226000	238000	255000	263000
218000	218000	208000	188000	183000	192000	214000	218000	216000	210000
213000	214000	248000	217000	242000	257000	257000	258000	241000	236000
263000	276000	264000	254000	267000	257000	240000	222000	223000	239000
229900	210300	205400	200800	191800					

MEAN ANNUAL FLOW-MISSISSIPPI R.(CFS, 1862-1965) 104

201300	124000	87950	162300	188000	234000	124700	206700	165500	137900
136700	178600	126700	186800	249900	202900	207800	137300	169700	232100
305100	236000	201000	239100	187400	145200	218700	120600	135400	157500
263500	185200	111700	81500	145400	198300	157200	171000	127200	133100
152600	261100	224200	181800	194300	207600	226000	204000	166400	95490
207700	145900	121100	256000	252180	168200	125000	171900	221400	149200
200700	132800	186900	150200	168500	324500	204100	264400	125200	78940
159800	136600	67700	187600	113700	146800	156800	148000	79110	120000
236200	234700	209100	223100	173200	237400	162900	169800	198900	264700
232200	141100	113500	130300	93990	123100	145100	136000	208400	166900
218900	113200	99560	198200						

MEAN ANNUAL FLOW-MISSOURI R.(CFS, 1898-1970) 73

78000	92000	62000	64000	77000	116000	113000	89000	86000	101000
107000	104000	82000	51000	91000	69000	66000	146000	119000	86000
67000	77000	97000	82000	91000	78000	93000	70000	69000	140000
99000	114000	55400	37100	70300	51900	29750	80010	41090	59090
56640	53170	31020	47360	104900	96540	95920	109700	61950	107700
79920	92740	92210	139400	103100	55300	40820	47200	35060	46990
73520	57100	79170	79180	84920	44980	47450	80110	59850	66460
66110	107500	84190							

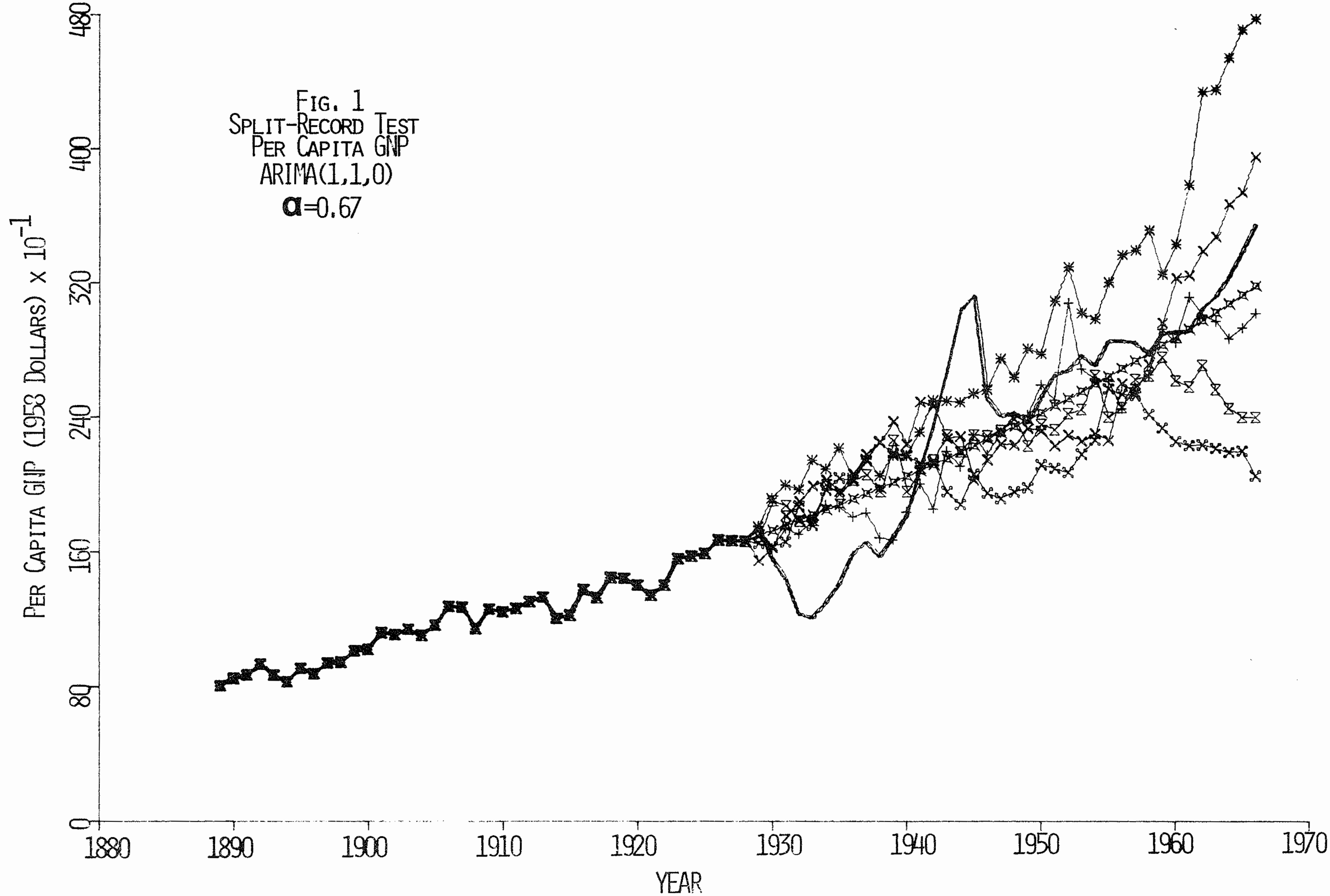


FIG. 1
SPLIT-RECORD TEST
PER CAPITA GNP
ARIMA(1,1,0)
α=0.67
PER CAPITA GNP (1953 DOLLARS) x 10^-1
0
80
160
240
320
400
480
1880
1890
1900
1910
1920
1930
1940
1950
1960
1970
YEAR

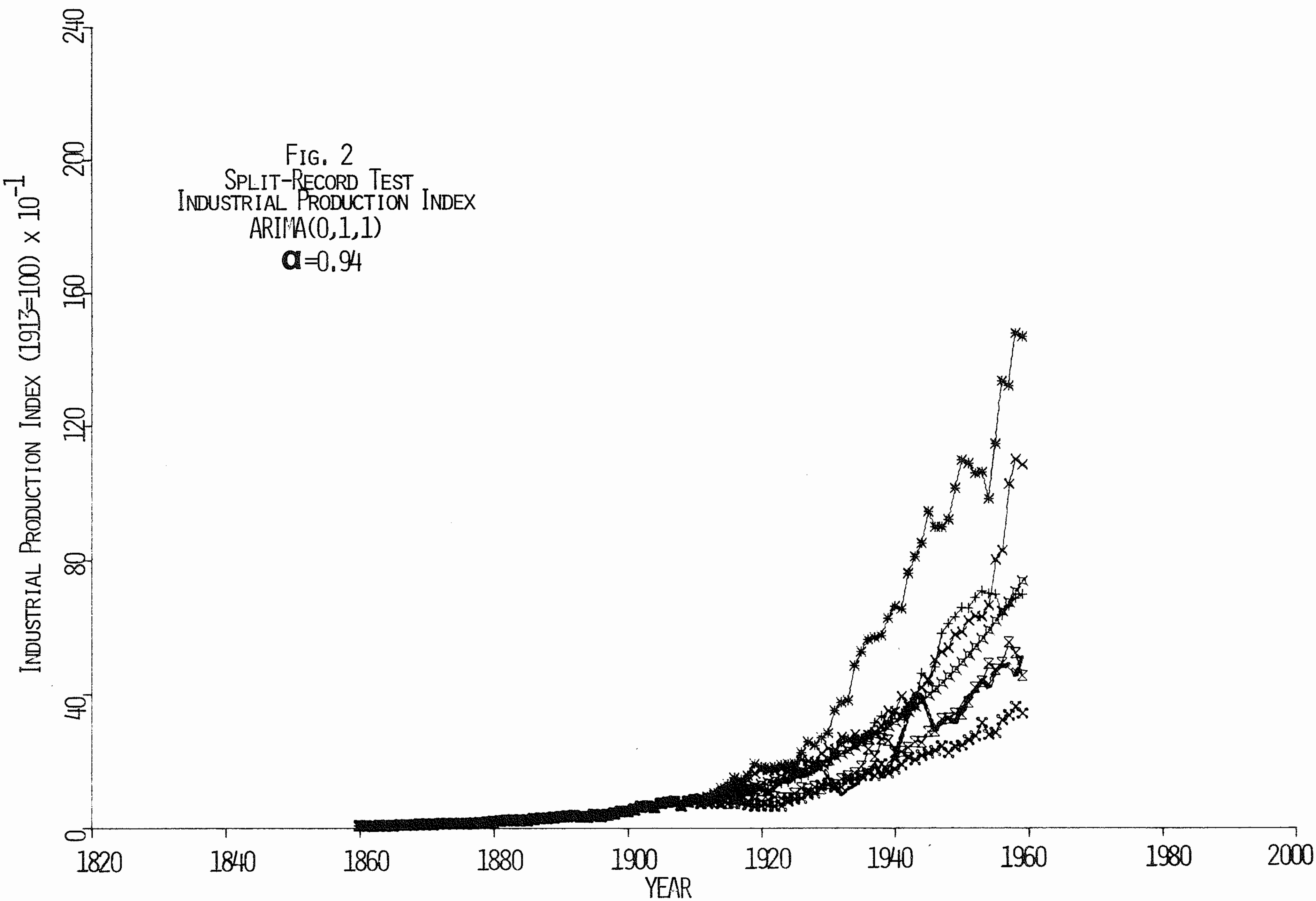


Fig. 2
Split-Record Test
Industrial Production Index
ARIMA(0,1,1)
α=0.94
Industrial Production Index (1913=100) x 10^-1
0
40
80
120
160
200
240
1820
1840
1860
1880
1900
1920
1940
1960
1980
2000
YEAR

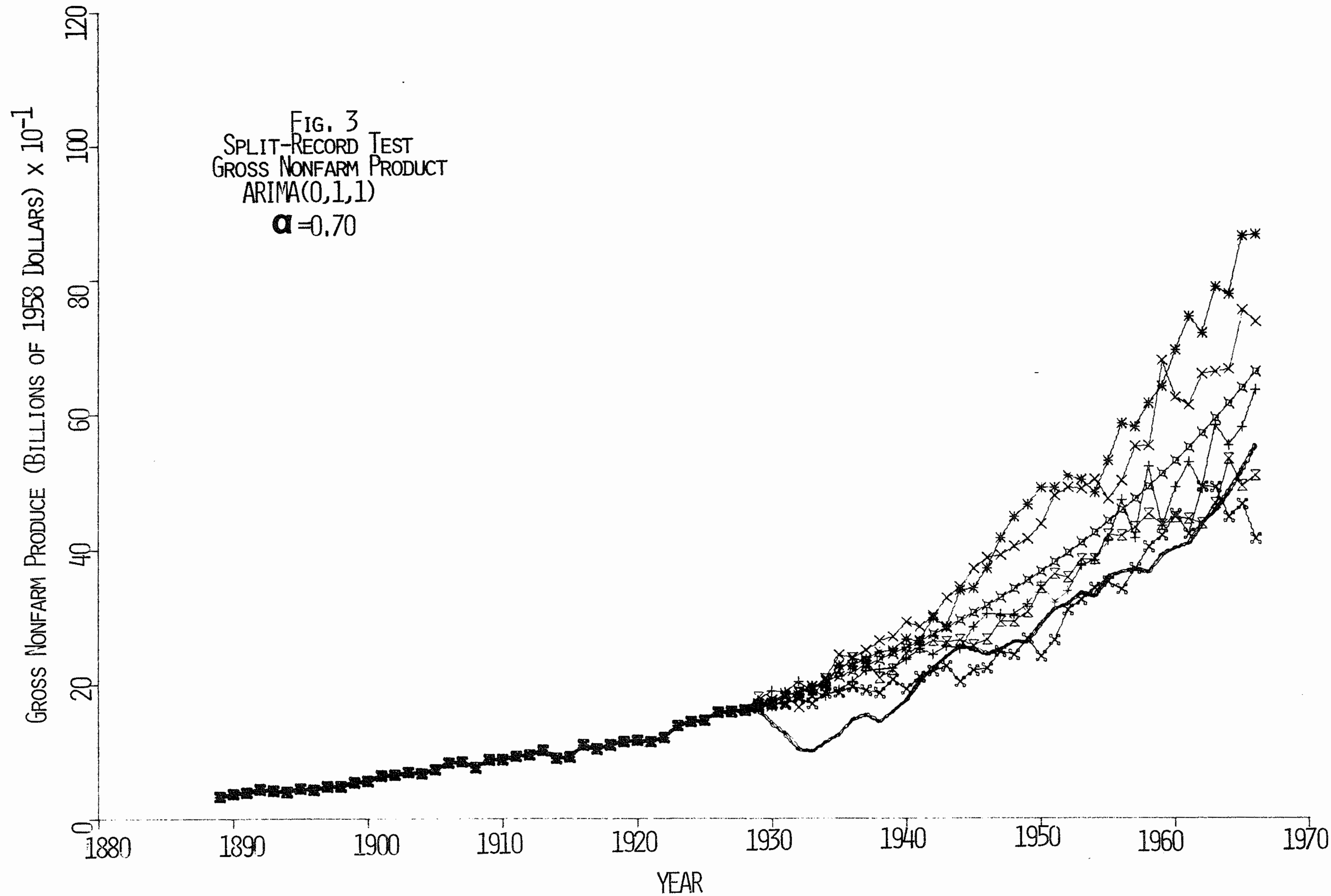


Fig. 3
Split-Record Test
Gross Nonfarm Product
ARIMA(0,1,1)
α =0.70
Gross Nonfarm Produce (Billions of 1958 Dollars) x 10^-1
0
20
40
60
80
100
120
1880
1890
1900
1910
1920
1930
1940
1950
1960
1970
YEAR

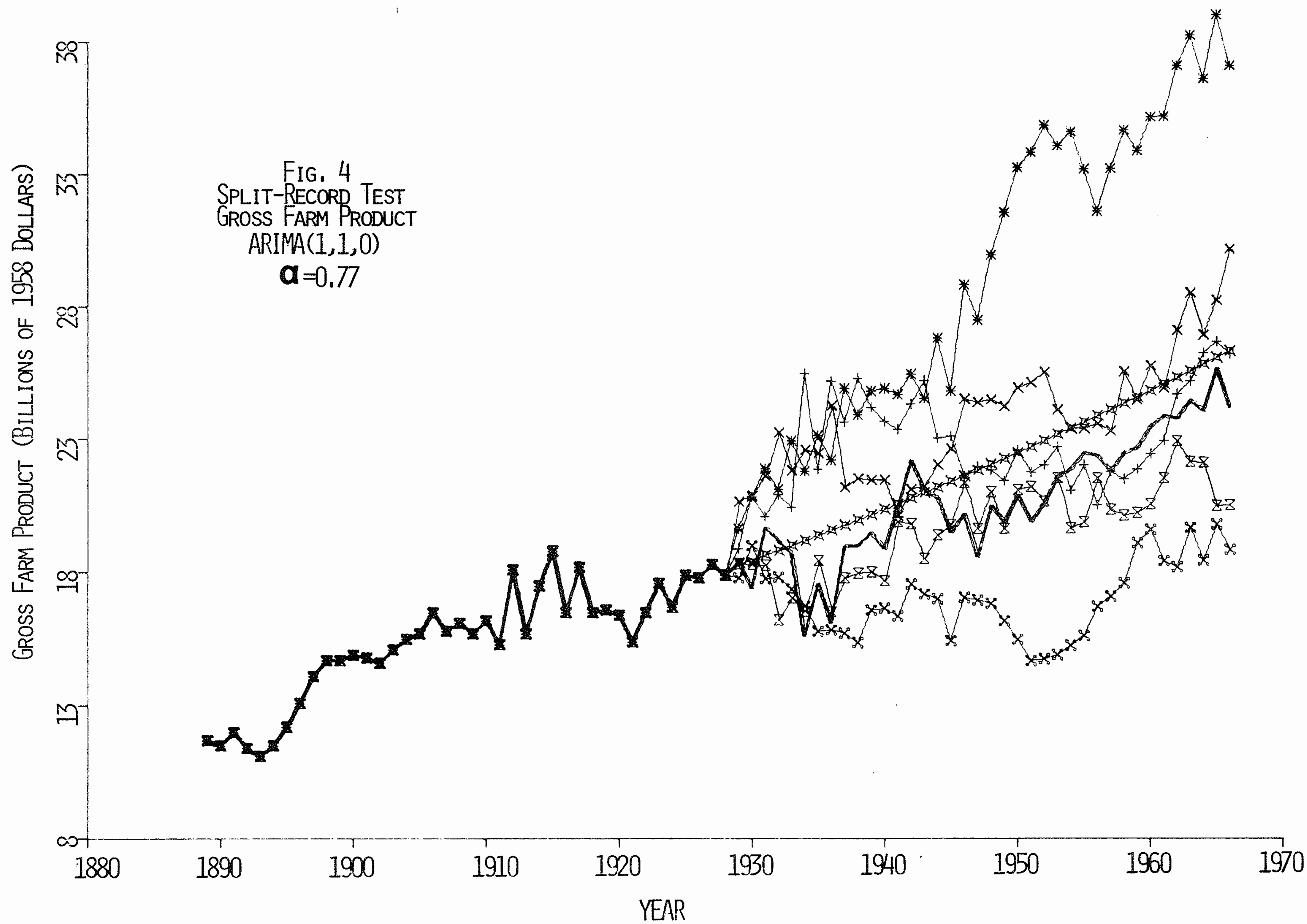


Fig. 4
Split-Record Test
Gross Farm Product
ARIMA(1,1,0)
α=0.77
Gross Farm Product (Billions of 1958 Dollars)
8
13
18
23
28
33
38
1880
1890
1900
1910
1920
1930
1940
1950
1960
1970
YEAR

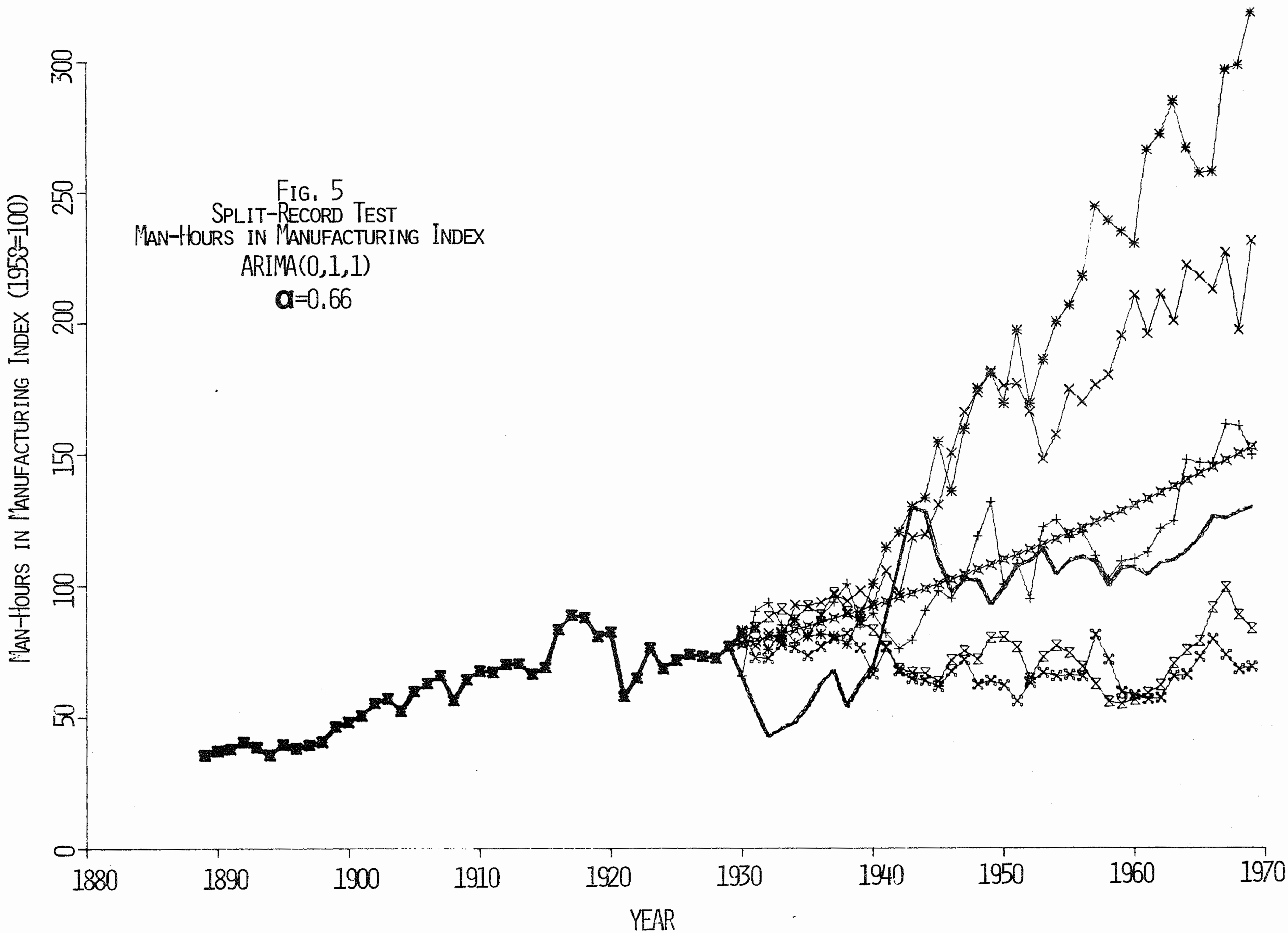


FIG. 5
SPLIT-RECORD TEST
MAN-HOURS IN MANUFACTURING INDEX
ARIMA(0,1,1)
α=0.66

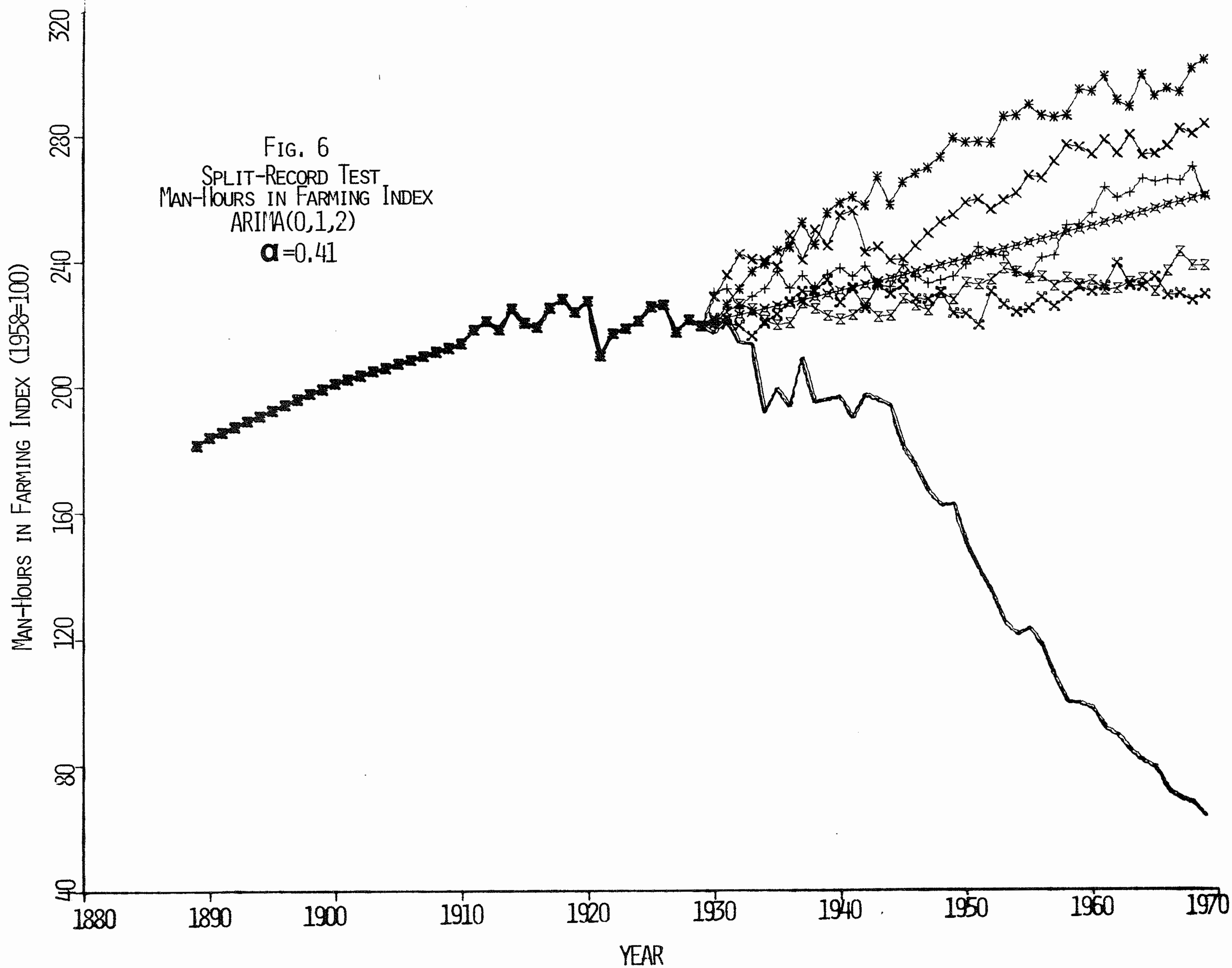


Fig. 6
Split-Record Test
Man-Hours in Farming Index
ARIMA(0,1,2)
α=0.41

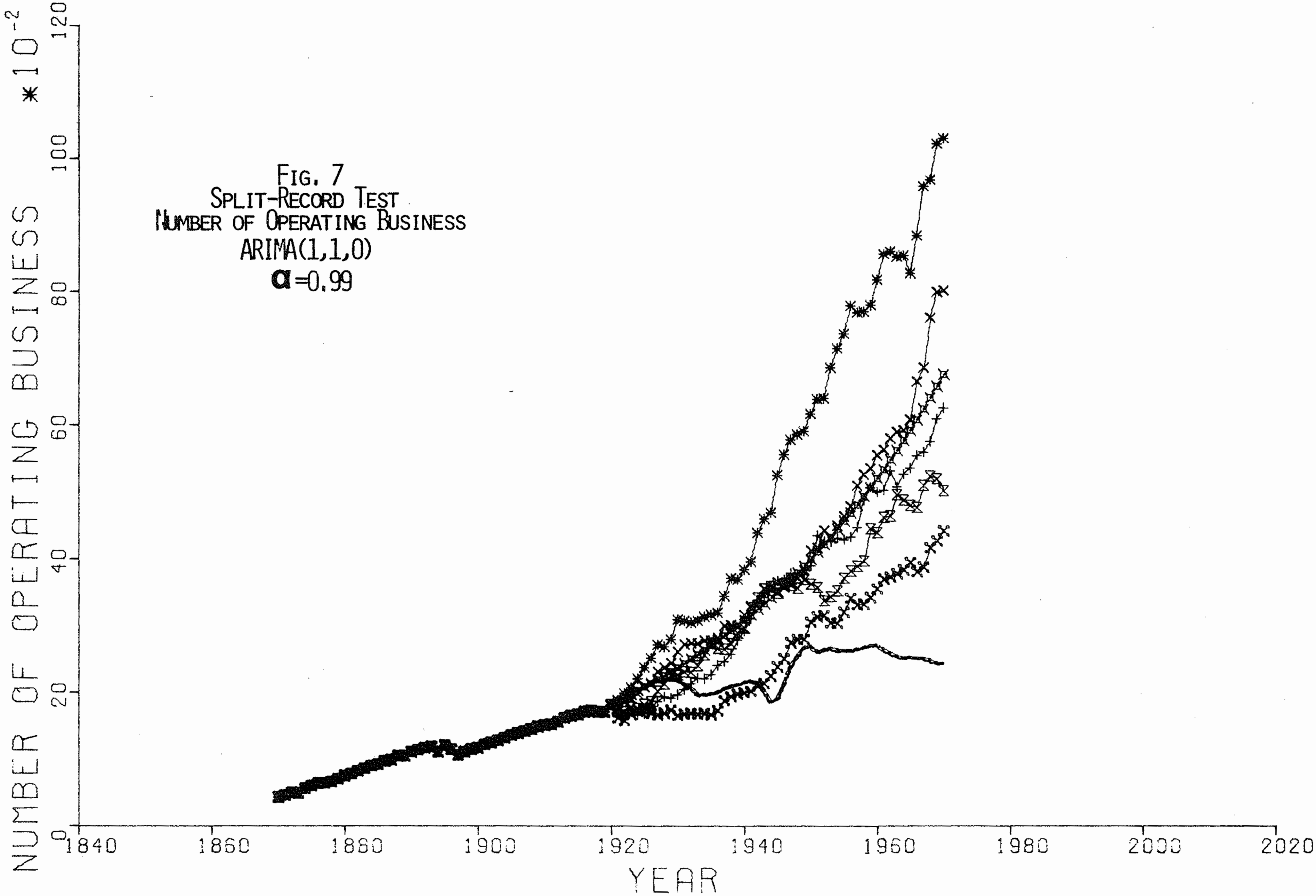


Fig. 7
Split-Record Test
Number of Operating Business
ARIMA(1,1,0)
α=0.99
NUMBER OF OPERATING BUSINESS *10^-2
0
20
40
60
80
100
120
1840
1860
1880
1900
1920
1940
1960
1980
2000
2020
YEAR

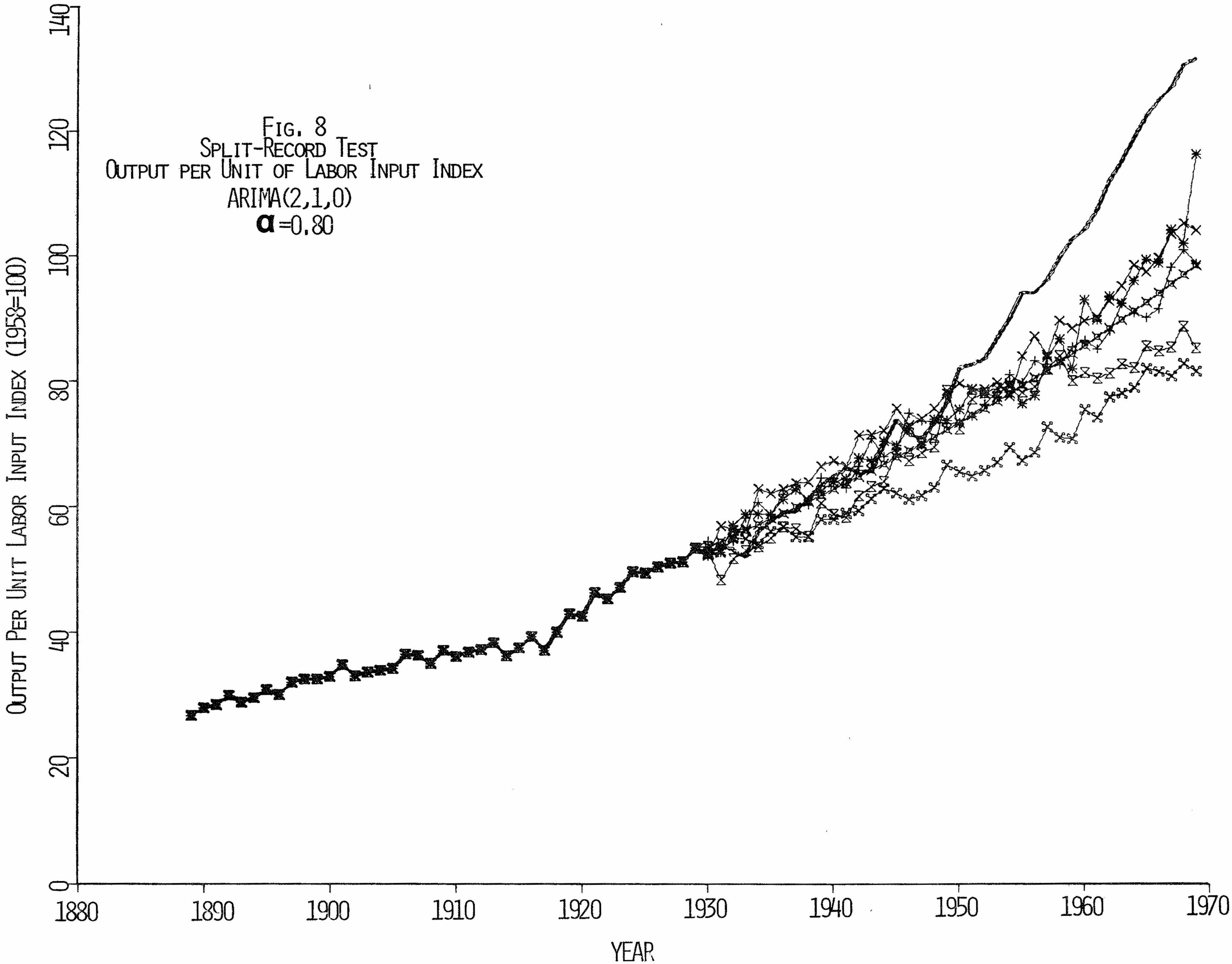


Fig. 8
Split-Record Test
Output per Unit of Labor Input Index
ARIMA(2,1,0)
$\alpha = 0.80$

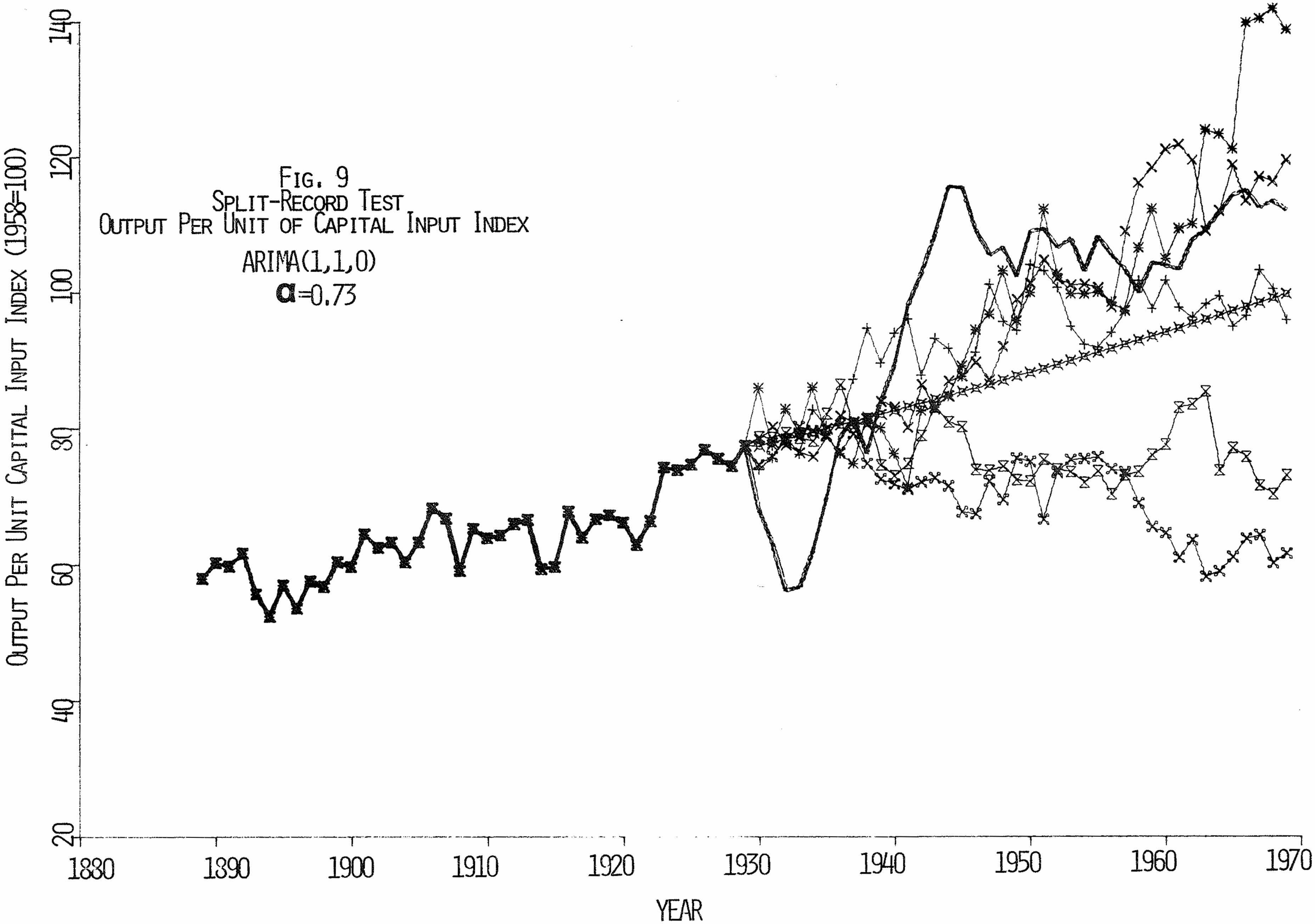


Fig. 9
Split-Record Test
Output Per Unit of Capital Input Index
ARIMA(1,1,0)
α=0.73
Output Per Unit Capital Input Index (1958=100)
20
40
60
80
100
120
140
1880
1890
1900
1910
1920
1930
1940
1950
1960
1970
YEAR

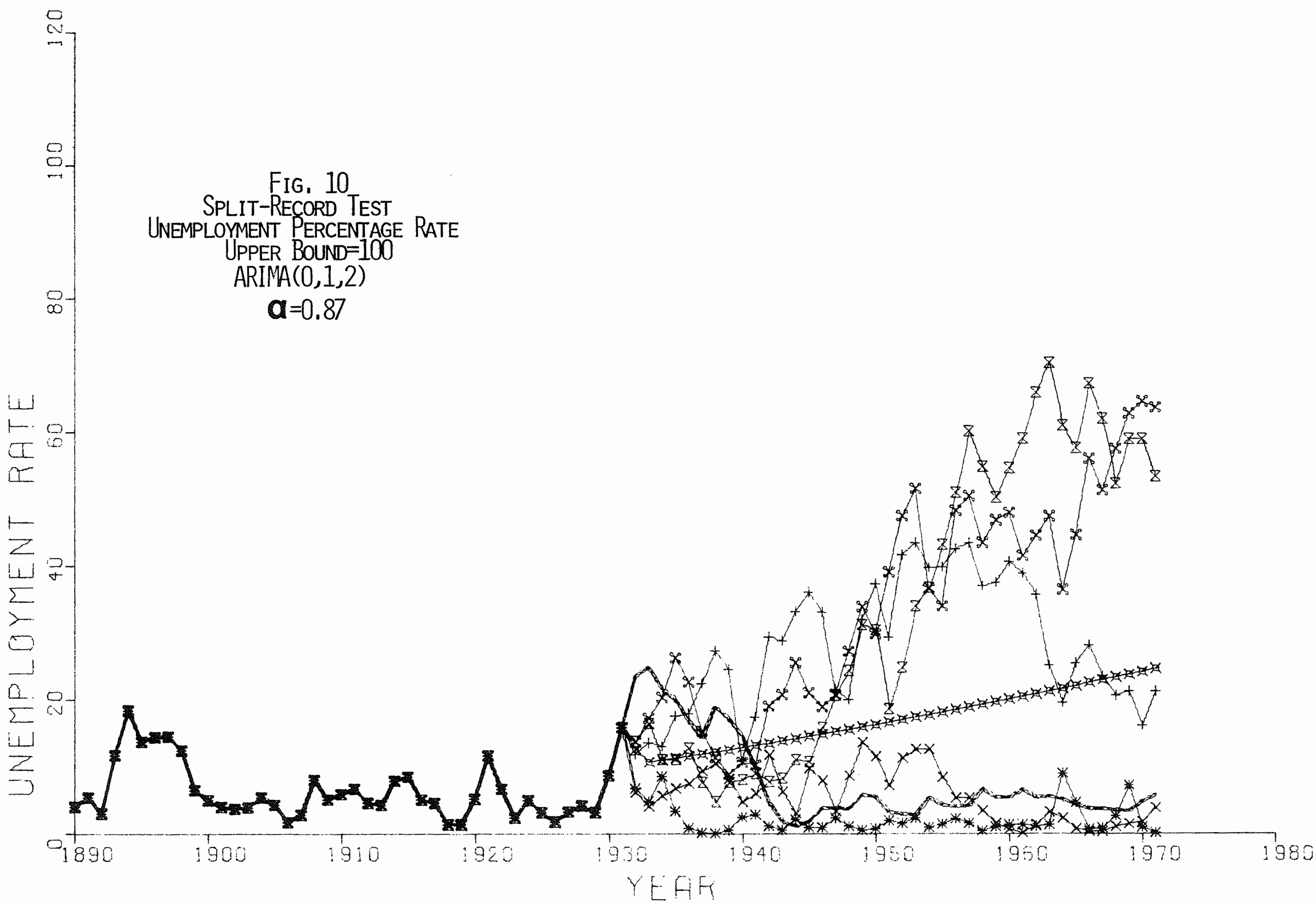


FIG. 10
SPLIT-RECORD TEST
UNEMPLOYMENT PERCENTAGE RATE
UPPER BOUND=100
ARIMA(0,1,2)
α=0.87

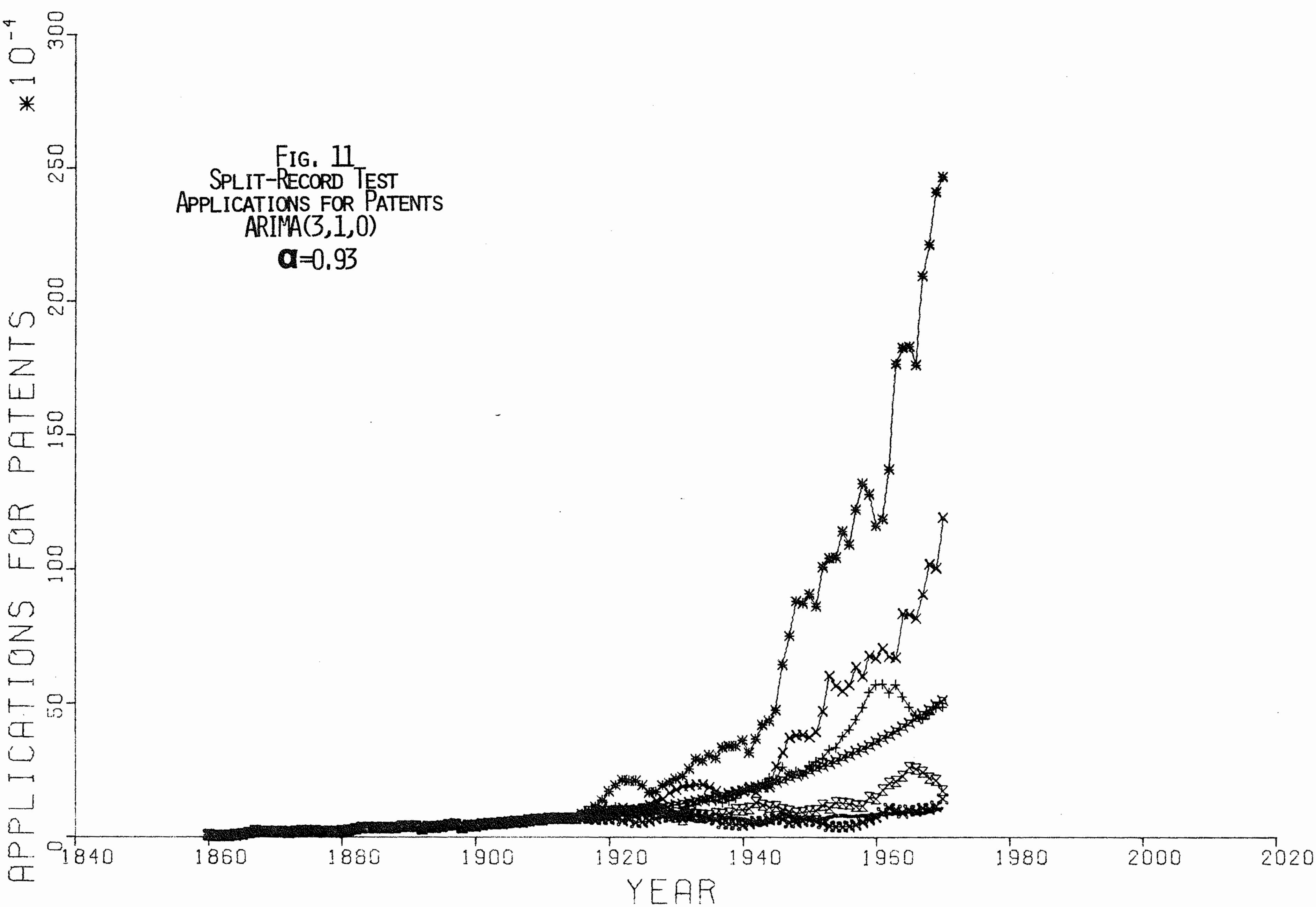


Fig. 11
Split-Record Test
Applications for Patents
ARIMA(3,1,0)
α=0.93

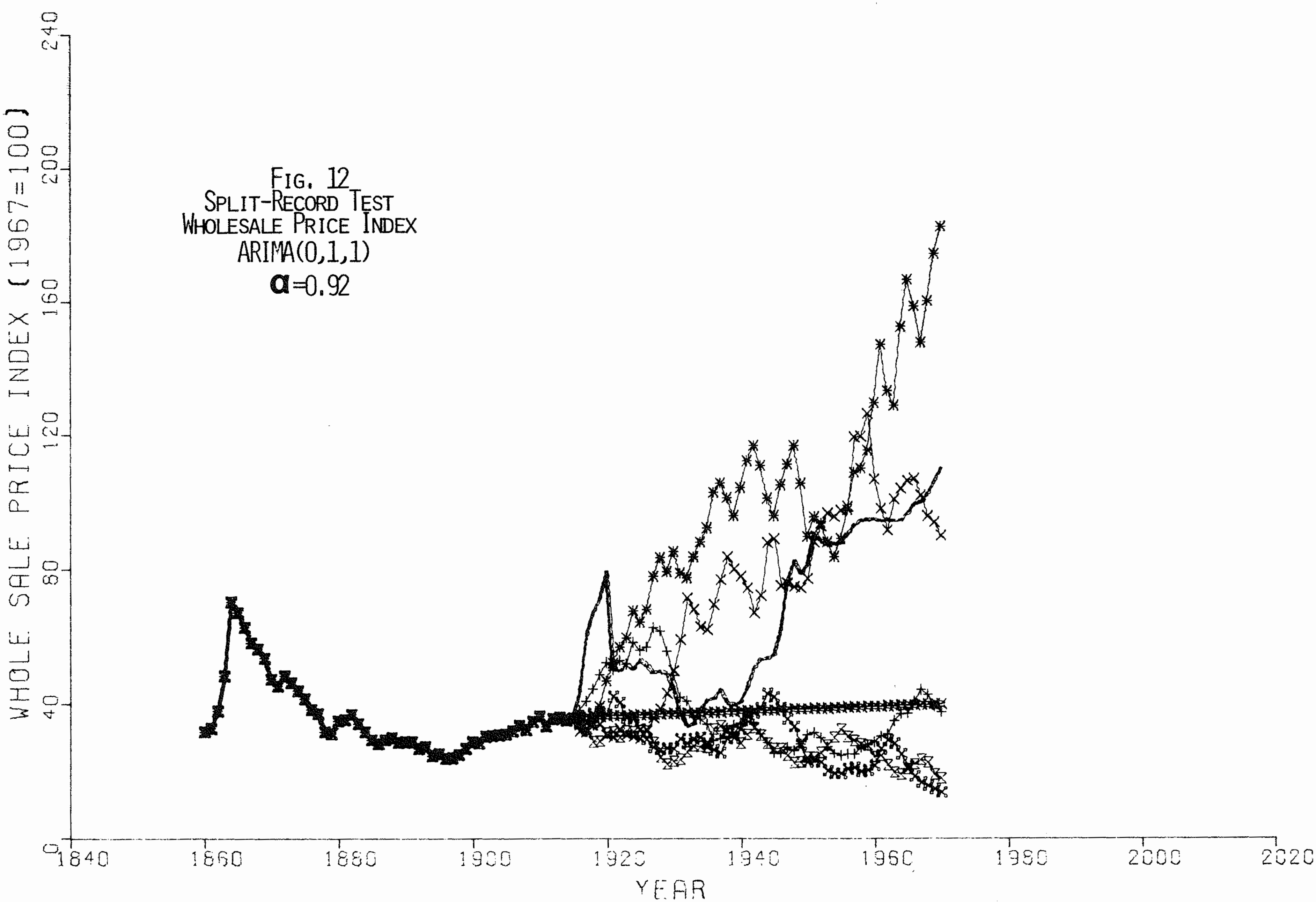


Fig. 12
Split-Record Test
Wholesale Price Index
ARIMA(0,1,1)
α=0.92

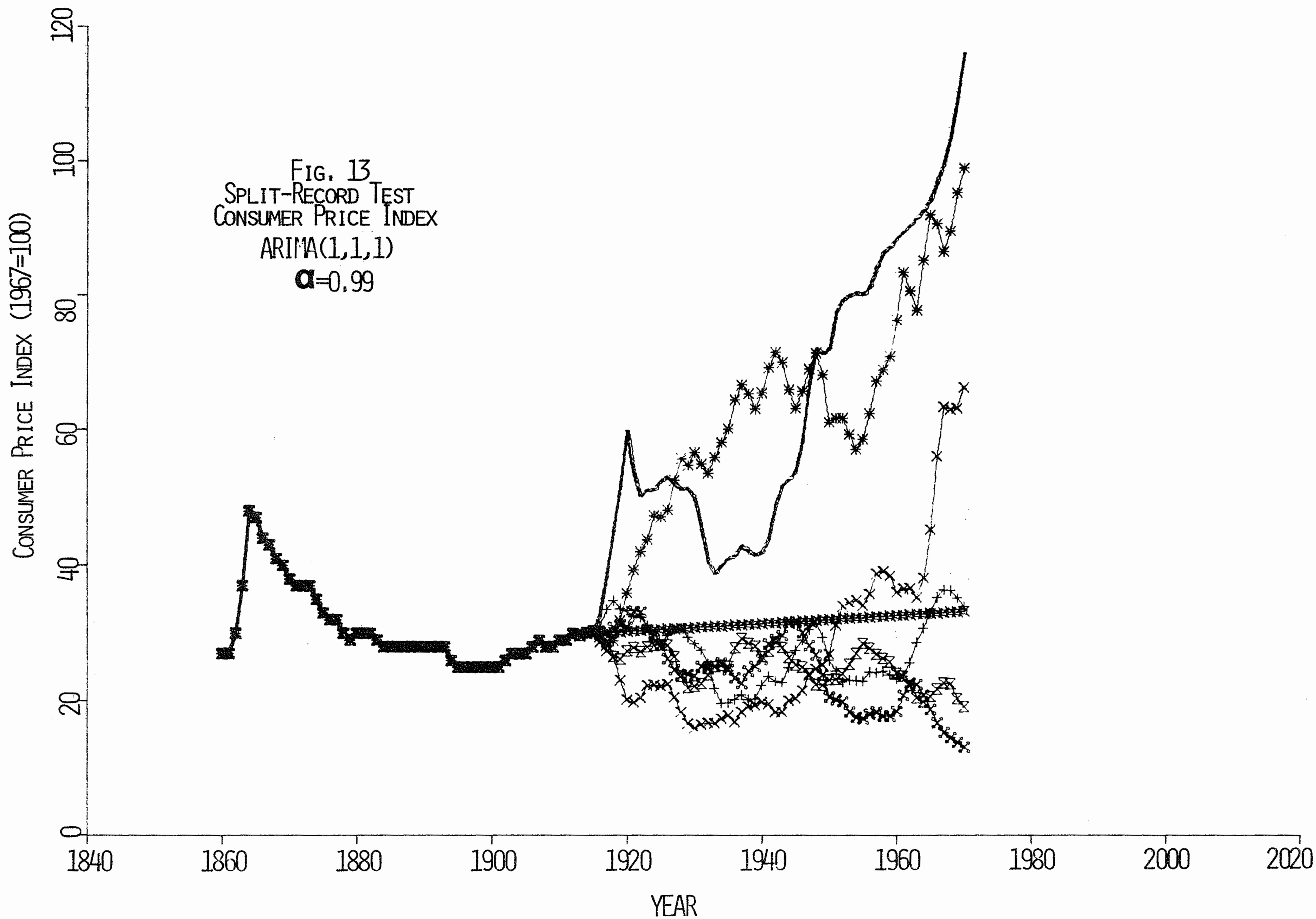


Fig. 13
Split-Record Test
Consumer Price Index
ARIMA(1,1,1)
α=0.99
Consumer Price Index (1967=100)
0
20
40
60
80
100
120
1840
1860
1880
1900
1920
1940
1960
1980
2000
2020
YEAR

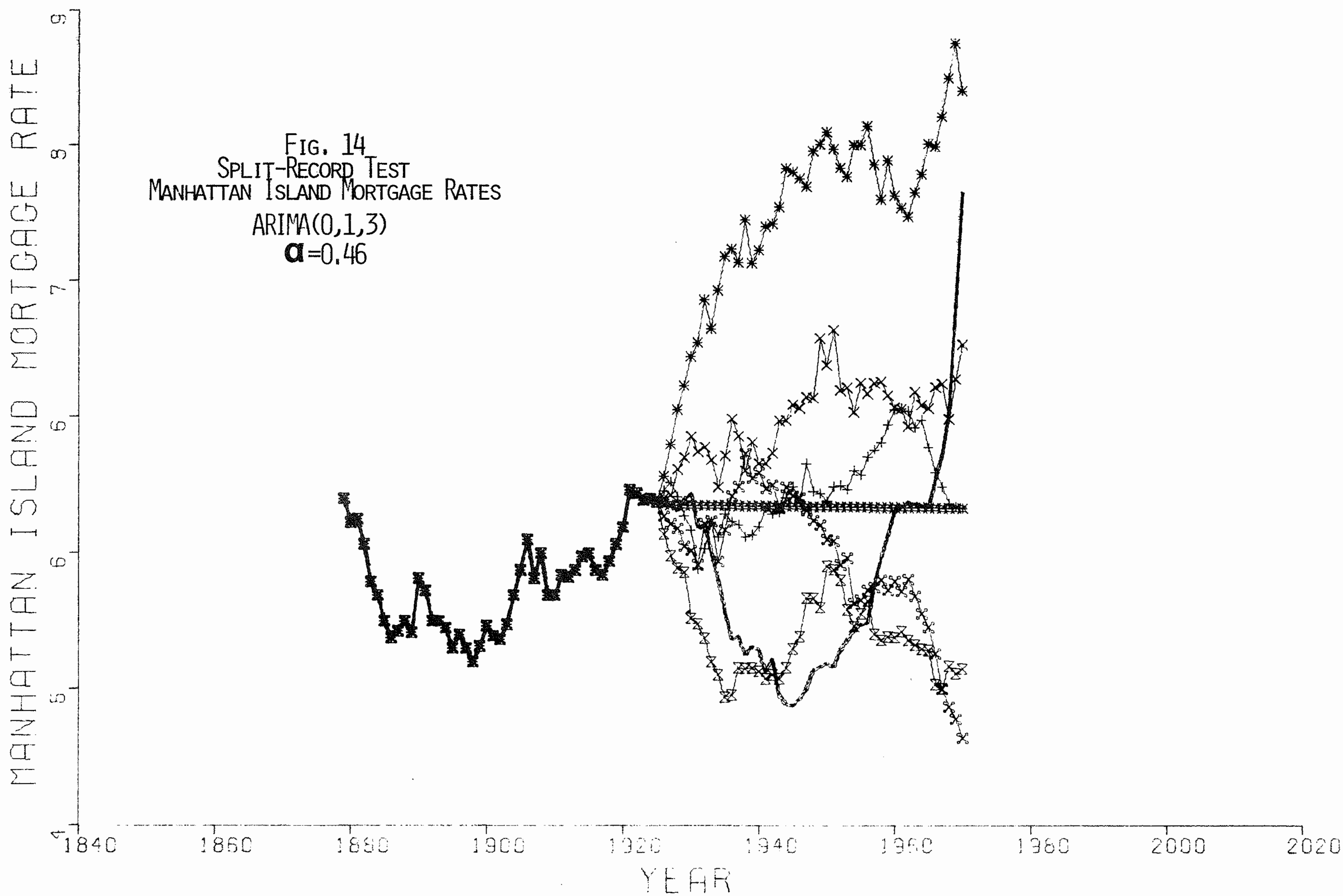


Fig. 14
Split-Record Test
Manhattan Island Mortgage Rates
ARIMA(0,1,3)
α=0.46
MANHATTAN ISLAND MORTGAGE RATE
4
5
6
7
8
9
1840
1860
1880
1900
1920
1940
1960
1980
2000
2020
YEAR

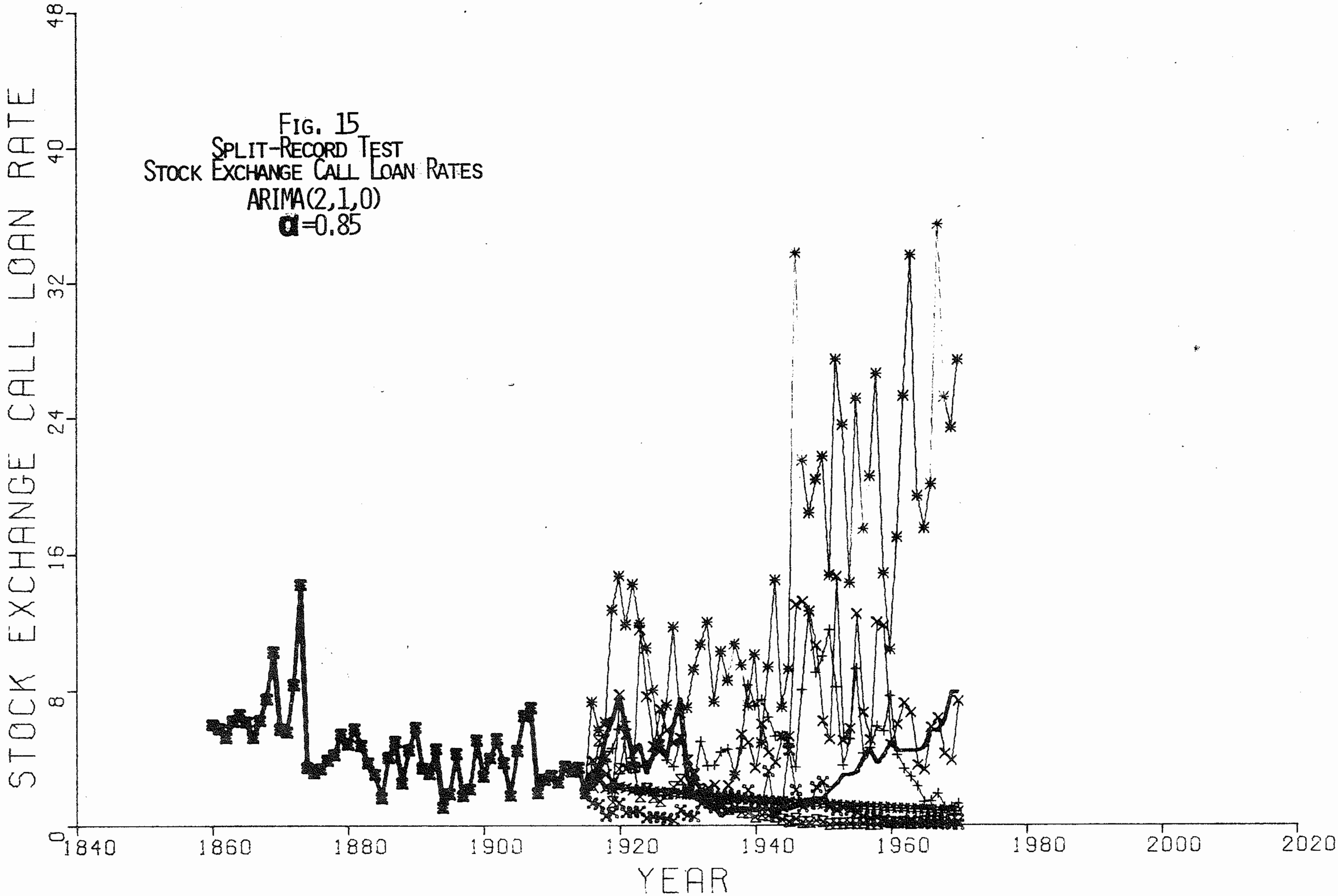
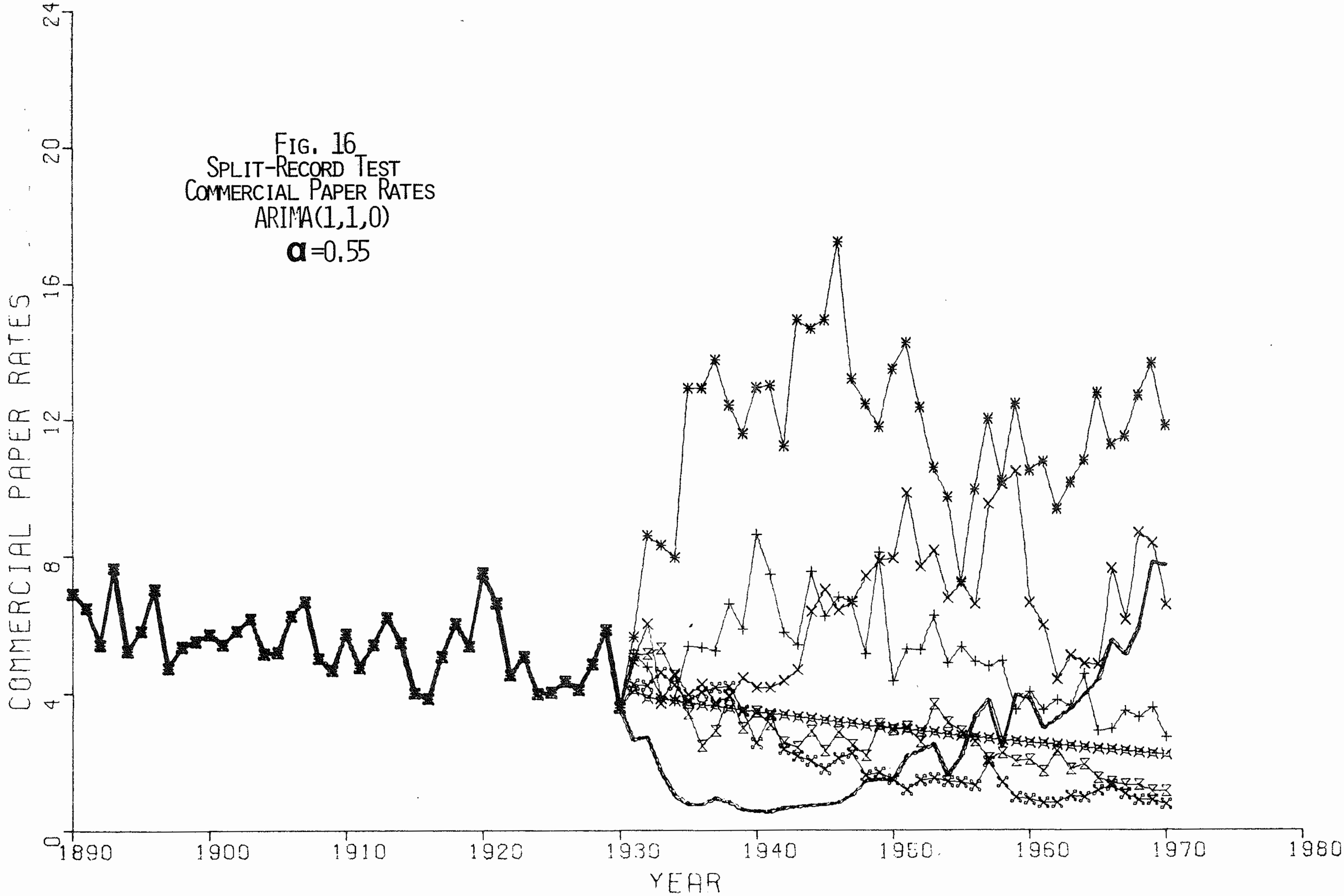


Fig. 15
Split-Record Test
Stock Exchange Call Loan Rates
ARIMA(2,1,0)
α=0.85

Fig. 16
Split-Record Test
Commercial Paper Rates
ARIMA(1,1,0)
α=0.55



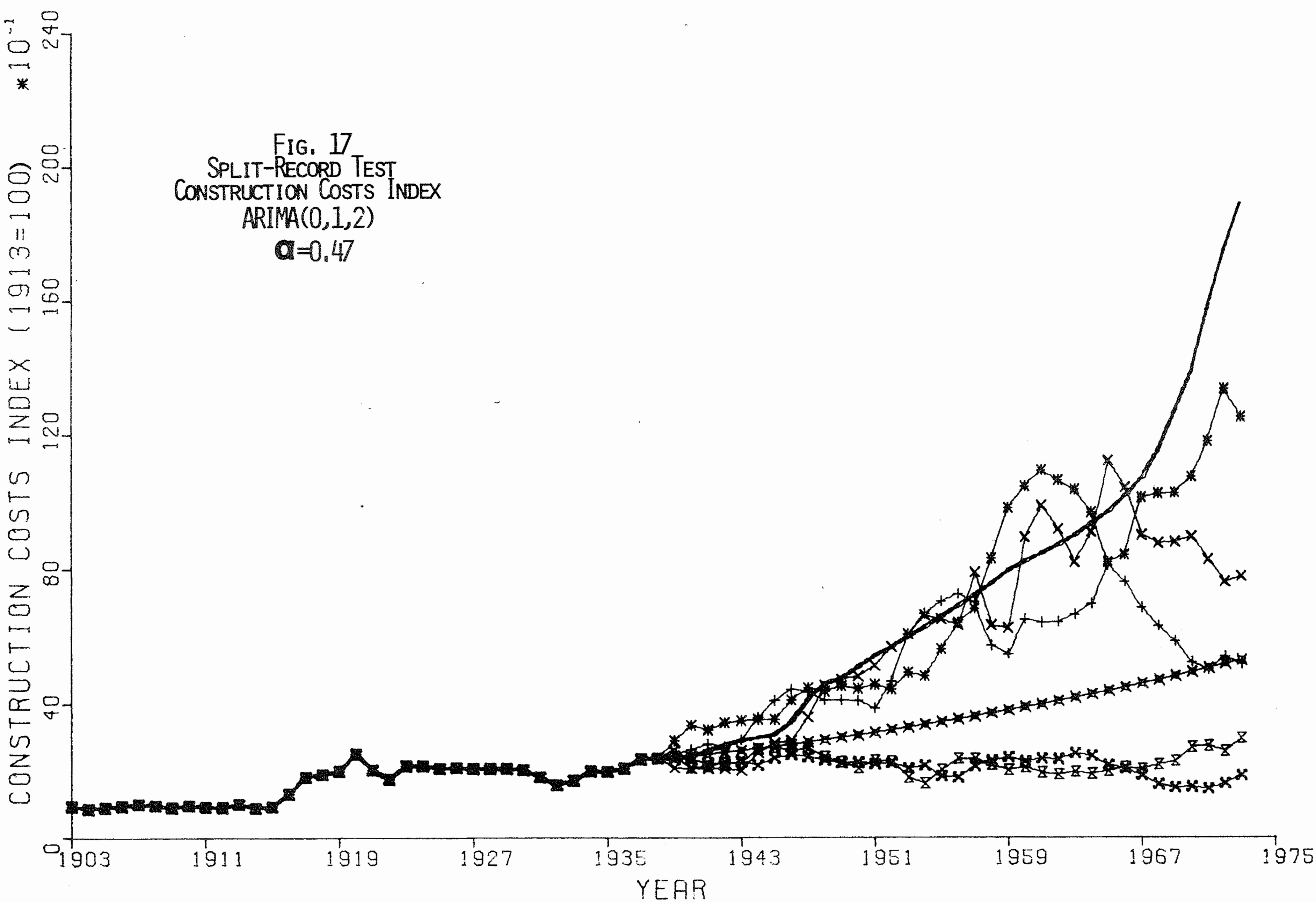


Fig. 17
Split-Record Test
Construction Costs Index
ARIMA(0,1,2)
α=0.47

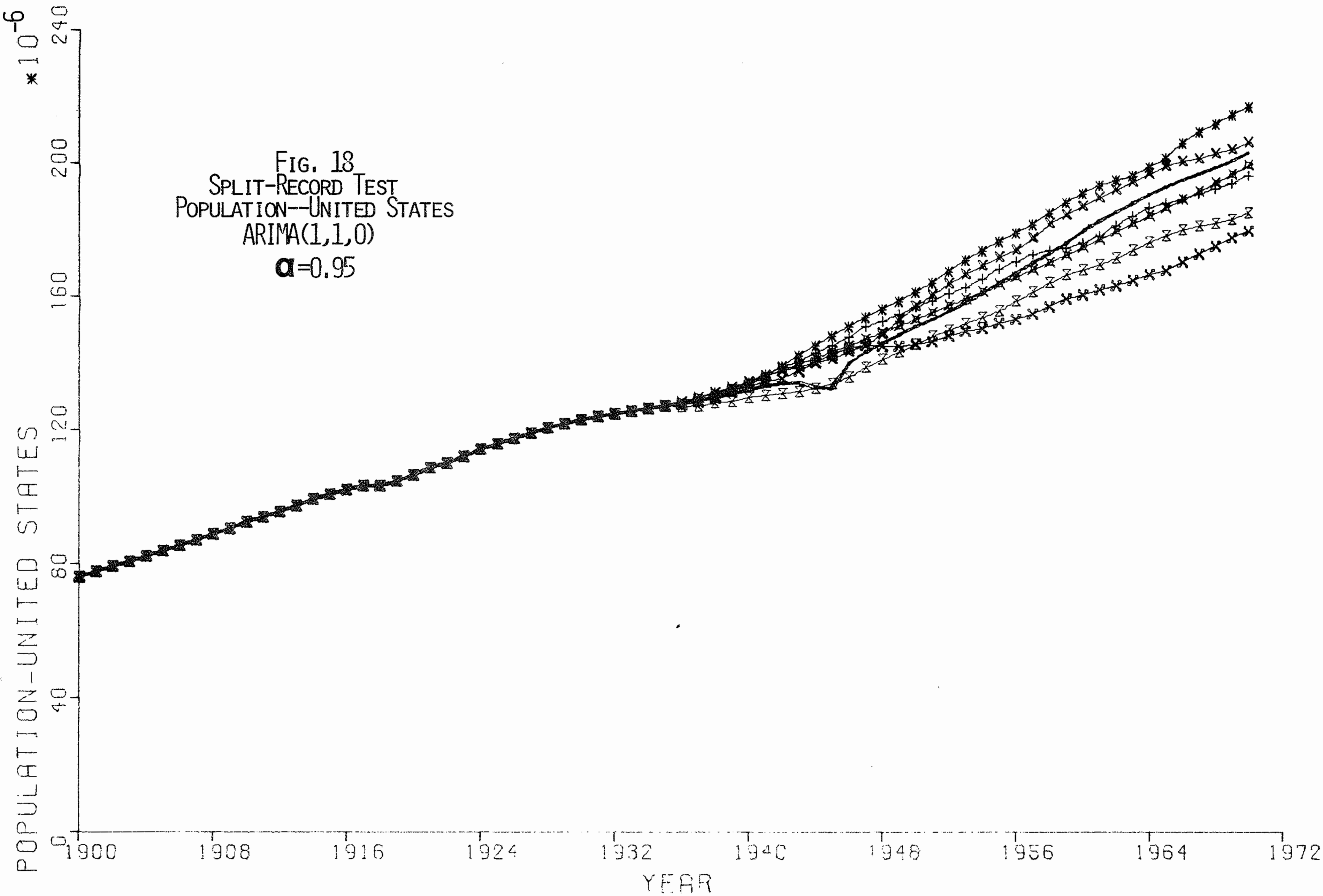


Fig. 18
Split-Record Test
Population--United States
ARIMA(1,1,0)
α=0.95

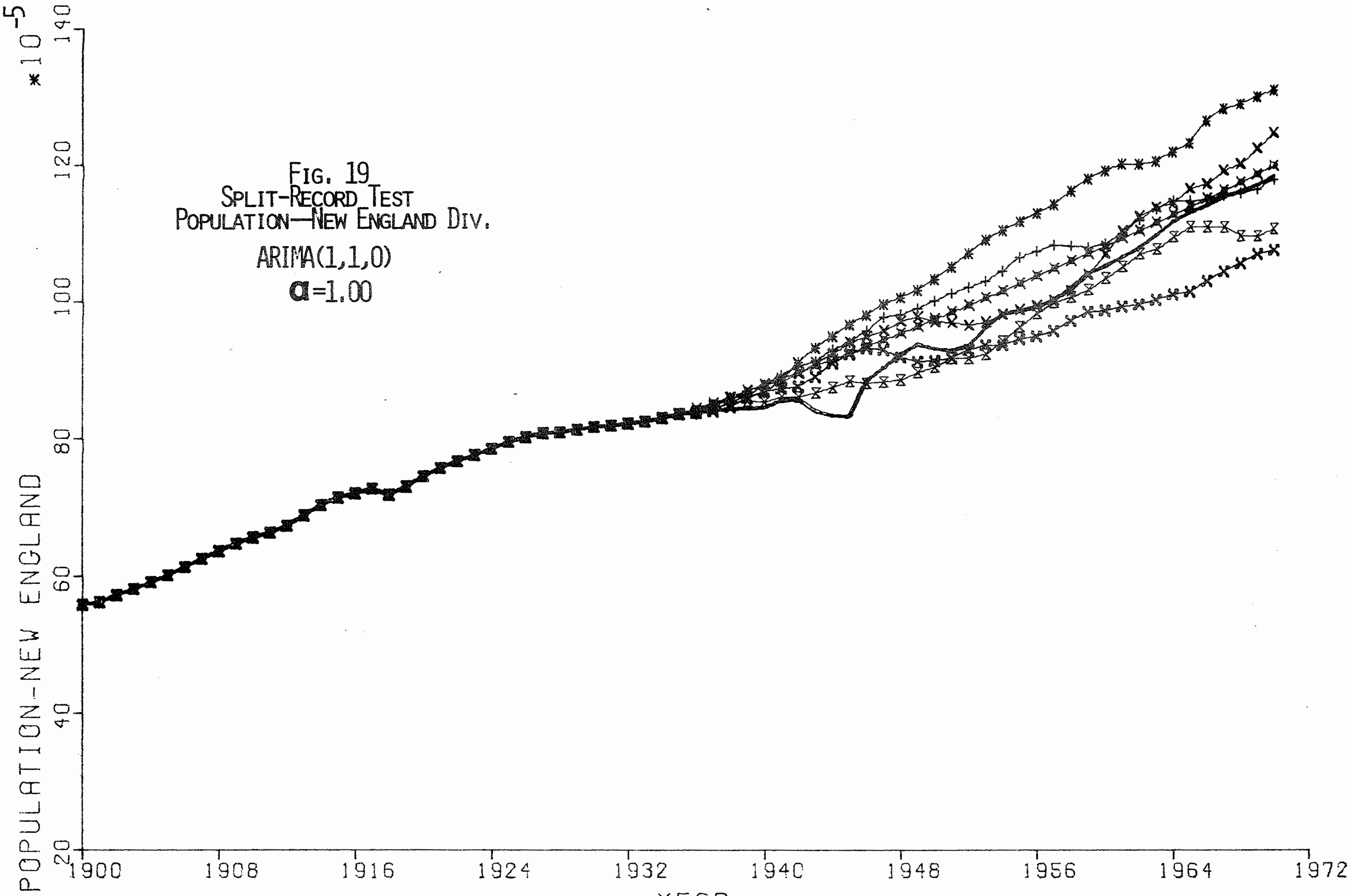
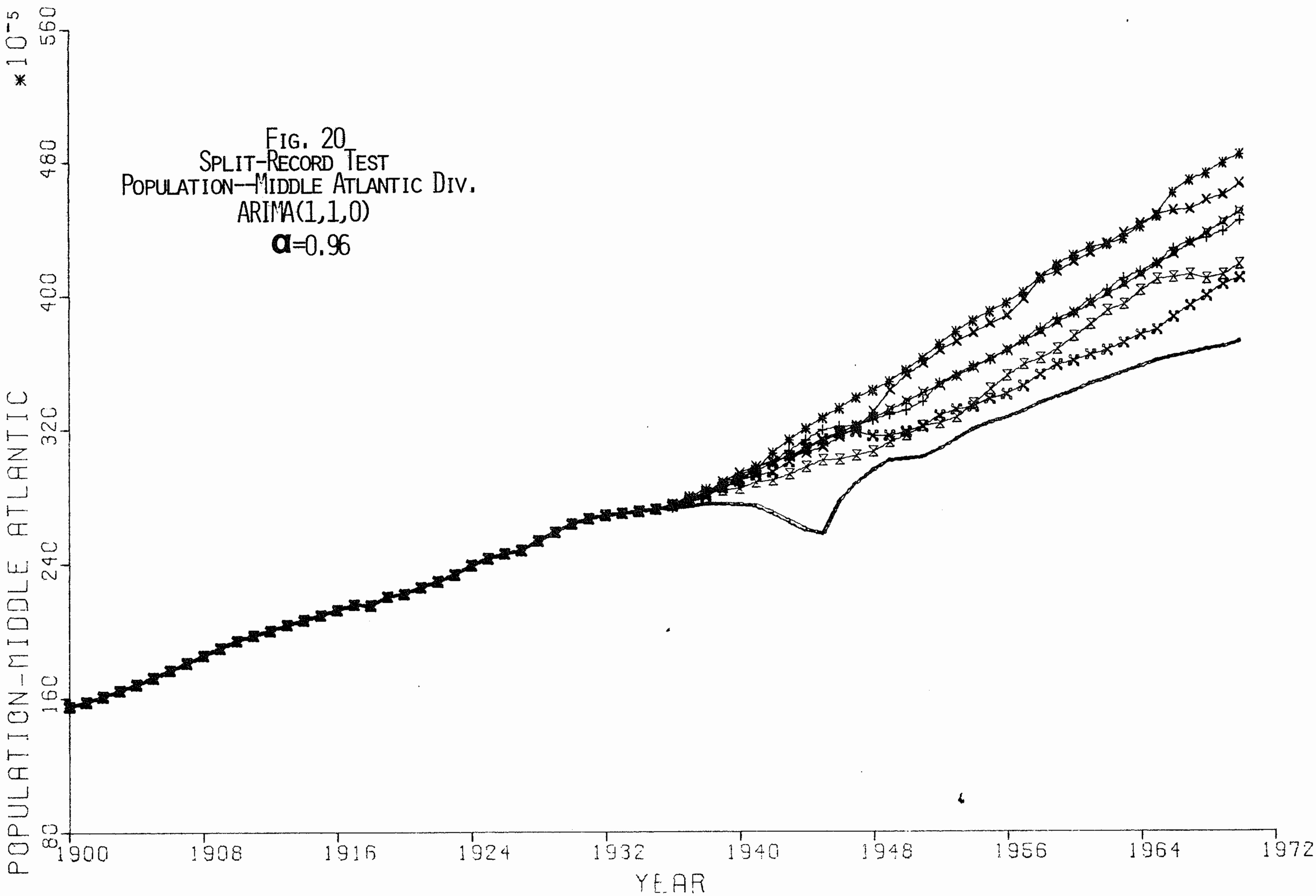


Fig. 19
Split-Record Test
Population—New England Div.
ARIMA(1,1,0)
α=1.00
POPULATION-NEW ENGLAND
*10^-5
20
40
60
80
100
120
140
1900
1908
1916
1924
1932
1940
1948
1956
1964
1972
YEAR

Fig. 20
Split-Record Test
Population--Middle Atlantic Div.
ARIMA(1,1,0)
α=0.96



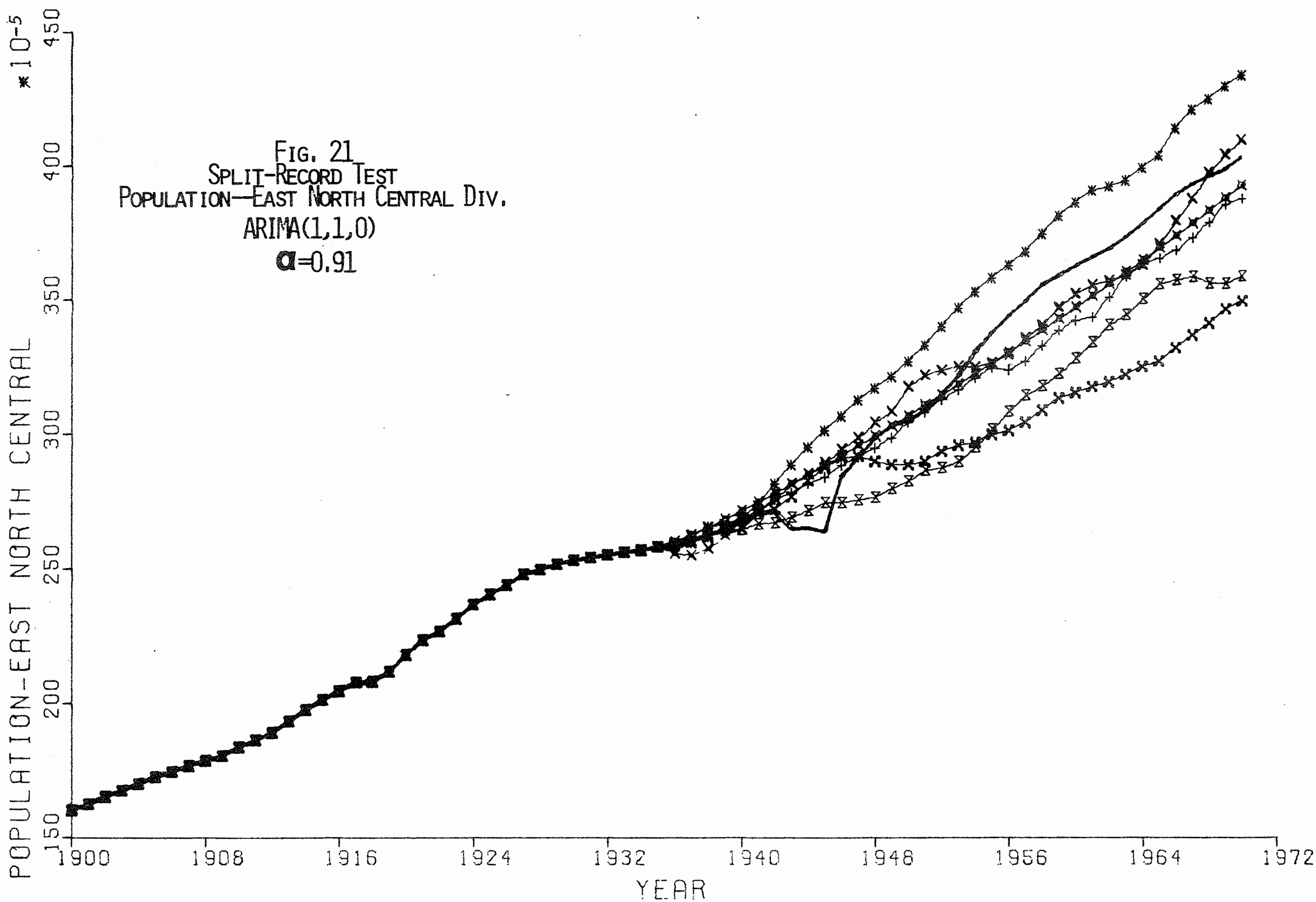


Fig. 21
Split-Record Test
Population—East North Central Div.
ARIMA(1,1,0)
α=0.91
POPULATION-EAST NORTH CENTRAL
*10-5
150
200
250
300
350
400
450
1900
1908
1916
1924
1932
1940
1948
1956
1964
1972
YEAR

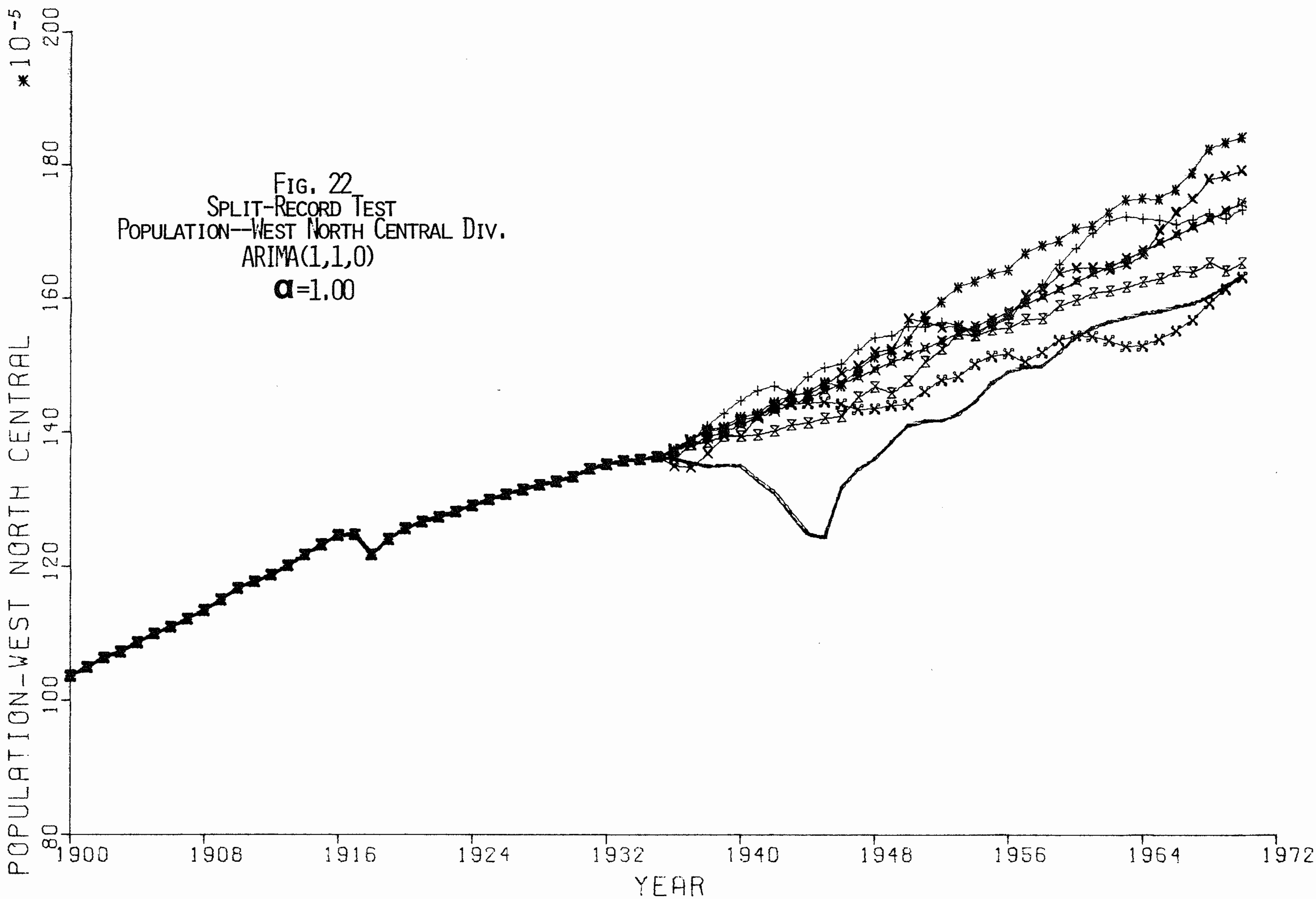


FIG. 22
SPLIT-RECORD TEST
POPULATION--WEST NORTH CENTRAL DIV.
ARIMA(1,1,0)
α=1.00

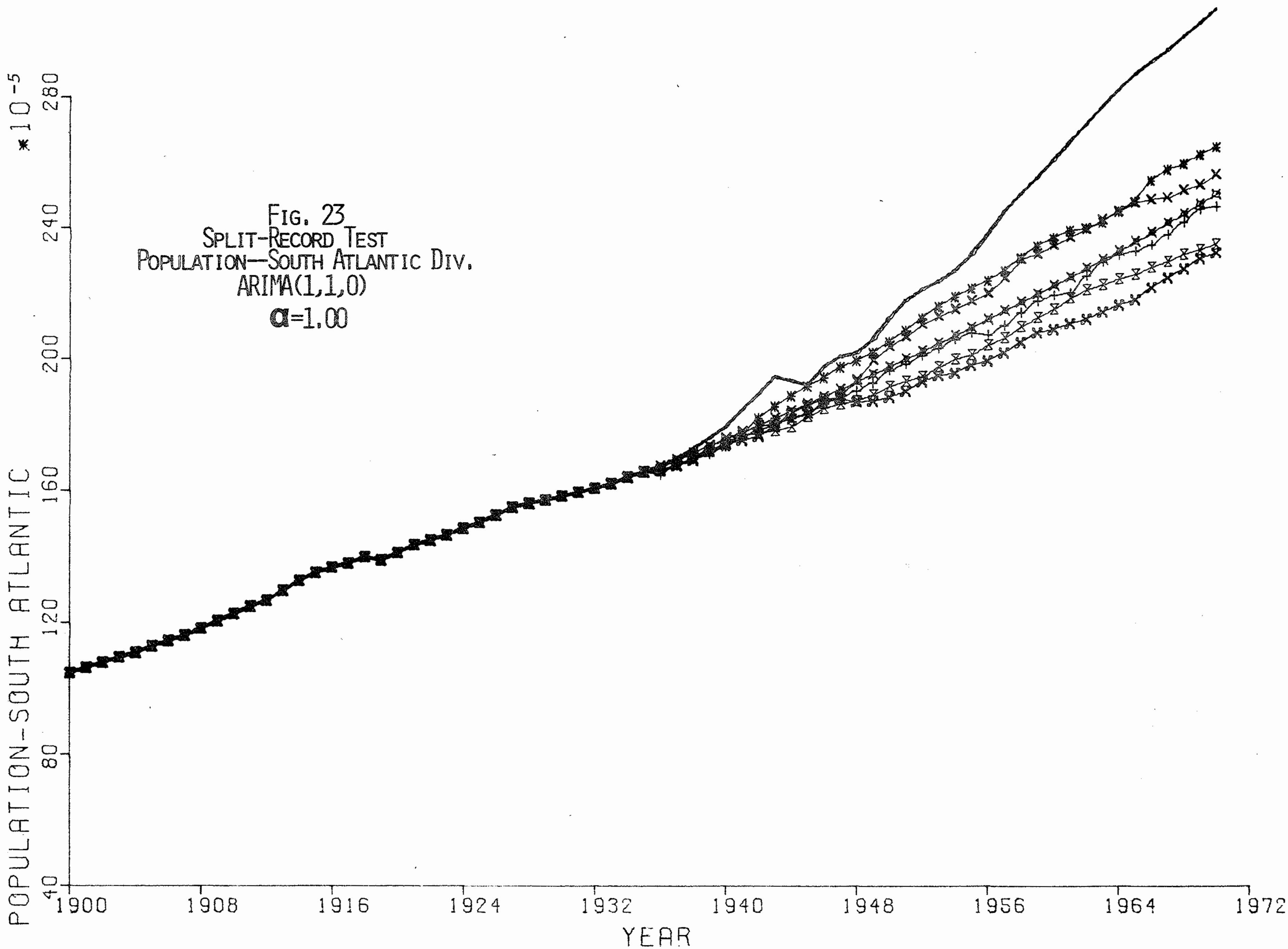
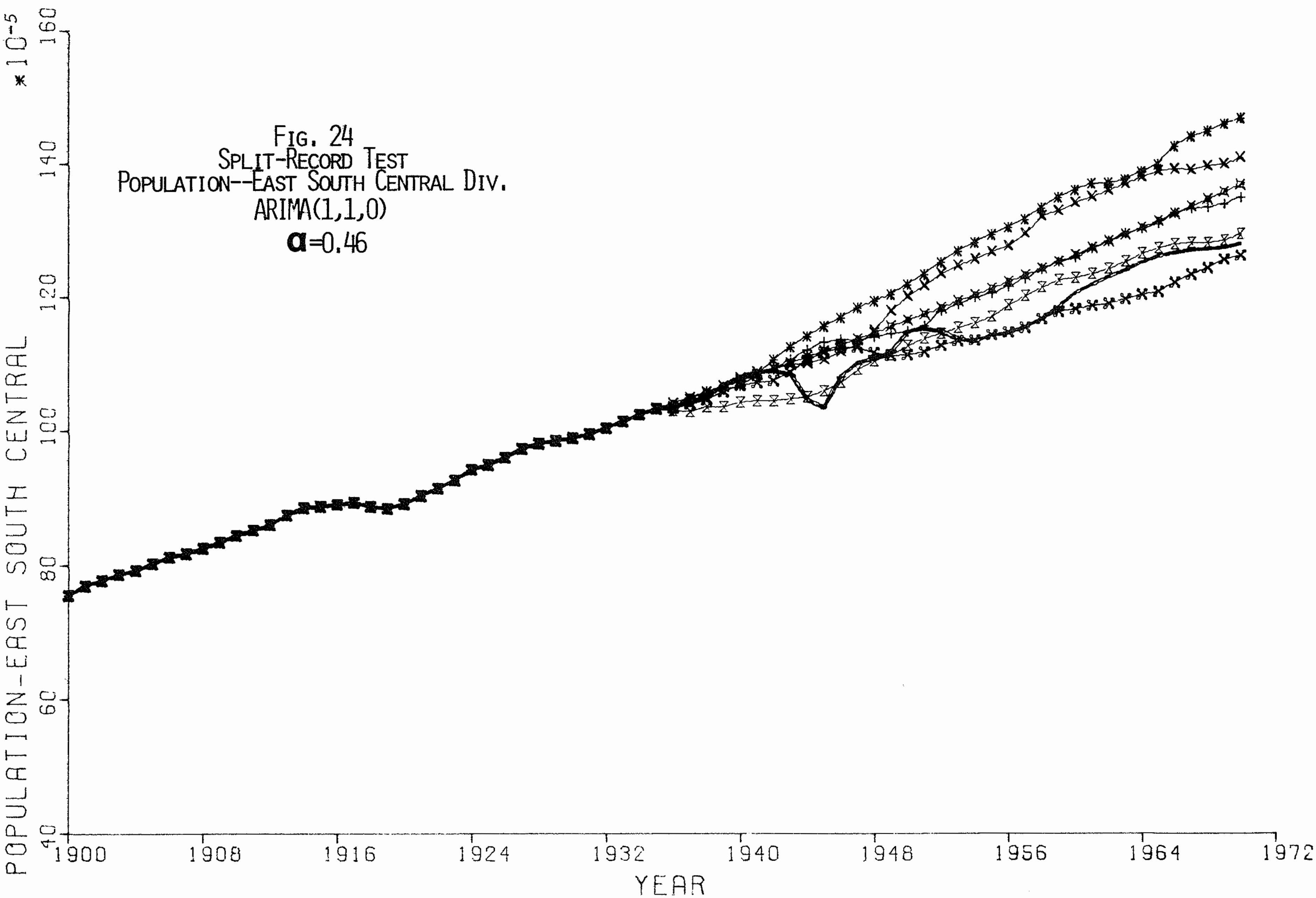


Fig. 23
Split-Record Test
Population—South Atlantic Div.
ARIMA(1,1,0)
α=1.00

Fig. 24
Split-Record Test
Population--East South Central Div.
ARIMA(1,1,0)
α=0.46



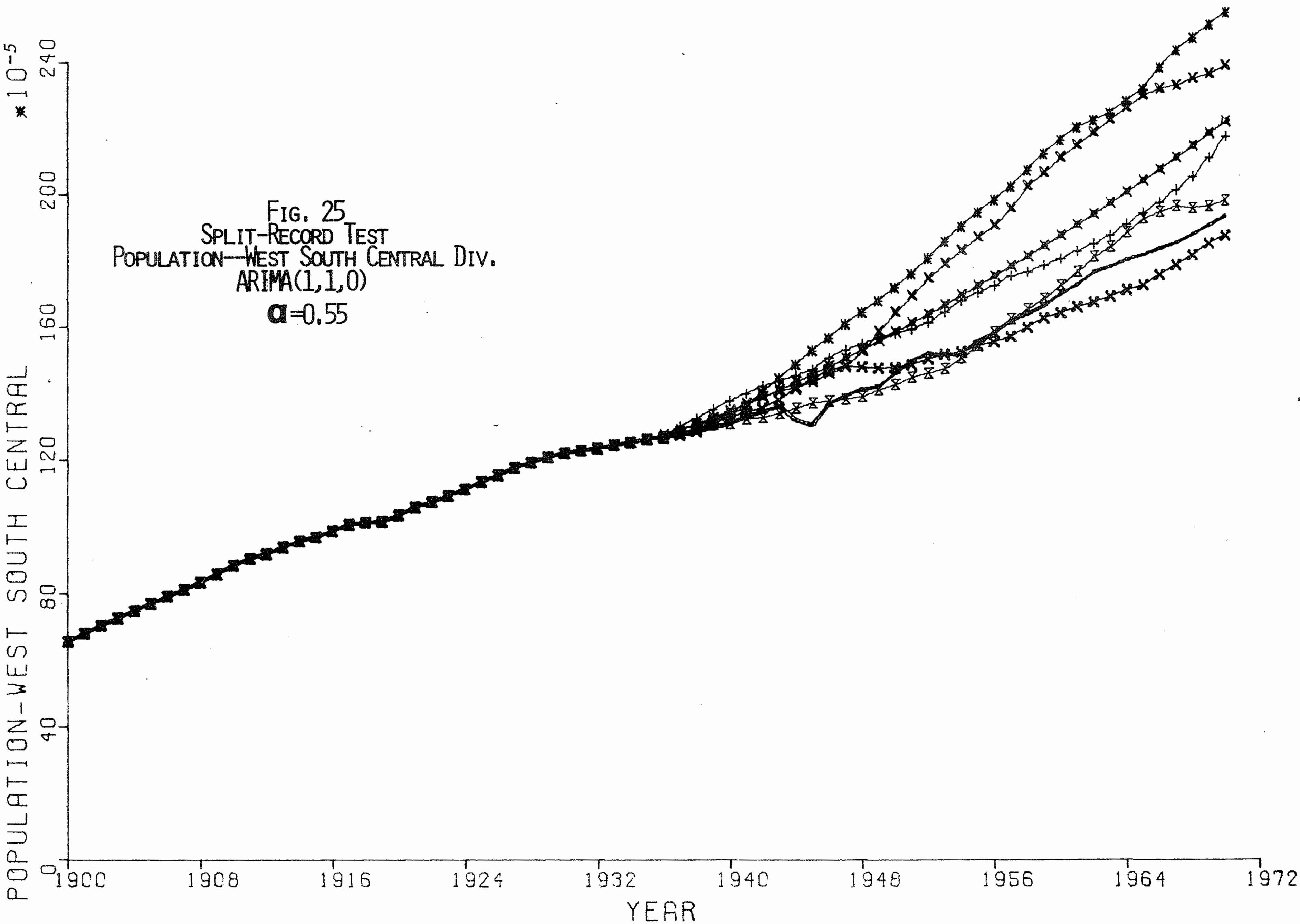


FIG. 25
SPLIT-RECORD TEST
POPULATION—WEST SOUTH CENTRAL DIV.
ARIMA(1,1,0)
α=0.55

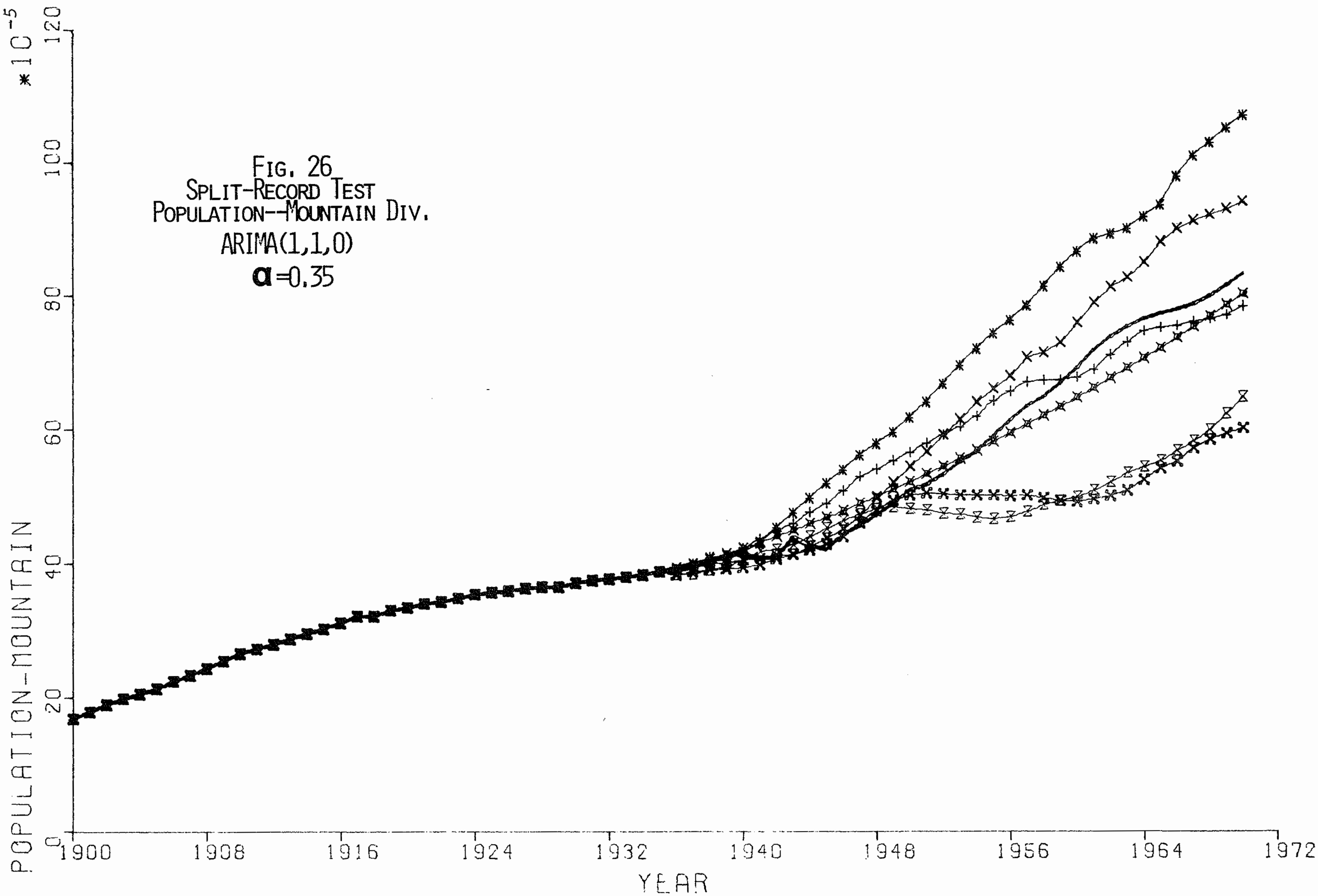


Fig. 26
Split-Record Test
Population--Mountain Div.
ARIMA(1,1,0)
α=0.35
*10^-5
120
100
80
60
40
20
0
POPULATION-MOUNTAIN
1900
1908
1916
1924
1932
1940
1948
1956
1964
1972
YEAR

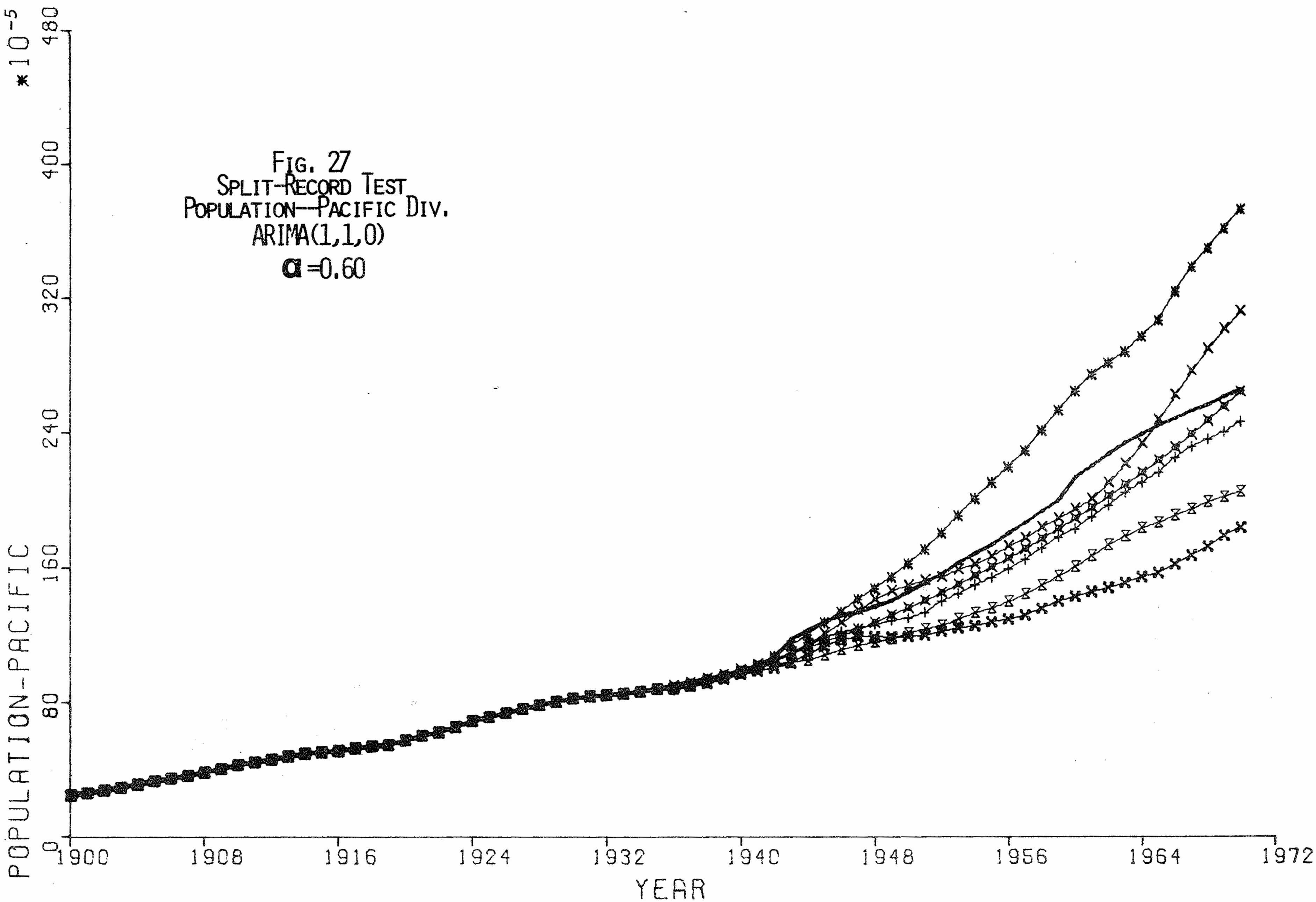


Fig. 27
Split-Record Test
Population--Pacific Div.
ARIMA(1,1,0)
α=0.60
*10^-5
480
400
320
240
160
80
0
1900
1908
1916
1924
1932
1940
1948
1956
1964
1972
POPULATION-PACIFIC
YEAR

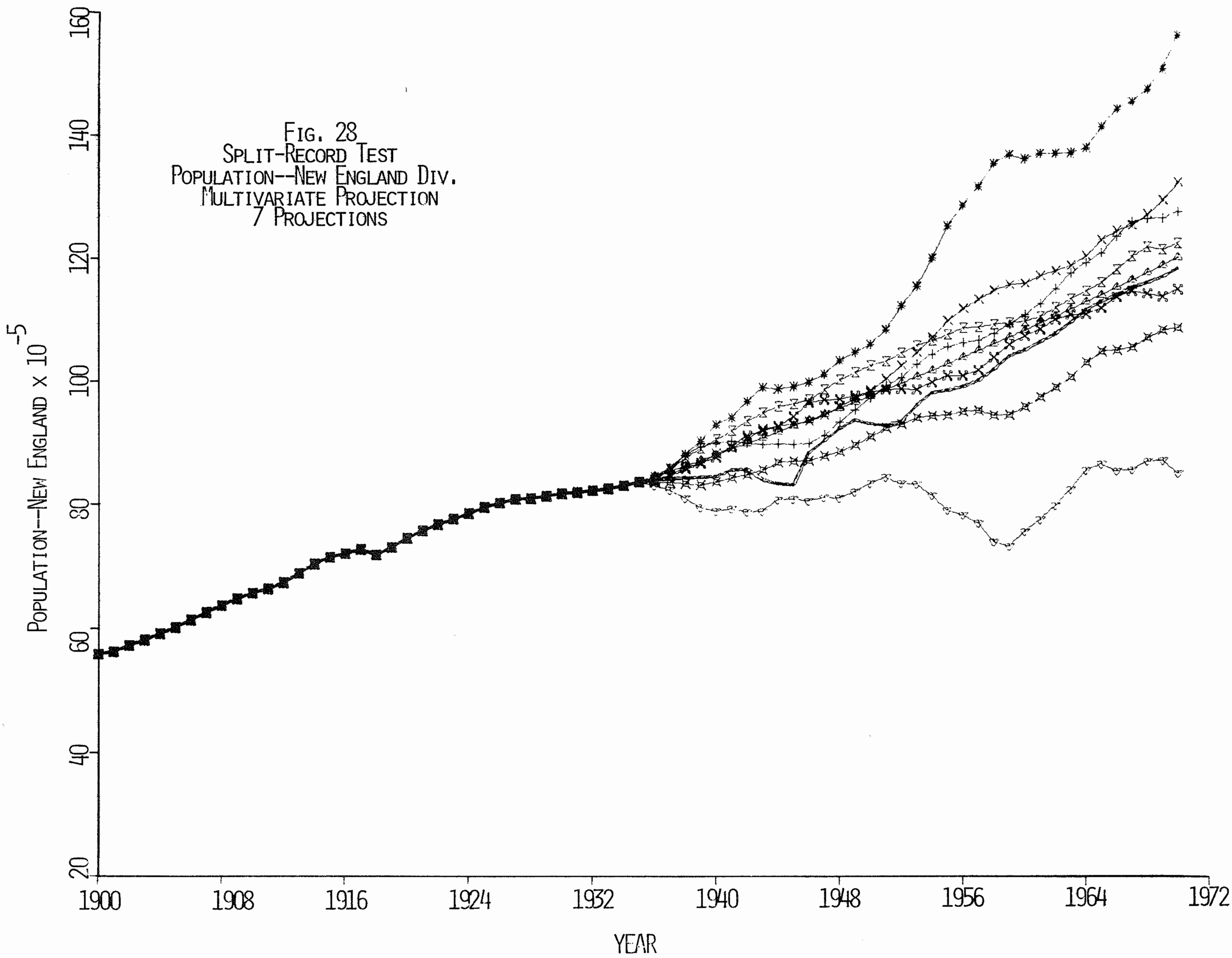


Fig. 28
Split-Record Test
Population--New England Div.
Multivariate Projection
7 Projections
Population--New England x 10^-5
160
140
120
100
80
60
40
20
1900
1908
1916
1924
1932
1940
1948
1956
1964
1972
YEAR

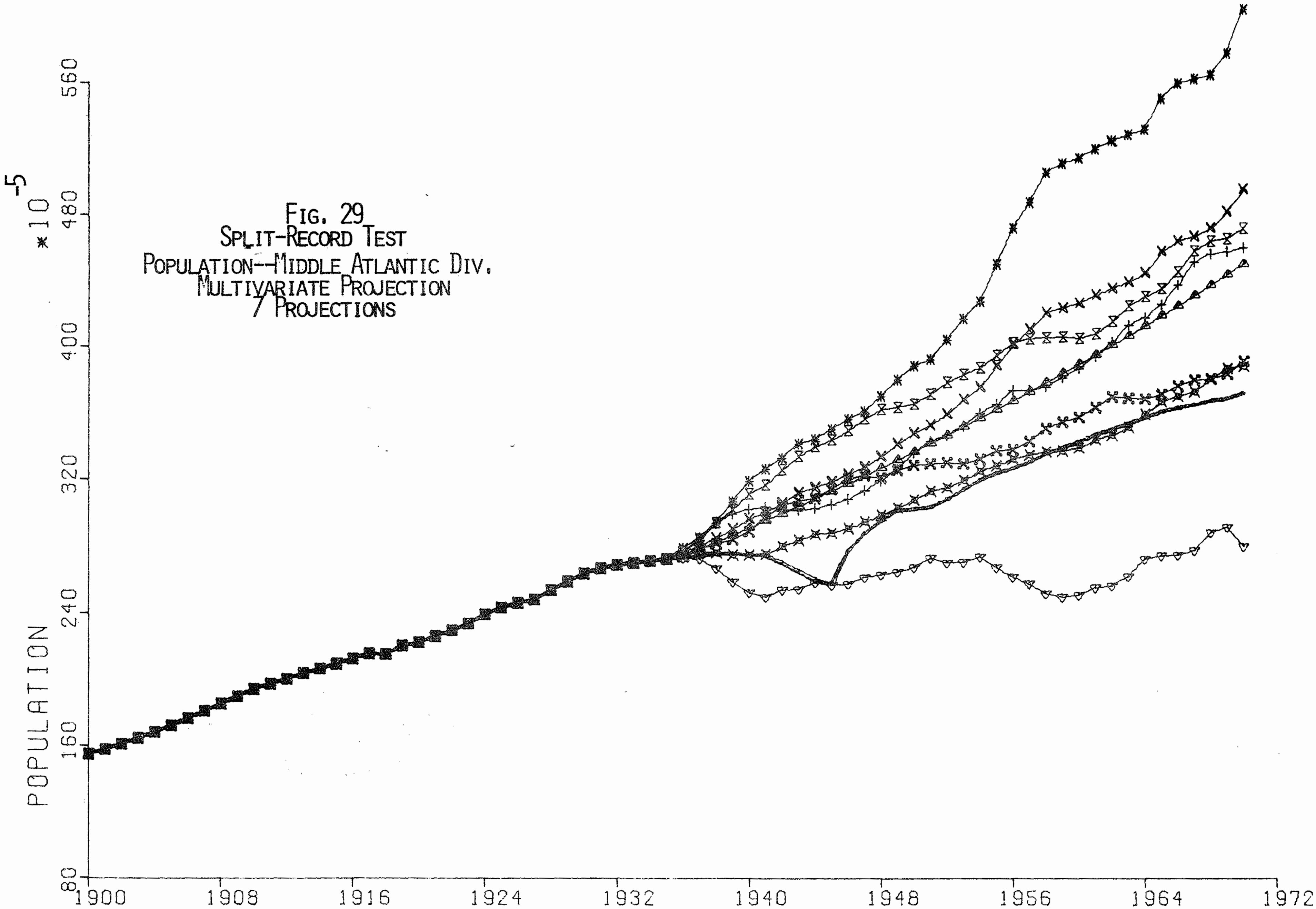


Fig. 29
Split-Record Test
Population--Middle Atlantic Div.
Multivariate Projection
7 Projections
POPULATION
*10^-5
80
160
240
320
400
480
560
1900
1908
1916
1924
1932
1940
1948
1956
1964
1972

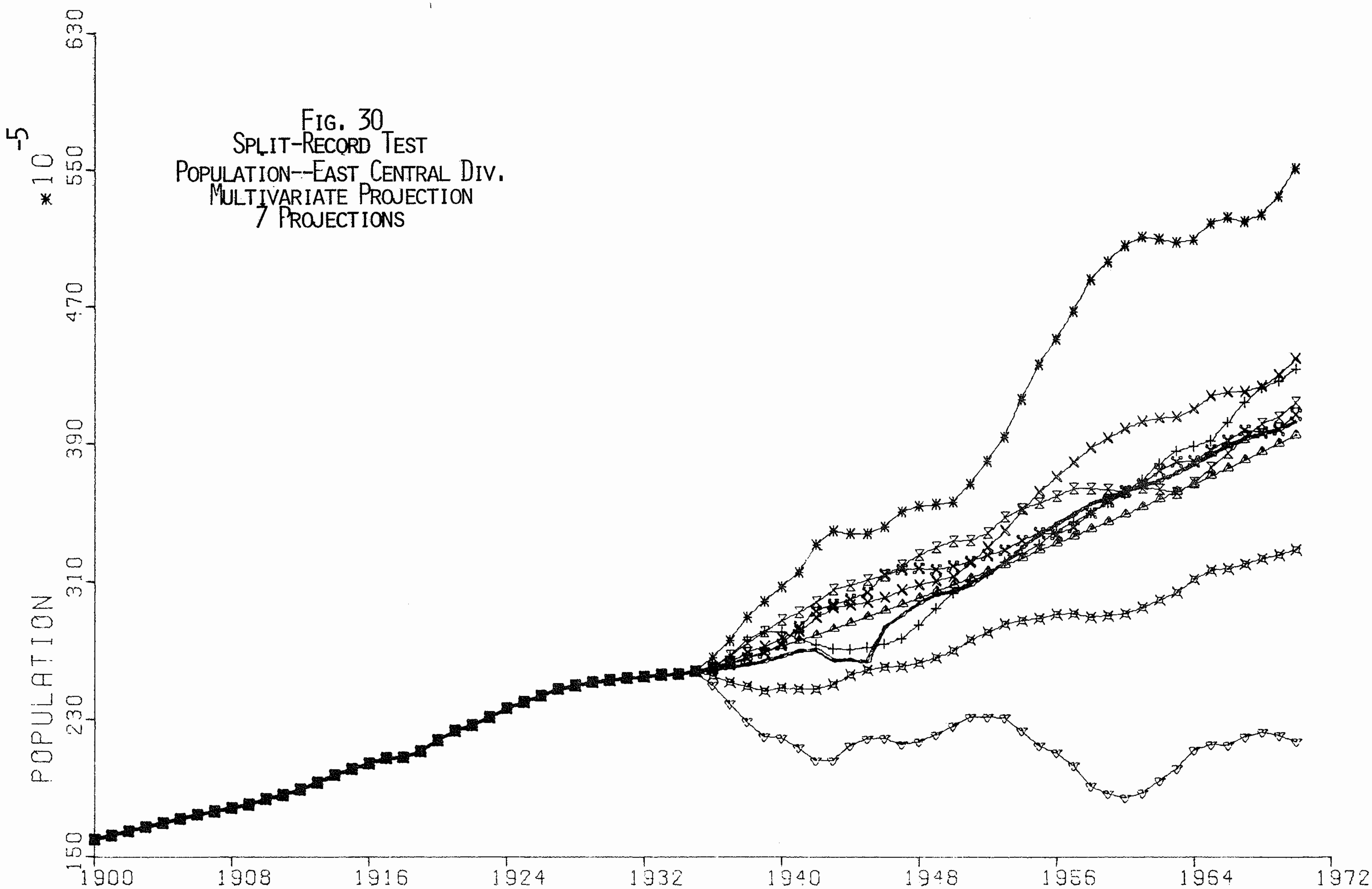


FIG. 30
SPLIT-RECORD TEST
POPULATION--EAST CENTRAL DIV.
MULTIVARIATE PROJECTION
7 PROJECTIONS

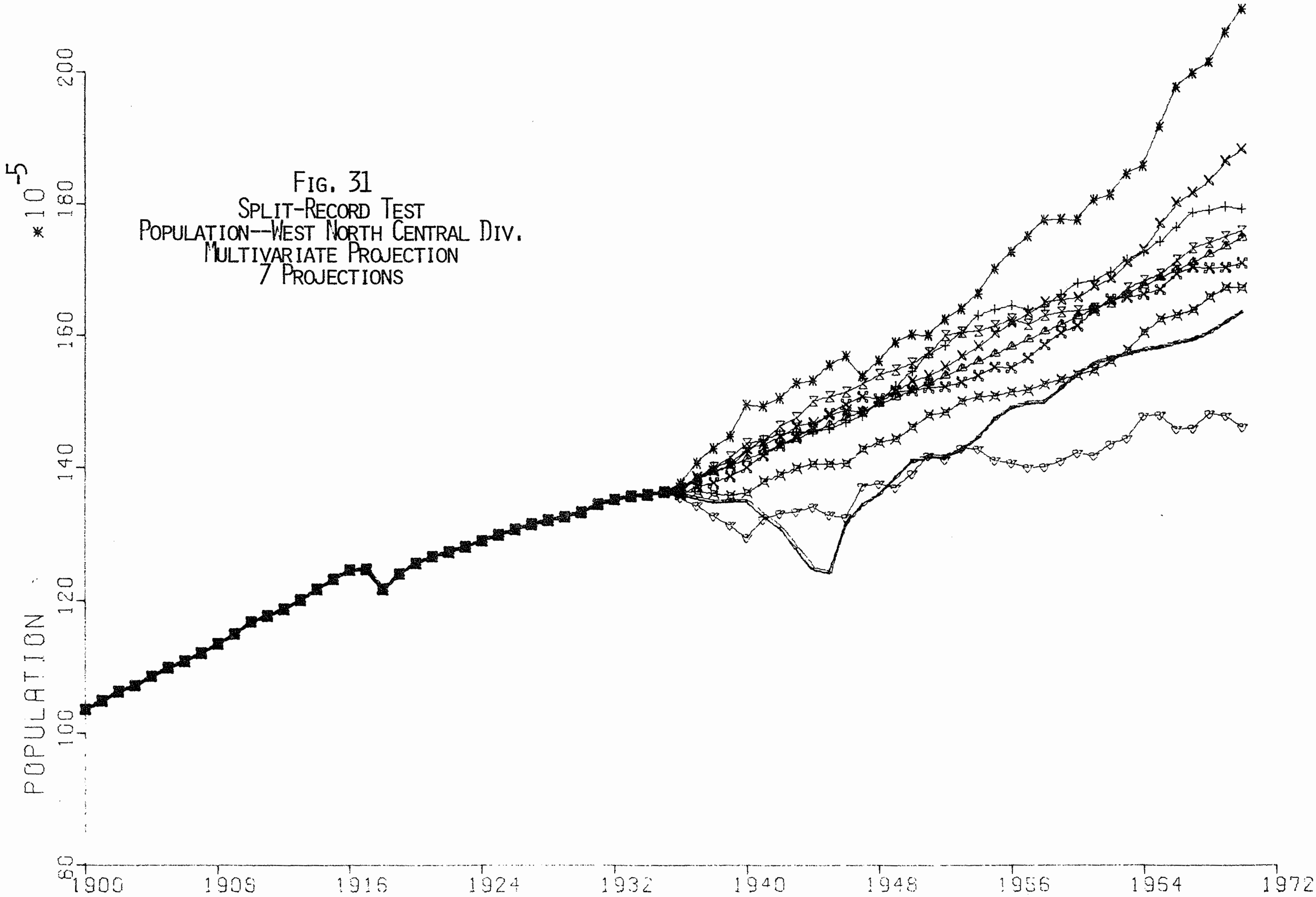


FIG. 31
SPLIT-RECORD TEST
POPULATION--WEST NORTH CENTRAL DIV.
MULTIVARIATE PROJECTION
7 PROJECTIONS
POPULATION
*10^-5
80
100
120
140
160
180
200
1900
1908
1916
1924
1932
1940
1948
1956
1964
1972

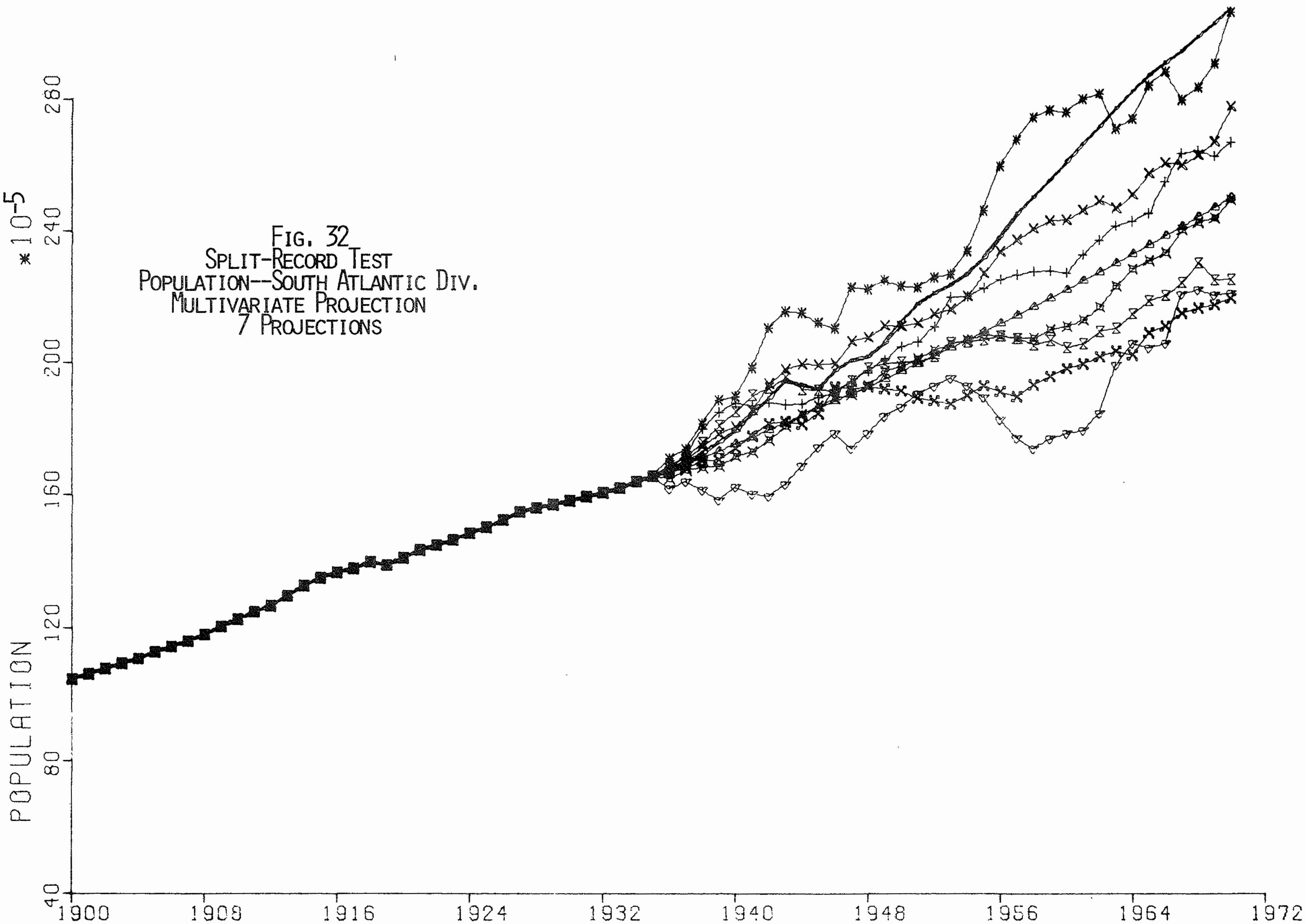


Fig. 32
Split-Record Test
Population--South Atlantic Div.
Multivariate Projection
7 Projections

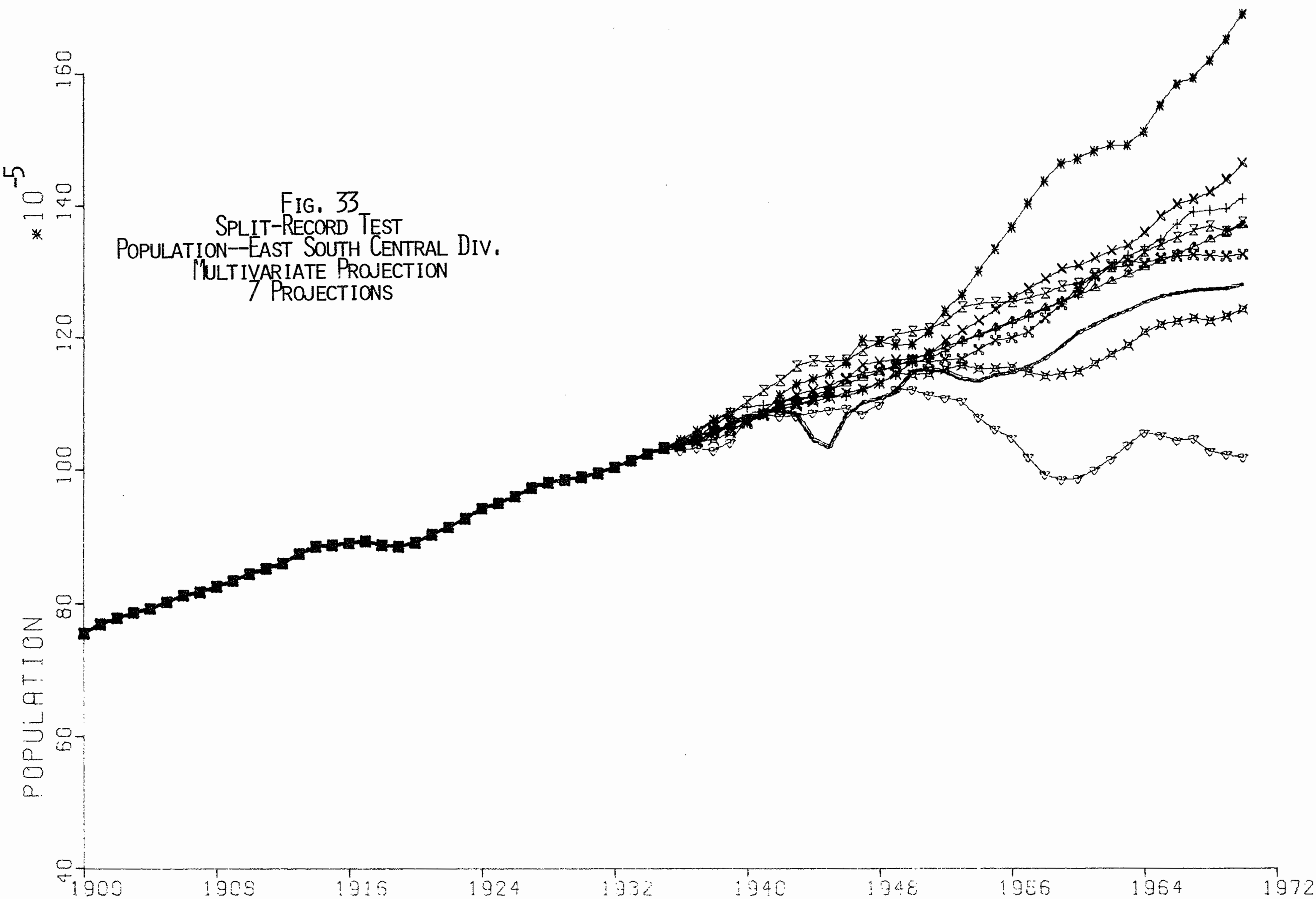


FIG. 33
SPLIT-RECORD TEST
POPULATION--EAST SOUTH CENTRAL DIV.
MULTIVARIATE PROJECTION
7 PROJECTIONS

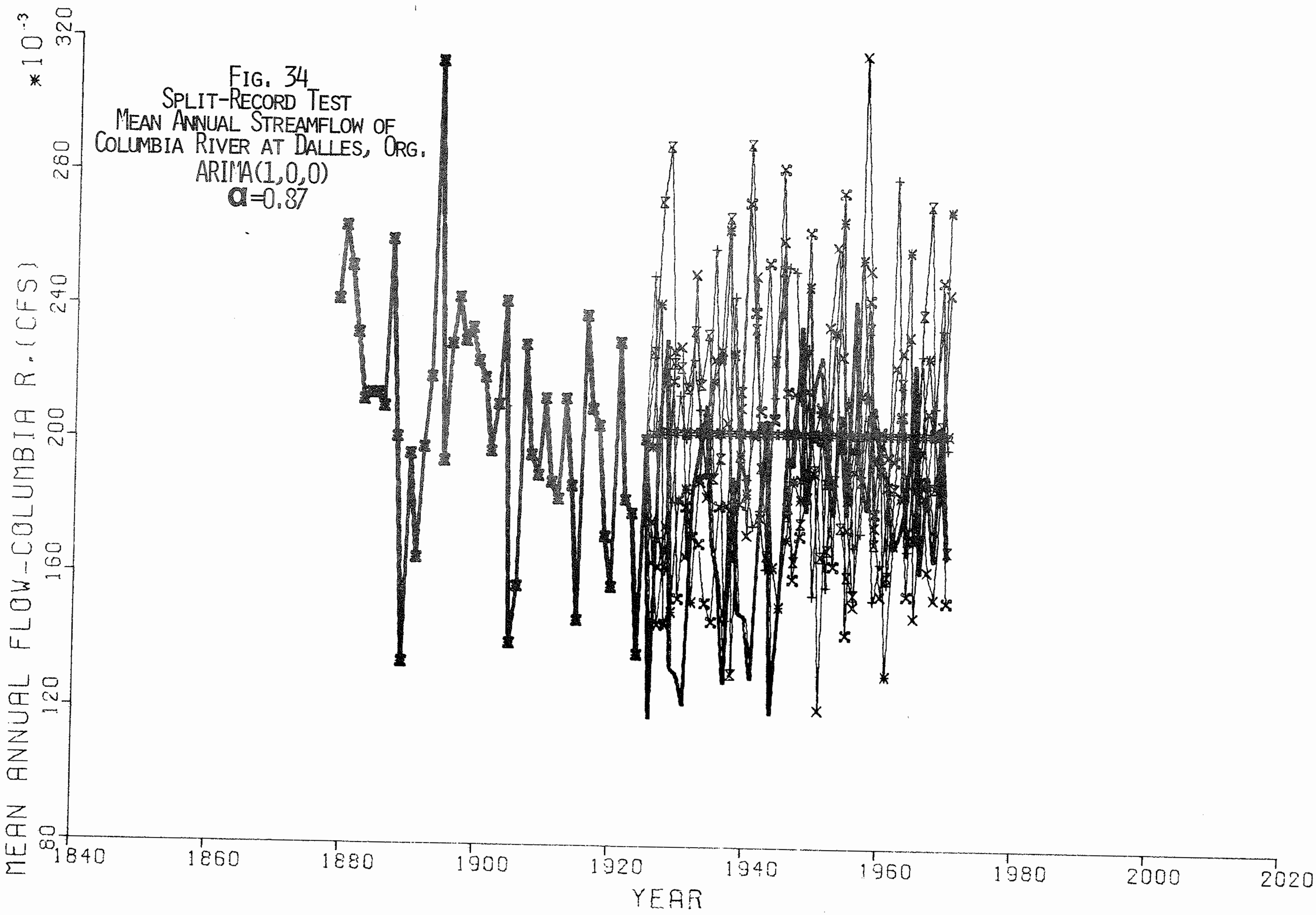
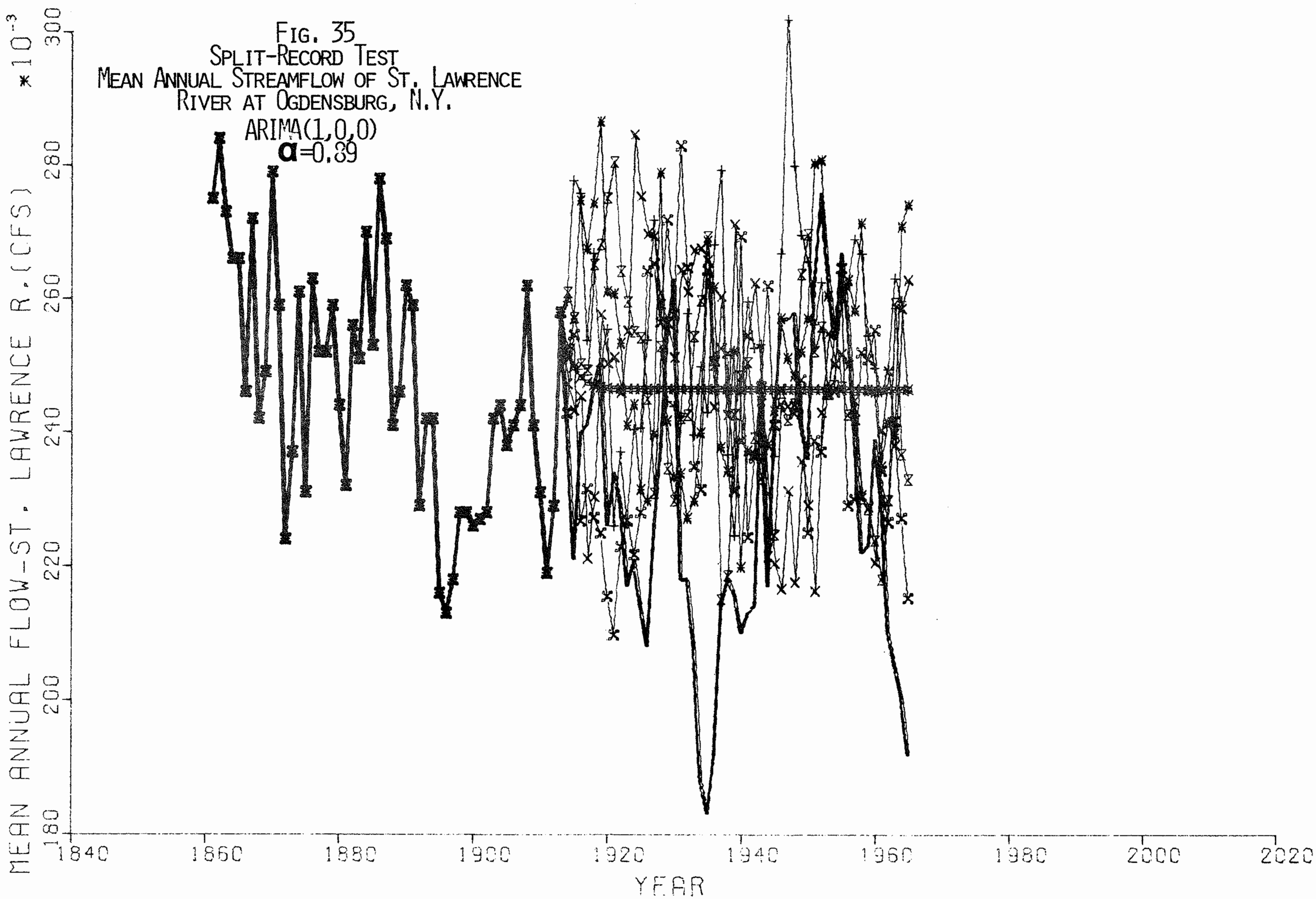


Fig. 34
Split-Record Test
Mean Annual Streamflow of
Columbia River at Dalles, Org.
ARIMA(1,0,0)
α=0.87
*10⁻³
MEAN ANNUAL FLOW-COLUMBIA R.(CFS)
80
120
160
200
240
280
320
1840
1860
1880
1900
1920
1940
1960
1980
2000
2020
YEAR

Fig. 35
Split-Record Test
Mean Annual Streamflow of St. Lawrence River at Ogdensburg, N.Y.
ARIMA(1,0,0)
α=0.89



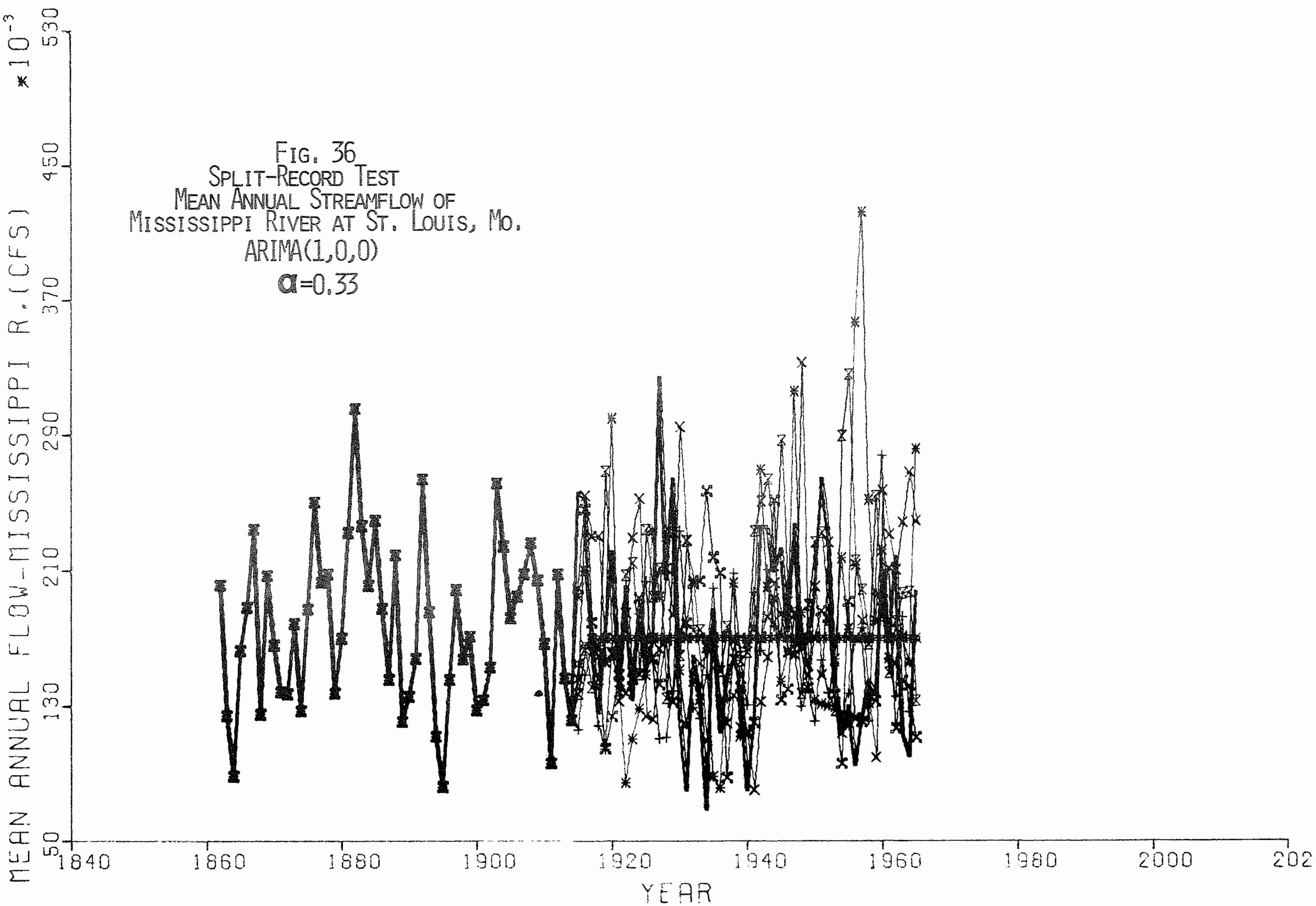


FIG. 36
SPLIT-RECORD TEST
MEAN ANNUAL STREAMFLOW OF
MISSISSIPPI RIVER AT ST. LOUIS, MO.
ARIMA(1,0,0)
α=0.33

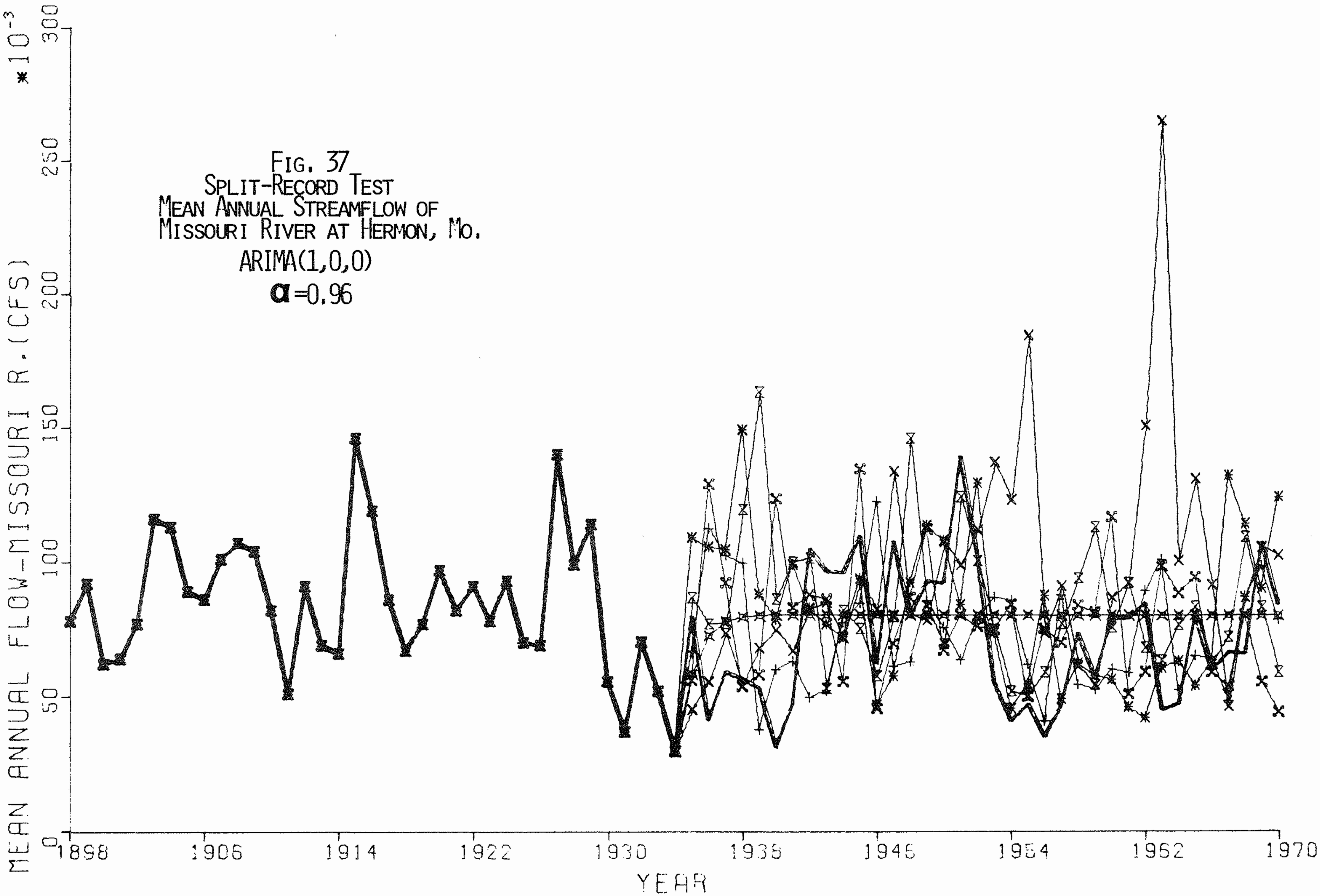


FIG. 37
SPLIT-RECORD TEST
MEAN ANNUAL STREAMFLOW OF
MISSOURI RIVER AT HERMON, MO.
ARIMA(1,0,0)
α =0.96

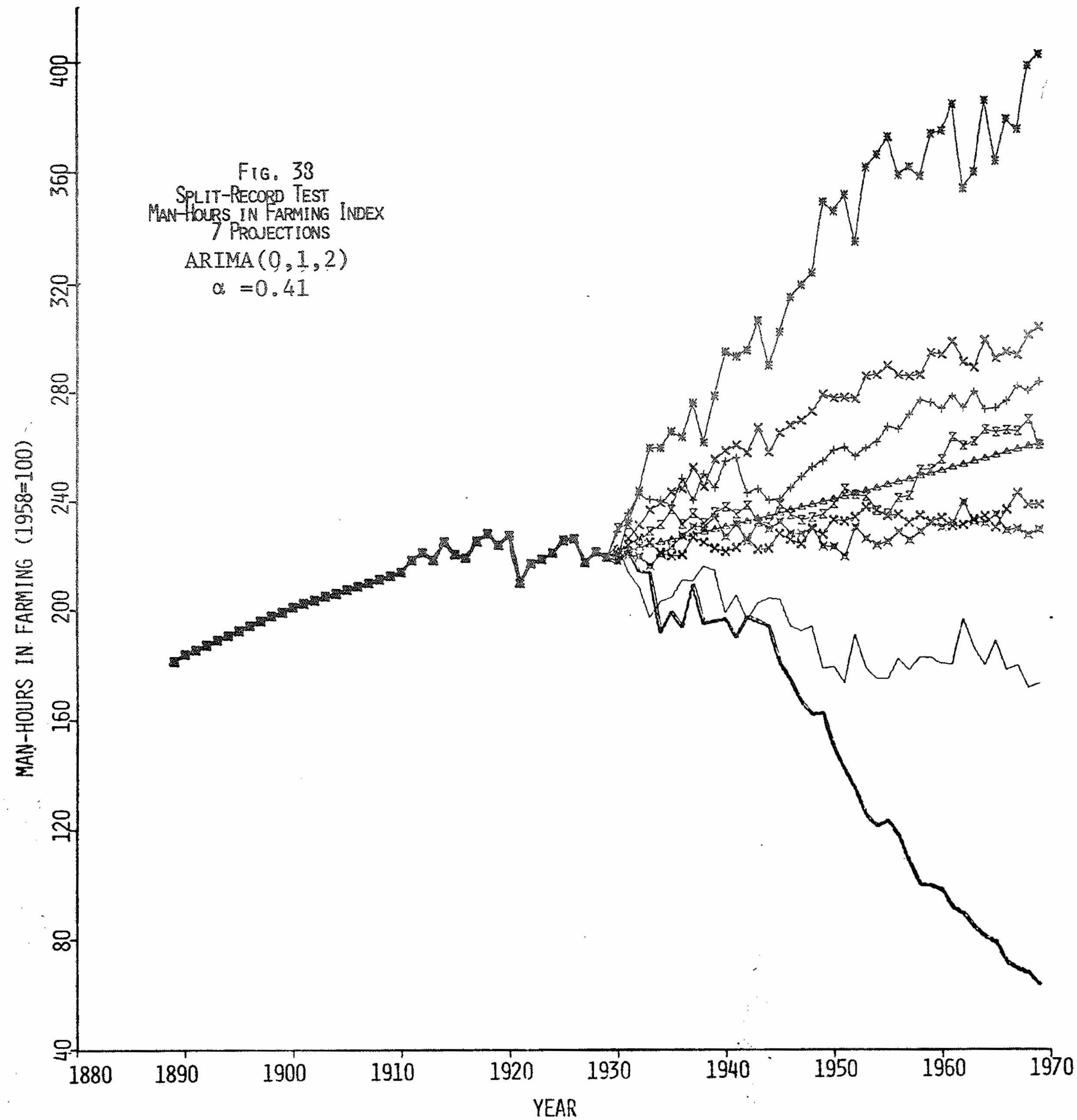


FIG. 38
SPLIT-RECORD TEST
MAN-HOURS IN FARMING INDEX
7 PROJECTIONS
ARIMA(0,1,2)
α =0.41
MAN-HOURS IN FARMING (1958=100)
400
360
320
280
240
200
160
120
80
40
1880
1890
1900
1910
1920
1930
1940
1950
1960
1970
YEAR

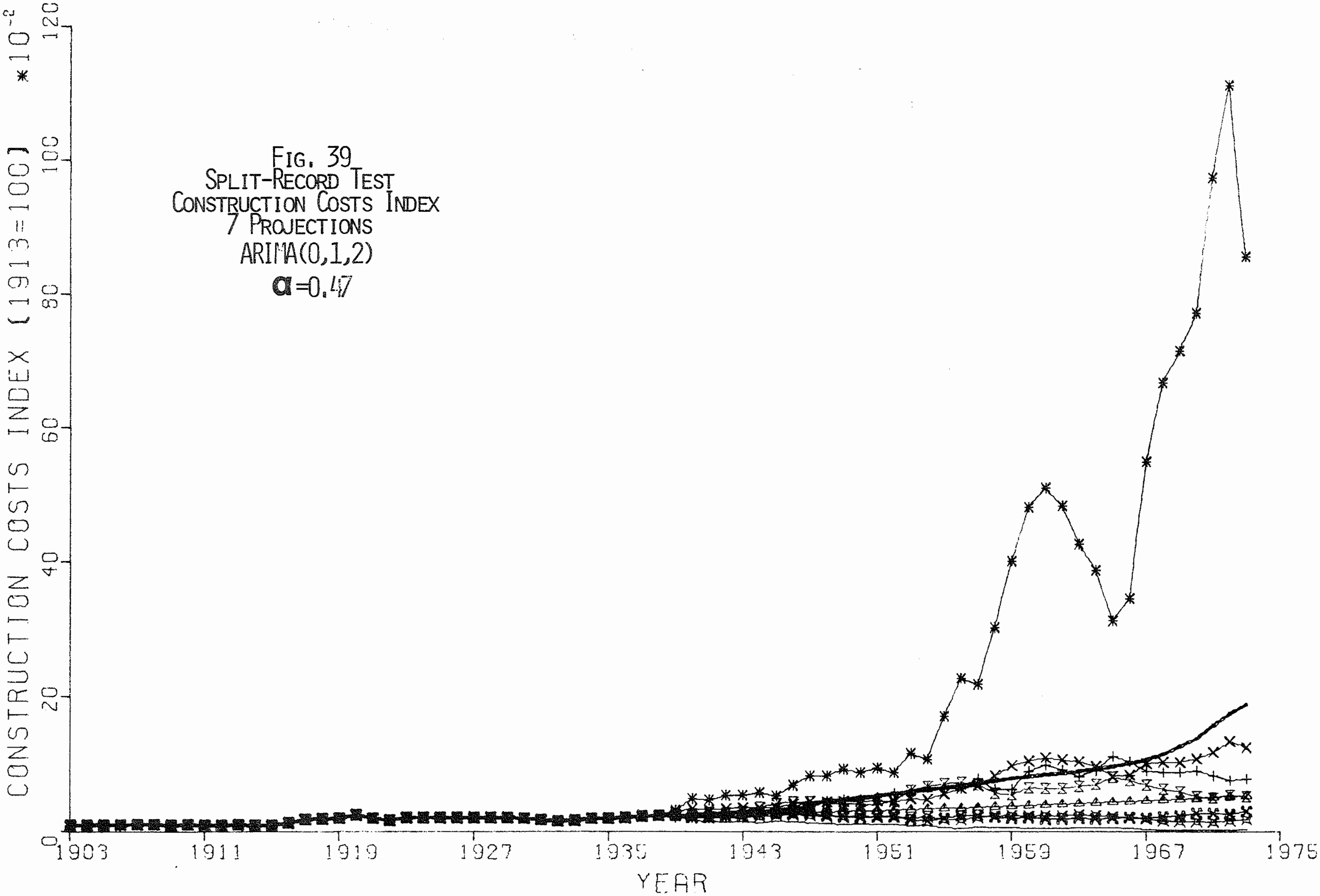


FIG. 39
SPLIT-RECORD TEST
CONSTRUCTION COSTS INDEX
7 PROJECTIONS
ARIMA(0,1,2)
α=0.47

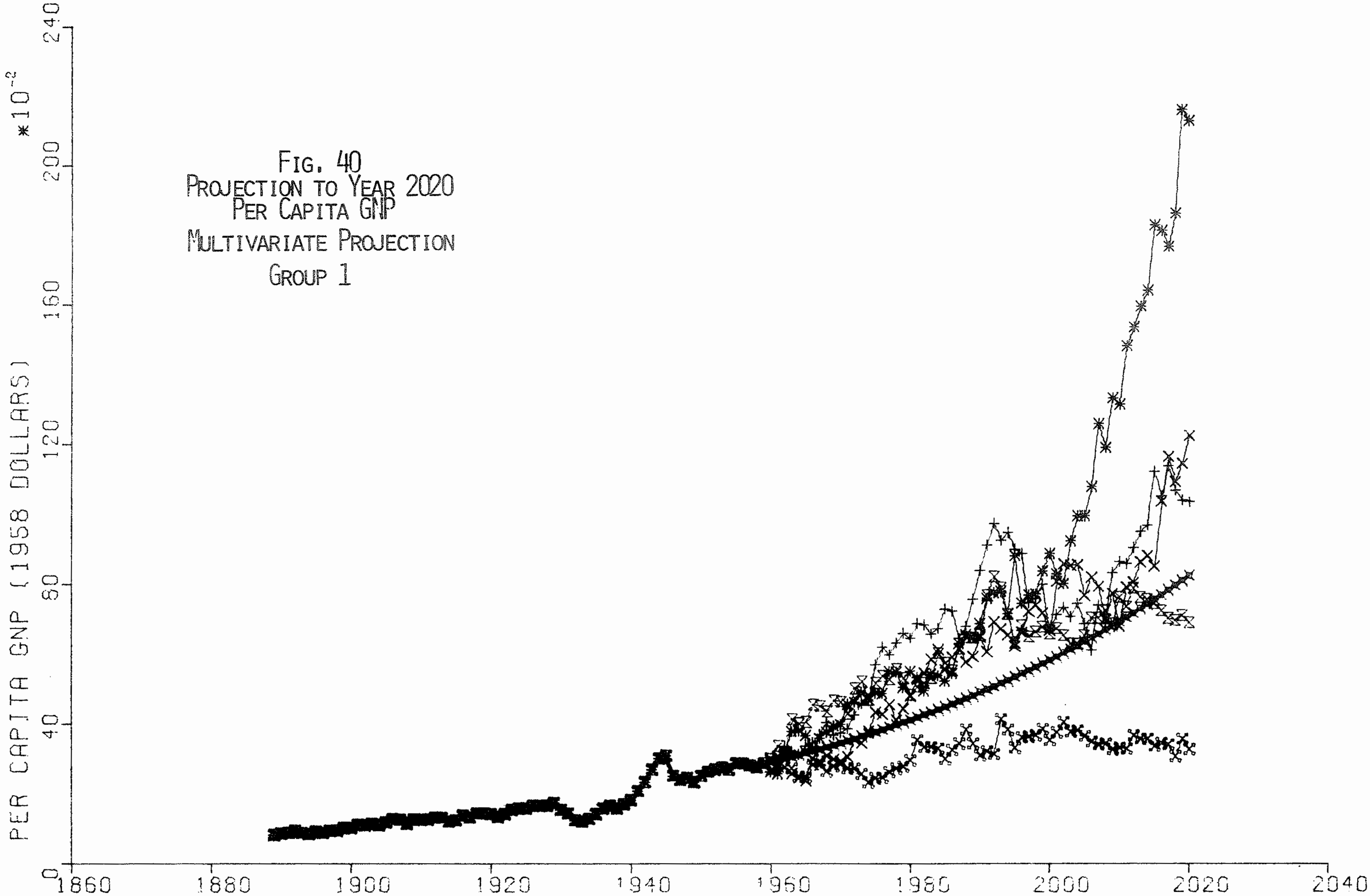


Fig. 40
Projection to Year 2020
Per Capita GNP
Multivariate Projection
Group 1

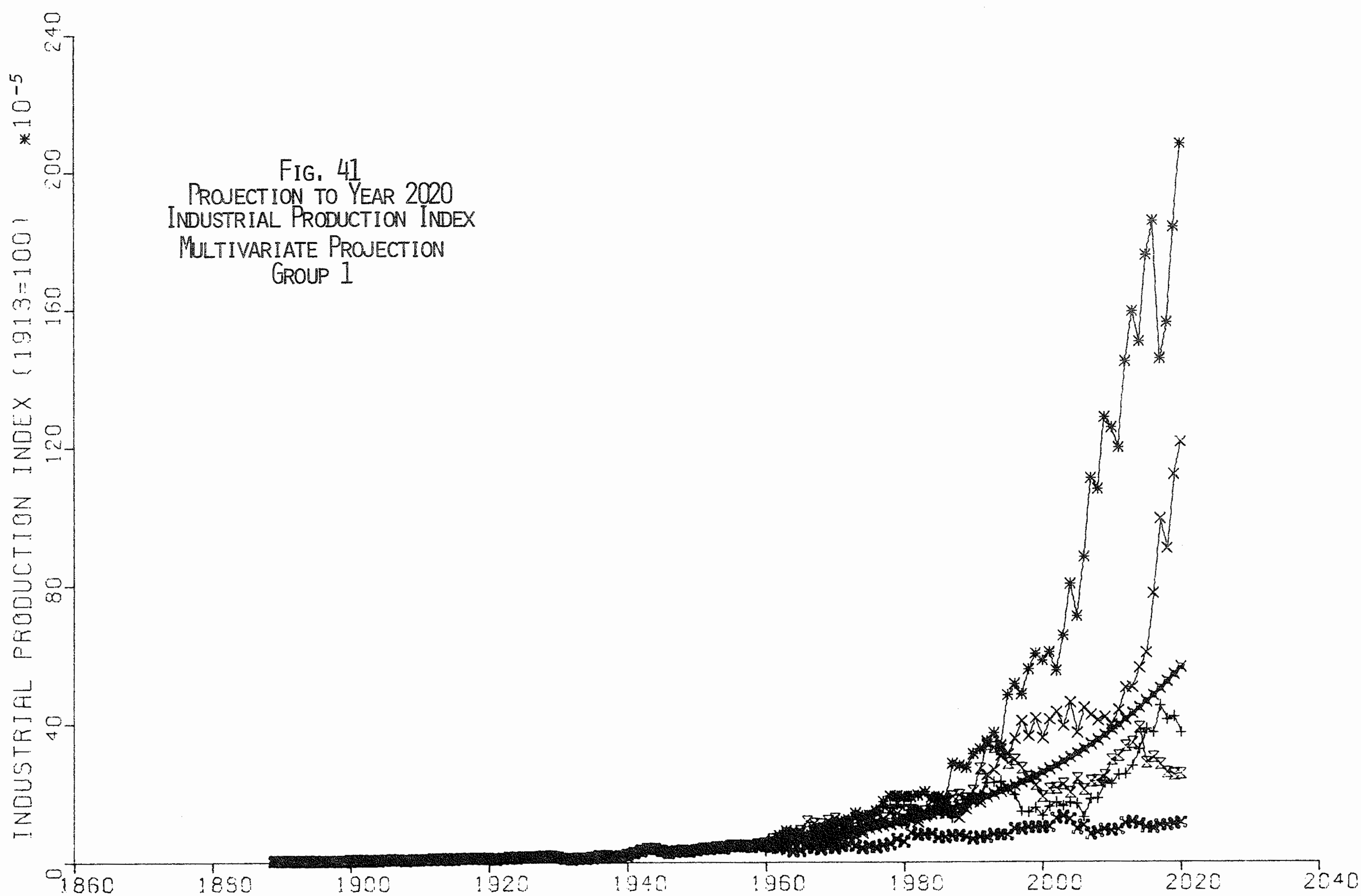


Fig. 41
Projection to Year 2020
Industrial Production Index
Multivariate Projection
Group 1
INDUSTRIAL PRODUCTION INDEX (1913=100)
*10-5
0
40
80
120
160
200
240
1860
1880
1900
1920
1940
1960
1980
2000
2020
2040

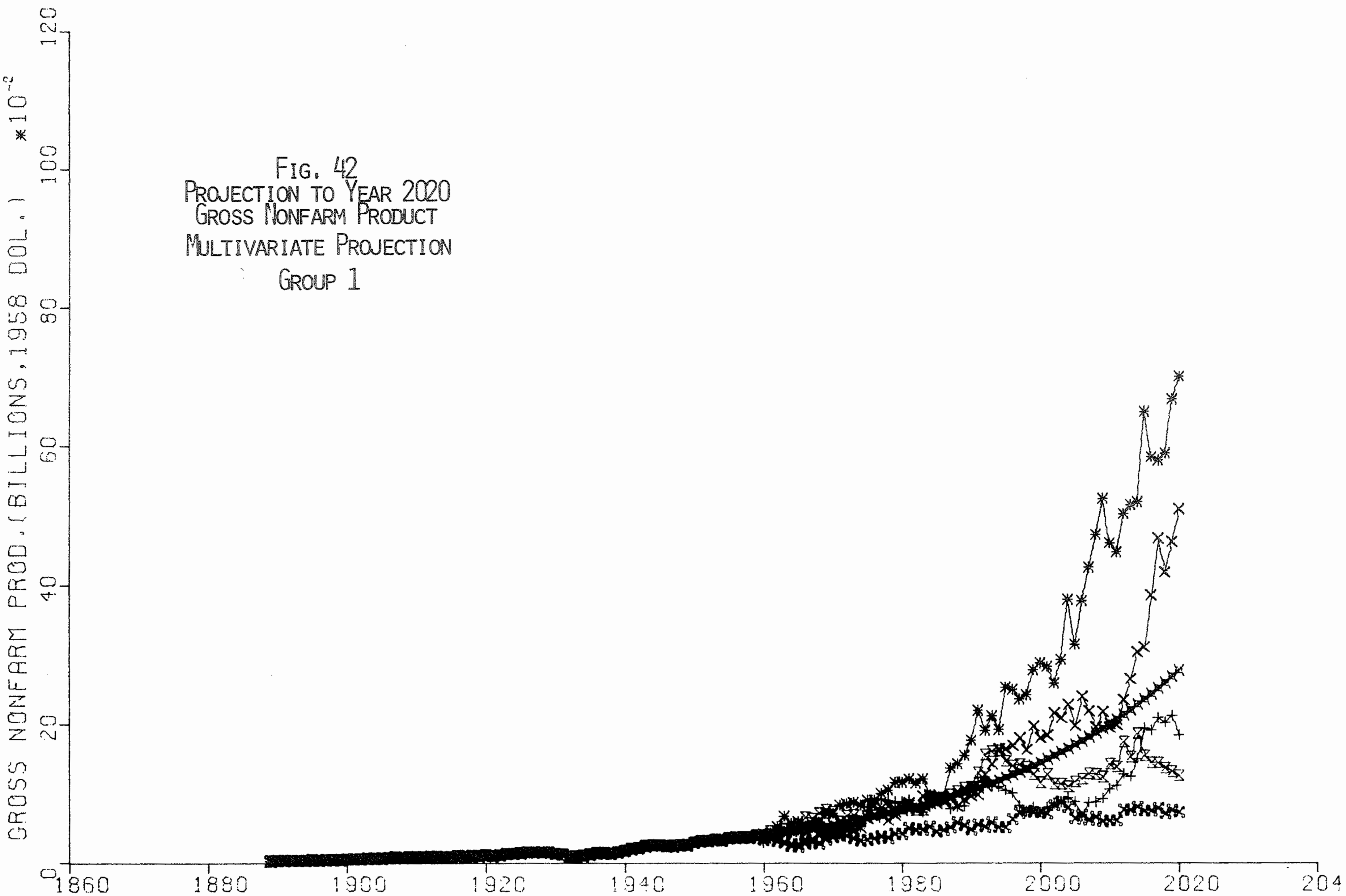


Fig. 42
Projection to Year 2020
Gross Nonfarm Product
Multivariate Projection
Group 1

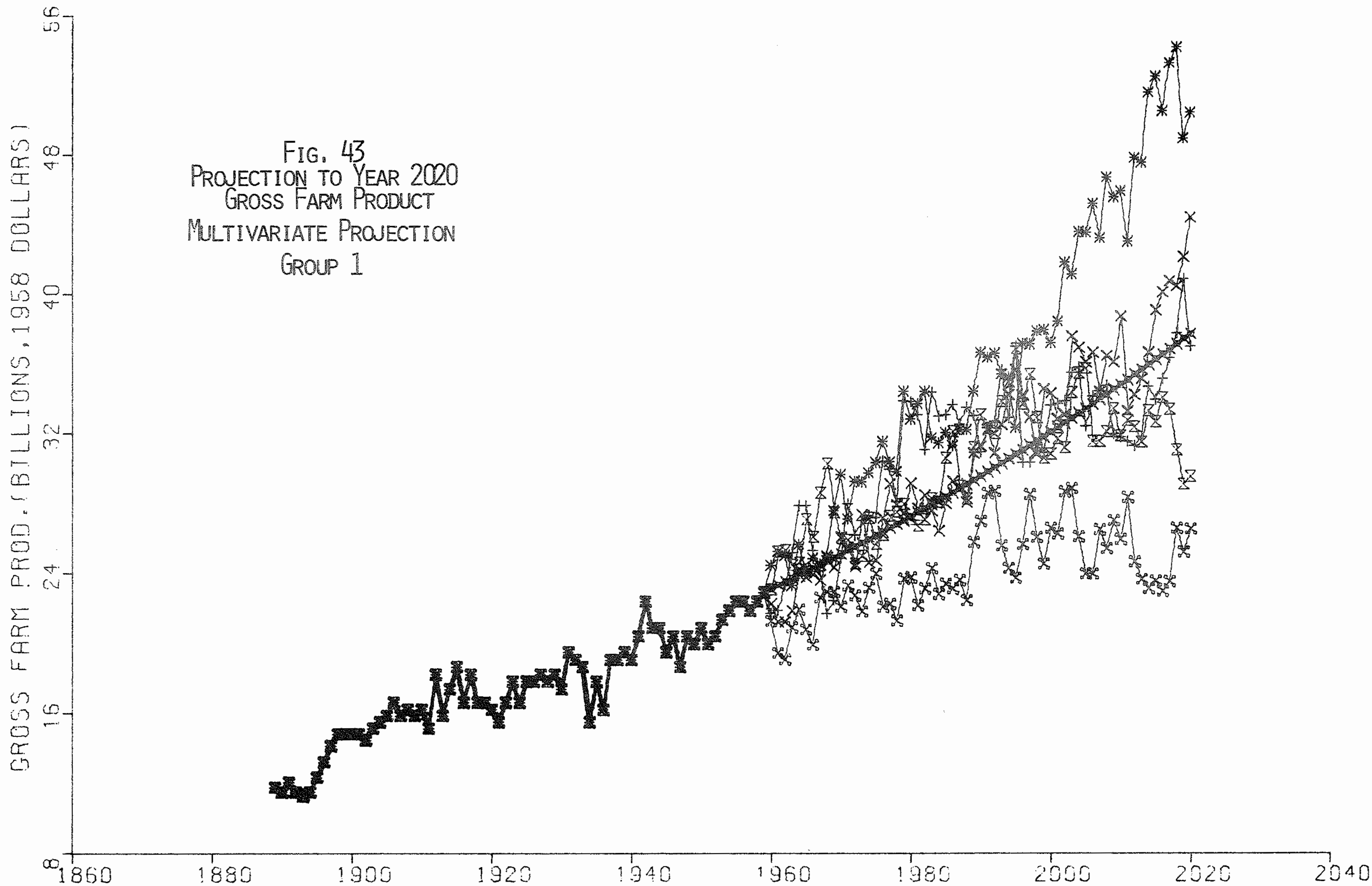


FIG. 43
PROJECTION TO YEAR 2020
GROSS FARM PRODUCT
MULTIVARIATE PROJECTION
GROUP 1
GROSS FARM PROD.(BILLIONS,1958 DOLLARS)
8
16
24
32
40
48
56
1860
1880
1900
1920
1940
1960
1980
2000
2020
2040

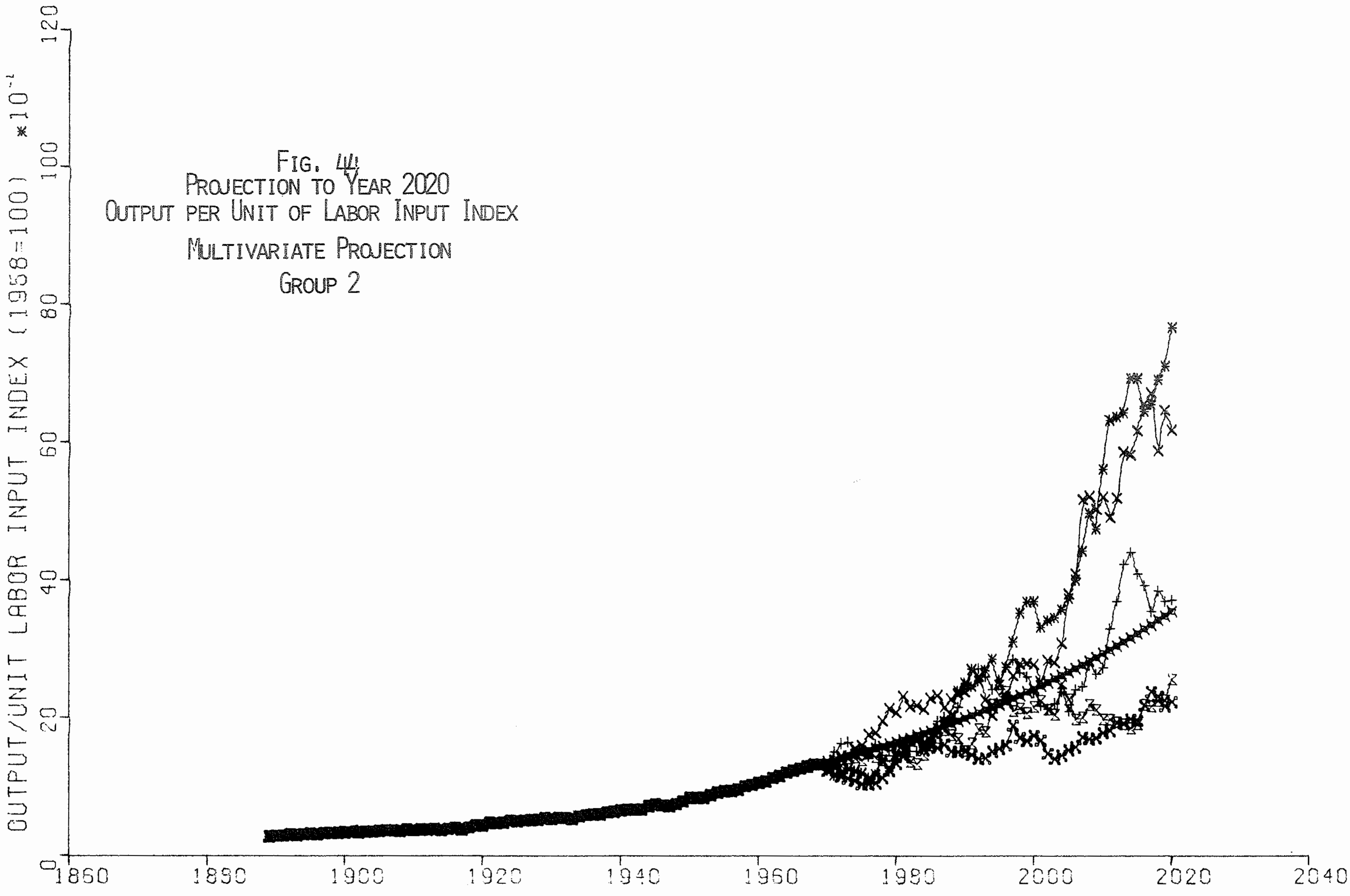


FIG. 44
PROJECTION TO YEAR 2020
OUTPUT PER UNIT OF LABOR INPUT INDEX
MULTIVARIATE PROJECTION
GROUP 2

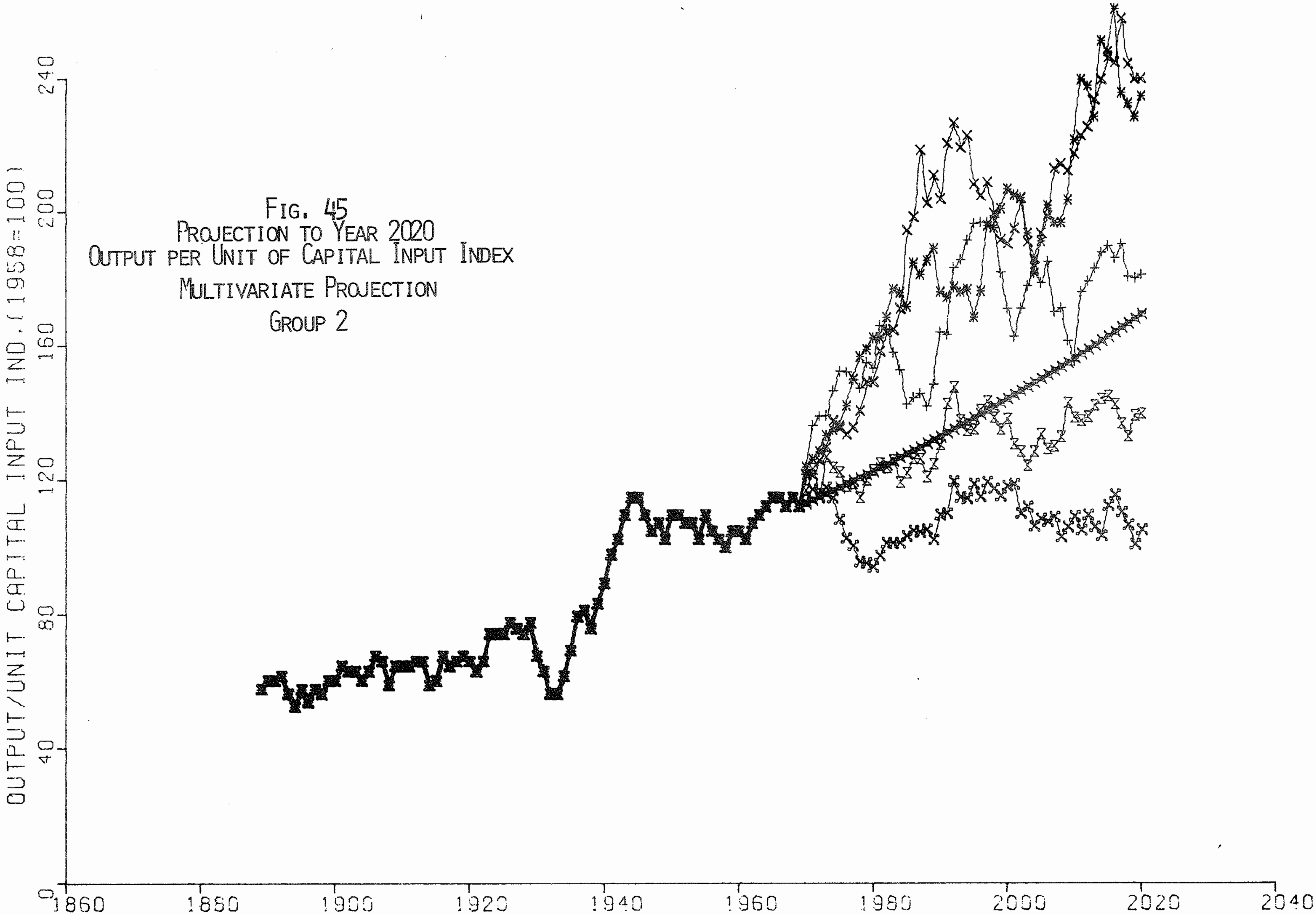


FIG. 45
PROJECTION TO YEAR 2020
OUTPUT PER UNIT OF CAPITAL INPUT INDEX
MULTIVARIATE PROJECTION
GROUP 2

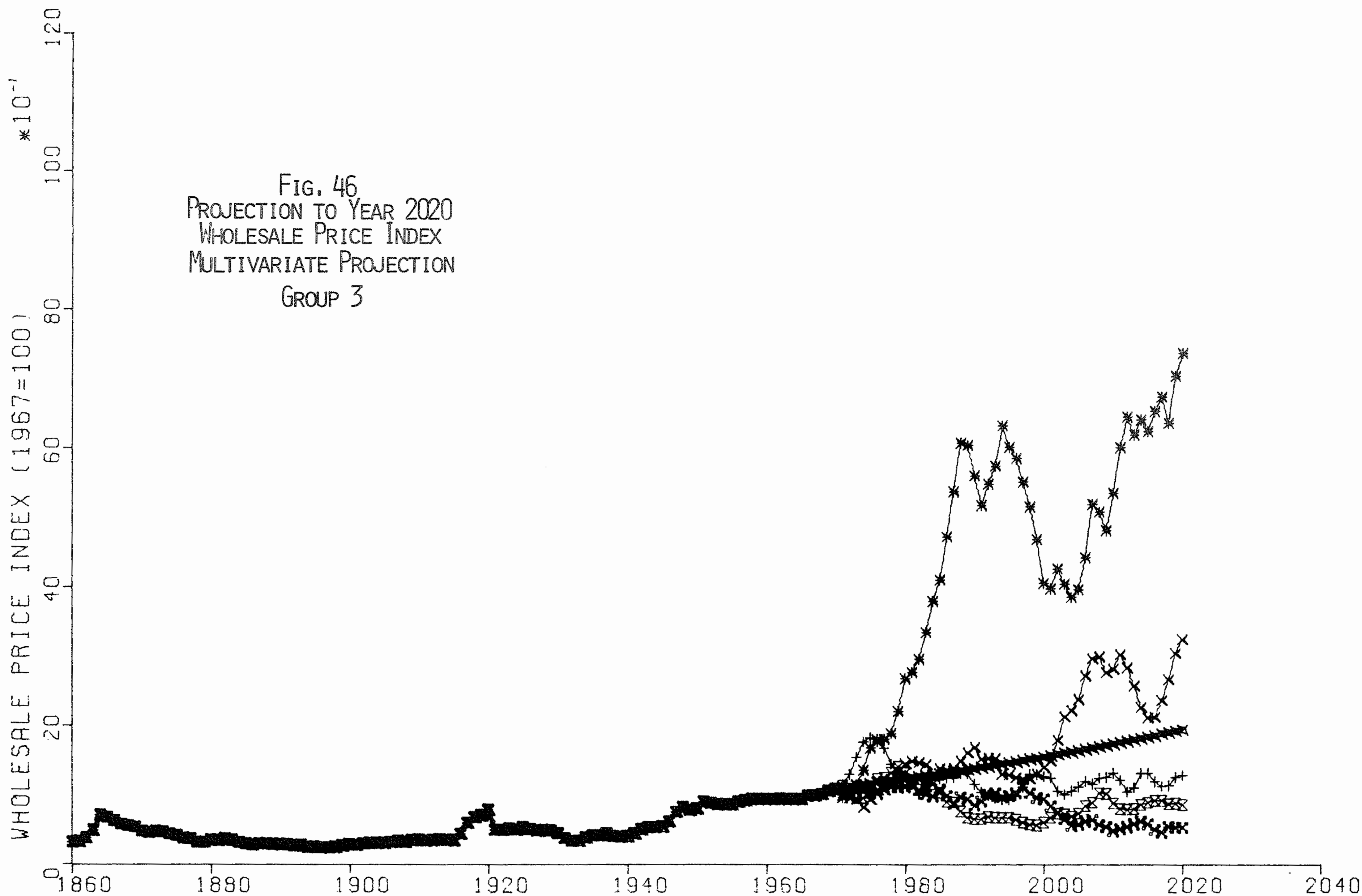


Fig. 46
Projection to Year 2020
Wholesale Price Index
Multivariate Projection
Group 3
WHOLESALE PRICE INDEX (1967=100)
*10⁻¹
0
20
40
60
80
100
120
1860
1880
1900
1920
1940
1960
1980
2000
2020
2040

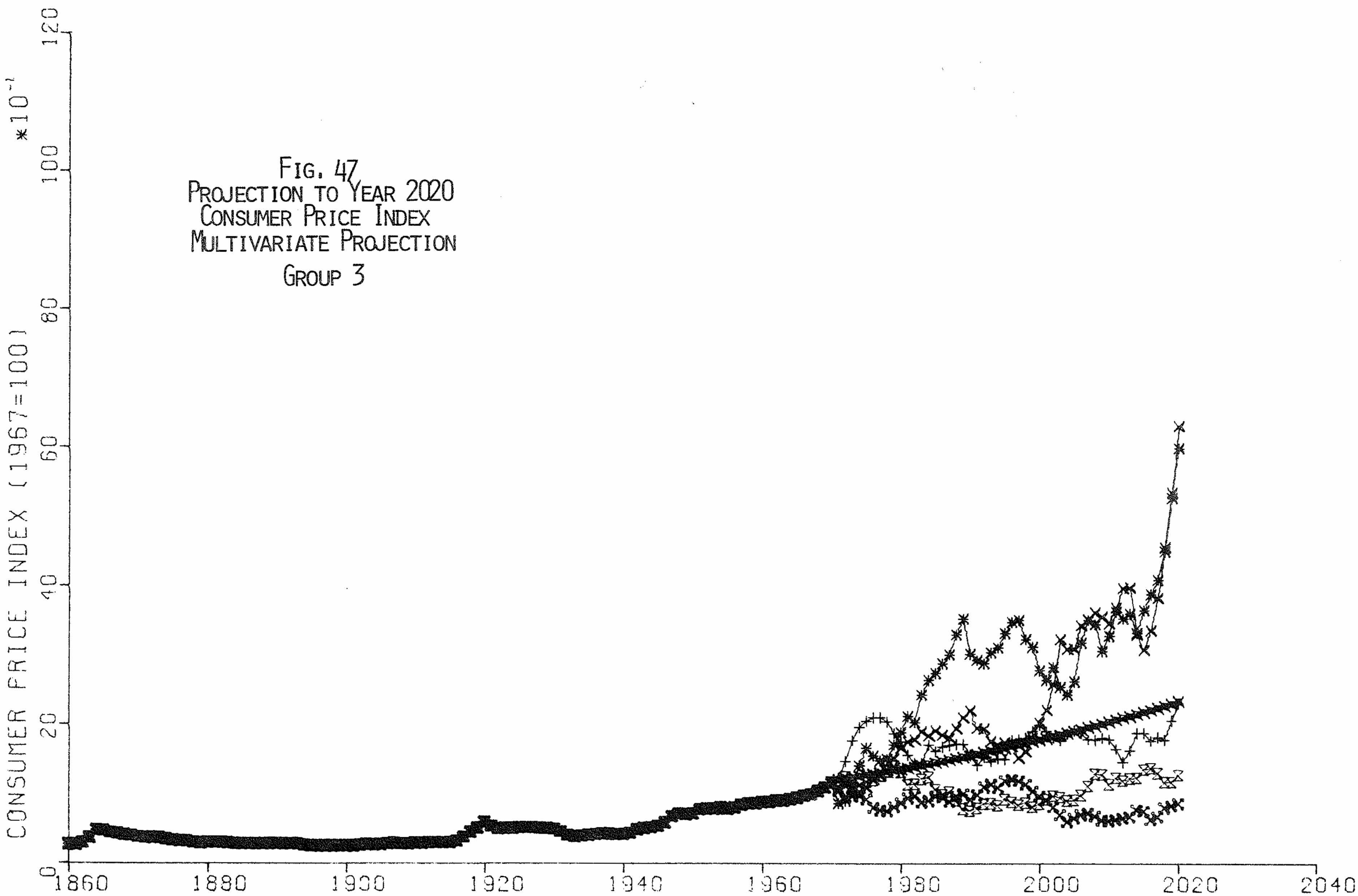


FIG. 47
PROJECTION TO YEAR 2020
CONSUMER PRICE INDEX
MULTIVARIATE PROJECTION
GROUP 3

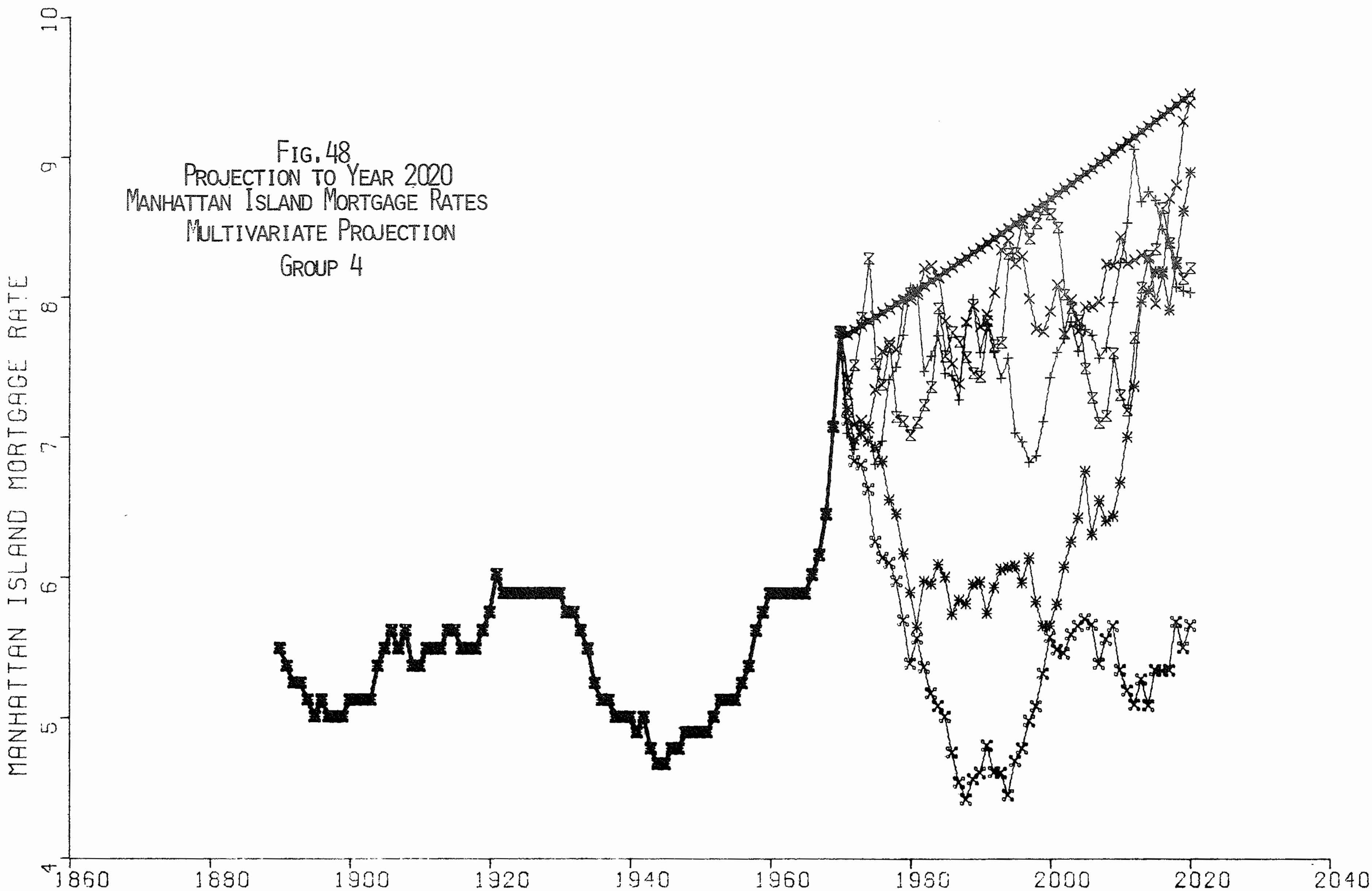


Fig. 48
Projection to Year 2020
Manhattan Island Mortgage Rates
Multivariate Projection
Group 4

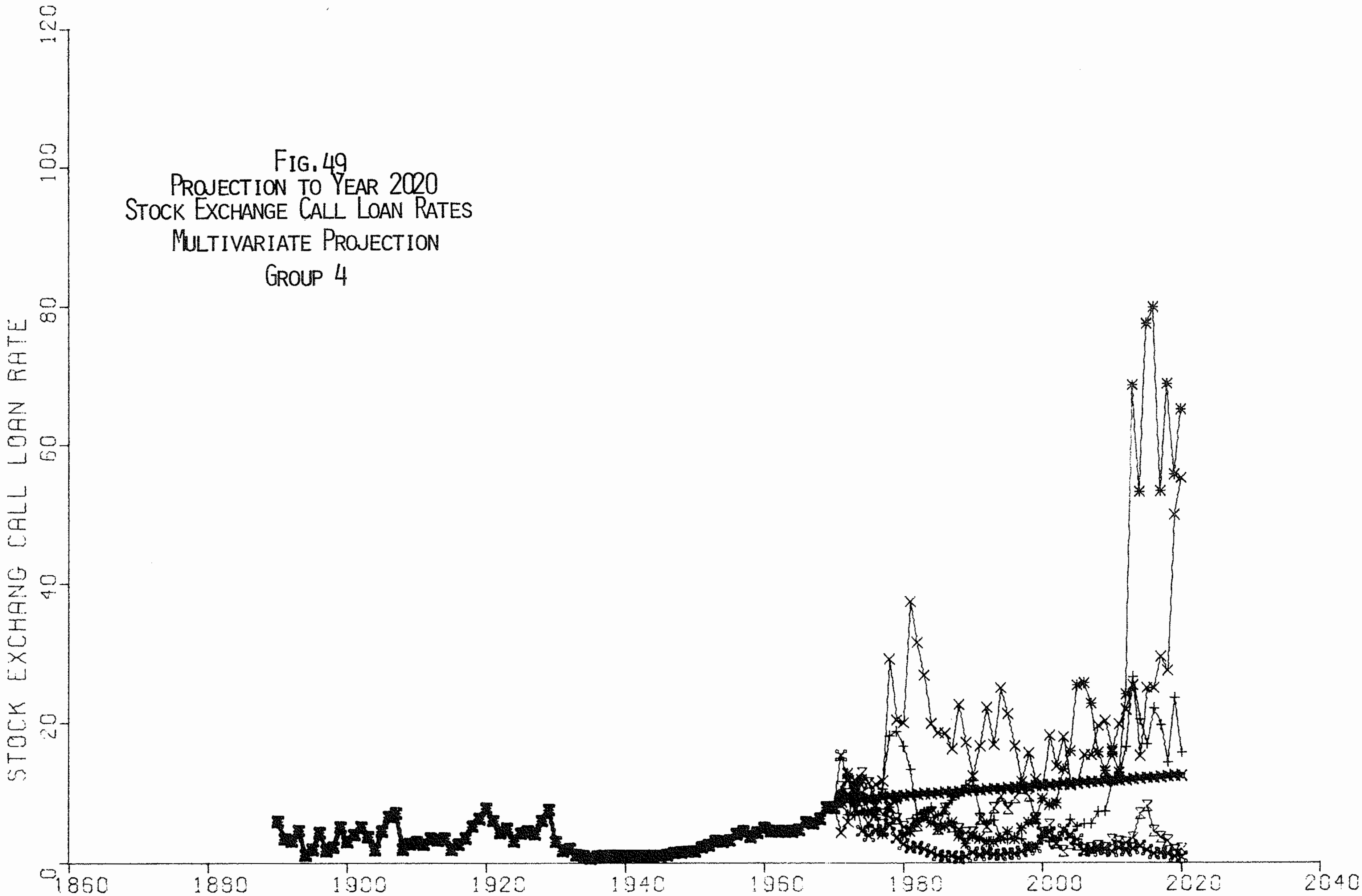
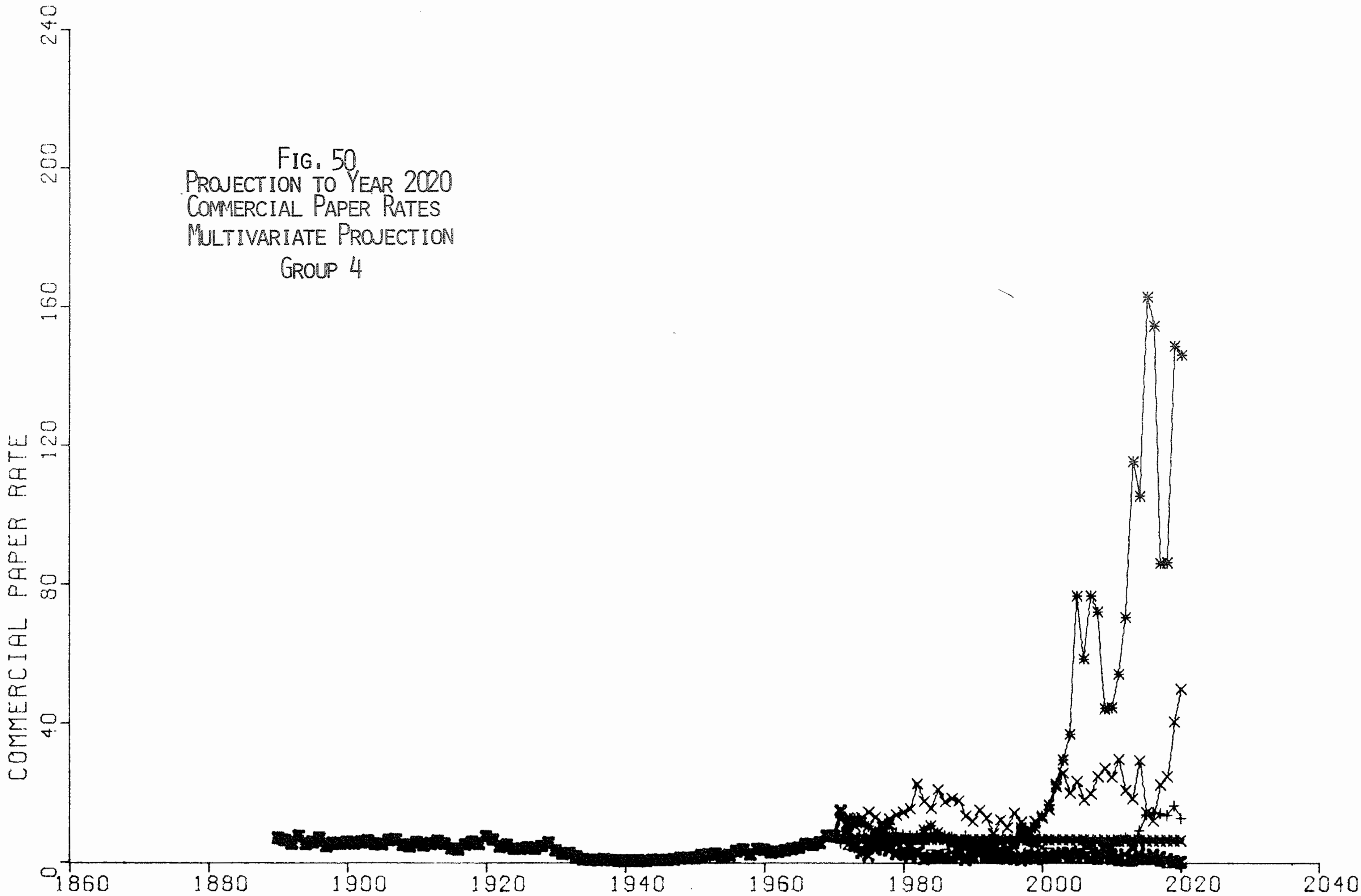


FIG. 49
PROJECTION TO YEAR 2020
STOCK EXCHANGE CALL LOAN RATES
MULTIVARIATE PROJECTION
GROUP 4
STOCK EXCHANG CALL LOAN RATE
0
20
40
60
80
100
120
1860
1880
1900
1920
1940
1960
1980
2000
2020
2040

Fig. 50
Projection to Year 2020
Commercial Paper Rates
Multivariate Projection
Group 4



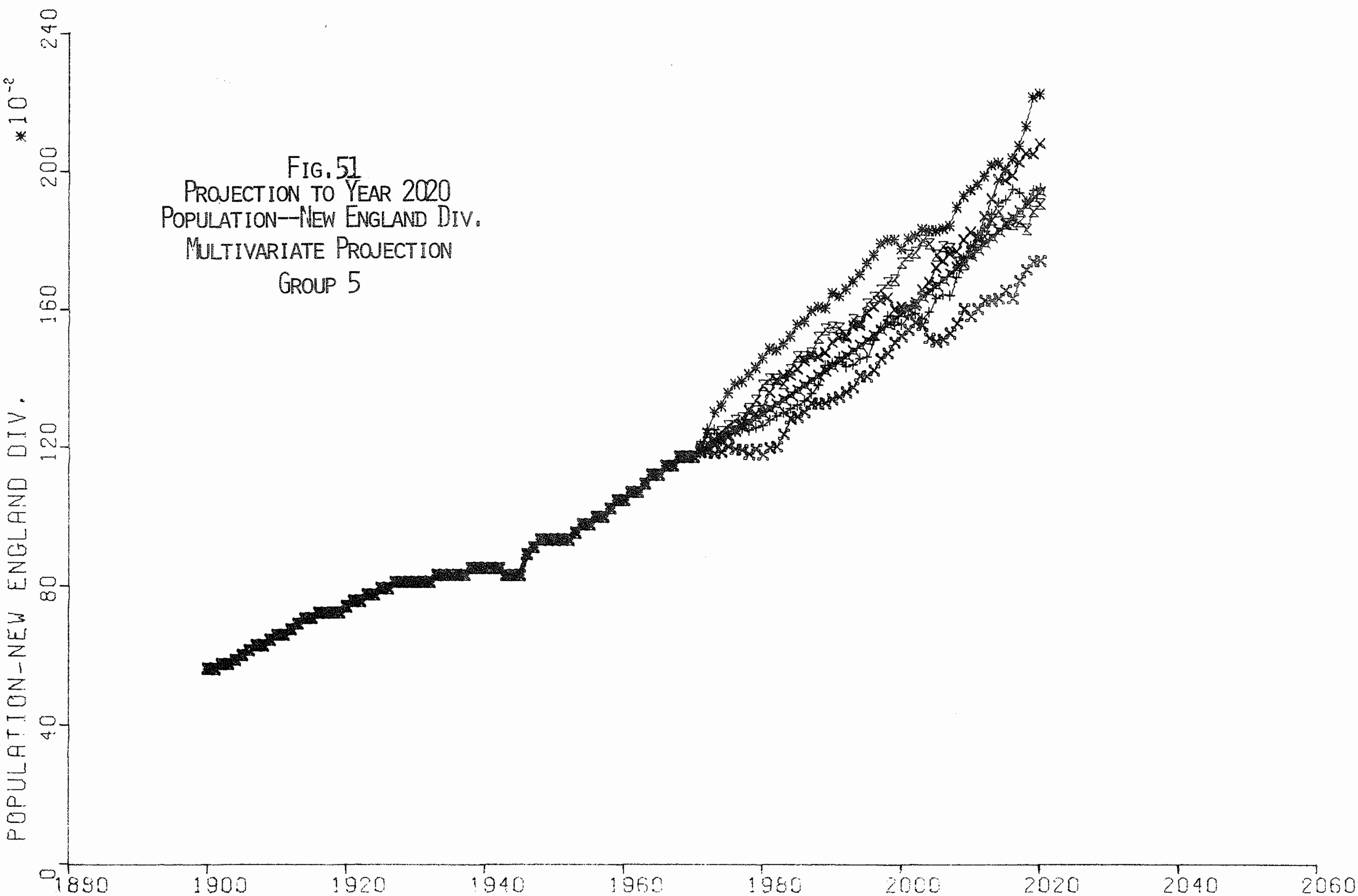


Fig. 51
Projection to Year 2020
Population--New England Div.
Multivariate Projection
Group 5

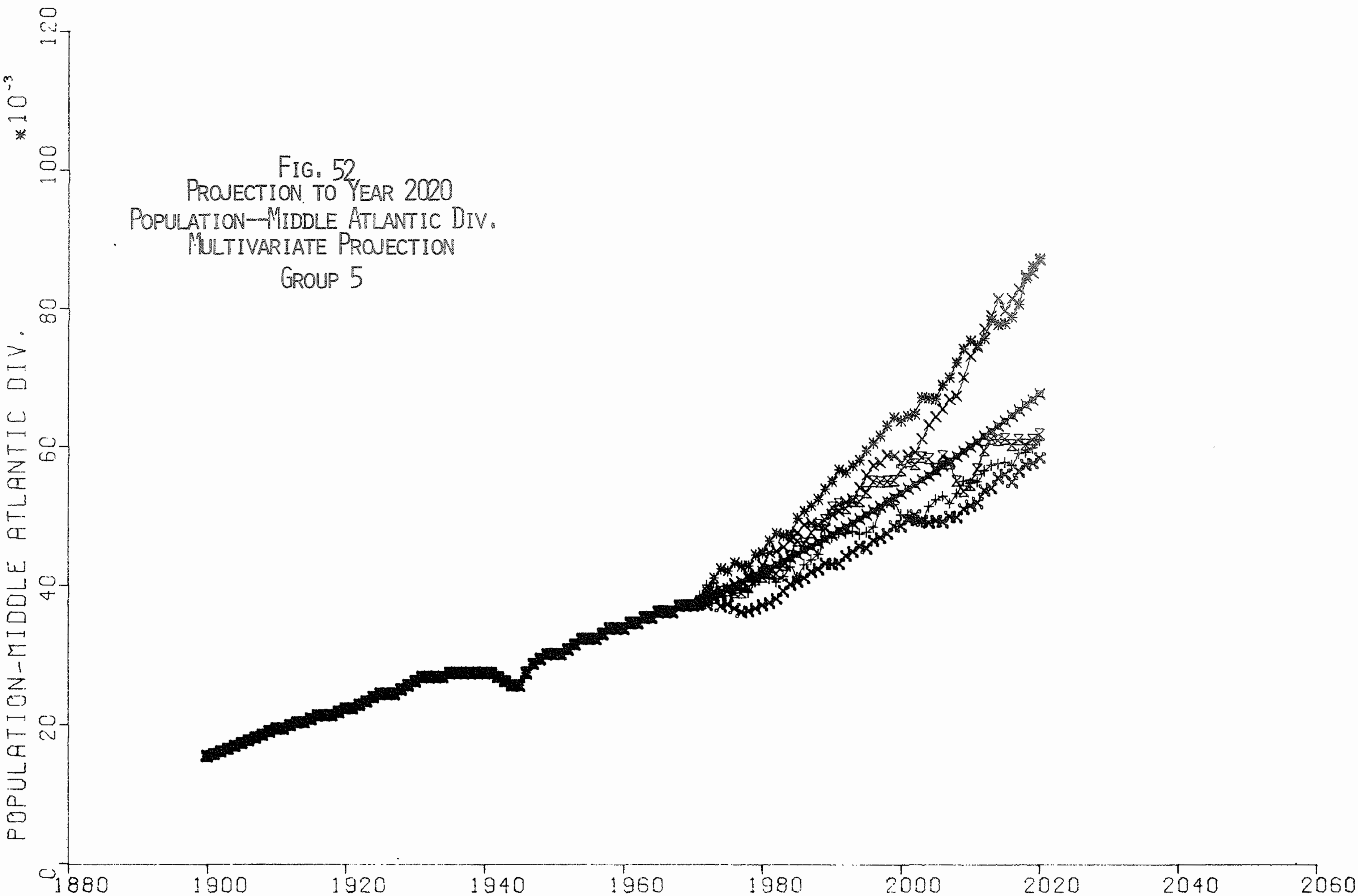
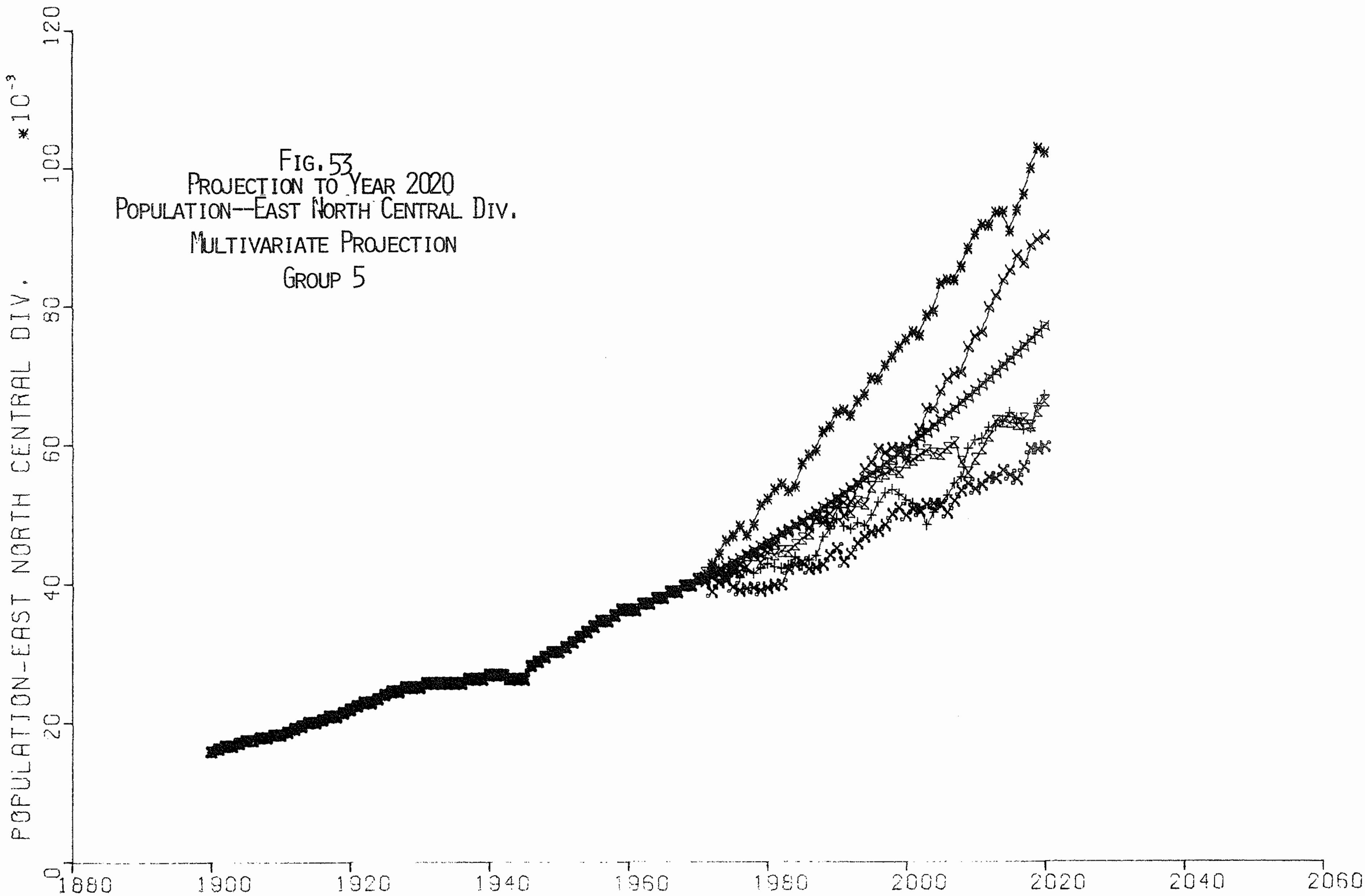


FIG. 52
PROJECTION TO YEAR 2020
POPULATION--MIDDLE ATLANTIC DIV.
MULTIVARIATE PROJECTION
GROUP 5

Fig. 53
Projection to Year 2020
Population--East North Central Div.
Multivariate Projection
Group 5



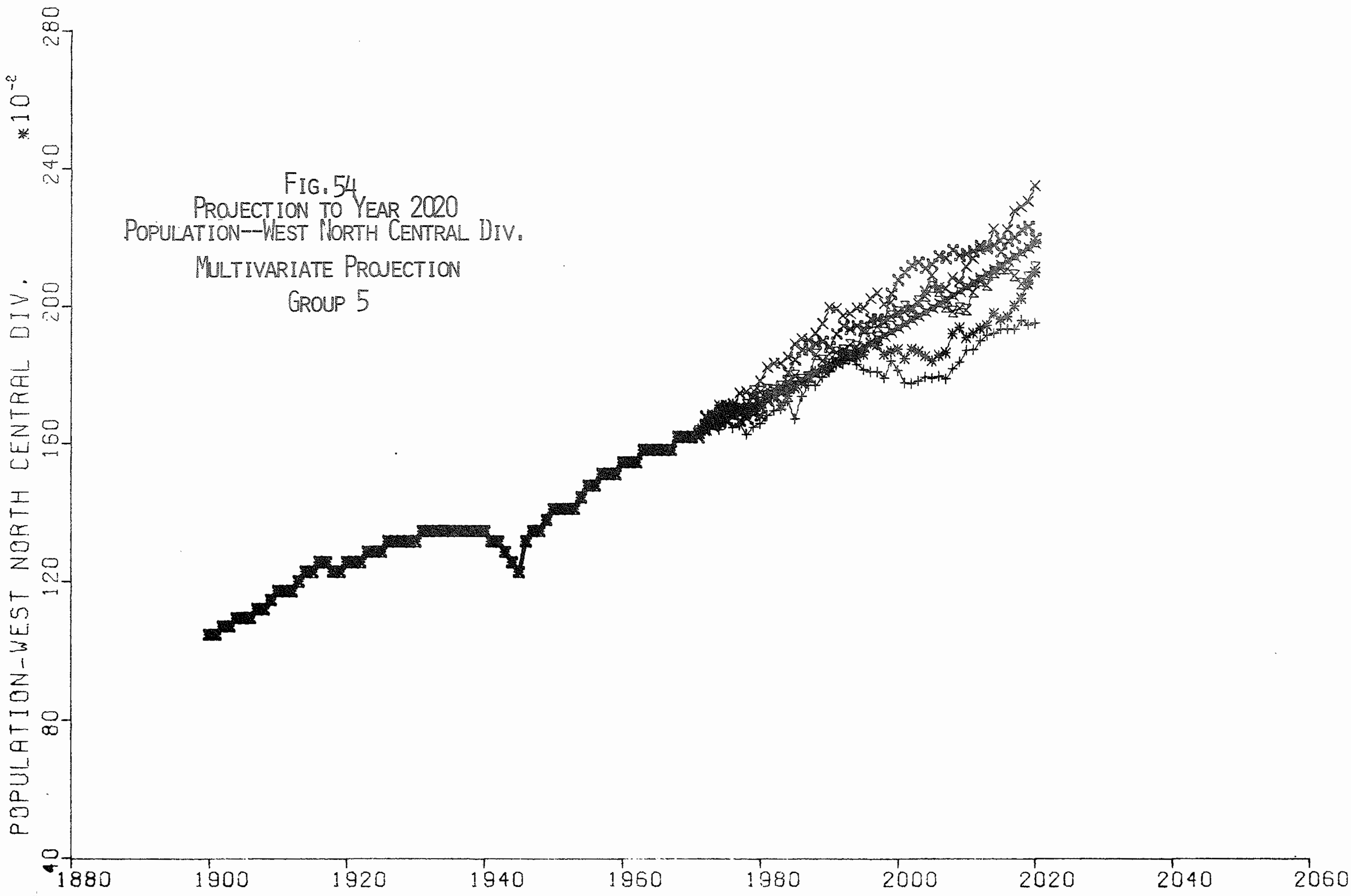


Fig. 54
Projection to Year 2020
Population--West North Central Div.
Multivariate Projection
Group 5
POPULATION-WEST NORTH CENTRAL DIV.
*10^-2
40
80
120
160
200
240
280
1880
1900
1920
1940
1960
1980
2000
2020
2040
2060

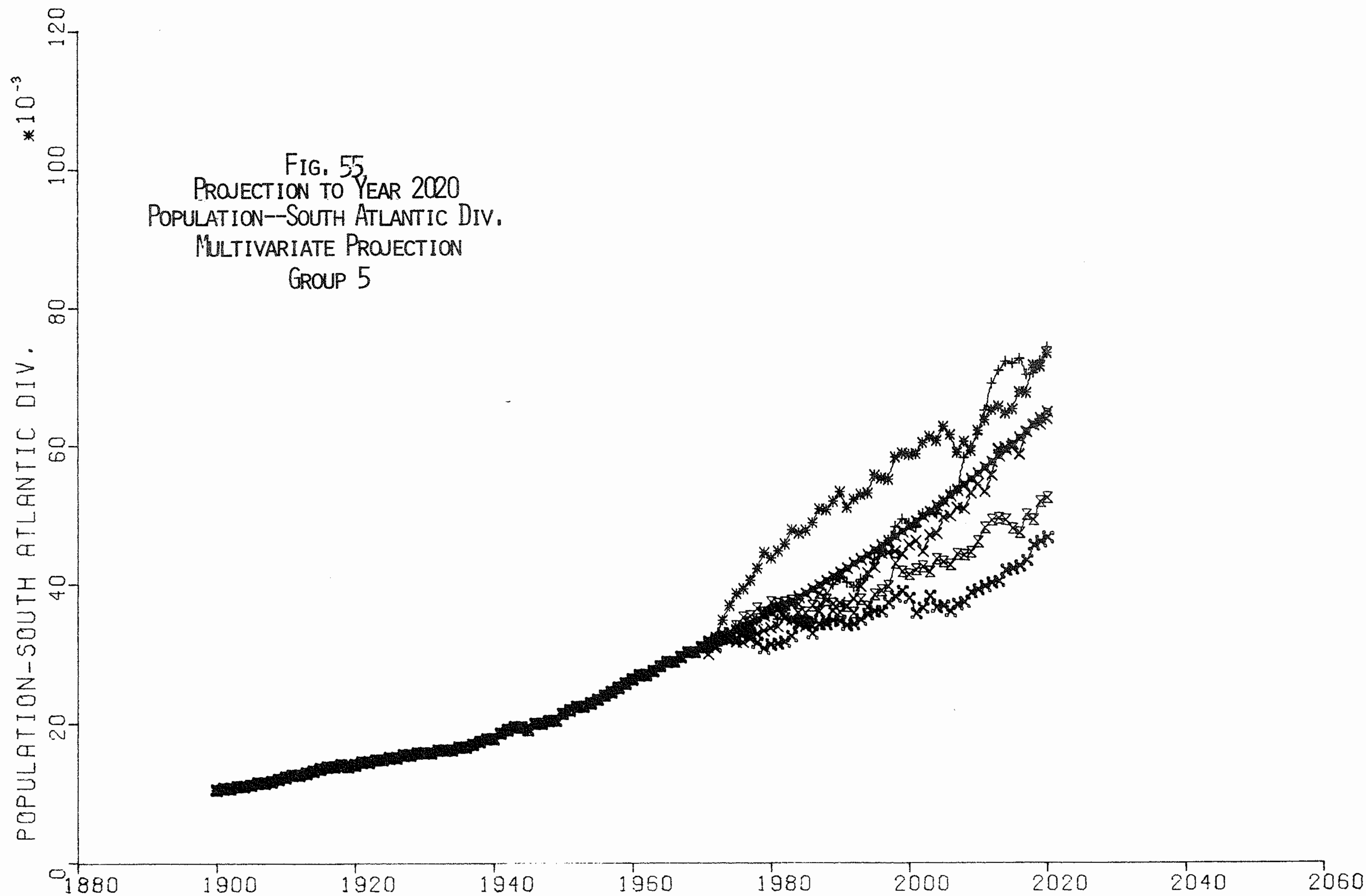


Fig. 55
Projection to Year 2020
Population--South Atlantic Div.
Multivariate Projection
Group 5
POPULATION-SOUTH ATLANTIC DIV.
*10^-3
0
20
40
60
80
100
120
1880
1900
1920
1940
1960
1980
2000
2020
2040
2060

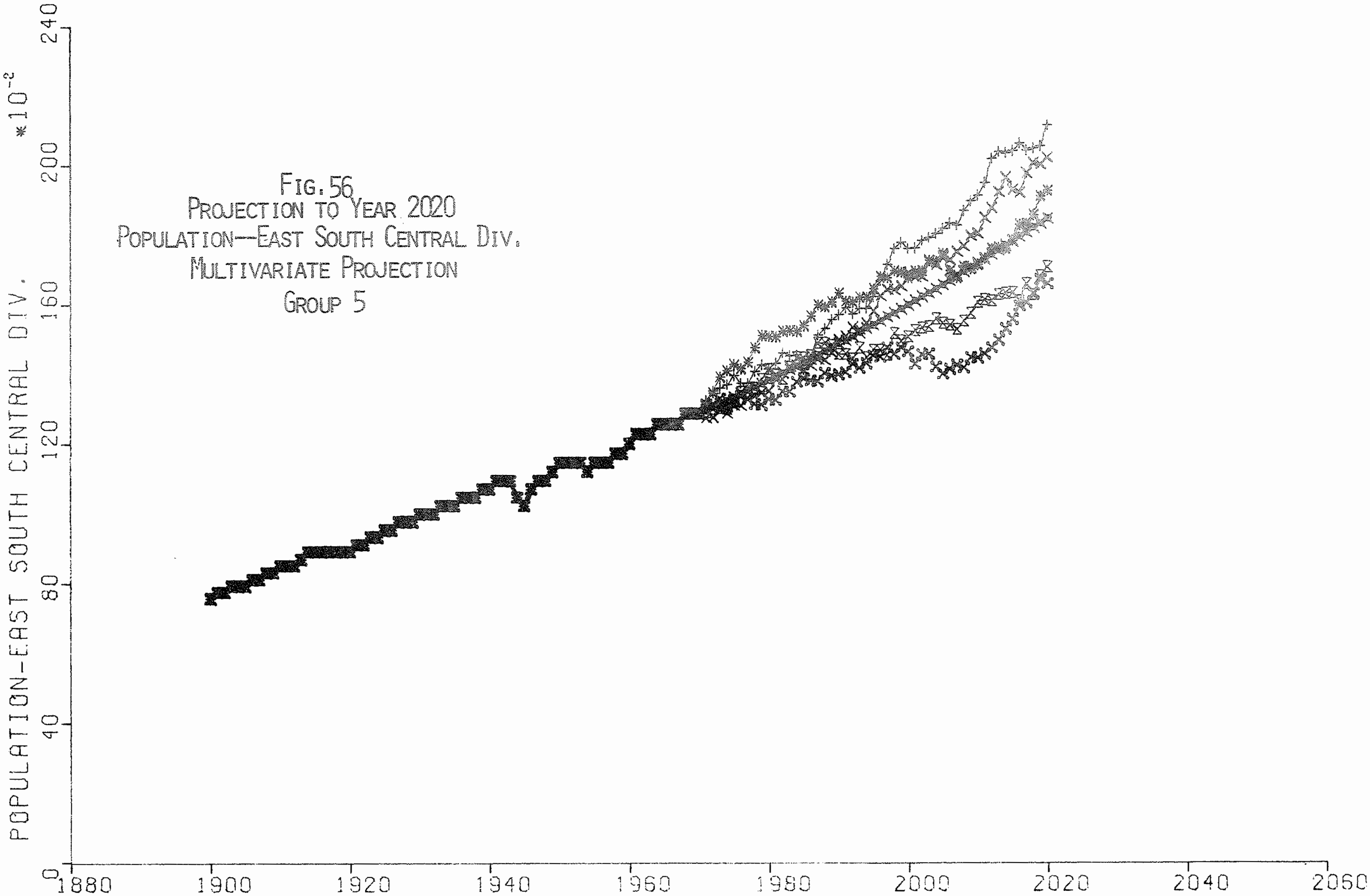


Fig. 56
Projection to Year 2020
Population--East South Central Div.
Multivariate Projection
Group 5
POPULATION-EAST SOUTH CENTRAL DIV.
*10⁻²
0
40
80
120
160
200
240
1880
1900
1920
1940
1960
1980
2000
2020
2040
2060

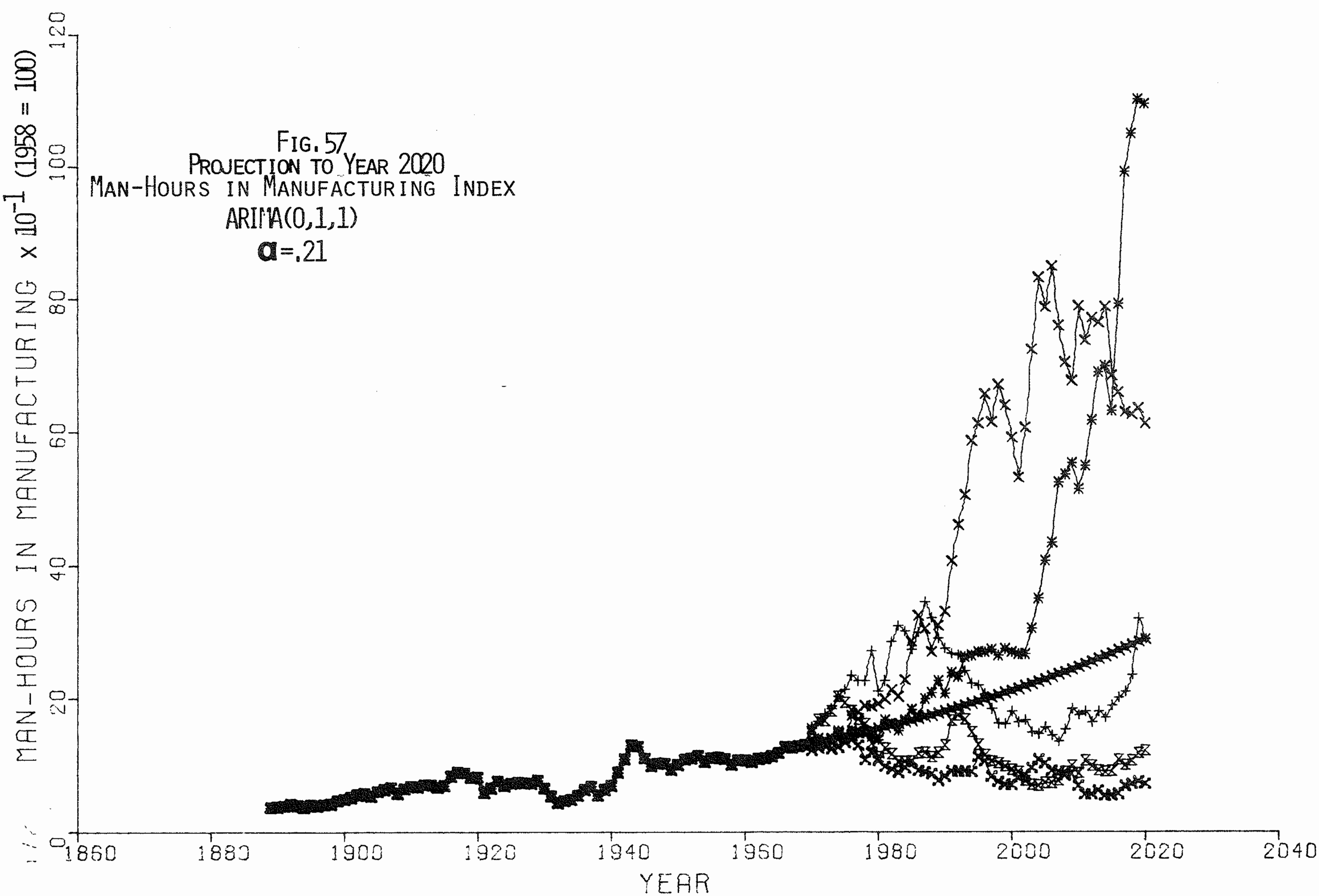


Fig. 57
Projection to Year 2020
Man-Hours in Manufacturing Index
ARIMA(0,1,1)
α=.21
MAN-HOURS IN MANUFACTURING x 10^-1 (1958 = 100)
0
20
40
60
80
100
120
1860
1880
1900
1920
1940
1960
1980
2000
2020
2040
YEAR

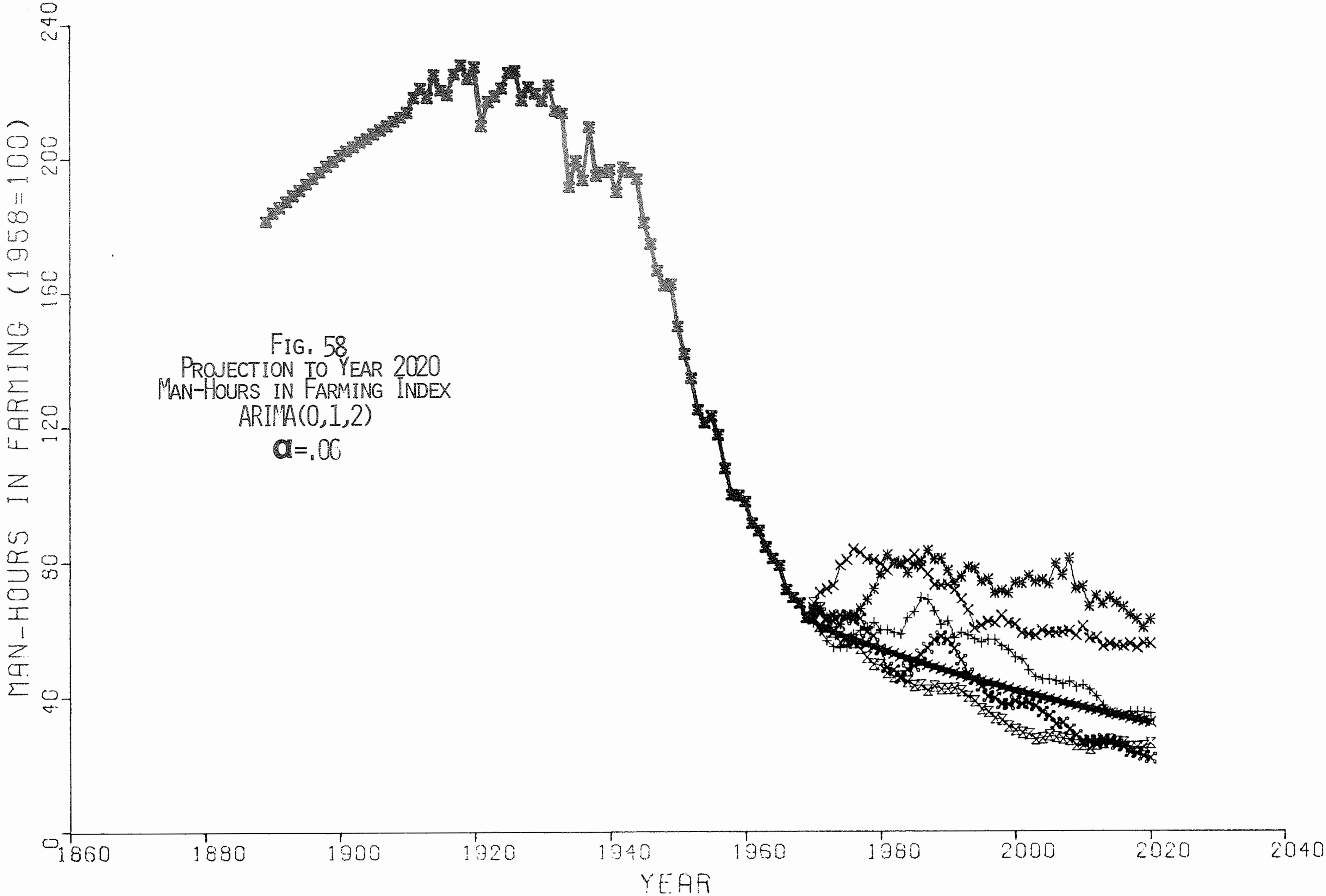


Fig. 58
Projection to Year 2020
Man-Hours in Farming Index
ARIMA(0,1,2)
α=.06
MAN-HOURS IN FARMING (1958=100)
0
40
80
120
160
200
240
1860
1880
1900
1920
1940
1960
1980
2000
2020
2040
YEAR

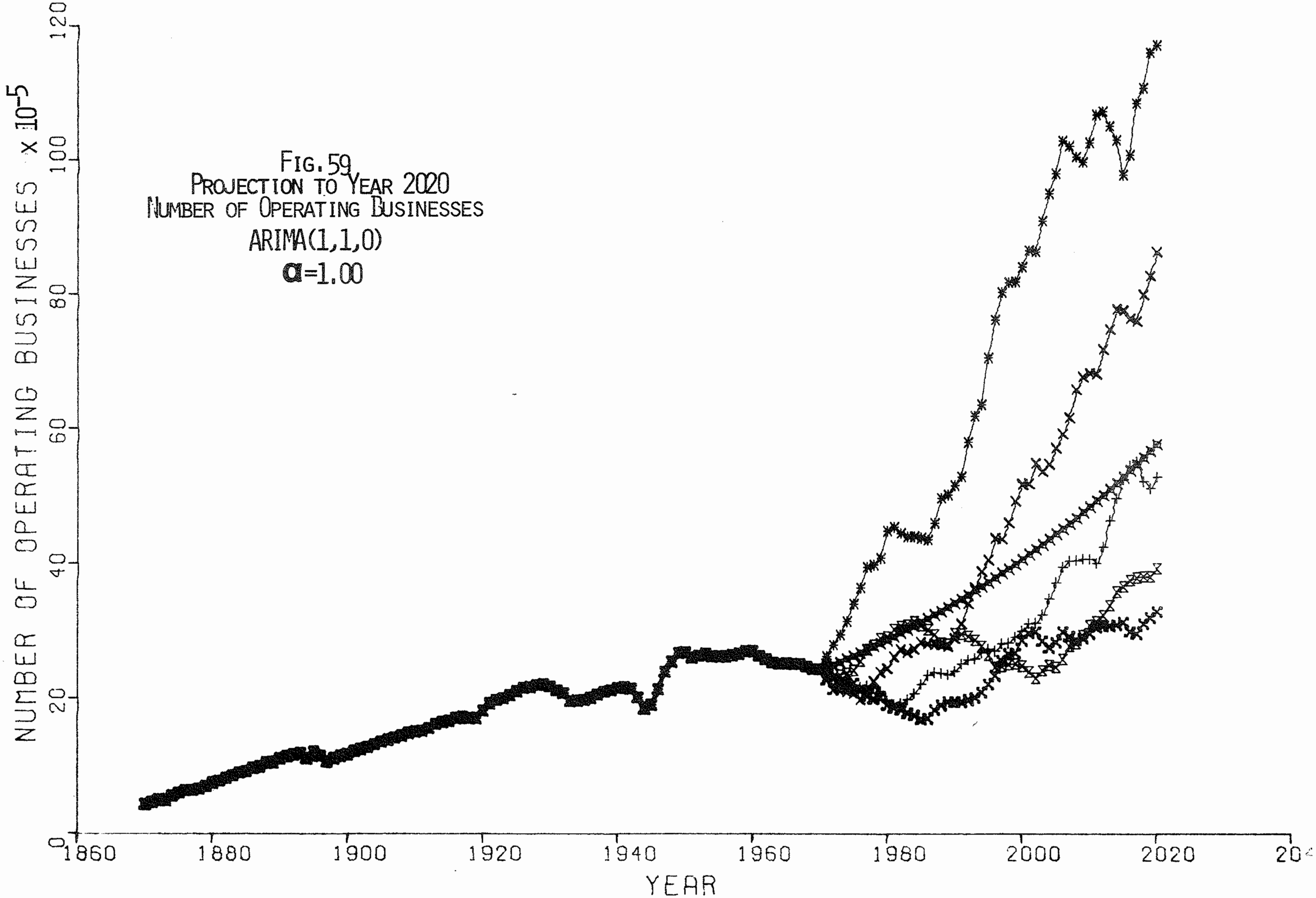


Fig. 59
Projection to Year 2020
Number of Operating Businesses
ARIMA(1,1,0)
α=1.00

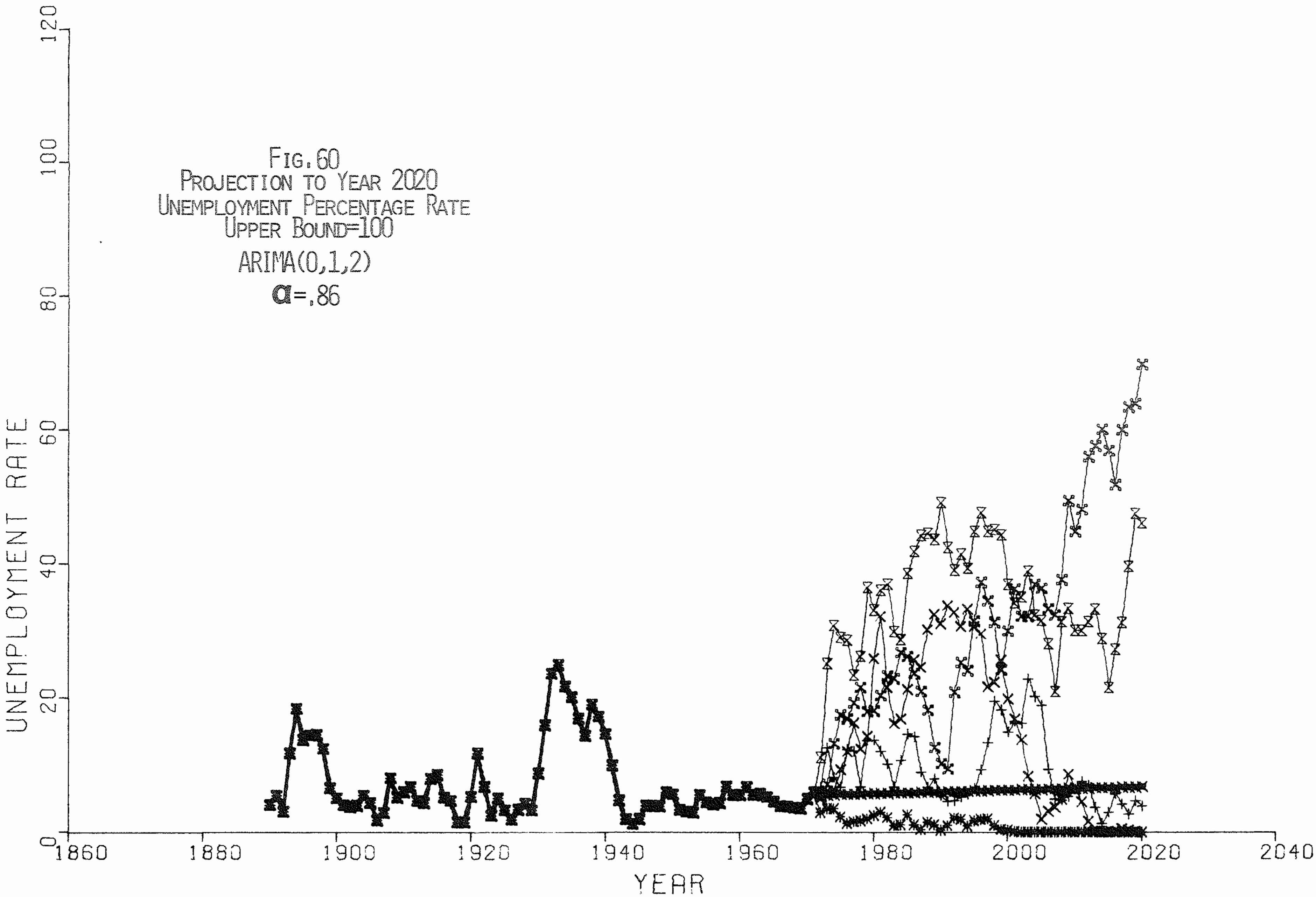


FIG. 60
PROJECTION TO YEAR 2020
UNEMPLOYMENT PERCENTAGE RATE
UPPER BOUND=100
ARIMA(0,1,2)
α=.86

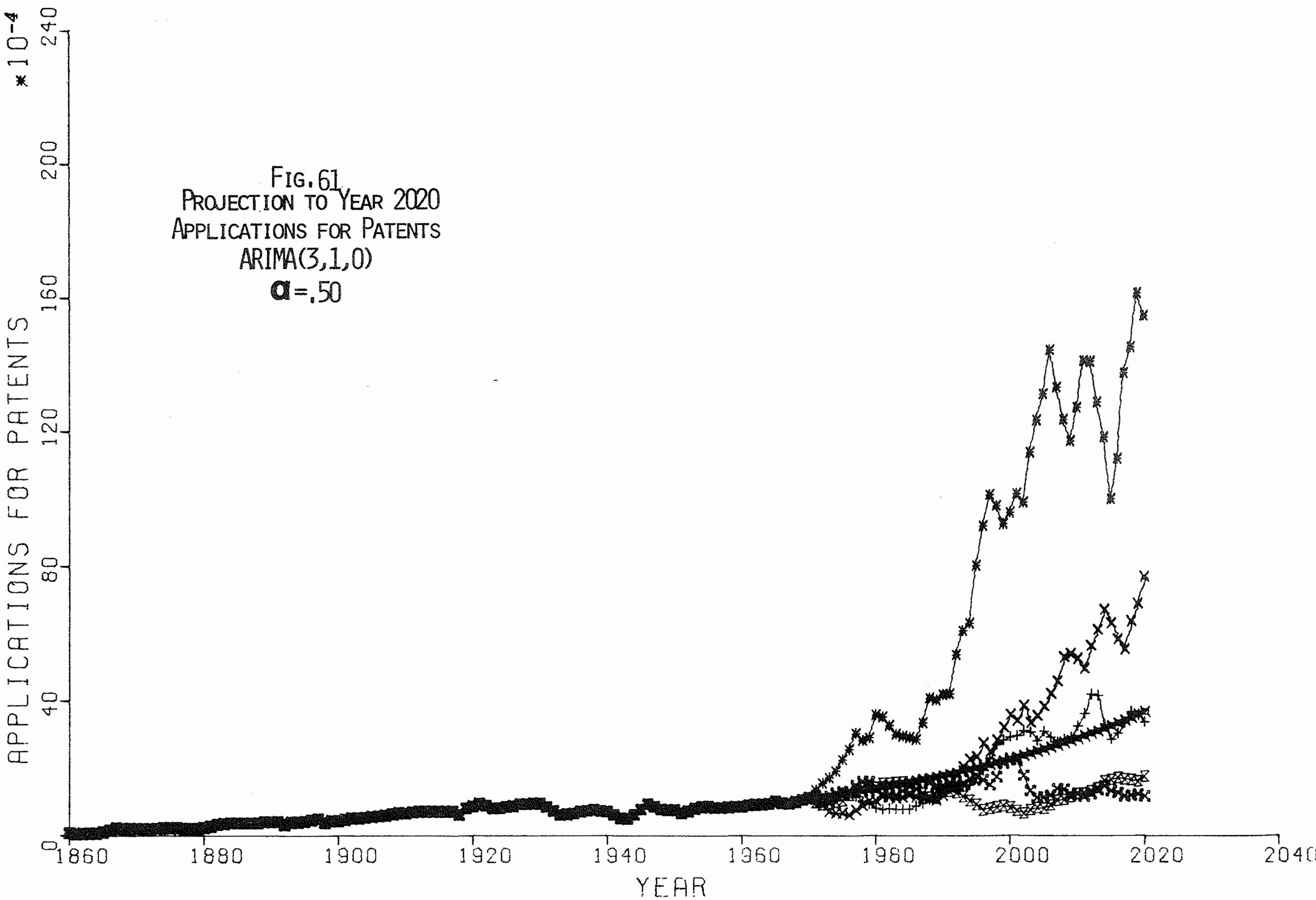
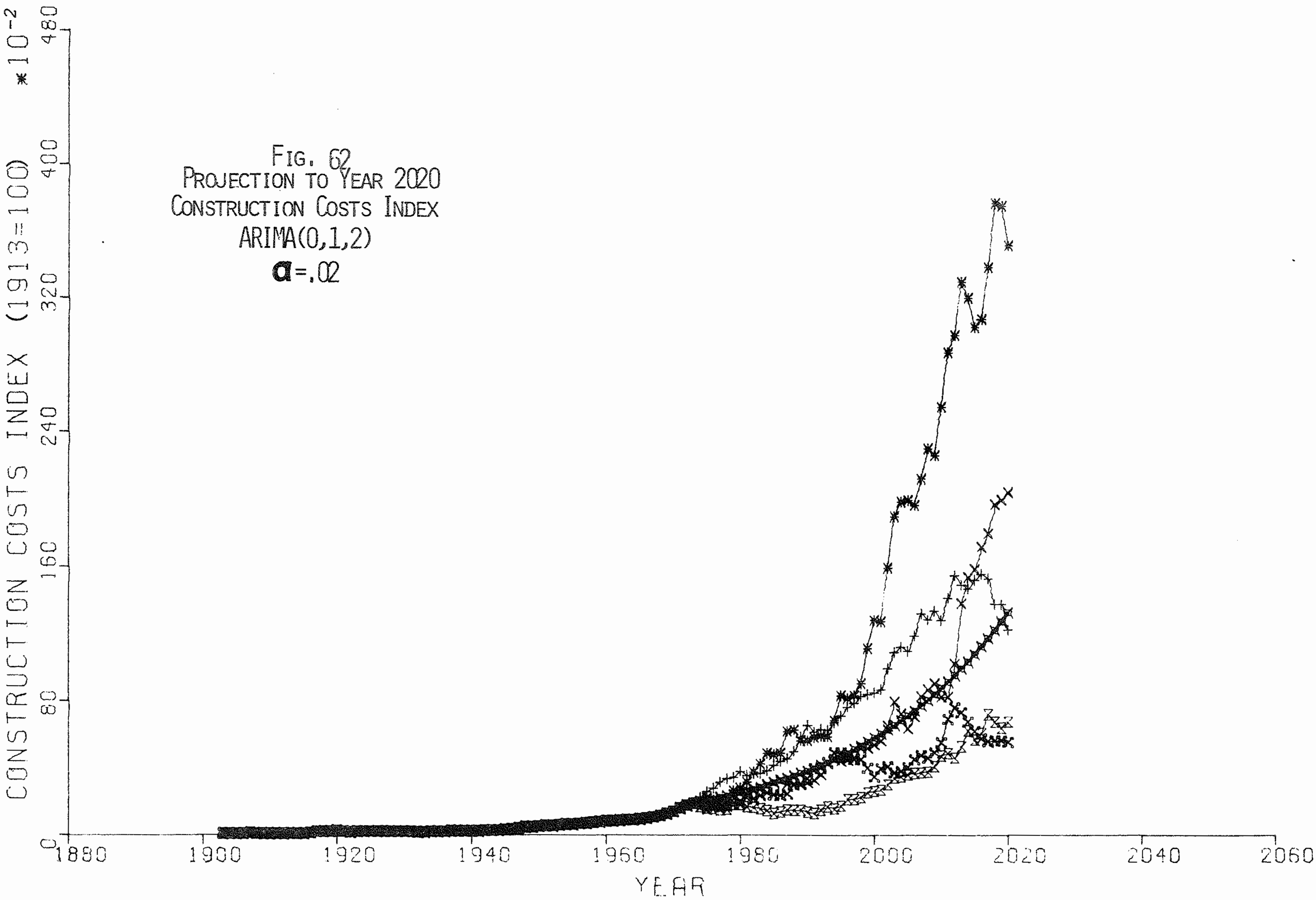


Fig. 61
Projection to Year 2020
Applications for Patents
ARIMA(3,1,0)
α =.50
APPLICATIONS FOR PATENTS
*10^-4
0
40
80
120
160
200
240
1860
1880
1900
1920
1940
1960
1980
2000
2020
2040
YEAR

Fig. 62
Projection to Year 2020
Construction Costs Index
ARIMA(0,1,2)
$\alpha = .02$



APPENDIX A

DESCRIPTION OF COMPUTER PROGRAM SVPROJ

INTRODUCTION

1. Origin of Program. This program was developed by Shin Chang of the Center for Research in Water Resources, the University of Texas at Austin under the supervision of Leo R. Beard, Technical Director. Subroutines obtained from other sources are:

   a. FTARPS, FTMAPS, FTAUTO, FTCAST, FTGENI, MDCH and their supporting subroutines FTFUNC, GGNOR, GGUI, LEQTIF, LUDATF, LUELMF, MERF, MERFI, MGAMMA, UERTST, VABMXF and ZSYSTM. All are included in the International Mathematics and Statistical Library (IMSL) computer package.

   b. PLOT was developed by Mr. David Ford and Mr. David Lott of the Center for Research in Water Resources, The University of Texas at Austin.

2. Capability of the SVPROJ Program. SVPROJ is capable of performing the following jobs:

   a. Identification of the degree of differencing, d, and number of parameters p and q of the best-fit ARIMA(p,d,q) model for time series of a single variable.

   b. Estimation of parameters of an identified ARIMA(p,d,q) model.

   c. Generation of 51 independent sets of projections using the best-fit ARIMA(p,d,q) model for a single variable.

   d. Plotting of up to 7 selected projections on one chart.

3. Hardware and Software Requirements. This program has been developed and tested on the CDC 64/6600 computer system. No tapes or disks are required. The Fortran IV complier and the IMSL library are used.

DESCRIPTION OF PROGRAM

4. Program Organization. The SVPROJ program consists of a main program and 23 subroutines, of which 18 are subroutines from IMSL. All input

data and input control parameters are read in the main program from punched cards. The program will make a normal exit (unless errors occur) only in the main program.

5. Method of Computation. The computational procedures and equations used in this program are described in chapter 4 of the main report under the title "Use of the ARIMA Model."

6. Description of Subroutines and Variables. The function of each subroutine and the definitions of variables in each subroutine are given in the program listings.

PROGRAM USE

7. Dimensional Limitations of the Program. For practical purpose, the following limitations are set in the program:

Variable	Size	Definition
LSPLIT	150	length of calibration series
LPROJ	100	length of projection
NSET	51	Sets of independent projections

These dimensions can be easily extended by increasing the dimension specification in related arrays, however, ultimate limitations are determined by the available central memory of the computer system used.

8. Control Parameters for Optional Jobs. The following parameters are used to control the execution of certain optional jobs in this program.

- IBOND: Controls the execution of lower and/or upper bound on the values of projections
- ISPLIT: Controls the execution of the split-record
- ISPY: Controls the execution of the model parameter identification for the calibration series
- LOGX: Controls the execution of natural logarithmic transformation of the calibration series.
- NPRINT: Controls the execution of the printed output of the selected projections (7 sets)

NPUNCH: Controls the execution of the punched-card output of the selected projections (7 sets).

NPLOT: Controls the execution of the plotted output of the selected projections (7 sets sets, or 5 sets).

9. Preparation Of Input Variables.

Variable		Description
IBOND:	-1	a nonzero lower bound control is used
(Requires LOGX=1)	0	a zero lower bound is used
	1	a zero lower bound and a nonzero upper bound is used
	2	both nonzero lower and upper bounds are used
ID	integer	degree of differencing, d, in an ARIMA (p,d,q) model
IP	integer	number of autoregressive parameters, p, in an ARIMA (p,d,q) model
IQ	integer	number of moving average parameters, q, in an ARIMA (p,d,q) model
ISPLIT	1	split-record test of the calibrated-ARIMA model is desired,
	0	not desired
ISPY	1	identification of ID, IP, and IQ for an ARIMA model by the program is desired
	0	ID, IP, IQ are provided by the user through input.
IYRB		beginning year of record of the calibration series
IYRE		ending year of record of the calibration series
IYRP		ending year of record of the projection series
LOGX	1	natural logarithmic transformation of the variable is desired,
NPLOT	0	no plot is desired as output
	1	plots of data and 7 projections depicted in one chart are desired
	2	plots of data and 7 projections, and data and 5 projections, depicted in 2 charts are desired

	3	plots of data and 5 projections depicted in one chart are desired
NPRINT	1	7 sets of projections of the variable being analyzed will be printed as output
	0	not desired
NPUNCH	1	7 sets of projections of the variable being analyzed will be punched as output
	0	not desired
NV	integer	number of variables to be analyzed sequentially in a multiple job run
YLABEL (8)	character	name of the variable
Z (150)	real	a vector of N elements containing data of the calibration series, and N is computed internally as IYRE-IYRB + 1
ZLO	real	lower bound of the variable being analyzed
ZUP	real	upper bound of the variable being analyzed

10. Preparation of Input Cards. The following eight types of input cards are required.

Card	Format	Variable(s)
A	16I5	NV
B	8A10	YLABEL
C	16I5	IYRB, IYRE, IYRP
D	8A10	IFMT
E	IFMT	Z
F	16I5	ISPLIT, LOGX, ISPY, IBOND, NPRINT, NPUNCH, NPLOT
G	16I5	IP, ID, IQ
H	2F20.4	ZUP, ZLO

It should be noted that the G card is not required if ISPY = 1, and the H card is not required if LOGX = 0 or IBOND = 0.

11. Arrangement of Input Cards.

a. Arrangement of the input deck for a single job is:

A,B,C,D,E,E···E,F,G,H

b. Arrangement of the input deck for a multiple job (maximum 14 jobs)

A,B,C,D,E,···E,F,G,H, B,C,D,E,···E,F,G,H

1st job 2nd job 3rd job

12. Overview of Output. The following printed output is shown for each job:

a. IP, ID, IQ of the identified ARIMA (p,d,q) model.

b. α , the significance level of goodness of fit

c. estimated parameters (Θ_o, Ψ_i and Θ_i) of the ARIMA (p,d,q) model

Printed and punched values of 7 sets of projections, and their plots in one curve are optional outputs.

13. Program Listing. Source program listing for SVPROJ and all its subroutines are shown in the following pages.

```
      PROGRAM SVPROJ(INPUT,OUTPUT,PUNCH,TAPE5=INPUT,TAPE6=OUTPUT,
     1 TAPE7=PUNCH)
      DIMENSION Z(150),ALPHA(2),DARPS(5),PMAS(5),FCST(3,100),
     1  SIM(100,51),WK(400),YLABEL(8),IPDQ(7),NODE(5),IFMT(8)
      DIMENSION ZLAST(56,14),  IND(3),NP(14),NQ(14),ND(14),IPARM(5,3)
      DIMENSION G(250,9), RANGEP(14),RLIMIT(14),ALFA(14),INDEX(55)
      DATA NODE/2,8,26,44,50/
      DOUBLE PRECISION SEED
C-----------------------------------------------------------------------
C   N              =NUMBER OF DATA FOR A VARIABLE
C   NV             =NUMBER OF VARIABLES TO BE ANALYZED
C   IYRB           =BEGINNING YEAR OF THE DATA
C   IYRE           =ENDING YEAR OF DATA
C   IYRP           =ENDING YEAR OF PROJECTION
C   LSPLIT         =FIRST LSPLIT DATA WILL BE USED TO CALIBRATE THE MODEL
C   LPROJ          =LENGTH OF PROJECTION
C   LSP            =LSPLIT+LPROJ
C   ZUP            =ASYMPTOTIC UPPER BOUND OF Z
C   ZLO            =ASYMPTOTIC LOWER BOUND OF Z
C   ISPLIT         =1, SPLIT-RECORD TEST IS DESIRED
C                   0, NOT DESIRED
C   LOGX           =1,LOGORITHMATIC TRANSFORMATION OF THE ORIGINAL VARIABL
C                     IS DESIRED
C                   0, IF NOT
C   ISPY           =1, IP(IQ( AND ID OF THE ARIMA(P,D,Q) MODEL WILL BE
C                     IDENTIFIED BY ROUTINE SPY
C                   0, IP,ID, AND IQ WILL BE INPUT BY USER
C   IBOND          =PARAMETER TO SPECIFY THE DESIRED BOUNDS CONTROL IN Z
C                  =-1, TO BOUND Z NOT LESS THAN ZLO
C                  = 0, TO BOUND Z NOT LESS THAN 0
C                  = 1, TO BOUND Z IN THE RANGE OF ZUP AND ZLO
C   NPRINT         =1, OPTIONAL PRINTED OUTPUT OF 7 PROJECTIONS AND THE
C                     MOST LIKELY FORECASTING IS DESIRED
C                   0, NOT DESIRED
C   NPUNCH         =0, NO PUNCHED OUTPUT OF FORECAST AND 7 SIMULATIONS IS
C                  =1, PUNCHED OUTPUT IS DESIRED
C   NPLOT          =PARAMETER TO SPECIFY THE DESIRED PLOTS OF PROJECTIONS
C                  =0, NO PLOT IS DESIRED
C                  =1, PLOTS OF DATA, FORECAST, AND 9 SETS OF SIMULATIONS
C                  =2,CURVES FOR NPLOT=1 AND 3 ARE BOTH DESIRED
C                  =3, PLOTS OF DATA, FORECAST AND 5 SIMULATIONS ARE DES*R
C   ALFA           =POSITIVE, SIGNIFICANCE LEVEL IN THE SUMMARY TABLE
C                  =NEGATIVE TO GIVE ERROR INDICATION
C                  =-1,0,NO SOLUTION FROM FTARPS
C                  =-2,0, ARPS OUT OF ADMISSIBLE RANGE FROM CHECK
C                  =-4,0, MAPS OUT OF ADMISSIBLE RANGE FROM CHECK
C                  =-3,0, NO SOLUTION FROM FTMAPS
C                  =-5,0, ALPHA(1) FAIL TO PASS THE MINIMUM LEVEL
C                  =-6,0, RANGE OF PROJECTION EXCEEDS RLIMIT
C   G(J,I)         -I=1,2,...,7 GIVES 7 PROJECTIONS,
C                   I=8, GIVES THE MOST LIKELY FORECASTS,
C                   I=9, GIVES THE DATA, BUT G(J,9)=G(J,8) WHEN J IS
C                     GREATER THAN N
C-----------------------------------------------------------------------
      NSET=51
      MSET=5
      MP1=MSET+1
      MP2=MSET+2
      MP3=MSET+3
      MP4=MSET+4
      READ 1390, NV
      DO 1380 ITT=1,NV
      ZLO=0.0
      ZUP=10000000000.0
      READ 1400, (YLABEL(J),J=1,8)
      READ 1390, IYRB,IYRE,IYRP
      N=IYRE-IYRB+1
      READ 1400, IFMT
      READ IFMT, (Z(J),J=1,N)
      READ 1390, ISPLIT,LOGX,ISPY,IBOND,NPRINT,NPUNCH,NPLOT
      PRINT 1390, ISPLIT,LOGX,ISPY,IBOND,NPRINT,NPUNCH,NPLOT
         IF (ISPY.EQ.0) READ 1390, IP,ID,IQ
         IF (IBOND.NE.0.AND.LOGX.NE.0) READ 1410, ZUP,ZLO
      PRINT 1400, (YLABEL(J),J=1,7)
      DO 1010 J=1,N
      G(J,MP4)=Z(J)
 1010 CONTINUE
      ZLAST(1,ITT)=Z(N)
      LSPLIT=N
      LPROJ=IYRP-IYRE
         IF (ISPLIT.EQ.0) GO TO 1020
      LSPLIT=N/2+1
      LPROJ=N-LSPLIT
 1020 CONTINUE
      LSP=LSPLIT+LPROJ
C              NATURAL-LOG TRANSFORMATION IS DESIRED
         IF (LOGX.EQ.0) GO TO 1070
      TEMP=ZUP-ZLO
      DO 1030 J=1,N
      Z(J)=Z(J)-ZLO
 1030 CONTINUE
         IF (IBOND.LE.0) GO TO 1050
      DO 1040 J=1,N
      Z(J)=ALOG(TEMP/Z(J))
 1040 CONTINUE
 1050 CONTINUE
      DO 1060 J=1,N
      Z(J)=ALOG(Z(J))
 1060 CONTINUE
 1070 CONTINUE
C             SEARCH THE RANGE OF TWO CALIBRATION SERIES
      ZMIN=Z(1)
      ZMAX=Z(1)
      DO 1080 JJ=2,LSPLIT
      IF (Z(JJ).LT.ZMIN) ZMIN=Z(JJ)
      IF (Z(JJ).GT.ZMAX) ZMAX=Z(JJ)
 1080 CONTINUE
      RC=ZMAX-ZMIN
      RLIMIT(ITT)=50.0*RC
C
C            KEEP GOING THE SEARCH TO GET THE RANGE OF THE WHOLE SERIES
C                AS RC AND SET THE RL FOR IDENTN ROUTINE
      DO 1090 JJ=LSPLIT,N
      IF (Z(JJ).GT.ZMAX) ZMAX=Z(JJ)
      IF (Z(JJ).LT.ZMIN) ZMIN=Z(JJ)
 1090 CONTINUE
      RC=ZMAX-ZMIN
      RL=50.0*RC
C            INITIALIZE PARAMETERS
C
      IPDQ(1)=1
      IPDQ(2)=LSPLIT
      IPDQ(6)=LPROJ
```

```
      DO 1110 JPC=1,NV
      REWIND ITAPE
      DO 1070 JT=1,N
      Y(JT)=C(JT,JPC)
 1070 CONTINUE
      CALL FOCAST (Y,N,JPC)
CC             STORE LAST MEMBERS OF SIMULATED P-C
      DO 1080 JS=1,NSET
      U(JS,JPC)=SIM(LPROJ,JS)
 1080 CONTINUE
      DO 1090 JT=1,LPROJ
      D(JT,MP1,JPC)=FCST(2,JT)
 1090 CONTINUE
CC             STORE ALL SIMULATIONS OF P-C IN DISK
      NTAPE=JPC+12
      DO 1100 JS=1,NSET
      WRITE (NTAPE,1430) (SIM(I,JS),I=1,LPROJ)
 1100 CONTINUE
 1110 CONTINUE
CC             CONSTRUCT LAST MEMBERS OF SIMULATED VARIABLE USING LAST
C                 MEMBERS OF SIMULATED P-C
      DO 1140 JV=1,NV
      DO 1130 JS=1,NSET
      TEMP=0.0
      DO 1120 JPC=1,NV
      TEMP=TEMP+U(JS,JPC)*Z(JV,JPC)
 1120 CONTINUE
      TEMP=TEMP*SDV(JV)+XMEAN(JV)
      IF (LOGX.NE.0) TEMP=10.0**TEMP
      ZL(JS,JV)=TEMP
 1130 CONTINUE
 1140 CONTINUE
CC             COMPUTE WEIGHTED SUM OF LAST MEMBERS OF SIMULATED VARIABLE
      DO 1160 JS=1,NSET
      TEMP=0.0
      DO 1150 JV=1,NV
      TEMP=TEMP+WT(JV)*ZL(JS,JV)
 1150 CONTINUE
      WTSUM(JS)=TEMP
 1160 CONTINUE
CC              RANK WTSUM( ) FROM MAX TO MIN
      A=WTSUM(1)
      DO 1170 JS=1,NSET
      IF (WTSUM(JS).LT.A) A=WTSUM(JS)
 1170 CONTINUE
      AMIN=A-10.0
      DO 1190 IS=1,NSET
      A=AMIN
      DO 1180 JS=1,NSET
         IF (WTSUM(JS).LT.A) GO TO 1180
      A=WTSUM(JS)
      ID=JS
 1180 CONTINUE
      WTSUM(ID)=AMIN
      INDEX(IS)=ID
 1190 CONTINUE
CC             CONSTRUCT SELECTED REPRESENTATIVES OF NSETS SIMULATED P-C
      DO 1230 JPC=1,NV
      NTAPE=JPC+12
      REWIND NTAPE
      DO 1200 JS=1,NSET
      READ (NTAPE,1430) (SIM(I,JS),I=1,LPROJ)
 1200 CONTINUE
      DO 1220 I=1,MSET
      J=NODE(I)
      JS=INDEX(J)
      DO 1210 JT=1,LPROJ
      D(JT,I,JPC)=SIM(JT,JS)
 1210 CONTINUE
 1220 CONTINUE
 1230 CONTINUE
C
C              FORECASTS AND SIMULATIONS VIA THE PROJECTED P-C
C
      DO 1410 JV=1,NV
      DO 1270 JT=1,LPROJ
      JJ=JT+LSPLIT
      DO 1250 JS=1,MSET
      TEMP=0.0
      DO 1240 JPC=1,NV
      TEMP=TEMP+D(JT,JS,JPC)*Z(JV,JPC)
 1240 CONTINUE
      TEMP=TEMP*SDV(JV)+XMEAN(JV)
      U(JJ,JS+1)=TEMP
 1250 CONTINUE
      U(JJ,1)=2.0*U(JJ,2)-U(JJ,MP1)
      U(JJ,MP2)=2.0*U(JJ,MP1)-U(JJ,2)
      TEMP=0.0
      DO 1260 JPC=1,NV
      TEMP=TEMP+D(JT,MP1,JPC)*Z(JV,JPC)
 1260 CONTINUE
      TEMP=TEMP*SDV(JV)+XMEAN(JV)
      U(JJ,MP3)=TEMP
      U(JJ,MP4)=TEMP
 1270 CONTINUE
C            SET U(JT,JS)=XX(JT,JV) AS IT SHOULD BE FOR JT=1,2,..LSPLIT
      DO 1280 JT=1,N
      U(JT,MP4)=XX(JT,JV)
 1280 CONTINUE
      DO 1300 JT=1,LSPLIT
      DO 1290 JS=1,MP3
      U(JT,JS)=U(JT,MP4)
 1290 CONTINUE
 1300 CONTINUE
         IF (LOGX.EQ.0) GO TO 1320
      DO 1310 JT=1,LSP
      DO 1310 JS=1,MP4
      U(JT,JS)=10.0**U(JT,JS)
 1310 CONTINUE
 1320 CONTINUE
      NTAPE=12+JV
      REWIND NTAPE
      DO 1330 KS=1,MP4
      WRITE (NTAPE,1430) (U(KT,KS),KT=1,LSP)
 1330 CONTINUE
      END FILE NTAPE
         IF (NPRINT.EQ.0) GO TO 1350
      PRINT 1460
      DO 1340 KS=1,MP3
      PRINT 1480, (U(KT,KS),KT=1,LSP)
 1340 CONTINUE
 1350 CONTINUE
         IF (NPUNCH.EQ.0) GO TO 1370
      DO 1360 KS=1,MP3
```

```
 1300 CONTINUE
C             STORE LAST MEMBERS OF SIMULATIONS IN ZLAST
      DO 1310 I=1,MP3
      ZLAST(I+1,ITT)=G(LSP,I)
 1310 CONTINUE
C             PUNCH SERIES NAME , 7 PROJECTIONS AND THE MOST LIKELY
C           FORECAESTS
         IF (NPUNCH.EQ.0) GO TO 1330
      WRITE (7,1400) (YLABEL(J),J=1,7)
      DO 1320 I=1,MP3
      WRITE (7,1420) (G(J,I),J=J1,LSP)
 1320 CONTINUE
 1330 CONTINUE
      PRINT 1430, (YLABEL(J),J=1,7)
      PRINT 1440
      PRINT 1450, IPDQ(3),IPDQ(5),IPDQ(4)
      PRINT 1500, ALPHA(1)
      PRINT 1460, PMAC
      PRINT 1470, (DARPS(J),J=1,4)
      PRINT 1480, (PMAS(J),J=1,4)
      PRINT 1490, WNV
      PRINT 1440
         IF (NPRINT.EQ.0) GO TO 1350
      PRINT 1510
      DO 1340 J=J1,LSP
      PRINT 1520, (G(J,I),I=1,MP3)
 1340 CONTINUE
 1350 CONTINUE
         IF (NPLOT.EQ.0) GO TO 1380
         IF (NPLOT.GT.2) GO TO 1360
      CALL PLOT (G,LSP,9,0,IYRB,YLABEL)
         IF (NPLOT.EQ.1) GO TO 1380
 1360 CONTINUE
      DO 1370 J=1,LSP
      G(J,7)=G(J,8)
      G(J,8)=G(J,9)
 1370 CONTINUE
      CALL PLOT (G(1,2),LSP,7,0,IYRB,YLABEL)
 1380 CONTINUE
         IF (LOGX.NE.0) PRINT 1530
         IF (ISPLIT.NE.0) PRINT 1540
 1390 FORMAT  (16I5)
 1400 FORMAT  (8A10)
 1410 FORMAT  (2F20.4)
 1420 FORMAT  (5F16.3)
 1430 FORMAT  (////,10X,*.......TIME SERIES--*,8A10,/)
 1440 FORMAT  (6X,*--------------------------------------------------------
     1 ---------------------*,/)
 1450 FORMAT  (12X,*(P,D,Q,) OF THE BEST-FIT ARIMA(P,D,Q) MODEL*,
     1  *IDENTIFIED BY IDENTN ARE*,3I4)
 1460 FORMAT  (14X,*TREND FACTOR(THETA ZERO)=*,1F8.4)
 1470 FORMAT  (14X,*AUTOREGRESSIVE PARAMETERS (SI(I))=*,5F8.4)
 1480 FORMAT  (14X,*MOVING AVERAGE PARAMETERS (THETA)=*,5F8.4)
 1490 FORMAT  (14X,*VARIANCE OF THE WHITE NOISE=*,5F10.4)
 1500 FORMAT  (14X,*LEVEL OF SIGNIFICANCE=*,1F5.2)
 1510 FORMAT  ( /,10X,*7 PROJECTIONS AND THE MOST LIKELY FORECASTING*)
 1520 FORMAT  (10X,12F10.2)
 1530 FORMAT  (    7X,*...ALOG-TRANSFORMATION IS USED* /)
 1540 FORMAT  (    7X,*...RECORD-SPLIT TEST* /)
      END
```

```
      SUBROUTINE CHECK (N,X,IER)
C----------------------------------------------------------------------------
C   FUNCTION      -CHECK VALUES OF PARAMETERS (EITHER AUTOREFRESSIVE OR
C                    MOVING AVERAGE TYPE) TO SEE IF THEY ARE IN THE
C                    ADMISSIBLE REGION
C         N       -NUMBER OF PARAMETERS (EITHER P OR Q OF THE ARIMA
C                    (P,D,Q) MODEL)
C         X       -INPUT VECTOR OF LENGTH N
C       IER       -0, PARAMETERS ARE IN THE ADMISSIBLE REGION
C                  139, IF NOT
C----------------------------------------------------------------------------
      DIMENSION X(3)
         IF (N.EQ.0) GO TO 1030
      ONE=1.
      IER=0
      TP=X(N)
         IF (ABS(TP).GT.1.) GO TO 1020
      TMP=0.
      TP=0.
      DO 1010 I=1,N
      TP=TP+X(I)
      ONE=-1.*ONE
      TMP=TMP+X(I)*ONE
 1010 CONTINUE
         IF (TP.GT.1..OR.TMP.GT.1.) GO TO 1020
         IF (N.LT.3) GO TO 1030
      TP=X(3)*(X(3)-X(1))-X(2)
         IF (TP.GT.1.) GO TO 1020
         GO TO 1030
 1020 CONTINUE
      IER=139
 1030 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE FINDD (U,N,M,NDD,EPSON)
C----------------------------------------------------------------------------
C   FUNCTION          -SEARCH THE ORDER OF DIFFERENCING D IN AN ARIMA(P,
C                        D,Q) MODEL SO THAT THE DIFFERENCED SERIES BECOMES
C                        STATIONARY
C     U               -INPUT TIME SERIES
C     N               -LENGTH OF VECTOR U
C     M               -NO. OF AUTOCORRELATION COEFFICIENTS USED IN THE
C                        CHI-SQUARE STATISTIC
C   NDD               -OUTPUT OPTIMAL ORDER OF DIFFERENCING D
C   EPSON             -CRITERION USED IN SELECTING NDD
C
C
C----------------------------------------------------------------------------
      DIMENSION  U(1),CHI(3),AC(36),ACV(36),PACV(36),WKAREA(120),X(120)
      DO 1010 J=1,N
      X(J)=U(J)
 1010 CONTINUE
      NN=N
      DO 1040 I=1,3
      CALL FTAUTO (X,NN,M,0.5,AMEAN,VAR,ACV,AC,PACV,WKAREA)
      TEMP=0.0
      DO 1020 J=1,M
      TEMP=TEMP+AC(J)*AC(J)
 1020 CONTINUE
      CHI(I)=TEMP*NN
C           DEFFERENCING THE SERIES
```

```
      NN=N-I
      DO 1030 K=1,NN
      X(K)=X(K+1)-X(K)
 1030 CONTINUE
 1040 CONTINUE
C               CALCULATE RATIOS R1 AND R2
      R1=CHI(2)/CHI(1)
      R2=CHI(3)/CHI(2)
C               SELECT NDD
         IF (R1.GT.EPSON) GO TO 1050
      NDD=1
      RETURN
 1050 CONTINUE
         IF (R2-EPSON) 1060,1060,1070
 1060 CONTINUE
      NDD=2
      RETURN
 1070 CONTINUE
      NDD=0
      RETURN
      END

      SUBROUTINE FTARPS (ACV,WBAR,IP,IQ,ARPS,PMAC,WA,IER)
C
C-FTARPS--------S--------LIBRARY 3--------------------------------------
C
C   FUNCTION            - PRELIMINARY ESTIMATION OF THE AUTOREGRESSIVE
C                            PARAMETERS IN AN ARIMA STOCHASTIC MODEL.
C   USAGE               - CALL FTARPS (ACV,WBAR,IP,IQ,ARPS,PMAC,WA,IER)
C   PARAMETERS   ACV(ID)- INPUT VECTOR OF AUTOCOVARIANCES OF TIME SERIES
C                            BEING MODELED. THE LENGTH ID=IP+IQ+1. ACV(I)
C                            IS THE AUTOCOVARIANCE CORRESPONDING TO A
C                            TIME LAG OF I-1 UNITS OF TIME. THE LENGTH ID
C                            IS NOT A PARAMETER IN THE CALLING SEQUENCE.
C                WBAR   - MEAN OF THE TIME SERIES (INPUT).
C                IP     - INPUT NUMBER OF AUTOREGRESSIVE PARAMETERS IN
C                            THE MODEL. IP MUST BE GREATER THAN OR
C                            EQUAL TO 1.
C                IQ     - INPUT NUMBER OF MOVING AVERAGE PARAMETERS IN
C                            THE MODEL.
C                ARPS   - OUTPUT VECTOR OF LENGTH IP.  ARPS CONTAINS
C                            THE PRELIMINARY ESTIMATES OF THE
C                            AUTOREGRESSIVE PARAMETERS OF THE MODEL.
C                PMAC   - OVERALL MOVING AVERAGE CONSTANT(OUTPUT).
C                WA(IE) - WORK AREA VECTOR OF LENGTH IE=IP**2+5*IP
C                            THE LENGTH IE IS NOT A PARAMETER IN THE
C                            CALLING SEQUENCE.
C                IER    - ERROR PARAMETER
C                         TERMINAL ERROR = 128 + N
C                            N = 1 INDICATES THE INPUT DATA GENERATED
C                              A SINGULAR SYSTEM. THIS USUALLY INDICATES
C                              INVALID VALUES IN THE ACV VECTOR OR A
C                              TIME SERIES WHICH IS NONSTATIONARY.
C   PRECISION           - SINGLE
C   REQD. IMSL ROUTINES - LEQT1F,LUDATF,LUELMF,UERTST,VABMXF
C   LANGUAGE            - FORTRAN
C-----------------------------------------------------------------------
C   LATEST REVISION     - MARCH 17, 1972
C
      DIMENSION          ACV(1),ARPS(1),WA(1)
      DATA               ZERO,ONE/0.,1./,IDGT/4/
      K = IQ + 1
      IER = 0
      IF (IP .GT. 2) GO TO 10
      IF (IP .EQ. 2) GO TO 5
      IF (ACV(K) .EQ. ZERO) GO TO 45
C                                  1 BY 1 SYSTEM
      ARPS(1) = ACV(K+1)/ACV(K)
      GO TO 35
C                                  2 BY 2 SYSTEM
    5 L = IABS(IQ-1) + 1
      DET = ACV(K)**2 - ACV(L)*ACV(K+1)
      IF (DET .EQ. ZERO) GO TO 45
      ARPS(1) = (ACV(K+1)*ACV(K) - ACV(L)*ACV(K+2))/DET
      ARPS(2) = (ACV(K)*ACV(K+2) - ACV(K+1)**2)/DET
      GO TO 35
C                                  SET UP MATRIX OF AUTO-
C                                  COVARIANCES (3 BY 3 OR MORE SYSTEM)
   10 KK = IP**2
      DO 15 I = 1,IP
         WA(I) = ACV(IQ+I)
         WA(KK+I) = ACV(K+I)
   15 CONTINUE
      M = 0
      N = IP + 1
      DO 25 J = 2,IP
         DO 20 I = 2,IP
            M = M+1
            N = N+1
            WA(N) = WA(M)
   20    CONTINUE
         M = M + 1
         N = N + 1
         KTY = IABS(K-J)+1
         WA(M+1) = ACV(KTY)
   25 CONTINUE
      N = KK + IP + 1
      CALL VABMXF(WA(KK+1),IP,1,J,VMAX1)
C                                  COMPUTE THE SOLUTION
      CALL LEQT1F(WA,1,IP,IP,WA(KK+1),IDGT,WA(N),IER)
      IF (IER .NE. 0) GO TO 45
      CALL VABMXF(WA(KK+1),IP,1,J,VMAX2)
      IF(VMAX2.GT.100.*VMAX1) GO TO 45
C                                  MOVE THE SOLUTION TO ARPS
      DO 30  I = 1,IP
         ARPS(I) = WA(KK+I)
   30 CONTINUE
C                                  COMPUTE PMAC FORM ARPS
   35 PMAC = ONE
      DO 40 I=1,IP
         PMAC = PMAC - ARPS(I)
   40 CONTINUE
      PMAC = PMAC*WBAR
      GO TO 9005
   45 IER = 129
 9000 CONTINUE
      CALL UERTST(IER,6HFTARPS)
 9005 RETURN
      END


      SUBROUTINE FTAUTO (M,LW,K,L,ISW,AMEAN,VAR,ACV,AC,PACV,WKAREA)
C
C-FTAUTO--------S--------LIBRARY 3--------------------------------------
```

```
C                                                                       FTAU0040
C     FUNCTION            - GIVEN A TIME SERIES  COMPUTE                FTAU0050
C                              1, THE MEAN AND VARIANCE,                FTAU0060
C                              2, THE AUTOCOVARIANCES,                  FTAU0070
C                              3, THE AUTOCOVARIANCES AND AUTOCORRELATIONS,FTAU0080
C                              4, THE PARTIAL AUTOCORRELATIONS.         FTAU0090
C     USAGE               - CALL FTAUTO(W,LW,K,L,ISW,AMEAN,VAR,ACV,AC,  FTAU0100
C                              PACV,WKAREA)                             FTAU0110
C     PARAMETERS   W(LW)  - INPUT VECTOR, TIMES SERIES.                 FTAU0120
C                  LW     - INPUT, LENGTH OF W.                         FTAU0130
C                  K      - INPUT, NUMBER OF AUTOCOVARIANCES AND        FTAU0140
C                              AUTOCORRELATIONS TO BE COMPUTED.         FTAU0150
C                  L      - INPUT, NUMBER OF PARTIAL AUTOCORRELATIONS TO FTAU0160
C                              BE COMPUTED.  L MUST BE LESS THAN OR EQUAL FTAU0170
C                              TO K.                                    FTAU0180
C                  ISW    - INPUT, CONTROL PARAMETER FOR DETERMINING TASK FTAU0190
C                              TO BE PERFORMED, IF                      FTAU0200
C                                ISW = 1  FIND MEAN AND VARIANCE.       FTAU0210
C                                ISW = 2  FIND AUTOCOVARIANCE.          FTAU0220
C                                ISW = 3  FIND MEAN, VARIANCE, AND      FTAU0230
C                                            AUTOCOVARIANCES.           FTAU0240
C                                ISW = 4  FIND AUTOCOVARIANCES AND      FTAU0250
C                                            AUTOCORRELATIONS.          FTAU0260
C                                ISW = 5  FIND MEAN, VARIANCE, AUTO-    FTAU0270
C                                            COVARIANCES, AND AUTOCORRELAT- FTAU0280
C                                            IONS.                      FTAU0290
C                                ISW = 6  FIND AUTOCOVARIANCES, AUTOCORREL-FTAU0300
C                                            ATIONS, AND PARTIAL AUTOCO- FTAU0310
C                                            RRELATIONS.                FTAU0320
C                                ISW = 7  FIND MEAN, VARIANCE, AUTOCOVAR- FTAU0330
C                                            IANCES, AUTOCORRELATIONS, AND FTAU0340
C                                            PARTIAL AUTOCORRELATIONS.  FTAU0350
C                  AMEAN  - OUTPUT FOR ISW = 1,3,5, AND 7.              FTAU0360
C                              INPUT FOR ISW = 2,4, AND 6, MEAN VALUE OF FTAU0370
C                              THE TIME SERIES W.                       FTAU0380
C                  VAR    - OUTPUT FOR ISW = 1,3,5, AND 7.              FTAU0390
C                              INPUT FOR ISW = 2,4, AND 6, VARIANCE OF  FTAU0400
C                              TIME SERIES W.                           FTAU0410
C                  ACV(K) - OUTPUT FOR ISW = 2,3,4,5,6 AND 7.           FTAU0420
C                              AUTOCOVARIANCES FOR TIME SERIES W.  ACV(I) FTAU0430
C                              CORRESPONDS TO A TIME LAG OF I TIME UNITS. FTAU0440
C                  AC(K)  - OUTPUT FOR ISW = 4,5,6, AND 7.              FTAU0450
C                              AUTOCORRELATIONS FOR TIME SERIES W.  AC(I) FTAU0460
C                              CORRESPONDS TO A TIME LAG OF I TIME UNITS. FTAU0470
C                  PACV(L)- OUTPUT FOR ISW = 6 AND 7.                   FTAU0480
C                              PARTIAL AUTOCORRELATIONS OF TIME SERIES W. FTAU0490
C                              PACV(1) = AC(1).                         FTAU0500
C                  WKAREA - WORK AREA VECTOR OF LENGTH L.               FTAU0510
C     PRECISION           - SINGLE                                      FTAU0520
C     LANGUAGE            - FORTRAN                                     FTAU0530
C-----------------------------------------------------------------------FTAU0540
C     LATEST REVISION     - MARCH 31, 1972                              FTAU0550
C                                                                       FTAU0560
      DIMENSION          W(1),ACV(1),AC(1),PACV(1),WKAREA(1)            FTAU0570
      DOUBLE PRECISION   TEMP,TEMP1                                     FTAU0580
      IW = (ISW/2)*2 - ISW                                              FTAU0590
C                                COMPUTE THE MEAN                       FTAU0600
      IF (IW .EQ. 0) GO TO 15                                           FTAU0610
      TEMP = 0.0D0                                                      FTAU0620
      DO 5 I=1,LW                                                       FTAU0630
    5 TEMP = TEMP + DBLE(W(I))                                          FTAU0640
      AMEAN = TEMP/LW                                                   FTAU0650
C                                COMPUTE THE VARIANCE                   FTAU0660
      TEMP = 0.0D0                                                      FTAU0670
      DO 10 I=1,LW                                                      FTAU0680
   10 TEMP = TEMP + (DBLE(W(I)-AMEAN)*DBLE(W(I)-AMEAN))                 FTAU0690
      VAR = TEMP/LW                                                     FTAU0700
      IF (ISW .EQ. 1) GO TO 9005                                        FTAU0710
C                                COMPUTE AUTOCOVARIANCES                FTAU0720
   15 DO 25 J=1,K                                                       FTAU0730
         KEND = LW-J                                                    FTAU0740
         TEMP = 0.0D0                                                   FTAU0750
         DO 20 I=1,KEND                                                 FTAU0760
   20    TEMP = TEMP+(DBLE(W(I)-AMEAN)*DBLE(W(I+J)-AMEAN))              FTAU0770
         ACV(J) = TEMP/LW                                               FTAU0780
   25 CONTINUE                                                          FTAU0790
      IF (ISW .LT. 4) GO TO 9005                                        FTAU0800
C                                COMPUTE AUTOCORRELATIONS               FTAU0810
      DO 30 J=1,K                                                       FTAU0820
   30 AC(J) = ACV(J)/VAR                                                FTAU0830
      IF (ISW .LT. 6) GO TO 9005                                        FTAU0840
C                                COMPUTE PARTIAL AUTOCOVARIANCE         FTAU0850
      PACV(1) = AC(1)                                                   FTAU0860
       DO 55 J=2,L                                                      FTAU0870
         J1 = J-1                                                       FTAU0880
         WKAREA(J1) = PACV(J1)                                          FTAU0890
         J2 = (J1)/2                                                    FTAU0900
         IF(J .EQ. 2) GO TO 40                                          FTAU0910
         DO 35 K0=1,J2                                                  FTAU0920
            J1MK = J1-K0                                                FTAU0930
            TEMP2 = WKAREA(K0)-PACV(J1)*WKAREA(J1MK)                    FTAU0940
            WKAREA(J1MK) = WKAREA(J1MK)-PACV(J1)*WKAREA(K0)             FTAU0950
   35    WKAREA(K0) = TEMP2                                             FTAU0960
   40    TEMP = 0.0D0                                                   FTAU0970
         TEMP1 = 0.0D0                                                  FTAU0980
         DO 45 I=1,J1                                                   FTAU0990
            TEMP = TEMP + (DBLE(AC(J-I)) * DBLE(WKAREA(I)))             FTAU1000
   45    TEMP1 = TEMP1 + (DBLE(AC(I)) * DBLE(WKAREA(I)))                FTAU1010
         PACV(J) = (AC(J)-TEMP)/(1.0D0-TEMP1)                           FTAU1020
   55 CONTINUE                                                          FTAU1030
 9005 RETURN                                                            FTAU1040
      END                                                               FTAU1050
```

```
      SUBROUTINE FTCAST (Z,ARPS,PMAS,PMAC,ALPHA,LV,DARPS,FCST,WNV,IER)  FTCA0010
C                                                                       FTCA0020
C-FTCAST--------S--------LIBRARY 3--------------------------------------FTCA0030
C                                                                       FTCA0040
C   FUNCTION            - USING A FITTED ARIMA STOCHASTIC MODEL, COMPUTEFTCA0050
C                           TIME SERIES FORECASTS AND PROBABILITY LIMITSFTCA0060
C                           FOR LEAD TIMES 1,2,3,...,LV(5).             FTCA0070
C   USAGE               - CALL FTCAST(Z,ARPS,PMAS,PMAC,ALPHA,LV,DARPS,  FTCA0080
C                           FCST,WNV,IER)                               FTCA0090
C   PARAMETERS   Z      - INPUT VECTOR OF LENGTH LV(1) CONTAINING THE   FTCA0100
C                           TIME SERIES.                                FTCA0110
C                ARPS   - INPUT VECTOR OF LENGTH LV(2) CONTAINING       FTCA0120
C                           ESTIMATES OF THE AUTOREGRESSIVE PARAMETERS. FTCA0130
C                PMAS   - INPUT VECTOR OF LENGTH 2*LV(3) CONTAINING     FTCA0140
C                           ESTIMATES OF THE MOVING AVERAGE PARAMETERS  FTCA0150
C                           IN THE FIRST LV(3) LOCATIONS.  THE REMAININGFTCA0160
C                           LOCATIONS ARE WORK STORAGE.                 FTCA0170
C                PMAC   - INPUT.  ESTIMATE OF OVERALL MOVING AVERAGE    FTCA0180
```

```
C                          CONSTANT.                                FTCA0190
C              ALPHA  - INPUT.  A VALUE IN THE EXCLUSIVE INTERVAL    FTCA0200
C                          (0,1) USED FOR COMPUTING 100(1-ALPHA) PER FTCA0210
C                          CENT PROBABILITY LIMITS FOR THE FORECASTS. FTCA0220
C                          THE VALUE 0.05 IS A COMMON CHOICE.        FTCA0230
C              LV     - INPUT VECTOR OF LENGTH 5. LV(I) CONTAINS, WHENFTCA0240
C                          I = 1, LENGTH OF TIME SERIES Z.           FTCA0250
C                          I = 2, NUMBER OF AUTOREGRESSIVE PARAMETERS FTCA0260
C                            IN THE MODEL.                           FTCA0270
C                          I = 3, NUMBER OF MOVING AVERAGE PARAMETERS FTCA0280
C                            IN THE MODEL.                           FTCA0290
C                          I = 4, NUMBER OF DIFFERENCING OPERATIONS  FTCA0300
C                            REQUIRED TO OBTAIN THE SERIES USED IN   FTCA0310
C                            FITTING THE ARIMA MODEL.                FTCA0320
C                          I = 5, MAXIMUM LEAD TIME DESIRED FOR A    FTCA0330
C                            FORECAST.                               FTCA0340
C              DARPS  - OUTPUT VECTOR OF LENGTH LV(2)+LV(4) CONTAININGFTCA0350
C                          THE CONSTANTS, CORRESPONDING TO THOSE     FTCA0360
C                          IN ARPS, FOR THE UNDIFFERENCED FORM OF    FTCA0370
C                          THE MODEL.                                FTCA0380
C              FCST   - OUTPUT MATRIX OF DIMENSION 3 BY LV(5),       FTCA0390
C                          FCST(I,J), FOR LEAD TIMES J=1,2,3,...,LV(5),FTCA0400
C                          CONTAINS WHEN                             FTCA0410
C                            I = 1, THE WEIGHTS FOR THE WEIGHTED SUM FTCA0420
C                              OF SHOCKS THAT GIVES THE FORECAST ERROR.FTCA0430
C                            I = 2, THE FORECASTS.                   FTCA0440
C                            I = 3, THE CORRESPONDING DEVIATIONS FROM FTCA0450
C                              EACH FORECAST FOR THE PROBABILITY     FTCA0460
C                              LIMITS.                               FTCA0470
C              WNV    - OUTPUT.  ESTIMATE OF WHITE NOISE VARIANCE.   FTCA0480
C              IER    - ERROR PARAMETER.                             FTCA0490
C                        TERMINAL ERROR = 128 + N                    FTCA0500
C                          N = 1 INDICATES PARAMETER ALPHA WAS NOT IN FTCA0510
C                            THE EXCLUSIVE INTERVAL (0,1).           FTCA0520
C                          N = 2 INDICATES ONE OR MORE OF LV(2), LV(3),FTCA0530
C                            OR LV(4) WERE LESS THAN ZERO OR LV(5) WAS FTCA0540
C                            LESS THAN ONE.                          FTCA0550
C                          N = 3 INDICATES LV(1) IS LESS THAN OR EQUAL FTCA0560
C                            TO LV(2)+LV(3)+LV(4).                   FTCA0570
C   PRECISION           - SINGLE                                     FTCA0580
C   REQD. IMSL ROUTINES - MERFI,UERTST                               FTCA0590
C   LANGUAGE            - FORTRAN                                    FTCA0600
C-------------------------------------------------------------------FTCA0610
C   LATEST REVISION     - DECEMBER 6,1974                            FTCA0620
C                                                                    FTCA0630
      DIMENSION        Z(1),ARPS(1),PMAS(1),LV(5),DARPS(1),FCST(3,1) FTCA0640
      DOUBLE PRECISION TEMP,TEMP1,TEMP2,S                            FTCA0650
      DATA             ONEN/-1.0/,ZERO/0.0/                          FTCA0660
C                                CHECK FOR ERRORS IN LV.             FTCA0670
      IF(LV(1) .LE. LV(2)+LV(3)+LV(4)) GO TO 205                     FTCA0680
      IF(LV(2).LT.0 .OR. LV(3).LT.0 .OR. LV(4).LT.0 .OR. LV(5).LT.1) FTCA0690
     *   GO TO 200                                                   FTCA0700
      LV2M = LV(2)-1                                                 FTCA0710
      IPD = LV(2)+LV(4)                                              FTCA0720
      LV1 = LV(1)                                                    FTCA0730
      LV2 = LV(2)                                                    FTCA0740
      LV3 = LV(3)                                                    FTCA0750
      LV4 = LV(4)                                                    FTCA0760
      LV5 = LV(5)                                                    FTCA0770
      LV24P = IPD+1                                                  FTCA0780
C                                COMPUTE THE CONSTANTS FOR THE       FTCA0790
C                                DIFFERENCE EQUATION FORM OF THE MODELFTCA0800
```

```
C                                CORRESPONDING TO THOSE IN ARPS      FTCA0810
      IF(LV2 .NE. 0) GO TO 20                                        FTCA0820
      IF(LV4 .EQ. 0) GO TO 30                                        FTCA0830
      LM = LV4-1                                                     FTCA0840
      DARPS(1) = -ONEN                                               FTCA0850
      IF(LM .EQ. 0) GO TO 30                                         FTCA0860
      ONE = ONEN                                                     FTCA0870
      DO 15 I=1,LM                                                   FTCA0880
         L = I                                                       FTCA0890
         IF(I .EQ. 1) GO TO 10                                       FTCA0900
         DO 5 J=2,I                                                  FTCA0910
            DARPS(L) = DARPS(L)-DARPS(L-1)                           FTCA0920
    5    L = L-1                                                     FTCA0930
   10    DARPS(I+1) = ONE                                            FTCA0940
         ONE = -ONE                                                  FTCA0950
   15 DARPS(1) = DARPS(1)-ONEN                                       FTCA0960
      DARPS(LV4) = ONEN                                              FTCA0970
      IF(DARPS(LM) .LT. ZERO) DARPS(LV4) = -DARPS(LV4)               FTCA0980
      GO TO 30                                                       FTCA0990
   20 DO 22 I=1,LV2                                                  FTCA1000
   22 DARPS(I) = ARPS(I)                                             FTCA1010
      IF(LV4 .EQ. 0) GO TO 30                                        FTCA1020
      LL = LV2+1                                                     FTCA1030
      ARP = ARPS(LV2)                                                FTCA1040
      DO 27 I=1,LV4                                                  FTCA1050
         DARPS(LL) = -ARP                                            FTCA1060
         L = LL-1                                                    FTCA1070
         M = LV2M+I                                                  FTCA1080
         IF(M .EQ. 1) GO TO 26                                       FTCA1090
         DO 24 J=2,M                                                 FTCA1100
            DARPS(L) = DARPS(L)-DARPS(L-1)                           FTCA1110
   24    L = L-1                                                     FTCA1120
   26    DARPS(1) = DARPS(1)-ONEN                                    FTCA1130
         ARP = -ARP                                                  FTCA1140
   27 LL = LL+1                                                      FTCA1150
      DARPS(LL-1) =  ARP                                             FTCA1160
   30 K = MIN0(LV(5),LV(3))                                          FTCA1170
      K1 = MAX0(LV(5),LV(3))                                         FTCA1180
      MQL = K                                                        FTCA1190
      MXQL = K1                                                      FTCA1200
      DO 35 I=1,LV5                                                  FTCA1210
   35 FCST(1,I) = ZERO                                               FTCA1220
C                                COMPUTE THE WEIGHTS FOR THE WEIGHTED FTCA1230
C                                SUM OF SHOCKS THAT GIVES THE FORECASTFTCA1240
C                                ERROR                               FTCA1250
      IF(K .EQ. 0) GO TO 45                                          FTCA1260
      DO 40 I=1,K                                                    FTCA1270
   40 FCST(1,I) = -PMAS(I)                                           FTCA1280
   45 K1 = LV24P                                                     FTCA1290
      KI = MIN0(IPD,LV5)                                             FTCA1300
      IF(KI .EQ. 0) GO TO 52                                         FTCA1310
      DO 50 I=1,KI                                                   FTCA1320
   50 FCST(1,I) = FCST(1,I)+DARPS(I)                                 FTCA1330
   52 K = MIN0(K1,LV(5))                                             FTCA1340
      K1 = LV5                                                       FTCA1350
      IF(K .LT. 2) GO TO 80                                          FTCA1360
      DO 60 I=2,K                                                    FTCA1370
         L = I-1                                                     FTCA1380
         TEMP = 0.0D0                                                FTCA1390
         DO 55 J=1,L                                                 FTCA1400
            L1 = I-J                                                 FTCA1410
   55    TEMP = TEMP+(DBLE(DARPS(J))*DBLE(FCST(1,L1)))               FTCA1420
```

```
   60 FCST(1,I) = FCST(1,I)+TEMP                                        FTCA1430
      IF(LV24P .GE. LV5) GO TO 80                                       FTCA1440
      KK = LV24P+1                                                      FTCA1450
      DO 75 I=KK,LV5                                                    FTCA1460
         TEMP = 0.0D0                                                   FTCA1470
         DO 70 J=1,IPD                                                  FTCA1480
            L1 = I-J                                                    FTCA1490
            IF(L1 .NE. 0) GO TO 65                                      FTCA1500
            TEMP = TEMP+DBLE(DARPS(J))                                  FTCA1510
            GO TO 70                                                    FTCA1520
   65       TEMP = TEMP+(DBLE(DARPS(J))*DBLE(FCST(1,L1)))               FTCA1530
   70    CONTINUE                                                       FTCA1540
   75 FCST(1,I) = FCST(1,I)+TEMP                                        FTCA1550
   80 TEMP = 0.0D0                                                      FTCA1560
      K = K-1                                                           FTCA1570
      IQ = LV(3)-1                                                      FTCA1580
      IQQ = LV(3)                                                       FTCA1590
      IQP = IQQ+1                                                       FTCA1600
      IQ22 = IQQ+IQQ                                                    FTCA1610
      IF(IQQ .EQ. 0) GO TO 90                                           FTCA1620
      DO 85 I=IQP,IQ22                                                  FTCA1630
   85 PMAS(I) = ZERO                                                    FTCA1640
C                                  COMPUTE THE ESTIMATE OF WHITE NOISE  FTCA1650
C                                  VARIANCE                             FTCA1660
   90 IF(LV1 .LT. LV24P) GO TO 125                                      FTCA1670
      DO 120 I=LV24P,LV1                                                FTCA1680
         TEMP1 = 0.0D0                                                  FTCA1690
         TEMP2 = 0.0D0                                                  FTCA1700
         IF(IPD .EQ. 0) GO TO 100                                       FTCA1710
         DO 95 J=1,IPD                                                  FTCA1720
   95    TEMP1 = TEMP1+(DBLE(DARPS(J))*DBLE(Z(I-J)))                    FTCA1730
  100    IF(IQQ .LT. 1) GO TO 115                                       FTCA1740
         IQ2 = IQ22                                                     FTCA1750
         DO 105 J=1,IQQ                                                 FTCA1760
  105    TEMP2 = TEMP2+(DBLE(PMAS(J))*DBLE(PMAS(J+IQQ)))                FTCA1770
         IF(IQ .LT. 1) GO TO 115                                        FTCA1780
         DO 110 J=1,IQ                                                  FTCA1790
            IQ2 = IQ2-1                                                 FTCA1800
  110    PMAS(IQ2+1) = PMAS(IQ2)                                        FTCA1810
  115    PMAS(IQP) = Z(I)-PMAC-TEMP1+TEMP2                              FTCA1820
  120 TEMP = TEMP+DBLE(PMAS(IQP))**2                                    FTCA1830
  125 WNV = TEMP/LV1                                                    FTCA1840
      TA = 1.0-ALPHA/2.0                                                FTCA1850
      CALL MDNRIS(TA,X,IER)                                             FTCA1860
      IF(IER .GT. 127) GO TO 195                                        FTCA1870
      S = X*SQRT(WNV)                                                   FTCA1880
C                                  COMPUTE THE CORRESPONDING DEVIATIONS FTCA1890
C                                  FROM EACH FORECAST FOR THE           FTCA1900
C                                  PROBABILITY LIMITS                   FTCA1910
      FCST(3,1) = S                                                     FTCA1920
      IF(LV5 .EQ. 1) GO TO 140                                          FTCA1930
      DO 135 I=2,LV5                                                    FTCA1940
         TEMP = 0.0D0                                                   FTCA1950
         L = I-1                                                        FTCA1960
         DO 130 J=1,L                                                   FTCA1970
  130    TEMP = TEMP+(DBLE(FCST(1,J))*DBLE(FCST(1,J)))                  FTCA1980
  135 FCST(3,I) = S*DSQRT(1.0D0+TEMP)                                   FTCA1990
  140 K = KI                                                            FTCA2000
      IF(K .EQ. 0) GO TO 155                                            FTCA2010
C                                  COMPUTE THE FORECASTS                FTCA2020
      DO 150 I=1,K                                                      FTCA2030
         N1 = LV1+I                                                     FTCA2040
         TEMP = 0.0D0                                                   FTCA2050
         IF(I .GT. IPD) GO TO 150                                       FTCA2060
         DO 145 J=I,IPD                                                 FTCA2070
  145    TEMP = TEMP+(DBLE(DARPS(J))*DBLE(Z(N1-J)))                     FTCA2080
  150 FCST(2,I) = PMAC+TEMP                                             FTCA2090
  155 K = K+1                                                           FTCA2100
      IF(K .GT. K1) GO TO 165                                           FTCA2110
      DO 160 I=K,K1                                                     FTCA2120
  160 FCST(2,I) = PMAC                                                  FTCA2130
  165 IF(MQL .EQ. 0) GO TO 180                                          FTCA2140
      LV3P = LV3+1                                                      FTCA2150
      DO 175 I=1,MQL                                                    FTCA2160
         IF(I .GT. LV(3)) GO TO 180                                     FTCA2170
         TEMP = 0.0D0                                                   FTCA2180
         DO 170 J=I,LV3                                                 FTCA2190
  170    TEMP = TEMP+DBLE(PMAS(J))*DBLE(PMAS(J-I+LV3P))                 FTCA2200
         FCST(2,I) = FCST(2,I)-TEMP                                     FTCA2210
  175 CONTINUE                                                          FTCA2220
  180 IF(LV5 .EQ. 1) GO TO 9005                                         FTCA2230
      DO 190 I=2,LV5                                                    FTCA2240
         KM1 = I-1                                                      FTCA2250
         TEMP = 0.0D0                                                   FTCA2260
         IF(IPD .EQ. 0) GO TO 9005                                      FTCA2270
         IF(KM1 .GT. IPD) KM1 = IPD                                     FTCA2280
         DO 185 J=1,KM1                                                 FTCA2290
            IJ = I-J                                                    FTCA2300
            TEMP = TEMP+(DBLE(DARPS(J))*DBLE(FCST(2,IJ)))               FTCA2310
  185    CONTINUE                                                       FTCA2320
         FCST(2,I) = FCST(2,I)+TEMP                                     FTCA2330
  190 CONTINUE                                                          FTCA2340
      GO TO 9005                                                        FTCA2350
  195 IER = 129                                                         FTCA2360
      GO TO 9000                                                        FTCA2370
  200 IER = 130                                                         FTCA2380
      GO TO 9000                                                        FTCA2390
  205 IER = 131                                                         FTCA2400
 9000 CONTINUE                                                          FTCA2410
      CALL UERTST(IER,6HFTCAST)                                         FTCA2420
 9005 CONTINUE                                                          FTCA2430
      RETURN                                                            FTCA2440
      END                                                               FTCA2450


      FUNCTION   FTFUNC (X,I,PAR)                                       FTFU0010
C                                                                       FTFU0020
C-FTFUNC--------S--------LIBRARY 3---------------------------------------FTFU0030
C                                                                       FTFU0040
C   FUNCTION            - TO PROVIDE FUNCTIONAL COMMUNICATION BETWEEN   FTFU0050
C                           FTMAPS AND ZSYSTM. NOT A STAND-ALONE ROUTINEFTFU0060
C   USAGE               - DUMMY = FTFUNC(X,I,PAR)                       FTFU0070
C   PARAMETERS   X(N)   - AN INPUT VECTOR OF LENGTH N WHICH CONTAINS    FTFU0080
C                           THE PRESENT APPROXIMATIONS TO AN            FTFU0090
C                           UNNORMALIZED FORM OF THE MOVING AVERAGE     FTFU0100
C                           PARAMETERS OF THE MODEL BEING ESTIMATED.    FTFU0110
C                           N EQUALS THE NUMBER OF MOVING AVERAGE       FTFU0120
C                           PARAMETERS PLUS 1.                          FTFU0130
C                I      - THE INDEX OF THE FUNCTION WHICH IS TO BE      FTFU0140
C                           COMPUTED.                                   FTFU0150
C                PAR(N) - INPUT VECTOR OF LENGTH N (DESCRIBED ABOVE)    FTFU0160
C                           CONTAINING COVARIANCES FROM IMSL ROUTINE    FTFU0170
```

```
C                        FTMAPS.
C   PRECISION          - SINGLE
C   LANGUAGE           - FORTRAN
C-----------------------------------------------------------------------
C   LATEST REVISION    - APRIL 1, 1974
C
      DIMENSION          X(1),PAR(1)
      COMMON             /FTCOMN/IQP1
      IM1 = I-1
      FTFUNC = -PAR(I)
      J = IQP1-I+1
      IF(J) 9005,9005,5
    5 DO 10 K=1,J
         L = K+IM1
   10 FTFUNC = FTFUNC+X(K)*X(L)
 9005 RETURN
      END


      SUBROUTINE FTMAPS (ACV,ARPS,IP,IQ,PMAS,WNV,WA,IER)
C
C-FTMAPS--------S-------LIBRARY 3---------------------------------------
C
C   FUNCTION            - PRELIMINARY ESTIMATION OF THE MOVING AVERAGE
C                           PARAMETERS IN AN ARIMA STOCHASTIC MODEL
C   USAGE               - CALL FTMAPS(ACV,ARPS,IP,IQ,PMAS,WNV,WA,IER)
C   PARAMETERS   ACV(ID)- INPUT VECTOR OF AUTOCOVARIANCES OF TIME SERIES
C                           BEING MODELED. THE LENGTH ID=IP+IQ+1. ACV(I)
C                           IS THE AUTOCOVARIANCE CORRESPONDING TO A
C                           TIME LAG OF I-1 UNITS OF TIME. THE LENGTH ID
C                           IS NOT A PARAMETER IN THE CALLING SEQUENCE.
C                ARPS   - INPUT VECTOR OF LENGTH IP.  THIS VECTOR
C                           CONTAINS THE PRELIMINARY ESTIMATES OF THE
C                           AUTOREGRESSIVE PARAMETERS OF THE MODEL.
C                           THESE ESTIMATES CAN BE COMPUTED BY CALLING
C                           SUBROUTINE FTARPS PRIOR TO CALLING FTMAPS.
C                IP     - INPUT NUMBER OF AUTOREGRESSIVE PARAMETERS IN
C                           THE MODEL.
C                IQ     - INPUT NUMBER OF MOVING AVERAGE PARAMETERS IN
C                           THE MODEL. IQ MUST BE GREATER THAN OR EQUAL
C                           TO 1.
C                PMAS   - OUTPUT VECTOR OF LENGTH IQ.  PMAS CONTAINS
C                           THE MOVING AVERAGE PARAMETERS.
C                WNV    - OUTPUT WHITE NOISE VARIANCE.
C                WA(IE) - WORK AREA VECTOR OF LENGTH IE = (5*(IQ+1))+
C                           (((IQ+3)*IQ)/2).  IE IS NOT A PARAMETER
C                           IN THE CALLING SEQUENCE.
C                IER    - ERROR PARAMETER
C                         TERMINAL ERROR = 128+N
C                           N=1 INDICATES FAILURE OF THE ALGORITHM
C                             TO CONVERGE TO ACCEPTABLE ESTIMATES
C                             FOR THE MOVING AVERAGE PARAMETERS.
C                           N=2 INDICATES A SINGULARITY IN THE NON-
C                             LINEAR EQUATIONS WHICH MUST BE SOLVED IN
C                             THE ITERATION PROCEDURE.THIS CONDITION MAY
C                             BE CAUSED BY EXTREMELY SMALL VALUES IN
C                             THE VECTOR ACV.
C                           N=3 INDICATES ACV(1) IS LESS THAN OR
C                             EQUAL TO ZERO.
C                           N=4 INDICATES A COMPUTED WHITE NOISE
C                             VARIANCE IS LESS THAN OR EQUAL TO ZERO.
C                             THIS IMPLIES THAT IQ IS LESS THAN OR
C                             EQUAL TO ZERO.
C   PRECISION           - SINGLE
C   REQD. IMSL ROUTINES - FTFUNC,UERTST,ZSYSTM
C   LANGUAGE            - FORTRAN
C-----------------------------------------------------------------------
C   LATEST REVISION     - APRIL 1, 1974
C
      COMMON             /FTCOMN/IQP1
      DIMENSION          ACV(1),ARPS(1),PMAS(1),WA(1)
      EXTERNAL           FTFUNC
      DATA               ZERO,ONEM,TOL,NSIG,ITMAX/0.,-1.,.001,3,100/
      IER = 0
      IQP1 = IQ+1
      ITMA1 = ITMAX
      IF (IP .GT. 0) GO TO 10
      DO 5  I=1,IQP1
    5 WA(I) = ACV(I)
      GO TO 30
C                                  COMPUTE MODIFIED COVARIANCES
   10 IPP1 = IP+1
      DO 25 I=1,IQP1
         IM = I
         WA(I) = ZERO
         AJ = ONEM
         DO 20 J=1,IPP1
            IF (J .GT. 1) AJ = ARPS(J-1)
            AK = ONEM
            DO 15  K=1,IPP1
               L = IABS(IM-K)+1
               IF (K .GT. 1) AK = ARPS(K-1)
   15       WA(I) = WA(I) + (AJ*AK*ACV(L))
   20    IM = IM+1
   25 CONTINUE
C                                  SET INITIAL GUESSES FOR ZSYSTM
   30 IQP2 = IQP1+1
      IF(WA(1) .LE. ZERO) GO TO 60
      WA(IQP2) = SQRT(WA(1))
      IQ1 = IQP2+1
      IQ2 = IQ+IQ+2
      DO 35  I=IQ1,IQ2
   35 WA(I) = ZERO
C                                  SET TOLERANCE AND CALL ZSYSTM
      EPS = WA(1) * TOL
      CALL ZSYSTM(FTFUNC,EPS,NSIG,IQP1,WA(IQP2),ITMA1,WA(IQ2+1),WA,IER)
      IF (IER) 40,40,9000
   40 WNV = WA(IQP2)
      IF (WNV) 45,45,50
C                                  ERROR, WHITE NOISE VARIANCE .LE. 0
   45 IER = 132
      GO TO 9000
C                                  COMPUTE MOVING AVERAGE PARAMETERS
   50 DO 55  I=1,IQ
   55 PMAS(I) = -WA(IQP2+I)/WNV
      WNV = WNV*WNV
      GO TO 9005
C                                  DATA ERROR, FIRST COVARIANCE .LE. 0
   60 IER = 131
 9000 CONTINUE
      CALL UERTST(IER,6HFTMAPS)
 9005 RETURN
```

```
      END                                                               FTMA1040


      SUBROUTINE FTRDIF (ID1,ID2,IP,IS,LZ,Z,SHIFT,LW,IER)               FTRD0010
C                                                                       FTRD0020
C-FTRDIF---------S--------LIBRARY 3-------------------------------------FTRD0030
C                                                                       FTRD0040
C   FUNCTION            - TIME SERIES TRANSFORMATION AND DIFFERENCING.  FTRD0050
C   USAGE               - CALL FTRDIF(ID1,ID2,IP,IS,LZ,Z,SHIFT,LW,IER)  FTRD0060
C   PARAMETERS   ID1    - INPUT.  ORDER OF NON-SEASONAL DIFFERENCE.     FTRD0070
C                ID2    - INPUT.  ORDER OF SEASONAL DIFFERENCE.         FTRD0080
C                IP     - INPUT.  TRANSFORMATION EXPONENT.              FTRD0090
C                           IF IP = 0, A LOGRITHMETIC TRANSFORMATION    FTRD0100
C                           IS PERFORMED ON THE INPUT SERIES.           FTRD0110
C                           IF IP NOT=0, AN EXPONENTIAL TRANSFORMATION  FTRD0120
C                           IS PERFORMED ON THE INPUT SERIES.           FTRD0130
C                IS     - INPUT.  LENGTH OF SEASONAL PERIOD.            FTRD0140
C                LZ     - INPUT. LENGTH OF VECTOR Z.                    FTRD0150
C                Z(LZ)  - INPUT AND OUTPUT VECTOR.                      FTRD0160
C                           ON INPUT, Z CONTAINS THE TIME SERIES.       FTRD0170
C                           ON OUTPUT, Z CONTAINS THE TRANSFORMED AND   FTRD0180
C                           DIFFERENCED TIME SERIES.                    FTRD0190
C                SHIFT  - OUTPUT.  WHEN IP=0, SHIFT WILL BE             FTRD0200
C                           (THE NEGATIVE OF THE MINIMUM VALUE OF Z)    FTRD0210
C                           PLUS ONE.  WHEN IP NOT=0, SHIFT WILL BE     FTRD0220
C                           ZERO.                                       FTRD0230
C                LW     - OUTPUT.  TIME SERIES LENGTH.  COMPUTED AS     FTRD0240
C                           LW = LZ-ID1-IS(ID2).                        FTRD0250
C                IER    - ERROR PARAMETER.                              FTRD0260
C                         TERMINAL ERROR = 128+N                        FTRD0270
C                           N = 1 INDICATES ID1 AND/OR ID2 LESS THAN 0. FTRD0280
C   PRECISION           - SINGLE                                        FTRD0290
C   REQD. IMSL ROUTINES - UERTST                                        FTRD0300
C   LANGUAGE            - FORTRAN                                       FTRD0310
C-----------------------------------------------------------------------FTRD0320
C   LATEST REVISION     - MARCH 31, 1972                                FTRD0330
C                                                                       FTRD0340
      DIMENSION          Z(1)                                           FTRD0350
      DATA               ZERO/0.0/,ONE/1.0/                             FTRD0360
      IF(ID1 .LT. 0 .OR. ID2 .LT. 0) GO TO 9000                         FTRD0370
      IER = 0                                                           FTRD0380
      K = IS+1                                                          FTRD0390
      SHIFT = ZERO                                                      FTRD0400
      IF (IP .NE. 0) GO TO 15                                           FTRD0410
C                                  FIND MINIMUM VALUE IN Z.             FTRD0420
      SHIFT = Z(1)                                                      FTRD0430
      DO 5  J=2,LZ                                                      FTRD0440
         IF (SHIFT .GT. Z(J)) SHIFT = Z(J)                              FTRD0450
    5 CONTINUE                                                          FTRD0460
      SHIFT = -SHIFT+ONE                                                FTRD0470
      DO 10  J=1,LZ                                                     FTRD0480
   10 Z(J) = ALOG(Z(J)+SHIFT)                                           FTRD0490
      GO TO 25                                                          FTRD0500
   15 IF (IP .EQ. 1) GO TO 25                                           FTRD0510
      DO 20 I=1,LZ                                                      FTRD0520
   20 Z(I) = Z(I)**IP                                                   FTRD0530
   25 LW = LZ                                                           FTRD0540
      IF (ID2 .EQ. 0) GO TO 40                                          FTRD0550
C                                  TRANSFORM TO TIME SERIES             FTRD0560
      DO 35  L=1,ID2                                                    FTRD0570
         DO 30  I=K,LW                                                  FTRD0580
   30    Z(I-IS) = Z(I)-Z(I-IS)                                         FTRD0590
         LW = LW - IS                                                   FTRD0600
   35 CONTINUE                                                          FTRD0610
   40 IF (ID1 .EQ. 0) GO TO 9005                                        FTRD0620
      DO 50  L=1,ID1                                                    FTRD0630
         DO 45  I=2,LW                                                  FTRD0640
   45    Z(I-1) = Z(I)-Z(I-1)                                           FTRD0650
         LW = LW-1                                                      FTRD0660
   50 CONTINUE                                                          FTRD0670
      GO TO 9005                                                        FTRD0680
 9000 CONTINUE                                                          FTRD0690
      IER = 129                                                         FTRD0700
      CALL UERTST(IER,6HFTRDIF)                                         FTRD0710
 9005 RETURN                                                            FTRD0720
      END                                                               FTRD0730


      SUBROUTINE GGNOR  (SEED,N,R)                                      GGNO0010
C                                                                       GGNO0020
C-GGNOR----------S--------LIBRARY 3-------------------------------------GGNO0030
C                                                                       GGNO0040
C   FUNCTION            - GENERATES PSEUDO NORMAL RANDOM NUMBERS.       GGNO0050
C   USAGE               - CALL GGNOR(SEED,N,R)                          GGNO0060
C   PARAMETERS   SEED   - A DOUBLE PRECISION NUMBER, NOT OF THE FORM    GGNO0070
C                           J/(2**K)(J,K INTEGERS) AND IN THE EXCLUSIVE GGNO0080
C                           RANGE (1,E-6,1,). SEED IS USED TO INITIATE  GGNO0090
C                           THE GENERATION, AND ON EXIT, HAS BEEN       GGNO0100
C                           REPLACED BY A NEW SEED FOR SUBSEQUENT USE.  GGNO0110
C                N      - INPUT NUMBER OF DEVIATES TO BE GENERATED.     GGNO0120
C                R      - OUTPUT VECTOR CONTAINING THE NORMAL           GGNO0130
C                           PSEUDO RANDOM NUMBERS.                      GGNO0140
C   PRECISION           - SINGLE                                        GGNO0150
C   REQD. IMSL ROUTINES - GGU1,MERFI,UERTST                             GGNO0160
C   LANGUAGE            - FORTRAN                                       GGNO0170
C-----------------------------------------------------------------------GGNO0180
C   LATEST REVISION     - FEBRUARY 20, 1973                             GGNO0190
C                                                                       GGNO0200
      DIMENSION          R(1)                                           GGNO0210
      DOUBLE PRECISION   SEED                                           GGNO0220
C                                  GET N RANDOM NUMBERS                 GGNO0230
      CALL GGU1(SEED,N,R)                                               GGNO0240
C                                  NORMALIZE EACH RANDOM NUMBER         GGNO0250
      DO 5 I=1,N                                                        GGNO0260
         CALL MDNRIS(R(I),R(I),IER)                                     GGNO0270
    5 CONTINUE                                                          GGNO0280
      RETURN                                                            GGNO0290
      END                                                               GGNO0300


      SUBROUTINE GGU1   (SEED,N,R)                                      GGU10010
C                                                                       GGU10020
C-GGU1-----------S--------LIBRARY 3-------------------------------------GGU10030
C-GGU1M                                                                 GGU10040
C                                                                       GGU10050
C   FUNCTION            - GENERATION OF PSEUDO-RANDOM UNIFORM (0,1)     GGU10060
C                           DEVIATES (ONE MULTIPLIER)                   GGU10070
C   USAGE               - CALL GGU1(SEED,N,R)                           GGU10080
```

```
C                          CALL GGU1M(SEED,N,R)
C                            ENTRY GGU1M IS USED TO CHANGE THE
C                            MULTIPLIER LABELED A.
C   PARAMETERS     SEED    - A DOUBLE PRECISION NUMBER, NOT OF THE FORM
C                            J/(2**K)(J,K INTEGERS) AND IN THE EXCLUSIVE
C                            RANGE (1.E-6,1.). SEED IS USED TO INITIATE
C                            THE GENERATION, AND ON EXIT, HAS BEEN
C                            REPLACED BY A NEW SEED FOR SUBSEQUENT USE.
C                          WHEN GGU1M IS CALLED SEED IS USED TO
C                            TEMPORARILY ALTER THE MULTIPLIER A WHICH
C                            IS USED IN GENERATING THE RANDOM DEVIATES.
C                            THE PARAMETERS N AND R ARE NOT USED WHEN
C                            GGU1M ENTRY IS USED.
C                  N       - THE NUMBER OF DEVIATES TO BE GENERATED
C                  R       - OUTPUT VECTOR OF LENGTH N, CONTAINING N
C                            PSEUDO UNIFORM FLOATING POINT RANDOM
C                            NUMBERS IN (0,1)
C   PRECISION              - SINGLE
C   LANGUAGE               - FORTRAN
C-----------------------------------------------------------------------
C   LATEST REVISION        - SEPTEMBER 10,1974
C
      DIMENSION            R(1)
      DOUBLE PRECISION     SEED,A1,A2,A3,A
      DATA                 A/11863237.0D0/
      DATA                 A1/ 1.164153218269348144531250000D-10/,
     *                     A2/ 9.223372036854775808000000000D+18/,
     *                     A3/ 3.961408125713216879677197516800+28/
      SEED = SEED*A1
      SEED=(SEED*A2)*A2
      DO 5 I=1,N
         SEED = SEED*A
         SEED = SEED-((SEED*A3)*A3)
         R(I) = SEED
    5 CONTINUE
      GO TO 10
C
      ENTRY GGU1M
C
      IF(SNGL(SEED) .EQ. 0.0) GO TO 10
      A = SEED
   10 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE GTSIMP (Z,IPDQ,SEED,ALPHA,DARPS,PMAS,PMAC,WNV,FCST,SIM,
     1        IS,WK,IER)
C-----------------------------------------------------------------------
C   FUNCTION      -NON-SEASONAL ARIMA STOCHASTIC MODEL FOR SINGLE SERIES
C     Z           -INPUT VECTOR OF LENGTH IPDQ(2) CONTAINING THE SERIES
C     IPDQ        -INPUT/OUTPUT VECTOR OF LENGTH 7. IPDQ(I) CONTAINS,
C                    I=1, IPDQ(1)=1 IS REQUIRED
C                    I=2, INPUT. LENGTH OF TIME SERIES Z.
C                    I=3, NUMBER OF AUTOREGRESSIVE PARAMETERS IN THE
C                       MODEL
C                    I=4, NUMBER OF MOVING AVERAGE PARAMETERS IN THE MODEL
C                    I=5, DEGREE OF DIFFERENCING OPERATION REQUIRED TO
C                       OBTAIN A STATIONARY TIME SERIES
C                    I=6, FORECASTING CONTROL PARAMETER.
C                       FORECASTS UP TO IPDQ(6) STEPS IN ADVANCE ARE
C                       CALCULATED . IPDQ(6) MUST BE POSITIVE
C                    I=7, SIMULATION OPTION.
C                       IPDQ(7) LESS THAN OR EQUAL TO ZERO IMPLIES
C                          SIMULATION NOT DESIRED
C                       IPDQ(7) POSITIVE IMPLIES IPDQ(7) SIMULATIONS OF
C                          THE TIME SERIES UP TO IPDQ(6) STEPS IN ADVANCE
C                          ARE DESIRED
C     SEED        -INPUT DOUBLE PRECISION NUMBER. SEED IS USED TO
C                    GENERATE THE SIMULATED TIME SERIES
C     ALPHA       -INPUT/OUTPUT VECTOR OF LENGTH 2. ALPHA CONTAINS, WHEN
C                    I=1, INPUT MINIMUM SIGNIFICANCE LEVEL OF THE MODEL
C                       IN THE EXCLUSIVE RANGE (0,1). ON OUTPUT , THE
C                       ESTIMATED SIGNIFICANCE LEVEL IS RETURND.
C                    I=2, INPUT VALUE IN THE EXCLUSIVE RANGE (0,1)
C                       USED FOR COMPUTING 100(1-ALPHA(2)) PERCENT
C                       PROBABILITY LIMITS FOR THE FORECASTS.
C     DARPS       -OUTPUT VECTOR OF LENGTH IPDQ(3)+IPDQ(5) CONTAINING
C                    ESTIMATES OF THE AUTOREGRESSIVE PARAMETERS.
C     PMAS        -OUTPUT VECTOR OF LENGTH IPDQ(4) CONTAINING ESTIMATES
C                    OF THE MOVING AVERAGE PARAMETERS
C     PMAC        -OUTPUT ESTIMATE OF THE TREND FACTOR
C     WNV         -OUTPUT ESTIMATE OF THE WHITE NOISE VARIANCE
C     FCST        -OUTPUT MATRIX OF DIMENSION 3 BY IPDQ(6)
C                    FCST(I,J) , FOR LEAD TIMES J=1,2,...,IPDQ(6),
C                    CONTAINS, WHEN
C                       I=1, THE WEIGHTS FOR THE WEIGHTED SUM OF SHOCKS
C                          THAT GIVES THE FORECAST ERROR
C                       I=2, THE FORECASTS
C                       I=3, THE CORRESPONDING DEVIATION FROM EACH
C                          FORSCAST FOR THE 100(1-ALPHA(2)) PERCENT
C                          PROBABILITY LIMITS
C     SIM         -OUTPUT MATRIX OF DIMENSION IPDQ(6) BY IPDQ(7)
C                    DEFINED ONLY WHEN IPDQ(7) IS POSITIVE. SIM(J,I)
C                    FOR LEAD TIMES J=1,2,...,IPDQ(6), CONTAINS THE
C                    RESULTS OF THE IPDQ(7) SIMULATIONS
C     IS          -FIRST DIMENSION OF SIM AS DIMENSIONED IN THE
C                    CALLING PROGRAM
C     WK          -WORK AREA,LENGTH DEFINED IN THE PROGRAMMING NOTES
C     IER         -ERROR PARAMETER.
C                    TERMINAL ERROR =128+N
C                       N=1, INDICATES A TERMINAL ERROR OCCURRED IN FTMAPS
C                          OR FTARPS
C                       N=2, INDICATES ONE OF THE INPUT PARAMETERS IPDQ(I)
C                          WAS OUT OF RANGE
C                       N=3, INDICATES ALPHA(I),I=1,2 WAS NOT IN THE
C                          EXCLUSIVE RANGE(0,1).
C                    WARNING SIGNAL =32+N
C                       N=4, INDICATES THE MODEL DID NOT PASS THE
C                          ALPHA(1) SIGNIFICANCE TEST. THE NEED FOR MORE
C                          PARAMETERS IS INDICATED
C   NOTE          1. THIS ROUTINE IS OBTAINED BY MODIFYING THE IMSL
C                    ROUTINE FTSIMP
C                 2. WK IS A WORK VECTOR OF LENGTH AT LEAST THE MAXIMUM OF
C                    A. 2*IPDQ(2)+42
C                    B. IPDQ(2)+P*(P+6)+Q+1
C                    C. IPDQ(2)+(((Q+3)*Q)/2)+6*Q+P36
C                 WHERE P,D,Q ARE NUMBER OF PARAMETERS OF THE ARIMA(P,D,Q)
```

```
C-----------------------------------------------------------------------
      DIMENSION           Z(1),IPDQ(1),ALPHA(1),DARPS(1),FCST(3,1),
     1                    SIM(I8,1),WK(1),PHAS(1)
       DIMENSION WMEAN(60)
      DOUBLE PRECISION   SEED
      DATA                I0/0/,I1/1/,I3/3/,I5/5/,I20/20/
C                                 SET OUTPUT PARAMETERS OF IPDQ
 1010 CONTINUE
      N=IPDQ(2)
      IP=IPDQ(3)
      IPP1=IP+1
      IQ=IPDQ(4)
      IQP1=IQ+1
      IF (IP.EQ.0.AND.IPDQ(5).EQ.0) IPDQ(5)=1
      ID=IPDQ(5)
      IR=MAX0(IP,IQ)+1
      NM1=N
C                                 ERROR CHECK ON PARAMETERS
      L=IP+IQ
      N2=IP
      DO 1020 I=1,N
      WK(I)=Z(I)
 1020 CONTINUE
      IER=0
         IF (IPDQ(5).EQ.0) GO TO 1050
 1030 CONTINUE
      N=N-1
C           DIFFERENCE  SERIES
      DO 1040 I=1,N
      WK(I)=WK(I+1)-WK(I)
 1040 CONTINUE
         IF (IPDQ(5).EQ.2.AND.NM1.EQ.N+1) GO TO 1030
 1050 CONTINUE
      N2=N+N
C                                 FIND COVARIANCES FOR FTARPS,FTMAPS
      CALL FTAUTO (WK,N,L,I0,I3,WBAR,WK(N+1),WK(N+2),WK,WK,WK)
C           STANDARD ERROR OF WBAR (STE)
      TMP=N
      STE=SQRT(WK(N+1)/(TMP-1.0))/SQRT(TMP)
C           RANDOMLY DRAW PROBABILITY AND FIND T STATISTIC
      ISET=IPDQ(7)
      DO 1060 I=1,ISET
      TP=1.0
      TEMP=RANF(0)
      IF (TEMP.GT.0.5) TP=-1.0
      IF (TEMP.GT.0.5) TEMP=1.0-TEMP
      TEMP=TEMP*2.0
      TMP=N-1
      CALL MDSTI (TEMP,TMP,X,IER)
      XA=X*TP
C           PROBABILISTIC SENSE OF WBAR
      WMEAN(I)=WBAR-XA*STE
 1060 CONTINUE
C                                 FIND DARPS,PHAS,PHAC,WNV
      J=N+L+2
      TEMP=1.0
         IF (IPDQ(3).EQ.0) GO TO 1070
      CALL FTARPS (WK(N+1),WBAR,IPDQ(3),IPDQ(4),DARPS,PHAC,WK(J),IER)
      TEMP=PHAC/WBAR
         IF (IER.GT.0) GO TO 1230
      CALL CHECK (IPDQ(3),DARPS(1),IER)
         IF (IER.GT.0) GO TO 1240
         IF (IPDQ(4).EQ.0) GO TO 1080
 1070 CONTINUE
      CALL FTMAPS (WK(N+1),DARPS,IPDQ(3),IPDQ(4),PHAS,WNV,WK(J),IER)
         IF (IER.GT.0) GO TO 1250
      IF (IPDQ(3).EQ.0) PHAC=WBAR
 1080 CONTINUE
      CALL CHECK (IPDQ(4),PHAS(1),IER)
         IF (IER.GT.0) GO TO 1260
C                                 FIND ONE-STEP FORECAST ERROR
      DO 1090 I=1,IR
      WK(N+I)=0.0
 1090 CONTINUE
         IF (ID.GT.0) GO TO 1110
      DO 1100 I=1,N
      WK(I)=WK(I)-WBAR
 1100 CONTINUE
 1110 CONTINUE
      DO 1150 I=IR,N
      NPI=N+I
      WK(NPI)=WK(I)
         IF (IP.EQ.0) GO TO 1130
      DO 1120 J=1,IP
      WK(NPI)=WK(NPI)-DARPS(J)*WK(I-J)
 1120 CONTINUE
         IF (IQ.EQ.0) GO TO 1150
 1130 CONTINUE
      DO 1140 J=1,IQ
      WK(NPI)=WK(NPI)+PHAS(J)*WK(NPI-J)
 1140 CONTINUE
 1150 CONTINUE
C                                 FIND COVARIANCE OF ERRORS
      CALL FTAUTO (WK(N+1),N,I20,I0,I5,WK(N2+21),WK(N2+22),WK(N2+23),
     1       WK(N2+1),WK,WK)
C                                 FIND ERROR SIGNIFICANCE STATISTIC
      T=0.0
      DO 1160 I=1,I20
      T=T+WK(N2+I)**2
 1160 CONTINUE
      XN=N-IPDQ(5)
      T=T*XN
      XM=I20-IP-IQ
      CALL MDCH (T,XM,CHI,IER)
C              MINIMUN SIGNIFICANCE LEVEL TEST
        IF ((1.0-CHI).GE.ALPHA(1)) GO TO 1170
        GO TO 1270
C                                 FORECASTS
 1170 CONTINUE
      ALPHA(1)=1.0-CHI
CC             LINEAR DECAY OF THE PHAC WHEN ID GT 0
      IF (ID.GT.0) PHAC=PHAC*(1.0-(0.2*IPDQ(6)/IPDQ(2)-0.1))
         IF (IPDQ(3).EQ.0) GO TO 1190
      DO 1180 I=1,IP
      WK(I)=DARPS(I)
 1180 CONTINUE
 1190 CONTINUE
         IF (IPDQ(4).EQ.0) GO TO 1210
      DO 1200 I=1,IQ
      WK(N+I)=PHAS(I)
 1200 CONTINUE
 1210 CONTINUE
      CALL FTCAST (Z,WK,WK(N+1),PHAC,ALPHA(2),IPDQ(2),DARPS,FCST,WNV,
     1       IERX)
```

```
C                                SIMULATIONS
         IF (IPDQ(7).LE.0) GO TO 1220
      M=IPDQ(7)
      J=IP+ID
      K=MAX0(J,1)
      DO 1220 I=1,M
      PMAC=TEMP*WMEAN(I)
      IF (ID.GT.0) PMAC=PMAC*(1.0-(0.2*IPDQ(6)/IPDQ(2)-0.1))
      CALL FTGEN1 (DARPS,PMAS,PMAC,Z(N-K+1),WNV,SEED,J,IQ,IPDQ(6),SIM(1,
     1        I),WK)
 1220 CONTINUE
      RETURN
 1230 CONTINUE
      ALPHA(1)=-1.0
      RETURN
 1240 CONTINUE
      ALPHA(1)=-2.0
      RETURN
 1250 CONTINUE
      ALPHA(1)=-3.0
      RETURN
 1260 CONTINUE
      ALPHA(1)=-4.0
      RETURN
 1270 CONTINUE
      ALPHA(1)=-5.0
      RETURN
      END

      SUBROUTINE IDENTN (Z,IND,RLIMIT,DISCON,ARF,IPARM,LOGX,NPRINT)
C-----------------------------------------------------------------------
C   FUNCTION    -TO IDENTIFY IP,ID AND IQ OF AN ARIMA(IP,ID,IQ) MODEL
C   Z           -INPUT TIME SERIES OF LENGTH N
C   IND(I)       I=1, LENGTH OF SERIES
C                 2, LENGTH OF PROJECTION (MAXIMUM 600)
C                 3, NUMBER OF SETS OF PROJECTION DESIRED (MAXIMUM 10)
C   RLIMIT      -MAXIMUM ALLOWABLE RANGE OF PROJECTION ,IT IS A FUNCTION
C                 OF THE RANGE OF THE CALIBRATION SERIES
C   DISCON      -DISCOUNT FACTOR (0.10 TO 0.30)IN COMPUTING DISCOUNTED
C                 SIGNIFICANT LEVEL AFA(.)
C   ARF         -OUTPUT VECTOR GIVE FIRST FIVE HIGHEST DISCOUNTED
C                 SIGNIFICANCE LEVELS
C   IPARM       -OUTPUT MATRIX GIVES THE P,D, AND Q FOR THE TOP FIVE
C                 ARIMA(P,D,Q) MODELS RANKED DECREASINGLY BY ARF VALUES
C   LOGX        =1, NATURAL-LOG TRANSFORMATION OF Z IS DESIRED
C                0, NOT DESIRED
C   NPRINT      =0, NO PRINT OUT IS DESIRED
C               =1, PRINT OUT IS DESIRED
C-----------------------------------------------------------------------
C
      DIMENSION Z(1  ),ALPHA(2),DARPS(5),PMAS(5),FCST(3,750),
     1 SIM(750, 5),WK(1500),IPDQ(7)
      DIMENSION ZLAST(11,14),NAMES(14),NP(14),NQ(14),ND(14)
      DIMENSION ALFA(14),ARF(5),RANGEC(14),RANGEP(14),
     1IND(3),IPARM(5,3)
      DOUBLE PRECISION SEED
      DATA NQ/3,2,3,2,1,3,2,1,0,2,1,0,1,0/
      DATA NP/2,3,1,2,3,0,1,2,3,0,1,2,0,1/
      N=IND(1)
      LPROJ=IND(2)
      NSET=IND(3)
C           SEARCH FOR OPTIMAL ID AS NDD
      CALL FINDD (Z,N,24,NDD,0,40)
C                INITIALIZE PARAMETERS TO BE USED IN GTSIMP
      IS=750
      IPDQ(1)=1
      IPDQ(2)=N
      IPDQ(6)=LPROJ
      IPDQ(7)=NSET
      DO 1080 ITT=1,14
      SEED=0.156789432587D0
      ALPHA(1)=0.01
      ALPHA(2)=0.05
      IPDQ(3)=NP(ITT)
      IPDQ(4)=NQ(ITT)
      IPDQ(5)=NDD
      DO 1010 IN=1,5
      DARPS(IN)=0.0
      PMAS(IN)=0.0
 1010 CONTINUE
      PMAC=0.0
      FCST(2,LPROJ)=0.0
      ZLAST(1,1)=Z(N)
      IF (LOGX.NE.0) ZLAST(1,1)=EXP(Z(N))
      DO 1020 I=1,NSET
      ZLAST(I+1,ITT)=0.0
 1020 CONTINUE
      CALL GTSIMP (Z,IPDQ,SEED,ALPHA,DARPS,PMAS,PMAC,WNV,FCST,SIM,IS,WK,
     1      IER)
      ALFA(ITT)=ALPHA(1)
      ND(ITT)=IPDQ(5)
C     PRINT 1130, ITT,(IPDQ(J),J=3,5),ALPHA(1),WNV,PMAC,(DARPS(J),J=1,
C    1       4),(PMAS(J),J=1,3)
         IF (IER.GT.0.OR.ALFA(ITT).LE.0.0) GO TO 1080
CC               COMPUTE THE RANGE OF PROJECT AS RANGEP
      XMIN=SIM(1,1)
      XMAX=SIM(1,1)
      RANGEP(ITT)=0.0
      DO 1040 J=1,NSET
      DO 1030 I=1,LPROJ
      IF (SIM(I,J).GT.XMAX) XMAX=SIM(I,J)
      IF (SIM(I,J).LT.XMIN) XMIN=SIM(I,J)
      RP=XMAX-XMIN
 1030 CONTINUE
      IF (RP.GT.RANGEP(ITT)) RANGEP(ITT)=RP
CC               CHECK WITH LIMIT ON RANGEP
         IF (RANGEP(ITT).LE.RLIMIT) GO TO 1040
      ALFA(ITT)=-6.0
         GO TO 1080
 1040 CONTINUE
         IF (LOGX.EQ.0) GO TO 1060
      FCST(2,LPROJ)=EXP(FCST(2,LPROJ))
      DO 1050 J=1,NSET
      SIM(LPROJ,J)=EXP(SIM(LPROJ,J))
 1050 CONTINUE
 1060 CONTINUE
      ZLAST(NSET+2,ITT)=FCST(2,LPROJ)
      DO 1070 I=1,NSET
      ZLAST(I+1,ITT)=SIM(LPROJ,I)
 1070 CONTINUE
 1080 CONTINUE
      J1=NSET+1
C     DO 1090 K=1,14
C     PRINT 1140, K,NP(K),ND(K),NQ(K),ALFA(K),ZLAST(1,1),(ZLAST(J,K),
C    1       J=2,7)
C1090 CONTINUE
```

```
CC      SEARCH HIGHEST ALFA VALUE AND STORE ITS LOCATION
C               DISCOUNTED SIGNIFICANCE LEVEL
      DO 1100 I=1,14
      ALFA(I)=ALFA(I)*(1.0-DISCON)**(NP(I)+NQ(I)-1)
 1100 CONTINUE
C               RANK ALFA(I) FROM MAX TO MIN AND STORE IN AFA
      A=ALFA(1)
      DO 1120 I=1,5
      DO 1110 J=1,14
         IF (ALFA(J).LT.A) GO TO 1110
      A=ALFA(J)
      IJ=J
 1110 CONTINUE
      ARF(I)=A
      ALFA(IJ)=-A
      A=-A
      IPARM(I,1)=NP(IJ)
      IPARM(I,2)=NDD
      IPARM(I,3)=NQ(IJ)
 1120 CONTINUE
      RETURN
 1130 FORMAT (4I4,11F9.3)
 1140 FORMAT (4I4,1F8.2,8F10.1)
      END


      SUBROUTINE PLOT (Q,N,NV,IFCTR,IBEGIN,NAME)
C------------------------------------------------------------------------
C   FUNCTION     - TO PLOT UP TO 9 CURVES IN ONE CHART
C       Q        -INPUT MATRIX OF DIMENSION N BY NV
C       N        -LENGTH OF EACH VARIABLE IN Q(I,J)
C       NV       - NUMBER OF VARIABLES IN Q(I,J) TO BE PLOTTED
C       IFCTR    -SCALING FACTOR
C       IBEGIN   -BEGINNING DATE OF THE RECORD (USED IN X-AXIS)
C       YLABEL   -LABEL TO BE USED IN THE Y-AXIS OF THE PLOT
C   NOTE         1.DIMENSIONS OF THE PARAMETERS IPLOT AND ISVAE (CONTROL
C                    THE PRINTING SPACE ) ARE NOT CHANGEABLE
C                2. DIMENSION OF Q AND THE FIRST IF STATEMENT SHOULD BE
C                    CHANGED ACCORDING TO THE SIZE OF THE MATRIX PLOTTED
C------------------------------------------------------------------------
      DIMENSION IPLOT(121),LTR(10),ISAVE(121),SCALE(13),Q(250,9),NAME(8)
      DATA IBLANK,IPER/1H ,1H./
      DATA LTR/1H1,1H2,1H3,1H4,1H5,1H6,1H7,1HX,1H*/
      WRITE (6,1200) (NAME(J),J=1,8)
         IF (N.GT.250.OR.NV.GT.9) GO TO 1190
      TMP=10.**IFCTR
      WRITE (6,1210) TMP
      DO 1010 I=1,121
      ISAVE(I)=IBLANK
      IPLOT(I)=IBLANK
 1010 CONTINUE
      WRITE (6,1220)
      AMIN=999999.
      AMAX=-999999.
      DO 1020 J=1,NV
      DO 1020 I=1,N
      Q(I,J)=Q(I,J)/TMP
      TEMP=Q(I,J)
      IF (TEMP.GT.AMAX) AMAX=TEMP
      IF (TEMP.LT.AMIN) AMIN=TEMP
 1020 CONTINUE
      IF (AMAX.LE.AMIN) AMAX=1.01*AMIN
      TMP=12.
      RATIO=(AMAX-AMIN)/TMP
      IFMT=1
      STEP=1.
 1030 CONTINUE
         IF (RATIO.LE..5) GO TO 1040
         IF (RATIO.GT.5.) GO TO 1050
      IF (RATIO.LE.2.AND.RATIO.GT.1.) STEP=2.*STEP
      IF (RATIO.LE.4.AND.RATIO.GT.2.) STEP=4.*STEP
      IF (RATIO.GT.4) STEP=5.*STEP
         GO TO 1060
 1040 CONTINUE
      STEP=STEP*.1
      RATIO=RATIO*10.
      IFMT=IFMT+1
         GO TO 1030
 1050 CONTINUE
      STEP=STEP*10.
      RATIO=RATIO*.1
         GO TO 1030
 1060 CONTINUE
      ITP=AMIN/STEP
      TMP=ITP
      TMP=(TMP-.01)*STEP
      IF (AMIN.LT.TMP) ITP=ITP-1
      ITMP=AMAX/STEP
      TMP=ITMP
      TMP=(TMP+.01)*STEP
      IF (AMAX.GT.TMP) ITMP=ITMP+1
      IF (IFMT.GT.3) IFMT=3
      ITEMP=1
      DO 1070 L=1,13
      SCALE(L)=0.
 1070 CONTINUE
      LX=1
      TEMP=ITP
      TEMP=TEMP*STEP
      ITMP=ITMP+1-ITP
 1080 CONTINUE
      SCALE(LX)=TEMP
      LX=LX+ITEMP
      TEMP=TEMP+STEP
         IF (LX.LE.13) GO TO 1080
         GO TO (1090,1100,1110), IFMT
 1090 CONTINUE
      WRITE (6,1230) SCALE
         GO TO 1120
 1100 CONTINUE
      WRITE (6,1240) SCALE
         GO TO 1120
 1110 CONTINUE
      WRITE (6,1250) SCALE
 1120 CONTINUE
      STP=STEP*.1
      TMP=ITP
      STRT=TMP*STEP
      ISTEP=1
```

```
      LX=1
      DO 1130 L=1,13
      ISAVE(LX)=IPER
      IPLOT(LX)=IPER
      LX=LX+10
 1130 CONTINUE
      KDEX=IBEGIN
      DO 1180 K=1,N
         IF (K.GT.1.AND.K.LT.N) GO TO 1150
      DO 1140 L=1,121
      IPLOT(L)=IPER
 1140 CONTINUE
 1150 CONTINUE
      DO 1160 I=1,NV
      TEMP=Q(K,I)
      ITMP=(TEMP-STRT)/STP+.5
      LX=ITMP+ISTEP+1
      IF (LX.GT.121) LX=121
      IF (LX.LT.1) LX=1
      IPLOT(LX)=LTR(I)
 1160 CONTINUE
      WRITE (6,1260) KDEX,IPLOT
      KDEX=1+KDEX
      LAG=LAG+1
      DO 1170 L=1,121
      IPLOT(L)=ISAVE(L)
 1170 CONTINUE
 1180 CONTINUE
      WRITE (6,1270)
      RETURN
 1190 CONTINUE
      WRITE (6,1280)
      RETURN
 1200 FORMAT   (1H1,10X,8A10)
 1210 FORMAT   (10X,20HQUANTITIES SCALED BY ,F8.0)
 1220 FORMAT   (1H0)
 1230 FORMAT   (4X,F6.0,12F10.0)
 1240 FORMAT   (4X,F6.1,12F10.1)
 1250 FORMAT   (4X,F6.2,12F10.2)
 1260 FORMAT  (I8,2X,121A1)
 1270 FORMAT   (1HR)
 1280 FORMAT   (1X,18HDIMENSION EXCEEDED)
      END
```

```
      SUBROUTINE LEQT1F (A,M,N,IA,B,IDGT,WKAREA,IER)
C
C-LEQT1F--------S--------LIBRARY 3------------------------------------------
C
C   FUNCTION            - LINEAR EQUATION SOLUTION - FULL STORAGE
C                           MODE - SPACE ECONOMIZER SOLUTION.
C   USAGE               - CALL LEQT1F (A,M,N,IA,B,IDGT,WKAREA,IER)
C   PARAMETERS   A      - INPUT MATRIX OF DIMENSION N BY N CONTAINING
C                           THE COEFFICIENT MATRIX OF THE EQUATION
C                           AX = B.
C                         ON OUTPUT, A IS REPLACED BY THE LU
C                           DECOMPOSITION OF A ROWWISE PERMUTATION OF
C                           A.
C                M      - NUMBER OF RIGHT-HAND SIDES.(INPUT)
C                N      - ORDER OF A AND NUMBER OF ROWS IN B.(INPUT)
C                IA     - NUMBER OF ROWS IN THE DIMENSION STATEMENT
C                           FOR A AND B IN THE CALLING PROGRAM. (INPUT)
C                B      - INPUT MATRIX OF DIMENSION N BY M CONTAINING
C                           RIGHT-HAND SIDES OF THE EQUATION AX = B.
C                         ON OUTPUT, THE N BY M SOLUTION X REPLACES B.
C                IDGT   - INPUT OPTION.
C                         IF IDGT IS GREATER THAN 0, THE ELEMENTS OF
C                           A AND B ARE ASSUMED TO BE CORRECT TO IDGT
C                           DECIMAL DIGITS AND THE ROUTINE PERFORMS
C                           AN ACCURACY TEST.
C                         IF IDGT EQUALS ZERO, THE ACCURACY TEST IS
C                           BYPASSED.
C                WKAREA - WORK AREA OF DIMENSION GREATER THAN OR EQUAL
C                           TO N.
C                IER    - ERROR PARAMETER
C                         TERMINAL ERROR = 128+N.
C                           N = 1 INDICATES THAT A IS ALGORITHMICALLY
C                             SINGULAR. (SEE THE CHAPTER L PRELUDE).
C                         WARNING ERROR = 32+N.
C                           N = 2 INDICATES THAT THE ACCURACY TEST
C                                 FAILED.
C                                 THE COMPUTED SOLUTION MAY BE IN ERROR
C                                 BY MORE THAN CAN BE ACCOUNTED FOR BY
C                                 THE UNCERTAINTY OF THE DATA.
C                                 THIS WARNING CAN BE PRODUCED ONLY IF
C                                 IDGT IS GREATER THAN 0 ON INPUT.
C                                 SEE CHAPTER L PRELUDE FOR FURTHER
C                                 DISCUSSION.
C   PRECISION           - SINGLE
C   REQD. IMSL ROUTINES - LUDATF,LUELMF,UERTST
C   LANGUAGE            - FORTRAN
C---------------------------------------------------------------------------
C   LATEST REVISION     - AUGUST 15, 1973
C
      DIMENSION          A(IA,1),B(IA,1),WKAREA(1)
C                                  INITIALIZE IER
      IER=0
```

```
C                                 DECOMPOSE A                           LE1F0530
      CALL LUDATF (A,A,N,IA,IDGT,D1,D2,WKAREA,WKAREA,WA,IER)            LE1F0540
      IF (IER .GT. 128) GO TO 9000                                      LE1F0550
C                                 CALL ROUTINE LUELMF (FORWARD AND      LE1F0560
C                                 BACKWARD SUBSTITUTIONS)               LE1F0570
      DO 10 J=1,M                                                       LE1F0580
         CALL LUELMF (A,B(1,J),WKAREA,N,IA,B(1,J))                      LE1F0590
   10 CONTINUE                                                          LE1F0600
      IF (IER .EQ. 0) GO TO 9005                                        LE1F0610
 9000 CONTINUE                                                          LE1F0620
      CALL UERTST (IER,6HLEQT1F)                                        LE1F0630
 9005 RETURN                                                            LE1F0640
      END                                                               LE1F0650


      SUBROUTINE LUDATF (A,LU,N,IA,IDGT,D1,D2,IPVT,EQUIL,WA,IER)        LUDA0010
C                                                                       LUDA0020
C-LUDATF--------S--------LIBRARY 3--------------------------------------LUDA0030
C                                                                       LUDA0040
C   FUNCTION            - L-U DECOMPOSITION BY THE CROUT ALGORITHM      LUDA0050
C                           WITH OPTIONAL ACCURACY TEST.                LUDA0060
C   USAGE               - CALL LUDATF(A,LU,N,IA,IDGT,D1,D2,IPVT,        LUDA0070
C                           EQUIL,WA,IER)                               LUDA0080
C   PARAMETERS   A      - INPUT MATRIX OF DIMENSION N BY N CONTAINING   LUDA0090
C                           THE MATRIX TO BE DECOMPOSED                 LUDA0100
C                LU     - REAL OUTPUT MATRIX OF DIMENSION N BY N        LUDA0110
C                           CONTAINING THE L-U DECOMPOSITION OF A       LUDA0120
C                           ROWWISE PERMUTATION OF THE INPUT MATRIX.    LUDA0130
C                           FOR A DESCRIPTION OF THE FORMAT OF LU, SEE  LUDA0140
C                           EXAMPLE.                                    LUDA0150
C                N      - INPUT SCALAR CONTAINING THE ORDER OF THE      LUDA0160
C                           MATRIX A.                                   LUDA0170
C                IA     - INPUT SCALAR CONTAINING THE ROW DIMENSION OF  LUDA0180
C                           MATRICES A AND LU IN THE CALLING PROGRAM.   LUDA0190
C                IDGT   - INPUT OPTION.                                 LUDA0200
C                           IF IDGT IS GREATER THAN ZERO, THE NON-ZERO  LUDA0210
C                           ELEMENTS OF A ARE ASSUMED TO BE CORRECT TO  LUDA0220
C                           IDGT DECIMAL PLACES.  LUDATF PERFORMS AN    LUDA0230
C                           ACCURACY TEST TO DETERMINE IF THE COMPUTED  LUDA0240
C                           DECOMPOSITION IS THE EXACT DECOMPOSITION    LUDA0250
C                           OF A MATRIX WHICH DIFFERS FROM THE GIVEN ONELUDA0260
C                           BY LESS THAN ITS UNCERTAINTY.               LUDA0270
C                         IF IDGT IS EQUAL TO ZERO, THE ACCURACY TEST ISLUDA0280
C                           BYPASSED.                                   LUDA0290
C                D1     - OUTPUT SCALAR CONTAINING ONE OF THE TWO       LUDA0300
C                           COMPONENTS OF THE DETERMINANT. SEE          LUDA0310
C                           DESCRIPTION OF PARAMETER D2, BELOW.         LUDA0320
C                D2     - OUTPUT SCALAR CONTAINING ONE OF THE           LUDA0330
C                           TWO COMPONENTS OF THE DETERMINANT. THE      LUDA0340
C                           DETERMINANT MAY BE EVALUATED AS (D1)(2**D2) LUDA0350
C                IPVT   - OUTPUT VECTOR OF LENGTH N CONTAINING THE      LUDA0360
C                           PERMUTATION INDICES. SEE DOCUMENT           LUDA0370
C                           (ALGORITHM).                                LUDA0380
C                EQUIL  - OUTPUT VECTOR OF LENGTH N CONTAINING          LUDA0390
C                           RECIPROCALS OF THE ABSOLUTE VALUES OF       LUDA0400
C                           THE LARGEST (IN ABSOLUTE VALUE) ELEMENT     LUDA0410
C                           IN EACH ROW.                                LUDA0420
C                WA     - ACCURACY TEST PARAMETER, OUTPUT ONLY IF       LUDA0430
C                           IDGT IS GREATER THAN ZERO.                  LUDA0440
C                           SEE ELEMENT DOCUMENTATION FOR DETAILS.      LUDA0450
C                IER    - ERROR PARAMETER                               LUDA0460
C                         TERMINAL ERROR=128+N                          LUDA0470
C                           N = 1 INDICATES THAT MATRIX A IS            LUDA0480
C                               ALGORITHMICALLY SINGULAR. (SEE THE      LUDA0490
C                               CHAPTER L PRELUDE).                     LUDA0500
C                         WARNING ERROR=32+N                            LUDA0510
C                           N = 2 INDICATES THAT THE ACCURACY TEST      LUDA0520
C                               FAILED.                                 LUDA0530
C                               THE COMPUTED SOLUTION MAY BE IN ERROR   LUDA0540
C                               BY MORE THAN CAN BE ACCOUNTED FOR BY    LUDA0550
C                               THE UNCERTAINTY OF THE DATA.            LUDA0560
C                               THIS WARNING CAN BE PRODUCED ONLY IF    LUDA0570
C                               IDGT IS GREATER THAN 0 ON INPUT.        LUDA0580
C                               SEE CHAPTER L PRELUDE FOR FURTHER       LUDA0590
C                               DISCUSSION.                             LUDA0600
C   PRECISION           - SINGLE                                        LUDA0610
C   REQD. IMSL ROUTINES - UERTST                                        LUDA0620
C   LANGUAGE            - FORTRAN                                       LUDA0630
C-----------------------------------------------------------------------LUDA0640
C   LATEST REVISION     - AUGUST 15, 1973                               LUDA0650
C                                                                       LUDA0660
      DIMENSION          A(IA,1),LU(IA,1),IPVT(1),EQUIL(1)              LUDA0670
      REAL               LU                                             LUDA0680
      DATA               ZERO,ONE,FOUR,SIXTN,SIXTH/0.0,1.,4.,16.,.0625/ LUDA0690
C                                  INITIALIZATION                       LUDA0700
      IER = 0                                                           LUDA0710
      RN = N                                                            LUDA0720
      WREL = ZERO                                                       LUDA0730
      D1 = ONE                                                          LUDA0740
      D2 = ZERO                                                         LUDA0750
      BIGA = ZERO                                                       LUDA0760
      DO 10 I=1,N                                                       LUDA0770
         BIG = ZERO                                                     LUDA0780
         DO 5 J=1,N                                                     LUDA0790
            P = A(I,J)                                                  LUDA0800
            LU(I,J) = P                                                 LUDA0810
            P = ABS(P)                                                  LUDA0820
            IF (P .GT. BIG) BIG = P                                     LUDA0830
    5    CONTINUE                                                       LUDA0840
         IF (BIG .GT. BIGA) BIGA = BIG                                  LUDA0850
         IF (BIG .EQ. ZERO) GO TO 110                                   LUDA0860
         EQUIL(I) = ONE/BIG                                             LUDA0870
   10 CONTINUE                                                          LUDA0880
      DO 105 J=1,N                                                      LUDA0890
         JM1 = J-1                                                      LUDA0900
         IF (JM1 .LT. 1) GO TO 40                                       LUDA0910
C                                  COMPUTE U(I,J), I=1,...,J-1          LUDA0920
         DO 35 I=1,JM1                                                  LUDA0930
            SUM = LU(I,J)                                               LUDA0940
            IM1 = I-1                                                   LUDA0950
            IF (IDGT .EQ. 0) GO TO 25                                   LUDA0960
C                                  WITH ACCURACY TEST                   LUDA0970
            AI = ABS(SUM)                                               LUDA0980
            WI = ZERO                                                   LUDA0990
            IF (IM1 .LT. 1) GO TO 20                                    LUDA1000
            DO 15 K=1,IM1                                               LUDA1010
               T = LU(I,K)*LU(K,J)                                      LUDA1020
               SUM = SUM-T                                              LUDA1030
               WI = WI+ABS(T)                                           LUDA1040
   15       CONTINUE                                                    LUDA1050
            LU(I,J) = SUM                                               LUDA1060
   20       WI = WI+ABS(SUM)                                            LUDA1070
```

```
      IF (AI .EQ. ZERO) AI = BIGA                                       LUDA1080
      TEST = WI/AI                                                      LUDA1090
      IF (TEST .GT. WREL) WREL = TEST                                   LUDA1100
      GO TO 35                                                          LUDA1110
C                                        WITHOUT ACCURACY               LUDA1120
   25 IF (JM1 .LT. 1) GO TO 35                                          LUDA1130
      DO 30 K=1,JM1                                                     LUDA1140
         SUM = SUM-LU(I,K)*LU(K,J)                                      LUDA1150
   30 CONTINUE                                                          LUDA1160
      LU(I,J) = SUM                                                     LUDA1170
   35 CONTINUE                                                          LUDA1180
   40 P = ZERO                                                          LUDA1190
C                               COMPUTE U(J,J) AND L(I,J), I=J+1,...,LUDA1200
      DO 70 I=J,N                                                       LUDA1210
         SUM = LU(I,J)                                                  LUDA1220
         IF (IDGT .EQ. 0) GO TO 55                                      LUDA1230
C                               WITH ACCURACY TEST                      LUDA1240
         AI = ABS(SUM)                                                  LUDA1250
         WI = ZERO                                                      LUDA1260
         IF (JM1 .LT. 1) GO TO 50                                       LUDA1270
         DO 45 K=1,JM1                                                  LUDA1280
            T = LU(I,K)*LU(K,J)                                         LUDA1290
            SUM = SUM-T                                                 LUDA1300
            WI = WI+ABS(T)                                              LUDA1310
   45    CONTINUE                                                       LUDA1320
         LU(I,J) = SUM                                                  LUDA1330
   50    WI = WI+ABS(SUM)                                               LUDA1340
         IF (AI .EQ. ZERO) AI = BIGA                                    LUDA1350
         TEST = WI/AI                                                   LUDA1360
         IF (TEST .GT. WREL) WREL = TEST                                LUDA1370
         GO TO 65                                                       LUDA1380
C                               WITHOUT ACCURACY TEST                   LUDA1390
   55    IF (JM1 .LT. 1) GO TO 65                                       LUDA1400
         DO 60 K=1,JM1                                                  LUDA1410
            SUM = SUM-LU(I,K)*LU(K,J)                                   LUDA1420
   60    CONTINUE                                                       LUDA1430
         LU(I,J) = SUM                                                  LUDA1440
   65    Q = EQUIL(I)*ABS(SUM)                                          LUDA1450
         IF (P .GE. Q) GO TO 70                                         LUDA1460
         P = Q                                                          LUDA1470
         IMAX = I                                                       LUDA1480
   70 CONTINUE                                                          LUDA1490
C                               TEST FOR ALGORITHMIC SINGULARITY        LUDA1500
      IF (RN+P .EQ. RN) GO TO 110                                       LUDA1510
      IF (J .EQ. IMAX) GO TO 80                                         LUDA1520
C                               INTERCHANGE ROWS J AND IMAX             LUDA1530
      D1 = -D1                                                          LUDA1540
      DO 75 K=1,N                                                       LUDA1550
         P = LU(IMAX,K)                                                 LUDA1560
         LU(IMAX,K) = LU(J,K)                                           LUDA1570
         LU(J,K) = P                                                    LUDA1580
   75 CONTINUE                                                          LUDA1590
      EQUIL(IMAX) = EQUIL(J)                                            LUDA1600
   80 IPVT(J) = IMAX                                                    LUDA1610
      D1 = D1*LU(J,J)                                                   LUDA1620
   85 IF (ABS(D1) .LE. ONE) GO TO 90                                    LUDA1630
      D1 = D1*SIXTH                                                     LUDA1640
      D2 = D2+FOUR                                                      LUDA1650
      GO TO 85                                                          LUDA1660
   90 IF (ABS(D1) .GE. SIXTH) GO TO 95                                  LUDA1670
      D1 = D1*SIXTN                                                     LUDA1680
      D2 = D2-FOUR                                                      LUDA1690
         GO TO 90                                                       LUDA1700
   95    CONTINUE                                                       LUDA1710
         JP1 = J+1                                                      LUDA1720
         IF (JP1 .GT. N) GO TO 105                                      LUDA1730
C                               DIVIDE BY PIVOT ELEMENT U(J,J)          LUDA1740
         P = LU(J,J)                                                    LUDA1750
         DO 100 I=JP1,N                                                 LUDA1760
            LU(I,J) = LU(I,J)/P                                         LUDA1770
  100    CONTINUE                                                       LUDA1780
  105 CONTINUE                                                          LUDA1790
C                               PERFORM ACCURACY TEST                   LUDA1800
      IF (IDGT .EQ. 0) GO TO 9005                                       LUDA1810
      P = 3*N+3                                                         LUDA1820
      WA = P*WREL                                                       LUDA1830
      IF (WA+10.0**(-IDGT) .NE. WA) GO TO 9005                          LUDA1840
      IER = 34                                                          LUDA1850
      GO TO 9000                                                        LUDA1860
C                               ALGORITHMIC SINGULARITY                 LUDA1870
  110 IER = 129                                                         LUDA1880
      D1 = ZERO                                                         LUDA1890
      D2 = ZERO                                                         LUDA1900
 9000 CONTINUE                                                          LUDA1910
C                               PRINT ERROR                             LUDA1920
      CALL UERTST(IER,6HLUDATF)                                         LUDA1930
 9005 RETURN                                                            LUDA1940
      END                                                               LUDA1950
```

```
      SUBROUTINE LUELMF (A,B,IPVT,N,IA,X)                               LUEF0010
C                                                                       LUEF0020
C-LUELMF--------S--------LIBRARY 3-------------------------------------LUEF0030
C                                                                       LUEF0040
C   FUNCTION            - ELIMINATION PART OF SOLUTION OF AX=B -        LUEF0050
C                           FULL STORAGE MODE                           LUEF0060
C   USAGE               - CALL LUELMF (A,B,IPVT,N,IA,X)                 LUEF0070
C   PARAMETERS   A      - THE RESULT, LU, COMPUTED IN THE SUBROUTINE    LUEF0080
C                           *LUDATF*, WHERE L IS A LOWER TRIANGULAR     LUEF0090
C                           MATRIX WITH ONES ON THE MAIN DIAGONAL. U IS LUEF0100
C                           UPPER TRIANGULAR. L AND U ARE STORED AS A   LUEF0110
C                           SINGLE MATRIX A, AND THE UNIT DIAGONAL OF   LUEF0120
C                           L IS NOT STORED                             LUEF0130
C                B      - B IS A VECTOR OF LENGTH N ON THE RIGHT HAND   LUEF0140
C                           SIDE OF THE EQUATION AX=B                   LUEF0150
C                IPVT   - THE PERMUTATION MATRIX RETURNED FROM THE      LUEF0160
C                           SUBROUTINE *LUDATF*, STORED AS AN N LENGTH  LUEF0170
C                           VECTOR                                      LUEF0180
C                N      - ORDER OF A AND NUMBER OF ROWS IN B            LUEF0190
C                IA     - NUMBER OF ROWS IN THE DIMENSION STATEMENT     LUEF0200
C                           FOR A IN THE CALLING PROGRAM.               LUEF0210
C                X      - THE RESULT X                                  LUEF0220
C   PRECISION           - SINGLE                                        LUEF0230
C   LANGUAGE            - FORTRAN                                       LUEF0240
C-----------------------------------------------------------------------LUEF0250
C   LATEST REVISION     - AUGUST 15, 1973                               LUEF0260
C                                                                       LUEF0270
      DIMENSION          A(IA,1),B(1),IPVT(1),X(1)                      LUEF0280
C                               SOLVE UX=Y FOR X                        LUEF0290
      DO 5 I=1,N                                                        LUEF0300
    5 X(I) = B(I)                                                       LUEF0310
      IW = 0                                                            LUEF0320
```

```
      DO 20 I=1,N
         IP = IPVT(I)
         SUM = X(IP)
         X(IP) = X(I)
         IF (IW .EQ. 0) GO TO 15
         IM1 = I-1
         DO 10 J=IW,IM1
            SUM = SUM-A(I,J)*X(J)
   10    CONTINUE
         GO TO 20
   15    IF (SUM .NE. 0.) IW = I
   20 X(I) = SUM
C                                 SOLVE LY=B FOR Y
      DO 30 IB=1,N
         I = N+1-IB
         IP1 = I+1
         SUM = X(I)
         IF (IP1 .GT. N) GO TO 30
         DO 25 J=IP1,N
            SUM = SUM-A(I,J)*X(J)
   25    CONTINUE
   30 X(I) = SUM/A(I,I)
      RETURN
      END
```

```
      SUBROUTINE MDCH   (CS,DF,P,IER)
C
C-MDCH-----------S--------LIBRARY 3-----------------------------------------
C
C   FUNCTION            - CHI-SQUARED PROBABILITY - NON-INTEGER DEGREES
C                           OF FREEDOM
C   USAGE               - CALL MDCH (CS,DF,P,IER)
C   PARAMETERS   CS     - INPUT VALUE FOR WHICH THE PROBABILITY IS
C                           COMPUTED. CS MUST BE GREATER THAN OR EQUAL
C                           TO ZERO.
C                DF     - INPUT VALUE CONTAINING NUMBER OF DEGREES OF
C                           FREEDOM OF THE CHI-SQUARED DISTRIBUTION. DF
C                           MUST BE GREATER THAN OR EQUAL TO .5 AND
C                           LESS THAN OR EQUAL TO 200,000.
C                P      - OUTPUT VALUE CONTAINING PROBABILITY
C                IER    - ERROR PARAMETER.
C                         TERMINAL ERROR=128+N.
C                           N = 1 INDICATES THAT CS OR DF WAS
C                             SPECIFIED INCORRECTLY.
C                         WARNING ERROR=32+N.
C                           N = 2 INDICATES THAT THE NORMAL PDF WOULD
C                             HAVE PRODUCED AN UNDERFLOW.
C   PRECISION           - SINGLE
C   REQD. IMSL ROUTINES - UERTST,MERF,MGAMMA
C   LANGUAGE            - FORTRAN
C--------------------------------------------------------------------------
C   LATEST REVISION     - APRIL 3,1974
C
      DATA              RINFM/Z7FFFFFFF/
      FUNC(W,A,Z)=W* EXP(A*ALOG(Z)-Z)
C                                 TEST FOR INVALID INPUT VALUES
      IF (DF .GE. .5 .AND. DF .LE. 2.E5 .AND. CS .GE. 0.0) GO TO 5
      IER=129
      P=RINFM
      GO TO 9000
    5 IER=0
C                                  SET P=0. IF CS IS LESS THAN OR
C                                  EQUAL TO 10.**(-12)
      IF (CS .GT. 1.E-12) GO TO 15
   10 P=0.0
      GO TO 9005
   15 IF (DF .LE. 100.) GO TO 20
C                                  USE NORMAL DISTRIBUTION APPROXIMATION
C                                  FOR LARGE DEGREES OF FREEDOM
      X=SQRT(CS+CS)-SQRT(DF+DF-1.)
      IF (X .GT. 5.0) GO TO 50
      IF (X .LT. -11.313708) GO TO 55
      CALL MDNOR (X,P)
      GO TO 9005
C                                  INITIALIZATION FOR CALCULATION USING
C                                  INCOMPLETE GAMMA FUNCTION
   20 IF (CS .GT. 200.) GO TO 50
      EPS = 1.E-14
      A = 5.E-1*DF
      Z = 5.E-1*CS
      CALL MGAMMA (A,DGAM,IER)
      W = AMAX1(.5*A,13.0)
      IF (Z .GE. W) GO TO 35
      IF (DF .GT. 25. .AND. CS .LT. 2.) GO TO 10
C                                  CALCULATE USING EQUATION NO. 6.5.29
      W = 1.00/(DGAM*A)
      W1=W
         DO 25 I=1,50
         B=I
         W1=W1*Z/(A+B)
         IF (W1 .LE. EPS*W) GO TO 30
         W=W+W1
   25    CONTINUE
   30 P=FUNC(W,A,Z)
      GO TO 9005
C                                  CALCULATE USING EQUATION NO. 6.5.32
   35 Z1 = 1./Z
      B = A-1.
      W1=B*Z1
      W = 1.0+W1
         DO 40 I=2,50
         B = B-1.0
         W1=W1*B*Z1
         IF (W1 .LE. EPS*W) GO TO 45
         W=W+W1
   40    CONTINUE
   45 W=Z1*FUNC(W,A,Z)
      P = 1.-W/DGAM
      GO TO 9005
   50 P=1.0
      GO TO 9005
C                                  WARNING ERROR - UNDERFLOW WOULD HAVE
C                                  OCCURRED
   55 P=0.0
      IER=34
 9000 CONTINUE
      CALL UERTST(IER,6HMDCH)
 9005 RETURN
      END
```

```
      SUBROUTINE MDCHI  (P,DF,X,IER)
C
C-MDCHI--------S--------LIBRARY 3--------------------------------------
C
C   FUNCTION            - INVERSE CHI-SQUARED PROBABILITY DISTRIBUTION
C                           FUNCTION
C   USAGE               - CALL MDCHI (P,DF,X,IER)
C   PARAMETERS   P      - INPUT PROBABILITY IN THE EXCLUSIVE RANGE
C                           (0,1)
C                DF     - INPUT VALUE CONTAINING NUMBER OF DEGREES OF
C                           FREEDOM.  DF MUST BE IN THE EXCLUSIVE RANGE
C                           (.5,200000.)
C                X      - OUTPUT CHI-SQUARED VALUE, SUCH THAT A RANDOM
C                           VARIABLE, DISTRIBUTED AS CHI-SQUARED WITH
C                           DF DEGREES OF FREEDOM, WILL BE LESS THAN OR
C                           EQUAL TO X WITH PROBABILITY P.
C                IER    - ERROR INDICATOR
C                         TERMINAL ERROR = 128+N
C                           N = 1 MEANS THAT BOUNDS WHICH ENCLOSED P
C                           COULD NOT BE FOUND WITHIN 20 (NCT)
C                           ITERATIONS
C                           N = 2 MEANS THAT AN ERROR OCCURRED IN
C                           MDNRIS (P IS NOT A VALID VALUE)
C                           N = 3 MEANS THAT AN ERROR OCCURRED IN MDCH
C                           N = 4 MEANS THAT THE VALUE X COULD NOT BE
C                           FOUND WITHIN 50 (ITMAX) ITERATIONS, SO
C                           THAT THE ABSOLUTE VALUE OF P1-P WAS NOT
C                           .LE. EPS, (P1= CALCULATED PROBABILITY AT X,
C                           EPS= .0001)
C   PRECISION           - SINGLE
C   REQD. IMSL ROUTINES - MDCH,MERFI,UERTST,MERF,MGAMMA
C   LANGUAGE            - FORTRAN
C----------------------------------------------------------------------
C   LATEST REVISION     - SEPTEMBER 1, 1973
C
      DATA              EPS/.0001/,ITMAX/50/,NCT/20/,NSIG/5/
      IC = 0
      IER = 0
C                                  ESTIMATE STARTING X
      CALL MDNRIS (P,XP,IER)
      IF (IER .NE. 0) GO TO 80
      DFF = .2222222222222222/DF
      D = SQRT(DFF)
      X  = DF*(1. -DFF + XP*D)**3
C                                  IS THE CASE ASYMPTOTICALLY NORMAL
      IF (DF .GE. 40.) GO TO 9005
C                                  FIND BOUNDS (IN X) WHICH ENCLOSE P
      NCNT = 0
      IST = 0
      ISW = 0
      DX = X * .125
    5 IF (X) 10,15,20
   10 X = 0.0
      DX = -DX
      GO TO 20
   15 DX = .1
   20 CALL MDCH (X,DF,P1,IER)
      DX = DX + DX
      IF (IER .NE. 0) GO TO 85
      CSS = X
      NCNT = NCNT + 1
      IF (NCNT .GT. NCT) GO TO 90
      IF (P1 - P) 25,9005,30
   25 X = X + DX
      ISW = 1
      IF (IST .EQ. 0) GO TO 5
      GO TO 35
   30 X = X - DX
      IST = 1
      IF (ISW .EQ. 0) GO TO 5
      XR = CSS
      XL = X
      GO TO 40
   35 XL = CSS
      XR = X
C                                  PREPARE FOR ITERATION TO FIND X
   40 EPSP = 10.**(-NSIG)
      IF (XL .LT. 0.) XL = 0.0
      CALL MDCH (XL,DF,P1,IER)
      IF (IER .NE. 0) GO TO 85
      FXL = P1 - P
      CALL MDCH (XR,DF,P1,IER)
      IF (IER .NE. 0) GO TO 85
      FXR = P1-P
      IF (FXL*FXR .NE. 0.0) GO TO 45
      X = XR
      IF (FXL .EQ. 0.0) X = XL
      GO TO 9005
   45 IF (DF .LE. 2. .OR. P .GT. .98) GO TO 50
C                                  REGULA FALSI METHOD
      X = XL + FXL*(XR-XL)/(FXL-FXR)
      GO TO 55
C                                  BISECTION METHOD
   50 X = (XL+XR) * .5
   55 CALL MDCH (X,DF,P1,IER)
      IF (IER .NE. 0) GO TO 85
      FCS = P1-P
      IF (ABS(FCS) .GT. EPS) GO TO 60
      GO TO 9005
   60 IF (FCS * FXL .GT. 0.0) GO TO 65
      XR = X
      FXR = FCS
      GO TO 70
   65 XL = X
      FXL = FCS
   70 IF (XR-XL .GT. EPSP*ABS(XR)) GO TO 75
      GO TO 9005
   75 IC = IC+1
      IF (IC .LE. ITMAX) GO TO 45
      IER = 132
C                                  ERROR RETURNED FROM MDNRIS
   80 IER = 130
      GO TO 9000
C                                  ERROR RETURNED FROM MDCH
   85 IER = 131
      GO TO 9000
   90 IER = 129
 9000 CONTINUE
      CALL UERTST (IER,6HMDCHI )
 9005 RETURN
      END
```

```
      SUBROUTINE MERF   (Y,ERF)
C
C-MERF-----------S--------LIBRARY 3--------------------------------------
C-MERFC
C-MDNOR
C
C   FUNCTION      MERF   - COMPUTE THE ERROR FUNCTION
C                 MERFC  - COMPUTE THE COMPLEMENTED ERROR FUNCTION
C                 MDNOR  - COMPUTE THE NORMAL PROBABILITY DENSITY
C                            FUNCTION
C   USAGE                - CALL MERF(Y,ERF)
C                          CALL MERFC(Y,ERF)
C                          CALL MDNOR(Y,ERF)
C   PARAMETERS   Y       - INPUT ARGUMENT
C                            Y MAY BE ANY FINITE REAL NUMBER
C                ERF     - FOR MERF ERF IS THE VALUE OF THE ERROR
C                            FUNCTION
C                          FOR MERFC ERF IS THE VALUE OF THE COMPLEMENTED
C                            ERROR FUNCTION
C                          FOR MDNOR ERF IS THE VALUE OF THE NORMAL
C                            PROBABILITY DENSITY FUNCTION
C   PRECISION            - SINGLE
C   LANGUAGE             - FORTRAN
C-----------------------------------------------------------------------
C   LATEST REVISION      - AUGUST 10,1971
C
      DIMENSION          A(10),P(8),Q(9)
      DATA               C,SQR1D2/1.1283791670955,.70710678118655/
      DATA               A/1.,-.33333333333333,.1,-.023809523809238,
     1                     .46296296296300E-2,-.75757575757576E-3,
     2                     1.0683760683761E-4,-1.3227513227513E-5,
     3                     1.4589169000934E-6,-1.4503855222315OE-7/
      DATA               P/883.47894260850,1549.6793124037,
     1                     1347.1941340976,723.04000277753,
     2                     255.50049469496,59.240010112914,
     3                     8.3765310814197,.56418955944261/
      DATA               Q/883.47894260850,2546.5785458098,
     1                     3337.2213699893,2606.7120152651,
     2                     1333.5699756800,460.28512369160,
     3                     105.50025439769,14.847012237523,1.0/
C                                   ERROR FUNCTION ENTRY
      KRET = 2
    5 X = Y
      MD = 0
   10 ISW = 2
      IF (X .GE. 0.) GO TO 15
      ISW=1
      X = -X
   15 IF (KRET .NE. 2) GO TO 60
      IF (X .LE. 5.5) GO TO 20
      ERF = 1.
      IF (ISW .EQ. 1) ERF = -1.
      GO TO 9005
C                                   ABS(X) LESS THAN .47 COMPUTE
C                                     ERF BY TAYLOR SERIES EXPANSION
   20 IF (X .LE. .47) GO TO 45
      KRET = 1
C                                   ABS(X) BETWEEN .47 AND 10. COMPUTE
C                                     COMPLEMENTED ERROR FUNCTION BY
C                                   A RATIONAL FUNCTION IN X
   25 SN = P(8)
      SD = Q(9)
      DO 30 J=1,7,1
         I = 8-J
         SN = SN * X + P(I)
   30 CONTINUE
      DO 35  J=1,8
         I = 9-J
         SD = SD * X + Q(I)
   35 CONTINUE
      ERF = (SN/SD) * EXP(-X*X)
      IF (KRET .NE. 1) GO TO 40
C                                   COMPUTE COMPLEMENTED ERROR FUNCTION
      ERF = 1.- ERF
      IF (ISW .NE. 2) ERF = -ERF
      GO TO 9005
   40 IF (ISW .NE. 2) ERF = 2.-ERF
      IF (MD .NE. 0) ERF = .5*ERF
      GO TO 9005
   45 XX = X*X
      ERF = A(10)
      DO 50  J=1,9
         I = 10-J
         ERF = ERF * XX + A(I)
   50 CONTINUE
      ERF = C * ERF * X
      IF (KRET .NE. 2) GO TO 55
      IF (ISW .NE. 2) ERF = -ERF
      GO TO 9005
   55 ERF = 1. - ERF
      GO TO 40
C                                   COMPLEMENTED ERROR FUNCTION ENTRY
      ENTRY MERFC
      KRET = 1
      GO TO 5
C                                   GAUSSIAN FUNCTION ENTRY POINT
      ENTRY MDNOR
C                                   COMPUTE GAUSSIAN FUNCTION FROM MERFC
      X = -Y * SQR1D2
      MD = 1
      KRET=1
      GO TO 10
   60 IF (X .LE. 8.0) GO TO 65
      ERF = 0.
      GO TO 40
   65 IF (X .LE. .47) GO TO 70
      KRET = 2
      GO TO 25
   70 IF (X .GT. 1.E-15) GO TO 45
C                                   ABS(X) LESS THAN 1.E-15 SET ERFC=1.
      ERF = 1.
      GO TO 40
 9005 RETURN
      END


      SUBROUTINE MERFI  (P,Y,IER)
C
C-MERFI----------S--------LIBRARY 3--------------------------------------
```

```
C-MERFCI
C-MDNRIS
C
C  FUNCTION         MERFI  - COMPUTE THE INVERSE ERROR FUNCTION.
C                   MERFCI - COMPUTE THE INVERSE COMPLEMENTED
C                               ERROR FUNCTION.
C                   MDNRIS - COMPUTE THE INVERSE GAUSSIAN INTEGRAL.
C
C  USAGE                   - CALL MERFI(P,X,IER)
C                          - CALL MERFCI(P,X,IER)
C                          - CALL MDNRIS(P,X,IER)
C  PARAMETERS       P      - INPUT VALUE
C                               FOR MERFI P MUST BE BETWEEN -1.0 AND 1.0
C                               FOR MERFCI P MUST BE BETWEEN 0.0 AND 2.0
C                               FOR MDNRIS P MUST BE BETWEEN 0.0 AND 1.0
C                   X      - OUTPUT RESULT OF COMPUTATION.
C                   IER    - ERROR INDICATOR
C                            TERMINAL ERROR = 128+N
C                              N=1 INDICATES P LIES OUTSIDE THE LEGAL
C                              DOMAIN. PLUS OR MINUS INFINITY IS GIVEN AS
C                              THE RESULT (SIGN IS THE SIGN OF THE FUNCTION
C                              VALUE OF THE NEAREST LEGAL ARGUMENT).
C   PRECISION              - SINGLE
C   REQD. IMSL ROUTINES    - UERTST
C   LANGUAGE               - FORTRAN
C-----------------------------------------------------------------------
C   LATEST REVISION        - AUGUST 15, 1971
C
      DIMENSION          A(65),C(17),D(25),E(23)
      EQUIVALENCE        (A(24),C(1)),(A(41),D(1)),(A(1),E(1))
      DATA               E/.992885376618940,.120467516143104,
     *                   .016078199342100,.002686704437162,
     *                   .000499634730235,.000098898218599,
     *                   .000020391812764,.000004327271618,
     *                   .000000938081413,.000000206734721,
     *                   .000000046159699,.000000010416680,
     *                   .000000002371501,.000000000543928,
     *                   .000000000125549,.000000000029138,
     *                   .000000000006795,.000000000001591,
     *                   .000000000000374,.000000000000088,
     *                   .000000000000021,.000000000000005,
     *                   .000000000000001/
      DATA               C/.912158803417554,-.016266281867664,
     *                   .000433556472949,.000214438570074,
     *                   .000002625751076,-.000003021091050,
     *                   -.000000012406062,.000000062406609,
     *                   -.000000000540125,-.000000001423208,
     *                   .000000000034384,.000000000033585,
     *                   -.000000000001458,-.000000000000810,
     *                   .000000000000053,.000000000000020,
     *                   -.000000000000002/
      DATA               D/.956679709020493,-.023107004309065,
     *                   -.004374236097508,-.000576503422651,
     *                   -.000010961022307,.000025108547025,
     *                   .000010562336068,.000002754412330,
     *                   .000000432484498,-.000000020530337,
     *                   -.000000043891537,-.000000017684010,
     *                   -.000000003991289,-.000000000186932,
     *                   .000000000272923,.000000000132817,
     *                   .000000000031834,.000000000001670,
     *                   -.000000000002036,-.000000000000965,
     *                   -.000000000000220,-.000000000000010,
     *                   .000000000000013,.000000000000006,
     *                   .000000000000001/
      DATA               H1,H2,H3,H4,R2/-1.5488130423753,
     *                   2.5654901231478,-.55945763132983,
     *                   2.2879157162634,-1.4142135623731/
      DATA               XINF/37767777777777777770/
C     INVERSE ERROR FUNCTION ENTRY
      INT = 1
      X = P
      GO TO 5
C     INVERSE COMPLEMENTED ERROR FUNCTION ENTRY
      ENTRY MERFCI
      INT = 2
      X = 1. - P
      GO TO 5
C     INVERSE GAUSSIAN INTEGRAL ENTRY
      ENTRY MDNRIS
      INT = 3
      X = 1.-P-P
    5 IER = 0
      SIGMA = SIGN(1.,X)
      IF (.NOT.(X.GT.-1..AND.X.LT.1.)) GO TO 45
      Z = ABS(X)
      IF(Z.GT..8) GO TO 25
      W = Z*Z/.32-1.
      N = 22
      IPP = 1
      L = 1
   10 LB2 = 1
      X3 = 1.
      X4 = W
      X6 = A(IPP)
   15 X6 = X6 + A(IPP+LB2) * X4
      X5 = X4 * W * 2.-X3
      X3 = X4
      X4 = X5
      LB2 = LB2 + 1
      IF (LB2 .LE. N) GO TO 15
      GO TO (20,35),L
   20 Y = Z * X6 * SIGMA
      IF (INT .NE. 3) GO TO 9005
      GO TO 40
   25 B = SQRT(-ALOG(1.-Z*Z))
      IF (X .GT. .9975) GO TO 30
      W = H1*B+H2
      IPP = 24
      L = 2
      N = 16
      GO TO 10
   30 W = H3 * B + H4
      IPP = 41
      N = 24
      L = 2
      GO TO 10
   35 Y = B * X6 * SIGMA
      IF (INT .NE. 3) GO TO 9005
   40 Y = R2*Y
      GO TO 9005
   45 Y = SIGMA*XINF
      IER = 129
 9000 CONTINUE
      CALL UERTST(IER,6HMERFI )
```

```
 9005 RETURN                                                       MFRI1280
      END                                                          MFRI1290

      SUBROUTINE MGAMMA (X,Y,IER)                                  MGAM0010
C                                                                  MGAM0020
C-MGAMMA--------S-------LIBRARY 3-----------------------------------MGAM0030
C-MLGAMA                                                           MGAM0040
C                                                                  MGAM0050
C   FUNCTION       MGAMMA - GAMMA FUNCTION OF A REAL ARGUMENT X    MGAM0060
C                  MLGAMA - NATURAL LOG OF THE GAMMA FUNCTION OF A REAL   MGAM0070
C                             ARGUMENT X                           MGAM0080
C   USAGE                 - CALL MGAMMA(X,Y,IER)                   MGAM0090
C                         - CALL MLGAMA(X,Y,IER)                   MGAM0100
C   PARAMETERS     X      - INPUT ARGUMENT                         MGAM0110
C                  Y      - OUTPUT RESULT. FOR MGAMMA, Y IS THE VALUE   MGAM0120
C                             OF THE GAMMA FUNCTION. FOR MLGAMA, Y IS   MGAM0130
C                             THE VALUE OF THE LOG OF THE ABSOLUTE VALUE  MGAM0140
C                             OF THE GAMMA FUNCTION.               MGAM0150
C                  IER    - ERROR PARAMETER.                       MGAM0160
C                           TERMINAL ERROR = 128+N                 MGAM0170
C                             N=1 INDICATES ABS(X) WAS GREATER THAN OR   MGAM0180
C                               EQUAL TO 177.803 FOR ENTRY MGAMMA  MGAM0190
C                               (1.723E319 FOR ENTRY MLGAMA). Y IS MGAM0200
C                               RETURNED AS THE LARGEST MACHINE NUMBER   MGAM0210
C                               (WITH A PROPER SIGN FOR THE MGAMMA ENTRY).MGAM0220
C                             N=2 INDICATES X WAS ZERO, OR A NEGATIVE   MGAM0230
C                               INTEGER. Y IS RETURNED AS THE LARGEST   MGAM0240
C                               MACHINE NUMBER. IF X IS AN EVEN NEGATIVE  MGAM0250
C                               INTEGER OR ZERO, THE SIGN IS NEGATIVE.   MGAM0260
C                               IF X IS AN ODD NEGATIVE INTEGER, THE SIGN MGAM0270
C                               IS POSITIVE.                       MGAM0280
C   PRECISION             - SINGLE                                 MGAM0290
C   REQD. IMSL ROUTINES   - UERTST                                 MGAM0300
C   LANGUAGE              - FORTRAN                                MGAM0310
C------------------------------------------------------------------MGAM0320
C   LATEST REVISION       - MARCH 12,1974                          MGAM0330
C                                                                  MGAM0340
      DIMENSION          P(7),Q(6),P1(8),Q1(7),P2(8),Q2(7),P3(8),  MGAM0350
     1                   Q3(7),P4(5)                               MGAM0360
      LOGICAL            IFLAG,MFLAG                               MGAM0370
C                                 COEFFICIENTS FOR MINIMAX         MGAM0380
C                                 APPROXIMATION TO GAMMA(X),       MGAM0390
C                                 2.0 .LE. X .LE. 3.0              MGAM0400
      DATA               P(1)/3.4109112397125E+01/,                MGAM0410
     1                   P(2)/-4.8341273405598E+01/,               MGAM0420
     2                   P(3)/4.3005887829594E+02/,                MGAM0430
     3                   P(4)/-5.5688734338586E+01/,               MGAM0440
     4                   P(5)/2.0585220673644E+03/,                MGAM0450
     5                   P(6)/7.7192407739800E-01/,                MGAM0460
     6                   P(7)/-3.1721064346240E+00/                MGAM0470
      DATA               Q(1)/2.4455138217658E+02/,                MGAM0480
     1                   Q(2)/-1.0174768492818E+03/,               MGAM0490
     2                   Q(3)/1.1615998272754E+03/,                MGAM0500
     3                   Q(4)/2.0512896777972E+03/,                MGAM0510
     4                   Q(5)/6.8080353498091E-01/,                MGAM0520
     5                   Q(6)/-2.5386729086746E+01/                MGAM0530
C                                 COEFFICIENTS FOR MINIMAX         MGAM0540
C                                 APPROXIMATION TO LN(GAMMA(X)),   MGAM0550
C                                 0.5 .LE. X .LE. 1.5              MGAM0560
      DATA               P1(1)/2.4317524352442E+02/,               MGAM0570
     1                   P1(2)/-2.6172185838561E+02/,              MGAM0580
     2                   P1(3)/-9.2226137288015E+02/,              MGAM0590
     3                   P1(4)/-5.1763834980232E+02/,              MGAM0600
     4                   P1(5)/-7.7410640713329E+01/,              MGAM0610
     5                   P1(6)/-2.2088439972161E+00/,              MGAM0620
     6                   P1(7)/4.1208431858477E+00/,               MGAM0630
     7                   P1(8)/8.5689820628313E+01/                MGAM0640
      DATA               Q1(1)/3.7783724848239E+02/,               MGAM0650
     1                   Q1(2)/9.5132359767970E+02/,               MGAM0660
     2                   Q1(3)/8.4607553620207E+02/,               MGAM0670
     3                   Q1(4)/2.6230834702694E+02/,               MGAM0680
     4                   Q1(5)/2.4435196625063E+01/,               MGAM0690
     5                   Q1(6)/4.0977929210926E-01/,               MGAM0700
     6                   Q1(7)/4.5646771875859E+01/                MGAM0710
C                                 COEFFICIENTS FOR MINIMAX         MGAM0720
C                                 APPROXIMATION TO LN(GAMMA(X)),   MGAM0730
C                                 1.5 .LE. X .LE. 4.0              MGAM0740
      DATA               P2(1)/5.268983255914E+03/,                MGAM0750
     1                   P2(2)/1.9553605540630E+04/,               MGAM0760
     2                   P2(3)/1.2043173809871E+04/,               MGAM0770
     3                   P2(4)/-2.0648294205325E+04/,              MGAM0780
     4                   P2(5)/-1.5086302287667E+04/,              MGAM0790
     5                   P2(6)/-1.5138318341150E+03/,              MGAM0800
     6                   P2(7)/5.1550576176408E+00/,               MGAM0810
     7                   P2(8)/3.7751067979721E+02/                MGAM0820
      DATA               Q2(1)/3.0399030414394E+03/,               MGAM0830
     1                   Q2(2)/2.2029562144156E+04/,               MGAM0840
     2                   Q2(3)/5.7120255396025E+04/,               MGAM0850
     3                   Q2(4)/5.2622863838411E+04/,               MGAM0860
     4                   Q2(5)/1.4402090371700E+04/,               MGAM0870
     5                   Q2(6)/6.9832741405735E+02/,               MGAM0880
     6                   Q2(7)/1.2890931890129E+02/                MGAM0890
C                                 COEFFICIENTS FOR MINIMAX         MGAM0900
C                                 APPROXIMATION TO LN(GAMMA(X)),   MGAM0910
C                                 4.0 .LE. X .LE. 12.0             MGAM0920
      DATA               P3(1)/-1.9718301158609E+07/,              MGAM0930
     1                   P3(2)/-8.7316754382383E+07/,              MGAM0940
     2                   P3(3)/1.1193853542998E+08/,               MGAM0950
     3                   P3(4)/4.8180771027736E+08/,               MGAM0960
     4                   P3(5)/-2.4483217690328E+08/,              MGAM0970
     5                   P3(6)/-2.4079869801733E+08/,              MGAM0980
     6                   P3(7)/-1.0377016517329E+04/,              MGAM0990
     7                   P3(8)/-9.8271022814204E+05/               MGAM1000
      DATA               Q3(1)/-3.1140628473406E+05/,              MGAM1010
     1                   Q3(2)/-1.0485775830499E+07/,              MGAM1020
     2                   Q3(3)/-1.1192541162633E+08/,              MGAM1030
     3                   Q3(4)/-4.0443592829143E+08/,              MGAM1040
     4                   Q3(5)/-4.3537071480437E+08/,              MGAM1050
     5                   Q3(6)/-7.9026111141876E+07/,              MGAM1060
     6                   Q3(7)/-2.0152751955004E+03/               MGAM1070
C                                 COEFFICIENTS FOR MINIMAX         MGAM1080
C                                 APPROXIMATION TO LN(GAMMA(X)),   MGAM1090
C                                 12.0 .LE. X                      MGAM1100
      DATA               P4(1)/9.1893853320467E-01/,               MGAM1110
     1                   P4(2)/8.3333333333267E-02/,               MGAM1120
     2                   P4(3)/-2.7777776505151E-03/,              MGAM1130
     3                   P4(4)/7.9358449768E-04/,                  MGAM1140
     4                   P4(5)/-5.82399983E-04/                    MGAM1150
      DATA               XINF/37767777777777777778/                MGAM1160
      DATA               PI/3.14159265358898/                      MGAM1170
C                                                                  MGAM1180
```

```
      IER = 0
      SIGN = -1.0
      Y = 0.0
      MFLAG = .FALSE.
      IFLAG = .TRUE.
      T = X
      IF (T.GT.0.0) GO TO 15
      IF (-T.GT.0.0) GO TO 5
      IER = 130
      MFLAG = .TRUE.
      GO TO 125
C
      ENTRY MLGAMA
C
      Y = 0.0
      MFLAG = .FALSE.
      IFLAG = .FALSE.
      T = X
      IF (T.GT.0.0) GO TO 20
      IF (T.LT.0.0) GO TO 5
      IER = 130
      GO TO 125
C                                  ARGUMENT IS NEGATIVE
    5 MFLAG = .TRUE.
      T = -T
      J = T
      R = J
      R = T-R
      I = J/2
      SIGN = 1.0
      IF (2*I.EQ.J) SIGN = -1.0
      IF (R.NE.0.0) GO TO 10
      IER = 130
      IF (IFLAG) GO TO 125
      MFLAG = .FALSE.
      GO TO 125
C                                  ARGUMENT IS NOT A NEGATIVE INTEGER
   10 R = PI/SIN(R*PI)*SIGN
      T = T+1.0
      IF (IFLAG) GO TO 15
      R = ALOG(ABS(R))
      GO TO 20
C                                  LOOKING FOR GAMMA(T), T .GE 0.0
   15 IF (T.LT.177.803E+0) GO TO 40
      IER = 129
      GO TO 125
C                                  LOOKING FOR LN(GAMMA(T)),T .GT. 0.0
   20 IF (T.LT.1.723E+319) GO TO 25
      IER = 129
      MFLAG = .FALSE.
      GO TO 125
   25 IER = 0
      IF (T.LE.4.0E+0) GO TO 30
      IF (T.LE.12.0E+0) GO TO 100
      GO TO 110
   30 IF (T.GE.1.5E+0) GO TO 90
      IF (T.GE.0.5E+0) GO TO 75
      B = -ALOG(T)
      IF (T.LE.1.0E-60) GO TO 35
      A = T
      T = T+1.0E+0
      GO TO 80
   35 Y = B
      GO TO 9005
   40 IER = 0
      IF (T.GT.12.0E0) GO TO 110
      I = T
      A = 1.0E+0
      IF (I.GT.2) GO TO 55
      I = I+1
      GO TO (45,50,65), I
C                                  0.0 .LT. T .LT. 1.0
   45 A = A/(T*(T+1.0E+0))
      T = T+2.0E+0
      GO TO 65
C                                  1.0 .LE. T .LT. 2.0
   50 A = A/T
      T = T+1.0E+0
      GO TO 65
C                                  3.0 .LE. T .LE. 12.0
   55 DO 60 J=3,I
         T = T-1.0E+0
         A = A*T
   60 CONTINUE
C                                  2.0 .LE. T .LE. 3.0
C                                  EVALUATE MINIMAX APPROXIMATION FOR
C                                  GAMMA.
   65 TOP = P(6)*T+P(7)
      DEN = T+Q(6)
      DO 70 J=1,5
         TOP = TOP*T+P(J)
         DEN = DEN*T+Q(J)
   70 CONTINUE
      Y = (TOP/DEN)*A
      IF (MFLAG) Y = R/Y
      GO TO 9005
C                                  0.5 .LE. T .LT. 1.5
C                                  EVALUATE MINIMAX APPROXIMATION FOR
C                                  LN(GAMMA(X)).
   75 B = 0
      A = (T-0.5E+0)-0.5E+0
   80 TOP = P1(7)*T+P1(8)
      DEN = T+Q1(7)
      DO 85 J=1,6
         TOP = TOP*T+P1(J)
         DEN = DEN*T+Q1(J)
   85 CONTINUE
      Y = (TOP/DEN)*A+B
      IF (MFLAG) Y = R-Y
      GO TO 9005
C                                  1.5 .LE. T .LE. 4.0
C                                  EVALUATE MINIMAX APPROXIMATION FOR
C                                  LN(GAMMA(X)).
   90 A = (T-1.0E+0)-1.0E+0
      TOP = P2(7)*T+P2(8)
      DEN = T+Q2(7)
      DO 95 J=1,6
         TOP = TOP*T+P2(J)
         DEN = DEN*T+Q2(J)
   95 CONTINUE
      Y = (TOP/DEN)*A
      IF (MFLAG) Y = R-Y
      GO TO 9005
C                                  4.0 .LT. T .LE. 12.0
```

```
C                                   EVALUATE MINIMAX APPROXIMATION FOR
C                                   LN(GAMMA(X)).
  100 TOP = P3(7)*T+P3(8)
      DEN = T+Q3(7)
      DO 105 J=1,6
         TOP = TOP*T+P3(J)
         DEN = DEN*T+Q3(J)
  105 CONTINUE
      Y = TOP/DEN
      IF (MFLAG) Y = R-Y
      GO TO 9005
C                                   12.0 .LT. X .LT. 1.723E+319
C                                   EVALUATE MINIMAX APPROXIMATION FOR
C                                   LN(GAMMA(X)).
  110 TOP = ALOG(T)
      TOP = (T-1.5E0)*(TOP-1.0E0)+TOP-1.5E0
      IF (T.GT.1.0E+50) GO TO 115
      T = 1.0E+0/T
      B = T*T
      TOP = (((P4(5)*B+P4(4))*B+P4(3))*B+P4(2))*T+P4(1)+TOP
  115 Y = TOP
      IF (IFLAG) GO TO 120
      IF (MFLAG) Y = R-Y
      GO TO 9005
  120 Y = EXP(Y)
      IF (MFLAG) Y = R/Y
      GO TO 9005
  125 Y = XINF
      IF (AMOD(X,2.).EQ.0.) Y = -Y
 9000 CONTINUE
      CALL UERTST (IER,6HMGAMMA)
 9005 RETURN
      END

      SUBROUTINE UERTST (IER,NAME)
C
C-UERTST----------------LIBRARY 3-------------------------------------------
C
C   FUNCTION            - ERROR MESSAGE GENERATION
C   USAGE               - CALL UERTST(IER,NAME)
C   PARAMETERS   IER    - ERROR PARAMETER. TYPE + N  WHERE
C                           TYPE= 128 IMPLIES TERMINAL ERROR
C                                  64 IMPLIES WARNING WITH FIX
C                                  32 IMPLIES WARNING
C                           N    = ERROR CODE RELEVANT TO CALLING ROUTINE
C                NAME   - INPUT SCALAR CONTAINING THE NAME OF THE
C                           CALLING ROUTINE AS A 6-CHARACTER LITERAL
C                           STRING.
C   LANGUAGE            - FORTRAN
C---------------------------------------------------------------------------
C   LATEST REVISION     - AUGUST 1, 1973
C
      DIMENSION          ITYP(2,4),IBIT(4)
      INTEGER            WARN,WARF,TERM,PRINTR
      EQUIVALENCE        (IBIT(1),WARN),(IBIT(2),WARF),(IBIT(3),TERM)
      DATA     ITYP       /10HWARNING   ,10H          ,
     *                     10HWARNING(WI,10HTH FIX)    ,
     *                     10HTERMINAL  ,10H          ,
     *                     10HNON-DEFINE,10HD         /,
     *         IBIT       / 32,64,128,0/
      DATA               PRINTR/6LOUTPUT/
      IER2=IER
      IF (IER2 .GE. WARN) GO TO 5
C                                   NON-DEFINED
      IER1=4
      GO TO 20
    5 IF (IER2 .LT. TERM) GO TO 10
C                                   TERMINAL
      IER1=3
      GO TO 20
   10 IF (IER2 .LT. WARF) GO TO 15
C                                   WARNING(WITH FIX)
      IER1=2
      GO TO 20
C                                   WARNING
   15 IER1=1
C                                   EXTRACT *N*
   20 IER2=IER2-IBIT(IER1)
C                                   PRINT ERROR MESSAGE
      WRITE (PRINTR,25) (ITYP(I,IER1),I=1,2),NAME,IER2,IER
   25 FORMAT(26H *** I M S L(UERTST) ***  ,2A10,4X,A6,4X,I2,
     1   8H (IER = ,I3,1H))
      RETURN
      END

      SUBROUTINE VABMXF (V,L,INC,J,VMAX)
C
C-VABMXF--------S-------LIBRARY 3-------------------------------------------
C
C   FUNCTION            - FIND THE MAXIMUM ABSOLUTE VALUE OF THE
C                           ELEMENTS OF A VECTOR OR A SUBSET OF THE
C                           ELEMENTS OF A VECTOR
C   USAGE               - CALL VABMXF (V,L,INC,J,VMAX)
C   PARAMETERS   V      - NAME OF THE VECTOR (ADDRESS OF THE FIRST
C                           ELEMENT) - INPUT
C                L      - LENGTH OF THE SUBSET VECTOR (INPUT)
C                INC    - INCREMENT BETWEEN SUCCESSIVE ELEMENTS OF THE
C                           SUBSET VECTOR (INPUT)
C                J      - INDEX IN THE SUBSET VECTOR OF THE ELEMENT
C                           HAVING THE MAXIMUM ABSOLUTE VALUE (OUTPUT)
C                VMAX   - MAXIMUM ABSOLUTE VALUE (OUTPUT)
C   PRECISION           - SINGLE
C   LANGUAGE            - FORTRAN
C---------------------------------------------------------------------------
C   LATEST REVISION     - JULY 19,1972
C
      DIMENSION          V(1)
      DATA               ZERO/0.0/
      VMAX = ZERO
      J=1
      IEND=1+(L-1)*INC
         DO 5 I=1,IEND,INC
         T=ABS(V(I))
         IF (T .LE. VMAX) GO TO 5
         VMAX=T
         J=I
    5    CONTINUE
      J=(J-1)/INC+1
      RETURN
      END

      SUBROUTINE ZSYSTM (F,EPS,NSIG,N,X,ITMAX,WA,PAR,IER)
C
```

```
C-ZSYSTM--------S-----LIBRARY 3-------------------------------------ZSYSTM
C                                                                   ZSYSTM
C   FUNCTION            - DETERMINATION OF A ROOT OF A SYSTEM OF N  ZSYSTM
C                           SIMULTANEOUS NONLINEAR EQUATIONS IN N   ZSYSTM
C                           UNKNOWNS, F(X)=0, IN VECTOR FORM. N CAN BE 1ZSYSTM
C   USAGE               - CALL ZSYSTM (F,EPS,NSIG,N,X,ITMAX,WA,PAR,IER) ZSYSTM
C   PARAMETERS   F      - F IS THE NAME OF THE FUNCTION CALLED      ZSYSTM
C                           BY ZSYSTM TO FURNISH THE VALUES OF THE  ZSYSTM
C                           FUNCTIONS WHICH DEFINE THE SYSTEM OF    ZSYSTM
C                           EQUATIONS BEING SOLVED. THE USER SPECIFIES ZSYSTM
C                           F BY WRITING A FUNCTION SUBPROGRAM      ZSYSTM
C                           F(X,K,PAR) WHICH COMPUTES THE K-TH COMPONENTZSYSTM
C                           OF F EVALUATED AT X. (INPUT)            ZSYSTM
C                EPS    - FIRST STOPPING CRITERION. A ROOT X(1),...,ZSYSTM
C                           X(N) IS ACCEPTED IF THE MAXIMUM ABSOLUTE ZSYSTM
C                           VALUE OF F(X,K,PAR) IS LESS THAN OR EQUAL ZSYSTM
C                           TO EPS, WHERE K=1,...,N. (INPUT)        ZSYSTM
C                NSIG   - SECOND STOPPING CRITERION. A ROOT IS ACCEPTED ZSYSTM
C                           IF TWO SUCCESSIVE APPROXIMATIONS TO A GIVEN ZSYSTM
C                           ROOT AGREE IN THE FIRST NSIG DIGITS. (INPUT)ZSYSTM
C                             NOTE. IF EITHER, OR BOTH, OF THE STOPPING ZSYSTM
C                               CRITERIA ARE FULFILLED, THE ROOT IS ZSYSTM
C                               ACCEPTED. (SEE PROGRAMMING NOTES)   ZSYSTM
C                N      - THE NUMBER OF EQUATIONS (= NUMBER OF UNKNOWNS)ZSYSTM
C                         (INPUT)                                   ZSYSTM
C                X      - THE VECTOR X OF LENGTH N, AS INPUT, IS THE ZSYSTM
C                           INITIAL GUESS TO THE ROOT.  AS OUTPUT, IT ZSYSTM
C                           IS THE COMPUTED SOLUTION.               ZSYSTM
C                ITMAX  - ON INPUT = THE MAXIMUM ALLOWABLE NUMBER OF ZSYSTM
C                           ITERATIONS AND ON OUTPUT = THE NUMBER OF ZSYSTM
C                           ITERATIONS USED IN FINDING THE COMPUTED ZSYSTM
C                WA     - AN ARRAY WORK AREA OF SIZE ((N+2)*(N-1))/2 + ZSYSTM
C                           SOLUTION,                               ZSYSTM
C                           3*N SUPPLIED BY THE USER                ZSYSTM
C                PAR    - PAR CONTAINS A PARAMETER SET (POSSIBLY A  ZSYSTM
C                           FUNCTION NAME) WHICH IS PASSED TO THE USER ZSYSTM
C                           SUPPLIED FUNCTION F. PAR MAY BE USED TO ZSYSTM
C                           PASS ANY AUXILIARY PARAMETERS NECESSARY FOR ZSYSTM
C                           COMPUTATION OF THE FUNCTION F.          ZSYSTM
C                IER    - ERROR PARAMETER                           ZSYSTM
C                         TERMINAL ERROR = 128 + N                  ZSYSTM
C                           N = 1 INDICATES FAILURE TO CONVERGE WITHIN ZSYSTM
C                             ITMAX ITERATIONS.                     ZSYSTM
C                           N = 2 SINGULAR SYSTEM (JACOBIAN)        ZSYSTM
C   PRECISION           - SINGLE                                    ZSYSTM
C   REQD. IMSL ROUTINES - UERTST                                    ZSYSTM
C   LANGUAGE            - FORTRAN                                   ZSYSTM
C-------------------------------------------------------------------ZSYSTM
C   LATEST REVISION     - SEPTEMBER 28, 1973                        ZSYSTM
C                                                                   ZSYSTM
      DIMENSION          X(1),WA(1),PAR(1)                          ZSYSTM
      DATA               PREC,DELTA/5.E-12,5.E-9/                   ZSYSTM
      DATA               ZERO,PM1,PT1,P2/0.0,.1,.0001,.002/         ZSYSTM
      IER=0                                                         ZSYSTM
C                                 PREC IS A FUNCTION OF THE MACHINE ZSYSTM
C                                 SIGNIFICANCE, SIG, AND SHOULD BE  ZSYSTM
C                                 COMPUTED AS PREC=5.*10.**(-SIG+2). ZSYSTM
C                                 IN THIS INSTANCE WE WERE DEALING  ZSYSTM
C                                 WITH A 14 DIGIT MACHINE.          ZSYSTM
C                                 DELTA SHOULD BE TAKEN AS          ZSYSTM
C                                 5.*10.**(-(SIG+4)/2), FOR SIG EVEN, ZSYSTM
C                                 AND 16.*10.**(-(SIG+5)/2, FOR SIG ODDZSYSTM
      N2 = N+N                                                      ZSYSTM
      RELCON=10.0**(-NSIG)                                          ZSYSTM
      JTEST = 1                                                     ZSYSTM
      IERROR=0                                                      ZSYSTM
      IPART=((N+2)*(N-1))/2                                         ZSYSTM
      ITMP=IPART+N                                                  ZSYSTM
      LKSUB=ITMP+N                                                  ZSYSTM
      DO 130 M = 1, ITMAX                                           ZSYSTM
         IQUIT=0                                                    ZSYSTM
         FMAX=ZERO                                                  ZSYSTM
         M1 = M-1                                                   ZSYSTM
         K1 = LKSUB + 1                                             ZSYSTM
         KMIN = LKSUB + N                                           ZSYSTM
         XTEMP = ZERO                                               ZSYSTM
C                                 THE ARRAY WA(LKSUB+1),...,WA(LKSUB+N)ZSYSTM
C                                 PERMITS A PARTIAL PIVOTING EFFECT ZSYSTM
C                                 WITHOUT HAVING TO PHYSICALLY      ZSYSTM
C                                 INTERCHANGE ROWS OR COLUMNS.      ZSYSTM
         DO 5 J = K1,KMIN                                           ZSYSTM
            XTEMP = XTEMP+1.0                                       ZSYSTM
            WA(J) = XTEMP                                           ZSYSTM
    5    CONTINUE                                                   ZSYSTM
         K = 1                                                      ZSYSTM
   10    IF(K .LE. 1) GO TO 30                                      ZSYSTM
         KMIN = K-1                                                 ZSYSTM
C                                 THE FOLLOWING CODE BACK-SOLVES THE ZSYSTM
C                                 FIRST KMIN ROWS OF A TRIANGULARIZED ZSYSTM
C                                 LINEAR SYSTEM FOR IMPROVED X VALUES ZSYSTM
C                                 IN TERMS OF PREVIOUS ONES.        ZSYSTM
         KK = 1                                                     ZSYSTM
   15    DO 25 K1=1,KMIN                                            ZSYSTM
            ISUB=K-K1                                               ZSYSTM
            MM=((ISUB-1)*(N2 -ISUB))/2                              ZSYSTM
            LIM=N-ISUB                                              ZSYSTM
            KPOINT = WA(LKSUB+ISUB)+PM1                             ZSYSTM
C                                 THE ADDITION OF .1 (PM1) IN THE LAST ZSYSTM
C                                 STATEMENT (AND OTHERS LIKE IT     ZSYSTM
C                                 BELOW) IS ESSENTIAL, SINCE WA     ZSYSTM
C                                 CONTAINS INTEGERS AS WELL AS FLOATINGZSYSTM
C                                 POINT NUMBERS.  FOR EXAMPLE, SUPPOSE ZSYSTM
C                                 THE INTEGER 3 WAS STORED AS       ZSYSTM
C                                 2.99999999999998                  ZSYSTM
            ISUB1 = ISUB-1                                          ZSYSTM
            X(KPOINT)=ZERO                                          ZSYSTM
            DO 20 L1=1,LIM                                          ZSYSTM
               JS1=ISUB1+L1                                         ZSYSTM
               LKJSUB=LKSUB + JS1 +1                                ZSYSTM
               IJ=MM+JS1                                            ZSYSTM
               JPOINT= WA(LKJSUB) + PM1                             ZSYSTM
               X(KPOINT)=X(KPOINT) + WA(IJ)*X(JPOINT)               ZSYSTM
   20       CONTINUE                                                ZSYSTM
            X(KPOINT)=X(KPOINT) + WA(MM+N)                          ZSYSTM
   25    CONTINUE                                                   ZSYSTM
         GO TO (30,45,105), KK                                      ZSYSTM
C                                                                   ZSYSTM
C                                 SET UP PARTIAL DERIVATIVES OF     ZSYSTM
C                                 KTH FUNCTION.                     ZSYSTM
C                                                                   ZSYSTM
   30    E=F(X,K,PAR)                                               ZSYSTM
         FMAX =AMAX1(FMAX,ABS(E))                                   ZSYSTM
         IF( ABS(E) .GE. EPS) GO TO 35                              ZSYSTM
         IQUIT=IQUIT+1                                              ZSYSTM
```

```
      IF(IQUIT .EQ. N) GO TO 140                                        ZSYSTM
   35 I = K                                                             ZSYSTM
   40 IP=IPART+I                                                        ZSYSTM
      ITEMP = WA(LKSUB+I) + PM1                                         ZSYSTM
      HOLD = X(ITEMP)                                                   ZSYSTM
      ETA=.001*ABS(HOLD)                                                ZSYSTM
      IF (ABS(HOLD) .LT.PREC)  ETA=DELTA                                ZSYSTM
      H =AMIN1(FMAX,ETA)                                                ZSYSTM
      IF(H .LT. PREC) H=PREC                                            ZSYSTM
      X (ITEMP)=HOLD+H                                                  ZSYSTM
      IF (K .LE. 1) GO TO 45                                            ZSYSTM
      KK = 2                                                            ZSYSTM
      GO TO 15                                                          ZSYSTM
   45 FPLUS=F(X,K,PAR)                                                  ZSYSTM
      TOP=FPLUS-E                                                       ZSYSTM
   50 WA(IP)=TOP/H                                                      ZSYSTM
   55 X(ITEMP)=HOLD                                                     ZSYSTM
      I = I + 1                                                         ZSYSTM
      IF (I .LE. N) GO TO 40                                            ZSYSTM
      IF (K .LT. N) GO TO 60                                            ZSYSTM
      IP=IPART+N                                                        ZSYSTM
      IF (ABS( WA(IP)) .EQ. ZERO)  GO TO 80                             ZSYSTM
      X(ITEMP) = -E/ WA(IP) + X(ITEMP)                                  ZSYSTM
      GO TO 100                                                         ZSYSTM
C                                                                       ZSYSTM
C                                FIND PARTIAL DERIVATIVE OF LARGEST     ZSYSTM
C                                ABSOLUTE VALUE..                       ZSYSTM
C                                                                       ZSYSTM
   60 KL=LKSUB+K                                                        ZSYSTM
      LOOK= WA(KL) + PM1                                                ZSYSTM
      KMAX=LOOK                                                         ZSYSTM
      IP=IPART+K                                                        ZSYSTM
      DERMAX= ABS( WA(IP))                                              ZSYSTM
      KPLUS = K+1                                                       ZSYSTM
      DO 65 I = KPLUS,N                                                 ZSYSTM
         TEST= ABS( WA(IPART+I))                                        ZSYSTM
         IF(TEST .LE. DERMAX) GO TO 65                                  ZSYSTM
         DERMAX = TEST                                                  ZSYSTM
         KMAX=I                                                         ZSYSTM
   65 CONTINUE                                                          ZSYSTM
      IF(LOOK .EQ. KMAX) GO TO 75                                       ZSYSTM
      LKMAX=LKSUB+KMAX                                                  ZSYSTM
      WA(KL)=WA(LKMAX)                                                  ZSYSTM
      WA(LKMAX)=LOOK                                                    ZSYSTM
      IP=IPART+KMAX                                                     ZSYSTM
      XTEMP= WA(IP)                                                     ZSYSTM
      IPK=IPART+K                                                       ZSYSTM
      WA(IP)=WA(IPK)                                                    ZSYSTM
      WA(IPK)=XTEMP                                                     ZSYSTM
      IF(K .LT. 2) GO TO 75                                             ZSYSTM
      KMIN=K-1                                                          ZSYSTM
      I1 = 0                                                            ZSYSTM
      DO 70 I=1,KMIN                                                    ZSYSTM
         L=((I1 )*(N2 -I))/2-1                                          ZSYSTM
         J=L+KMAX                                                       ZSYSTM
         XTEMP= WA(J)                                                   ZSYSTM
         JJ=L+K                                                         ZSYSTM
         WA(J)=WA(JJ)                                                   ZSYSTM
         WA(JJ)=XTEMP                                                   ZSYSTM
         I1 = I                                                         ZSYSTM
   70 CONTINUE                                                          ZSYSTM
   75 IF ( ABS(WA(IPART+K)).NE. ZERO) GO TO 90                          ZSYSTM
   80 IF(IERROR .EQ. 1) GO TO 135                                       ZSYSTM
      DO 85  I=1,N                                                      ZSYSTM
C                                IF THE MODIFIED JACOBIAN IS SINGULAR   ZSYSTM
C                                AT X, CHANGE THE COMPONENTS OF X AND   ZSYSTM
C                                PROCEED WITH THE ITERATIONS.  IF IT    ZSYSTM
C                                HAPPENS A SECOND TIME, TERMINATE.      ZSYSTM
         X(I) = 0.9*X(I) + .12345                                       ZSYSTM
   85 CONTINUE                                                          ZSYSTM
      IERROR=1                                                          ZSYSTM
      GO TO 105                                                         ZSYSTM
C                                                                       ZSYSTM
C                                SET UP COEFFICIENTS FOR KTH ROW        ZSYSTM
C                                OF TRIANGULAR LINEAR SYSTEM USED       ZSYSTM
C                                TO BACK-SOLVE FOR THE FIRST K X(I)     ZSYSTM
C                                VALUES...                              ZSYSTM
C                                                                       ZSYSTM
   90 L=((K-1)*(N2 -K))/2                                               ZSYSTM
      KN=L+N                                                            ZSYSTM
      I1 = L-1                                                          ZSYSTM
      WA(KN)=ZERO                                                       ZSYSTM
      IPK=IPART+K                                                       ZSYSTM
      DO 95  J = KPLUS,N                                                ZSYSTM
         JSUB= WA(LKSUB+J) + PM1                                        ZSYSTM
         JJ=I1+J                                                        ZSYSTM
         IPJ=IPART+J                                                    ZSYSTM
         WA(JJ)=-WA(IPJ)/WA(IPK)                                        ZSYSTM
         WA(KN)=WA(KN)+WA(IPJ)*X(JSUB)                                  ZSYSTM
   95 CONTINUE                                                          ZSYSTM
      LK= WA(LKSUB+K) + PM1                                             ZSYSTM
      WA(KN)=(WA(KN)-E)/WA(IPK) + X(LK)                                 ZSYSTM
      K = K+1                                                           ZSYSTM
      IF (K .LE. N) GO TO 10                                            ZSYSTM
C                                BACK SUBSTITUTE TO OBTAIN NEXT         ZSYSTM
C                                APPROXIMATION TO X                     ZSYSTM
  100 IF( N .EQ. 1) GO TO 105                                           ZSYSTM
      KMIN=N-1                                                          ZSYSTM
      KK = 3                                                            ZSYSTM
      GO TO 15                                                          ZSYSTM
  105 IF (M .LE. 1) GO TO 120                                           ZSYSTM
C                                                                       ZSYSTM
C                                TEST FOR CONVERGENCE..                 ZSYSTM
C                                                                       ZSYSTM
      DO 110 I = 1,N                                                    ZSYSTM
         IF (ABS(WA(ITMP+I)-X(I)) .GT. ABS(X(I))*RELCON) GO TO 115      ZSYSTM
  110 CONTINUE                                                          ZSYSTM
      JTEST = JTEST+1                                                   ZSYSTM
      IF (JTEST-3)120,140,140                                           ZSYSTM
  115 JTEST = 1                                                         ZSYSTM
  120 DO 125 I = 1,N                                                    ZSYSTM
         WA(ITMP+I)=X(I)                                                ZSYSTM
  125 CONTINUE                                                          ZSYSTM
  130 CONTINUE                                                          ZSYSTM
      M=ITMAX                                                           ZSYSTM
      IER = 129                                                         ZSYSTM
      GO TO 240                                                         ZSYSTM
  135 IER = 130                                                         ZSYSTM
  140 FMAX=ZERO                                                         ZSYSTM
      TEST=1.0E+15                                                      ZSYSTM
      IF (N .GT. 1) GO TO 145                                           ZSYSTM
      WA(IPART+2) = F(X,1,PAR)                                          ZSYSTM
      FMAX =AMAX1(FMAX,ABS(WA(IPART+2)))                                ZSYSTM
      GO TO 155                                                         ZSYSTM
```

```
  145 DO 150 I = 1,N                                                    ZSYSTM
          IP=IPART+I                                                    ZSYSTM
          WA(IP)=F(X,I,PAR)                                             ZSYSTM
          FMAX=AMAX1(FMAX, ABS(WA(IP)))                                 ZSYSTM
  150 CONTINUE                                                          ZSYSTM
C                                 CHECK TO SEE IF SMALL COMPONENTS ARE ZSYSTM
C                                 ACTUALLY ZERO                         ZSYSTM
  155 K=1                                                               ZSYSTM
      DO 160 I=1,N                                                      ZSYSTM
          WA(I)=X(I)                                                    ZSYSTM
          IF (ABS(X(I)) .GT. P2) GO TO 160                              ZSYSTM
          K=2                                                           ZSYSTM
          WA(I)=ZERO                                                    ZSYSTM
  160 CONTINUE                                                          ZSYSTM
      IF(K .EQ. 1) GO TO 195                                            ZSYSTM
      KK = 1                                                            ZSYSTM
      GO TO 205                                                         ZSYSTM
  165 IF(FMAX .LT. TEST) GO TO 190                                      ZSYSTM
C                                 NOTE THAT SMALL COMPONENTS ARE SET    ZSYSTM
C                                 TO ZERO ONLY IF THE NORM OF THE       ZSYSTM
C                                 FUNCTION VECTOR IS REDUCED AS A       ZSYSTM
C                                 RESULT OF THIS PROCESS.               ZSYSTM
      DO 170 I=1,N                                                      ZSYSTM
          X(I)= WA(I)                                                   ZSYSTM
  170 CONTINUE                                                          ZSYSTM
      IF (N .GT. 1) GO TO 175                                           ZSYSTM
      WA(IPART+2) = WA(ITMP+2)                                          ZSYSTM
      GO TO 185                                                         ZSYSTM
  175 DO 180 I = 1,N                                                    ZSYSTM
          WA(IPART+I) = WA(ITMP+I)                                      ZSYSTM
  180 CONTINUE                                                          ZSYSTM
  185 FMAX=TEST                                                         ZSYSTM
C                                 CHECK FOR INTEGER COMPONENTS          ZSYSTM
  190 K=1                                                               ZSYSTM
  195 ITEST=0                                                           ZSYSTM
      DO 200 I=1,N                                                      ZSYSTM
          WA(I)=X(I)                                                    ZSYSTM
          IF (ABS(X(I)) .LE. P2) GO TO 200                              ZSYSTM
          L=X(I)+PT1                                                    ZSYSTM
          J=X(I)-PT1                                                    ZSYSTM
          IF(L .EQ. J) GO TO 200                                        ZSYSTM
          WA(I) =ISIGN(1,J)*MAX0(IABS(L),IABS(J))                       ZSYSTM
          K=2                                                           ZSYSTM
  200 CONTINUE                                                          ZSYSTM
      IF(K .EQ. 1) GO TO 235                                            ZSYSTM
      KK = 2                                                            ZSYSTM
  205 TEST=ZERO                                                         ZSYSTM
      IF (N .GT. 1) GO TO 210                                           ZSYSTM
      WA(ITMP+2) = F(WA,1,PAR)                                          ZSYSTM
      TEST =AMAX1(TEST,ABS(WA(ITMP+2)))                                 ZSYSTM
      GO TO 220                                                         ZSYSTM
  210 DO 215 I=1,N                                                      ZSYSTM
          IT=ITMP+I                                                     ZSYSTM
          WA(IT)=F(WA,I,PAR)                                            ZSYSTM
          TEST=AMAX1(TEST, ABS(WA(IT)))                                 ZSYSTM
  215 CONTINUE                                                          ZSYSTM
  220 GO TO (165,225), KK                                               ZSYSTM
  225 IF(FMAX .LT. TEST) GO TO 235                                      ZSYSTM
C                                 NOTE THAT NEAR-INTEGER COMPONENTS     ZSYSTM
C                                 ARE SET TO BE INTEGERS ONLY IF THE    ZSYSTM
C                                 NORM OF THE FUNCTION VECTOR IS        ZSYSTM
C                                 REDUCED AS A RESULT OF THIS PROCESS.  ZSYSTM
      DO 230 I=1,N                                                      ZSYSTM
          X(I)= WA(I)                                                   ZSYSTM
  230 CONTINUE                                                          ZSYSTM
      ITEST=1                                                           ZSYSTM
C                                 TEST FOR CONVERGENCE                  ZSYSTM
  235 IF(FMAX .LT. EPS .OR. TEST .LT. EPS)  IER = 0                     ZSYSTM
  240 ITMAX=M1 + 1                                                      ZSYSTM
 9000 CONTINUE                                                          ZSYSTM
      IF (IER .NE. 0) CALL UERTST(IER,6HZSYSTM)                         ZSYSTM
 9005 RETURN                                                            ZSYSTM
      END                                                               ZSYSTM
```

APPENDIX B

DESCRIPTION OF COMPUTER PROGRAM MVPROJ

INTRODUCTION

1. Origin of Program. This program was developed by Shin Chang of The Center for Research in Water Resources, The University of Texas at Austin under the supervision of Leo R. Beard, Technical Director. Subroutines obtained from other sources are BMDOIM of Biomedical Computer Programs (BMD) and all those subroutines used in Program SVPROJ.

2. Capability of the MVPROJ Program. MVPROJ is capable of performing the following jobs:

   a. Construction of the principal components of a set of original variables.

   b. For each of these principal components:

      1. Idenficiation of p, d, and q of the best-fit ARIMA(p,d,q) model for each principal component being analyzed.
      2. Estimation of parameters for the models identified in 1.
      3. Generation of 51 independent sets of projections for each principal component.

   c. Computation of 51 projections for each of the original variables as functions of projections of all principal components.

   d. Plotting up to 7 selected projections of each of the original variables in one chart for each variable.

3. Hardware and Software Requirements. This program has been developed and tested on the CDC 64/6600 computer system. Thirteen disk files are required. The Fortran IV complier and the IMSL library are used.

DESCRIPTION OF THE PROGRAM

4. Program Organization. The MVPROJ program consists of a main program and 25 subroutines. All input data and input control parameters are read

in the main program from punched cards. The program will make a normal exit (unless errors occur) only in the main program.

5. Method of Computation. The computation procedures and equations used in this program are described in Chapter 5 of the main report under the title "Multivariate Projection Model."

6. Description of Subroutines and Variables. The function of each subroutine and the definitions of variables in each subroutine are given in the program listings. However, subroutines used in SVPROJ are not repeated here.

7. Dimensional Limitations of the Program. For practical purpose, the following limitations are set in the program:

Variable	Size	Definition
NV	12	number of variables
LSPLIT	150	length of calibration variables
LPROJ	100	length of projection
NSET	51	sets of independent projections

These dimensions can be easily extended by increasing the dimension specification in related arrays, however, ultimate limitations are determined by the available control memory of the computer system used.

8. Control Parameters for Optional Jobs. The following parameters are used to control the execution of certain optional jobs in the program.

ISPLIT = controls the execution of the split-record test

LOGX = Controls the execution of the logarithmic (base 10) transformation of the original variables

NPRINT = Controls the execution of the printed output of the selected projections (7 sets)

NPUNCH = Controls the execution of the punched-card output of the selected projections (7 sets)

NPLOT = Controls the execution of the plotted output of the selected projections (7 sets or 5 sets)

9. Preparation of Input Variables.

Variable		Description
IFMT	character	variable format for input data x
ISPLIT	1	split-record test is desired,
	0	not desired
IYRB	integer	beginning year of record of the calibration series
IYRE	integer	ending year of record of the calibration series
IYRP	integer	ending year of projection
LOGX	1	logarithmic (base 10) transformation of the variables is desired
	2	not desired
NPLOT	0	no plot is desired as output
	1	plots of data and 7 projections depicted in one chart are desired
	2	plots of data and 7 projections, and data and 5 projections, depicted in 2 charts are desired
	3	plots of data and 5 projections depicted in one chart are desired
NV	integer	number of variables
WT (12)	real	relative weights to be applied to the last member of projection of each variable for ranking the 51 sets of projections.
YLABEL (8)	character	name of variable to be appeared in the chart
X (150, 12)	real	input data of calibration series, the first dimension refers to time and the second dimension refers to variables, (X(j,i) = jth record of ith variable)

10. Preparation of Input Cards. The following types of input cards are required.

Card	Format	Variable(s)
A	16I5	NV, IYRB, IYRE, IYRP, LOGX, ISPLIT, NPRINT, NPUNCH, NPLOT

B	12F5.0	WT
C	8A10	IFMT
D	IFMT	X
E	———	job card for BMD01M
F	———	label card for BMD01M
G	IFMT	data format card for BMD01M
H	8A10	YLABEL

Preparation of E and F cards of BMD01M:

E card (Job card):

col. 1-6	Problem (mandatory)
col. 7-12	Alphanumeric job code
col. 13,14	Number of original variables ($2 \leq NV \leq 12$)
col. 15-17	Length of calibration variables ($3 \leq LSPLIT \leq 150$)
col. 18-20	Blank
col. 21-23	Yes
col. 24-28	Blank
col. 29-30	Number of variables to be labeled
col. 31-68	Blank
col. 68-70	07 (data input is from logical BCD Tape 7)
col. 71,72	K Number of Variable Format Cards ($1 \leq K \leq 10$)

F Card (Labels card):

col. 1-6	LABELS (mandatory
col. 7-10	The number of the variable to be named. The number must be right-justified
col. 11-16	The corresponding alphanumeric name
col. 17-20	The number of another variable
col. 21-26	The corresponding alphanumeric name.

⋮

col. 67-70 The number of another variable

col. 71-76 The corresponding alphanumeric name (up to 7 per card)

Example = The following Labels Card (F Card) is prepared for the case of Labelling X(j,2) as name 01 and X(j,5) as name 02.

Labels0002NAME010005NAME02.

Note 1. The number of a variable is i if the variable to be labeled is the ith variable of the input vector X(j,i).

2. More than one labels card can be used.

3. The number of labels appearing on all the labels cards must equal to the number of labels specified on the job card.

11. Arrangement of Input-deck. The following sequence of the input deck is used:

$$A\ B\ C\ \underbrace{D\ D\ D\ \ldots D}_{X}\ E\ F\ G\ \underbrace{H\ H\ H\ \ldots H}_{YLABEL}$$

12. Overview of Output. The following printout will be shown:

a. Normalized and transformed (LOGX = 1) input data

b. Correlation coefficient matrix

c. Eigenvalues

d. Eigenvectors

e. For each principal component, a summary table showing the best-fit ARIMA (p,d,q) model, estimations of parameters, level of significance.

Optional printed and/or punched output of the 7 sets of projections and chart (s) of projections will be shown upon the specification of related output control parameters.

13. Program Listing. Source program listing for MVPROJ and its subroutines are shown in the following pages. Note that subroutines shared by both programs SVPROJ and MVPROJ will be shown in appendix A only.

```
      PROGRAM MVPROJ (ALTTAPE=1001,INPUT=1001,OUTPUT=1001,PUNCH=1001,
     1 TAPE1=ALTTAPE,TAPE2=ALTTAPE,TAPE3=PUNCH,TAPE4=ALTTAPE,
     2 TAPE5=INPUT,TAPE6=OUTPUT,TAPE7=ALTTAPE,TAPE8=ALTTAPE,
     3 TAPE9=ALTTAPE,TAPE10=ALTTAPE,TAPE11=ALTTAPE,TAPE12=ALTTAPE,
     4 TAPE13=1001,TAPE14=1001,TAPE15=1001,TAPE16=1001,TAPE17=1001,
     5 TAPE18=1001,TAPE19=1001,TAPE20=1001,TAPE21=1001,TAPE22=1001,
     6  TAPE23=1001, TAPE24=1001                                    )
C-----------------------------------------------------------------------
C   FUNCTION--1, FIND PRINCIPAL COMPONENTS
C             2, FORECASTING AND SIMULATING OF PRINCIPAL COMPONENTS
C             3, FORECASTING AND SIMULATING OF ORIGINAL VARIABLES USING
C                PROJECTED VALUES OF PRINCIPAL COMPONENTS
C        NV     -NUMBER OF VARIABLES TO BE ANALYZED
C        LOGX   =1, LOG10 TRANSFORMATION OF RECORDS IS DESIRED
C               0, IF NOT
C        ISPLIT =1, SPLIT-RECORD TEST IS DESIRED
C               0, IF NOT
C        NPRINT =1, PRINT OUTPUT OF PROJECTIONS IS DESIRED
C               0, IF NOT
C        NPUNCH =1, PUNCHED CARDS OUTPUT OF PROJECTION IS DESIRED
C               0, IF NOT
C        NPLOT  =0, NO PLOT OF PROJECTION IS DESIRED
C               1, PLOTS OF 7 PROJECTIONS IN ONE CHART IS DESIRED
C               2, PLOTS OF BOTH CURVES IN NPLOT=1 AND 3 ARE DESIRED
C               3, PLOTS OF 5 PROJECTIONS IN ONE CHART IS DESIRED
C        IYRB   -BEGINNING YEAR OF THE DATA
C        IYRE   -ENDING YEAR OF DATA
C        IYRP   -ENDING YEAR OF PROJECTION
C        LSPLIT -FIRST LSPLIT DATA WILL BE USED TO CALIBRATE THE MODEL
C        LPROJ  -LENGTH OF PROJECTION
C        LSP    -LSPLIT+LPROJ
C        XX(I,J) -ORIGINAL JTH VARIABLE AT TIME I
C        X(I,J) -AS STANDARDIZE JTH VARIABLE AT TIME I
C        C(I,J) -JTH PRINCIPAL COMPONENT AT TIME I
C        XMEAN(J)-MEAN VALUE OF ORIGNAL VARIABLE(TRANSFORMED)
C        SDV(J) -STANDARD DEVIATION OF ORIGINAL VARIABLE (TRANSFORMED)
C        YLABEL(J) -NAME OF THE VARIABLE
C        Z(I,J) -COMPONENT-PATTERN COEFFICIENT FOR VARIABLE
C        WT(J)  -A VECTOR CONTAINING WEIGHTS USED IN THE COMPUTATION
C                OF THE WEIGHTED SUM OF LAST MEMBERS OF PROJECTIONS
C                OF THE ORIGINAL VARIABLES
C        U(I,J) -J=1,2,..7 CONTAINS 7 PROJECTIONS , J=8 CONTAINS
C                THE MOST LIKELY PROJECTION LEVEL AND, J=9 CONTAINS
C                THE DATA OF ORIGINAL VARIABLE (IF I.LE.N) AND IT
C                BECOME THE MOST LIKELY PROJECTION  IF I.GT.N
C        ALFA   =POSITIVE, SIGNIFICANCE LEVEL IN THE SUMMARY TABLE
C               =NEGATIVE TO GIVE ERROR INDICATION
C               =-1,0,NO SOLUTION FROM FTARPS
C               =-2,0, ARPS OUT OF ADMISSIBLE RANGE FROM CHECK
C               =-4,0, MAPS OUT OF ADMISSIBLE RANGE FROM CHECK
C               =-3,0, NO SOLUTION FROM FTMAPS
C               =-5,0, ALPHA(1) FAIL TO PASS THE MINIMUM LEVEL
C               =-6,0, RANGE OF PROJECTION EXCEEDS RLIMIT
C-----------------------------------------------------------------------
      DIMENSION X(150,12),C(150,12),Z(12,12),XMEAN(12)
      DIMENSION NAMES(12),SDV(12),COV(12,12),XX(150,12)
      DIMENSION Y(150),D(100,6,12),U(250,9 ),WT(12),WTSUM(55),ZL(55,12)
      DIMENSION YLABEL(8)
      DIMENSION NODE(5),INDEX(55),NP(12),NQ(12),ND(12),IFMT(8)
      COMMON/ARA/SIM(100,51),FCST(3,100),ALFA(12),RANGEP(12),RLIMIT(12)
      COMMON X,C,Z,COV,XMEAN,NV,N,NAMES,SDV,XX
      COMMON  /IQ/ISPLIT,NPRINT,NPLOT,NSET,IYRB,IYRE,IYRP
      COMMON/MODEL/NP,NQ,ND,LSPLIT,LPROJ,LSP
      DOUBLE PRECISION SEED
      COMMON /DRAND/ SEED
      DATA NODE/2,8,26,44,50/
C          FOR A BETTER UNDERSTANDIND OF THE ALGORITHM, THE FOLLOWING
C           INTEGERS ARE USED,
C         JT- THE J-TH TIME
C         JV- THE J-TH VARIABLE
C         JPC- THE J-TH PRINCIPLE COMPONENT
C         JS- THE J-TH SIMULATION OF A VARIABLE OR A P-C
C
C   D(JT,JS,JPC)=J-TH PRINCIPLE COMPONENT AT TIME I, AND
C              JS=1TO5, 5-SIMULATIONS OF THIS P-C
C              JS=6, FORECASTS OF THIS P-C
C              JS=7, ORIGINAL P-C FROM BMD
C   U(JT,JS)    =J-TH SIMULATION OF A VARIABLE AT TIME I
      NSET=51
      READ 1420, NV,IYRB,IYRE,IYRP,LOGX,ISPLIT,NPRINT,NPUNCH,NPLOT
      PRINT 1420, NV,IYRB,IYRE,IYRP,LOGX,ISPLIT,NPRINT,NPUNCH,NPLOT
      N=IYRE-IYRB+1
      READ 1450, (WT(J),J=1,12)
      READ 1470, IFMT
      DO 1010 J=1,N
      READ IFMT, (X(J,I),I=1,NV)
 1010 CONTINUE
        IF (LOGX.EQ.0) GO TO 1030
      DO 1020 J=1,N
      DO 1020 I=1,NV
      X(J,I)=ALOG10(X(J,I))
 1020 CONTINUE
 1030 CONTINUE
C          STORE X(J,I) IN TAPE7 FOR BMD USESAS INPUT
      DO 1040 J=1,N
      WRITE (7,1440) (X(J,I),I=1,NV)
 1040 CONTINUE
      END FILE 7
      REWIND 7
      ITAPE=4
      CALL BMD01M
      TEMP=SQRT(FLOAT(N-1))
      DO 1050 I=1,NV
      SDV(I)=SDV(I)/TEMP
 1050 CONTINUE
C          IF LOG10-TRANSFORMATION IS USED IN THE PRINCIPAL COMPONENT
C           ANALYSIS, THEN LOGX IS NOT EQUAL 0 (BY READ 2)
C
C           FORECASTS AND SIMULATIONS OF THE P-C
C
      MSET=5
      MP1=MSET+1
      MP2=MSET+2
      MP3=MSET+3
      MP4=MSET+4
      LSPLIT=N
      LPROJ=IYRP-IYRE
        IF (ISPLIT.EQ.0) GO TO 1060
      LSPLIT=N/2+1
      LPROJ=N-LSPLIT
 1060 CONTINUE
      LSP=LSPLIT+LPROJ
      SEED=0.5234678540D0
```

```
      IPDQ(7)=NSET
      JTT=1
        IF (ISPY-0) 1100,1100,1110
 1100 CONTINUE
      IPDQ(3)=IP
      IPDQ(4)=IQ
      IPDQ(5)=ID
        GO TO 1130
 1110 CONTINUE
C               PARAMETERS IP,ID AND IQ OF THE ARIMA (IP,ID,IQ) MODEL ARE
C                 IDENTIFIED BY IDENTN ROUTINE USING WHOLE RECORD OF SERIE
      IND(1)=N
      IND(2)=5*N
      IND(3)=5
      CALL IDENTN (Z,IND,RL,0.30,ARF,IPARM,1,1)
 1120 CONTINUE
      IPDQ(3)=IPARM(JTT,1)
      IPDQ(4)=IPARM(JTT,3)
      IF (IPDQ(1).GT.0) IPDQ(5)=IPARM(JTT,2)
 1130 CONTINUE
      SEED=0.523467854D0
      ALPHA(1)=0.01
      ALPHA(2)=0.05
      IS=100
      DO 1140 IN=1,5
      DARPS(IN)=0.0
      PMAS(IN)=0.0
 1140 CONTINUE
      PMAC=0.00
      CALL GTSIMP (Z,IPDQ,SEED,ALPHA,DARPS,PMAS,PMAC,WNV,FCST,SIM,IS,WK,
     1       IER)
      ALFA(ITT)=ALPHA(1)
      NP(ITT)=IPDQ(3)
      NQ(ITT)=IPDQ(4)
      ND(ITT)=IPDQ(5)
C                RANK THE LAST MEMBERS OF SIMULATIONS SIM(I,J) INTO ZLAST
C                FROM MAX TO MIN
      DO 1150 J=1,NSET
      WK(J)=SIM(LPROJ,J)
 1150 CONTINUE
C             SEARCH FOR THE MINIMUN AND THEN SET AMIN FOR RANKING OPERAT
      A=WK(1)
      DO 1160 J=1,NSET
      IF (WK(J).LT.A) A=WK(J)
 1160 CONTINUE
      AMIN=A-10.0
      DO 1180 L=1,NSET
      A=AMIN
      DO 1170 J=1,NSET
        IF (WK(J).LE.A) GO TO 1170
      A=WK(J)
      IJ=J
 1170 CONTINUE
      WK(IJ)=AMIN
      INDEX(L)=IJ
 1180 CONTINUE
CC              COMPUTE THE RANGE OF PROJECT AS RANGEP
      RANGEP(ITT)=0.0
      ZMAX=SIM(1,J)
      ZMIN=SIM(1,J)
      DO 1200 I=1,5
      L=NODE(I)
      J=INDEX(L)
      DO 1190 II=1,LPROJ
      IF (SIM(II,J).GT.ZMAX) ZMAX=SIM(II,J)
      IF (SIM(II,J).LT.ZMIN) ZMIN=SIM(II,J)
 1190 CONTINUE
      RP=ZMAX-ZMIN
      IF (RP.GT.RANGEP(ITT)) RANGEP(ITT)=RP
CC                CHECK WITH LIMIT ON RANGEP
        IF (RANGEP(ITT).LE.RLIMIT(ITT)) GO TO 1200
      ALFA(ITT)=-6.0
 1200 CONTINUE
C
C             ITERATED CYCLE IN CASE THE FIRST CHOICE FROM IDENTN CAN
C            NOT GIVE SATISFIED SOLUTION
C
        IF (IER.GT.0.OR.ALFA(ITT).LE.0.0) GO TO 1210
        GO TO 1220
C             SOLUTION IS UNSATISFIED, ITERATED IF ISPY NOT EQUAL TO
C            ZERO AND ITERATION NUMBER JTT IS NOT EXCEED 5
 1210 CONTINUE
        IF (ISPY.EQ.0) GO TO 1380
        IF (JTT.GE.5) GO TO 1380
      JTT=JTT+1
        GO TO 1120
 1220 CONTINUE
C             CONSTRUCTION OF G-ARRAY FOR PLOTTING
      DO 1230 I=1,MP3
      DO 1230 J=1,LSPLIT
      G(J,I)=G(J,MP4)
 1230 CONTINUE
      DO 1250 J=1,LPROJ
      JJ=J+LSPLIT
      DO 1240 I=1,MSET
      L=NODE(I)
      II=INDEX(L)
      G(JJ,I+1)=SIM(J,II)
 1240 CONTINUE
      TEMP1=G(JJ,2)
      TEMP2=G(JJ,MP1)
      G(JJ,1)=2.0*TEMP1-TEMP2
      G(JJ,MP2)=2.0*TEMP2-TEMP1
      G(JJ,MP3)=FCST(2,J)
 1250 CONTINUE
C             REVERSE TRANSFORMATION OF VARIABLES IF APPLICABLE
      J1=LSPLIT+1
        IF (LOGX.EQ.0) GO TO 1280
      TEMP=ZUP-ZLO
      DO 1260 J=J1,LSP
      DO 1260 I=1,MP3
      G(J,I)=EXP(G(J,I))+ZLO
 1260 CONTINUE
        IF (IBOND.LE.0) GO TO 1280
      DO 1270 J=J1,LSP
      DO 1270 I=1,MP3
      G(J,I)=TEMP/EXP(G(J,I)-ZLO)+ZLO
 1270 CONTINUE
 1280 CONTINUE
C             IF ISPLIT=0, SET G(J,MP4)=G(J,MP3) FOR J GREATER THAN N
        IF (ISPLIT.NE.0) GO TO 1300
      DO 1290 J=J1,LSP
      G(J,MP4)=G(J,MP3)
 1290 CONTINUE
```

```
      WRITE (3,1460) (U(KT,K8),KT=1,LSP)
 1360 CONTINUE
 1370 CONTINUE
      READ 1470, (YLABEL(J),J=1,8)
         IF (NPLOT.EQ.0) GO TO 1400
         IF (NPLOT.GT.2) GO TO 1380
      CALL PLOT (U,LSP,9,0,IYRB,YLABEL)
         IF (NPLOT.EQ.1) GO TO 1400
 1380 CONTINUE
      DO 1390 JT=1,LSP
      U(JT,MP2)=U(JT,MP3)
      U(JT,MP3)=U(JT,MP4)
 1390 CONTINUE
      CALL PLOT (U(1,2),LSP,7,0,IYRB,YLABEL)
 1400 CONTINUE
 1410 CONTINUE
 1420 FORMAT  (16I5)
 1430 FORMAT  (5F16.6)
 1440 FORMAT  (10F8.2)
 1450 FORMAT  (12F5.2)
 1460 FORMAT  (//,*7 PROJECTIONS AND THE MOST LIKELY FORECASTING*/)
 1470 FORMAT  (8A10)
 1480 FORMAT  (8F10.2)
      END
```

```
      SUBROUTINE CHECK (N,X,IER)
C-----------------------------------------------------------------------
C   FUNCTION      -CHECK VALUES OF PARAMETERS (EITHER AUTOREFRESSIVE OR
C                   MOVING AVERAGE TYPE) TO SEE IF THEY ARE IN THE
C                   ADMISSIBLE REGION
C        N        -NUMBER OF PARAMETERS (EITHER P OR Q OF THE ARIMA
C                   (P,D,Q) MODEL)
C        X        -INPUT VECTOR OF LENGTH N
C        IER      -0, PARAMETERS ARE IN THE ADMISSIBLE REGION
C                   139, IF NOT
C-----------------------------------------------------------------------
      DIMENSION X(3)
         IF (N.EQ.0) GO TO 1030
      ONE=1.
      IER=0
      TP=X(N)
         IF (ABS(TP).GT.1.) GO TO 1020
      TMP=0.
      TP=0.
      DO 1010 I=1,N
      TP=TP+X(I)
      ONE=-1.*ONE
      TMP=TMP+X(I)*ONE
 1010 CONTINUE
         IF (TP.GT.1..OR.TMP.GT.1.) GO TO 1020
         IF (N.LT.3) GO TO 1030
      TP=X(3)*(X(3)-X(1))-X(2)
         IF (TP.GT.1.) GO TO 1020
         GO TO 1030
 1020 CONTINUE
      IER=139
 1030 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE FINDD (U,N,M,NDD,EPSON)
C-----------------------------------------------------------------------
C   FUNCTION        -SEARCH THE ORDER OF DIFFERENCING D IN AN ARIMA(P,
C                      D,Q) MODEL SO THAT THE DIFFERENCED SERIES BECOMES
C                      STATIONARY
C    U              -INPUT TIME SERIES
C    N              -LENGTH OF VECTOR U
C    M              -NO. OF AUTOCORRELATION COEFFICIENTS USED IN THE
C                      CHI-SQUARE STATISTIC
C   NDD             -OUTPUT OPTIMAL ORDER OF DIFFERENCING D
C   EPSON           -CRITERION USED IN SELECTING NDD
C
C
C-----------------------------------------------------------------------
      DIMENSION  U(1),CHI(3),AC(36),ACV(36),PACV(36),WKAREA(120),X(120)
      DO 1010 J=1,N
      X(J)=U(J)
 1010 CONTINUE
      NN=N
      DO 1040 I=1,3
      CALL FTAUTO (X,NN,M,0,5,AMEAN,VAR,ACV,AC,PACV,WKAREA)
      TEMP=0.0
      DO 1020 J=1,M
      TEMP=TEMP+AC(J)*AC(J)
 1020 CONTINUE
      CHI(I)=TEMP*NN
C             DEFFERENCING THE SERIES
      NN=N-I
      DO 1030 K=1,NN
      X(K)=X(K+1)-X(K)
 1030 CONTINUE
 1040 CONTINUE
C             CALCULATE RATIOS R1 AND R2
      R1=CHI(2)/CHI(1)
      R2=CHI(3)/CHI(2)
C             SELECT NDD
         IF (R1.GT.EPSON) GO TO 1050
      NDD=1
      RETURN
 1050 CONTINUE
         IF (R2-EPSON) 1060,1060,1070
 1060 CONTINUE
      NDD=2
      RETURN
```

```
 1070 CONTINUE                                                          FIN 450
      NDD=0                                                             FIN 460
      RETURN                                                            FIN 470
      END                                                               FIN 480-
```

```
      SUBROUTINE FOCASI (Z,N,ITT)                                       FOC  10
C-----------------------------------------------------------------------FOC  20
C   FUNCTION     -1. USING WHOLE DATA OF Z TO IDENTIFY THE OPTIMAL VALUEFOC  30
C                  P,Q, AND D OF THE ARIMA MODEL                        FOC  40
C                 2. USING THE MODEL IDENTIFIED IN 1, AND ONLY LSPLIT   FOC  50
C                   OF THE DATA OF Z TO FORECAST LPROJ FUTURE VALUES OF ZFOC  60
C        Z       -INPUT VECTOR (TIME SERIES) OF LENGTH N                FOC  70
C        N       -LENGTH OF VECTOR Z                                    FOC  80
C        ITT     -INPUT INTEGER TO IDENTIFY THE CURRENT INPUT SERIES    FOC  90
C                  AS PRINCIPAL NO. ITT                                 FOC 100
C-----------------------------------------------------------------------FOC 110
      DIMENSION Z(1),ALPHA(2),DARPS(5),PMAS(5),WK(400)                  FOC 120
      DIMENSION IPDQ(7),IND(3),IPARM(5,3)                               FOC 130
      COMMON/ARA/SIM(100,51),FCST(3,100),ALFA(12),RANGEP(12),RLIMIT(12) FOC 140
      DIMENSION NP(12),NQ(12),ND(12)                                    FOC 150
      COMMON  /IQ/ISPLIT,NPRINT,NPLOT,NSET,IYRB,IYRE,IYRP               FOC 160
      COMMON/MODEL/NP,NQ,ND,LSPLIT,LPROJ,LSP                            FOC 170
      DOUBLE PRECISION SEED                                             FOC 180
      COMMON /DRAND/ SEED                                               FOC 190
C             SEARCH THE RANGE OF THE CALIBRATION SERIES                FOC 200
      ZMIN=Z(1)                                                         FOC 210
      ZMAX=Z(1)                                                         FOC 220
      DO 1010 JJ=2,LSPLIT                                               FOC 230
      IF (Z(JJ).LT.ZMIN) ZMIN=Z(JJ)                                     FOC 240
      IF (Z(JJ).GT.ZMAX) ZMAX=Z(JJ)                                     FOC 250
 1010 CONTINUE                                                          FOC 260
      RC=ZMAX-ZMIN                                                      FOC 270
      RLIMIT(ITT)=50.0*RC                                               FOC 280
C            KEEP GOING THE SEARCH TO GET THE RANGE OF THE WHOLE SERIESFOC 290
C             AS RC AND SET THE RL FOR IDENTN ROUTINE                   FOC 300
      DO 1020 JJ=LSPLIT,N                                               FOC 310
      IF (Z(JJ).LT.ZMIN) ZMIN=Z(JJ)                                     FOC 320
      IF (Z(JJ).GT.ZMAX) ZMAX=Z(JJ)                                     FOC 330
 1020 CONTINUE                                                          FOC 340
      RC=ZMAX-ZMIN                                                      FOC 350
      RL=50.0*RC                                                        FOC 360
C            INITIALIZE PARAMETERS                                      FOC 370
      IPDQ(1)=1                                                         FOC 380
      IPDQ(2)=LSPLIT                                                    FOC 390
      IPDQ(6)=LPROJ                                                     FOC 400
      IPDQ(7)=NSET                                                      FOC 410
      JTT=1                                                             FOC 420
C            PARAMETERS IP,ID AND IQ OF THE ARIMA (IP,ID,IQ) MODEL ARE FOC 430
C              IDENTIFIED BY IDENTN ROUTINE USING WHOLE RECORD OF SERIEFOC 440
      RL=RLIMIT(ITT)                                                    FOC 450
      IND(1)=N                                                          FOC 460
      IND(2)=5*N                                                        FOC 470
      IND(3)=5                                                          FOC 480
      CALL IDENTN (Z,IND,RL,0.30,ARF,IPARM,1,1)                         FOC 490
 1030 CONTINUE                                                          FOC 500
      IPDQ(3)=IPARM(JTT,1)                                              FOC 510
      IPDQ(4)=IPARM(JTT,3)                                              FOC 520
      IF (IPDQ(1).GT.0) IPDQ(5)=IPARM(JTT,2)                            FOC 530
      ALPHA(1)=0.05                                                     FOC 540
      ALPHA(2)=0.05                                                     FOC 550
      IS=100                                                            FOC 560
      DO 1040 IN=1,5                                                    FOC 570
      DARPS(IN)=0.0                                                     FOC 580
      PMAS(IN)=0.0                                                      FOC 590
 1040 CONTINUE                                                          FOC 600
      PMAC=0.00                                                         FOC 610
      CALL GTSIMP (Z,IPDQ,SEED,ALPHA,DARPS,PMAS,PMAC,WNV,FCST,SIM,IS,WK,FOC 620
     1     IER)                                                         FOC 630
      ALFA(ITT)=ALPHA(1)                                                FOC 640
      NP(ITT)=IPDQ(3)                                                   FOC 650
      NQ(ITT)=IPDQ(4)                                                   FOC 660
      ND(ITT)=IPDQ(5)                                                   FOC 670
CC             COMPUTE THE RANGE OF THE FORCAST AS RANGEP               FOC 680
      RANGEP(ITT)=0.0                                                   FOC 690
      ZMAX=FCST(2,1)                                                    FOC 700
      ZMIN=FCST(2,1)                                                    FOC 710
      DO 1050 J=1,LPROJ                                                 FOC 720
      IF (FCST(2,J).GT.ZMAX) ZMAX=FCST(2,J)                             FOC 730
      IF (FCST(2,J).LT.ZMIN) ZMIN=FCST(2,J)                             FOC 740
 1050 CONTINUE                                                          FOC 750
      RANGEP(ITT)=ZMAX-ZMIN                                             FOC 760
CC               CHECK WITH LIMIT ON RANGEP                             FOC 770
         IF (RANGEP(ITT).LE.RLIMIT(ITT)) GO TO 1060                     FOC 780
      ALFA(ITT)=-6.0                                                    FOC 790
 1060 CONTINUE                                                          FOC 800
C                                                                       FOC 810
C             ITERATED CYCLE IN CASE THE FIRST CHOICE FROM IDENTN CAN   FOC 820
C            NOT GIVE SATISFIED SOLUTION                                FOC 830
C                                                                       FOC 840
         IF (IER.GT.0.OR.ALFA(ITT).LE.0.0) GO TO 1070                   FOC 850
         GO TO 1080                                                     FOC 860
C             SOLUTION IS UNSATISFIED, ITERATED IF ISPY NOT EQUAL TO    FOC 870
C            ZERO AND ITERATION NUMBER JTT IS NOT EXCEED 5              FOC 880
 1070 CONTINUE                                                          FOC 890
         IF (JTT.GE.5) GO TO 1090                                       FOC 900
      JTT=JTT+1                                                         FOC 910
         GO TO 1030                                                     FOC 920
 1080 CONTINUE                                                          FOC 930
 1090 CONTINUE                                                          FOC 940
      PRINT 1100, ITT                                                   FOC 950
      PRINT 1110                                                        FOC 960
      PRINT 1120, IPDQ(3),IPDQ(5),IPDQ(4)                               FOC 970
      PRINT 1130, PMAC                                                  FOC 980
      PRINT 1140, (DARPS(J),J=1,4)                                      FOC 990
      PRINT 1150, (PMAS(J),J=1,4)                                       FOC1000
      PRINT 1160, WNV                                                   FOC1010
      PRINT 1170, ALPHA(1)                                              FOC1020
      PRINT 1110                                                        FOC1030
      RETURN                                                            FOC1040
 1100 FORMAT  (////,10X,*......PRINCIPAL COMPONENT NUMBER*,I13,*....*,/)FOC1050
 1110 FORMAT  (8X,*-----------------------------------------------------FOC1060
     1 ----------------------*,/)                                       FOC1070
 1120 FORMAT  (12X,*(P,D,Q,) OF THE BEST-FIT ARIMA(P,D,Q) MODEL*,       FOC1080
     1  *IDENTIFIED BY IDENTN ARE*,3I4)                                 FOC1090
 1130 FORMAT  (14X,*TREND FACTOR(THETA ZERO)=*,1F8.4)                   FOC1100
 1140 FORMAT  (14X,*AUTOREGRESSIVE PARAMETERS (SI(I))=*,5F8.4)          FOC1110
 1150 FORMAT  (14X,*MOVING AVERAGE PARAMETERS (THETA)=*,5F8.4)          FOC1120
 1160 FORMAT  (14X,*VARIANCE OF THE WHITE NOISE=*,5F10.4)               FOC1130
 1170 FORMAT  (14X,*LEVEL OF SIGNIFICANCE=*,1F5.2)                      FOC1140
      END                                                               FOC1150-
```

```
      SUBROUTINE IDENTN (Z,IND,RLIMIT,DISCON,ARF,IPARM,LOGX,NPRINT)
C-----------------------------------------------------------------------
C   FUNCTION    -TO IDENTIFY IP,ID AND IQ OF AN ARIMA(IP,ID,IQ) MODEL
C   Z           -INPUT TIME SERIES OF LENGTH N
C   IND(I)       I=1, LENGTH OF SERIES
C                 2, LENGTH OF PROJECTION (MAXIMUM 600)
C                  3, NUMBER OF SETS OF PROJECTION DESIRED (MAXIMUM 10)
C   RLIMIT      -MAXIMUM ALLOWABLE RANGE OF PROJECTION ,IT IS A FUNCTION
C                OF THE RANGE OF THE CALIBRATION SERIES
C   DISCON      -DISCOUNT FACTOR (0.10 TO 0.30)IN COMPUTING DISCOUNTED
C                SIGNIFICANT LEVEL AFA(.)
C   ARF         -OUTPUT VECTOR GIVE FIRST FIVE HIGHEST DISCOUNTED
C                SIGNIFICANCE LEVELS
C   IPARM       -OUTPUT MATRIX GIVES THE P,D, AND Q FOR THE TOP FIVE
C                ARIMA(P,D,Q) MODELS RANKED DECREASINGLY BY ARF VALUES
C   LOGX        =1, LOG10 TRANSFORMATION OF Z IS DESIRED
C                0, NOT DESIRED
C   NPRINT      =0, NO PRINT OUT IS DESIRED
C               =1, PRINT OUT IS DESIRED
C-----------------------------------------------------------------------
C
      DIMENSION Z(1  ),ALPHA(2),DARPS(5),PMAS(5),FCST(3,750),
     1 SIM(750,5 ),WK(1500),IPDQ(7)
      DIMENSION ZLAST(11,14),NAMES(14),NP(14),NQ(14),ND(14)
      DIMENSION ALFA(14),ARF(5),RANGEC(14),RANGEP(14),
     1IND(3),IPARM(5,3)
      DOUBLE PRECISION SEED
      DATA NP/2,3,1,2,3,0,1,2,3,0,1,2,0,1/
      DATA NQ/3,2,3,2,1,3,2,1,0,2,1,0,1,0/
C             FOR PRINCIPLE COMPONENT ANALYSIS, NO ALOG WILL BE USED
      LOGX=0
      N=IND(1)
      LPROJ=IND(2)
      NSET=IND(3)
C             SEARCH FOR OPTIMAL ID AS NDD
      CALL FINDD (Z,N,24,NDD,1,0.40)
C               INITIALIZE PARAMETERS TO BE USED IN GTSIMP
      IS=750
      IPDQ(1)=1
      IPDQ(2)=N
      IPDQ(6)=LPROJ
      IPDQ(7)=NSET
      DO 1080 ITT=1,14
      SEED=0.15678943258700
      ALPHA(1)=0.01
      ALPHA(2)=0.05
      IPDQ(3)=NP(ITT)
      IPDQ(4)=NQ(ITT)
      IPDQ(5)=NDD
      DO 1010 IN=1,5
      DARPS(IN)=0.0
      PMAS(IN)=0.0
 1010 CONTINUE
      PMAC=0.0
      FCST(2,LPROJ)=0.0
      ZLAST(1,1)=Z(N)
      DO 1020 I=1,NSET
      ZLAST(I+1,ITT)=0.0
 1020 CONTINUE
      CALL GTSIMP (Z,IPDQ,SEED,ALPHA,DARPS,PMAS,PMAC,WNV,FCST,SIM,IS,WK,
     1       IER)
      ALFA(ITT)=ALPHA(1)
         IF (IER.GT.0.OR.ALFA(ITT).LE.0.0) GO TO 1080
CC            COMPUTE THE RANGE OF PROJECT AS RANGEP
      XMIN=SIM(1,1)
      XMAX=SIM(1,1)
      RANGEP(ITT)=0.0
      DO 1040 J=1,NSET
      DO 1030 I=1,LPROJ
      IF (SIM(I,J).GT.XMAX) XMAX=SIM(I,J)
      IF (SIM(I,J).LT.XMIN) XMIN=SIM(I,J)
      RP=XMAX-XMIN
 1030 CONTINUE
      IF (RP.GT.RANGEP(ITT)) RANGEP(ITT)=RP
CC              CHECK WITH LIMIT ON RANGEP
         IF (RANGEP(ITT).LE.RLIMIT) GO TO 1040
      ALFA(ITT)=-6.0
         GO TO 1080
 1040 CONTINUE
         IF (LOGX.EQ.0) GO TO 1060
      FCST(2,LPROJ)=EXP(FCST(2,LPROJ))
      DO 1050 J=1,NSET
      SIM(LPROJ,J)=EXP(SIM(LPROJ,J))
 1050 CONTINUE
 1060 CONTINUE
      DO 1070 I=1,NSET
      ZLAST(I+1,ITT)=SIM(LPROJ,I)
 1070 CONTINUE
 1080 CONTINUE
      J1=NSET+1
CC     SEARCH HIGHEST ALFA VALUE AND STORE ITS LOCATION
C              DISCOUNTED SIGNIFICANCE LEVEL
      DO 1090 I=1,14
      ALFA(I)=ALFA(I)*(1.0-DISCON)**(NP(I)+NQ(I)-1)
 1090 CONTINUE
C              RANK ALFA(I) FROM MAX TO MIN AND STORE IN AFA
      A=ALFA(1)
      DO 1110 I=1,5
      DO 1100 J=1,14
         IF (ALFA(J).LT.A) GO TO 1100
      A=ALFA(J)
      IJ=J
 1100 CONTINUE
      ARF(I)=A
      ALFA(IJ)=-A
      A=-A
      IPARM(I,1)=NP(IJ)
      IPARM(I,2)=NDD
      IPARM(I,3)=NQ(IJ)
 1110 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE GTSIMP (Z,IPDQ,SEED,ALPHA,DARPS,PMAS,PMAC,WNV,FCST,SIM,
     1       IS,WK,IER)
C-----------------------------------------------------------------------
C   FUNCTION      -NON-SEASONAL ARIMA STOCHASTIC MODEL ANALYSIS FOR A
C                    SINGLE TIME SERIES
C         Z       -INPUT VECTOR OF LENGTH IPDQ(2) CONTAINING THE
C                    TIME SERIES
C         IPDQ    -INPUT/OUTPUT VECTOR OF LENGTH 7, IPDQ(I) CONTAINS,
```

```
C                 I=1, IPDQ(1)=1 IS REQUIRED
C                 I=2, INPUT. LENGTH OF TIME SERIES Z.
C                 I=3, NUMBER OF AUTOREGRESSIVE PARAMETERS IN THE
C                   MODEL
C                 I=4, NUMBER OF MOVING AVERAGE PARAMETERS IN THE MODEL
C                 I=5, DEGREE OF DIFFERENCING OPERATION REQUIRED TO
C                   OBTAIN A STATIONARY TIME SERIES
C                 I=6, FORECASTING CONTROL PARAMETER.
C                   FORECASTS UP TO IPDQ(6) STEPS IN ADVANCE ARE
C                   CALCULATED . IPDQ(6) MUST BE POSITIVE
C                 I=7, SIMULATION OPTION.
C                   IPDQ(7) LESS THAN OR EQUAL TO ZERO IMPLIES
C                     SIMULATION NOT DESIRED
C                   IPDQ(7) POSITIVE IMPLIES IPDQ(7) SIMULATIONS OF
C                     THE TIME SERIES UP TO IPDQ(6) STEPS IN ADVANCE
C                     ARE DESIRED
C        SEED   -INPUT DOUBLE PRECISION NUMBER, SEED IS USED TO
C                 GENERATE THE SIMULATED TIME SERIES
C        ALPHA  -INPUT/OUTPUT VECTOR OF LENGTH 2. ALPHA CONTAINS, WHEN
C                 I=1, INPUT MINIMUM SIGNIFICANCE LEVEL OF THE MODEL
C                   IN THE EXCLUSIVE RANGE (0,1). ON OUTPUT , THE
C                   ESTIMATED SIGNIFICANCE LEVEL IS RETURND.
C                 I=2, INPUT VALUE IN THE EXCLUSIVE RANGE (0,1)
C                   USED FOR COMPUTING 100(1-ALPHA(2)) PERCENT
C                   PROBABILITY LIMITS FOR THE FORECASTS.
C        DARPS  -OUTPUT VECTOR OF LENGTH IPDQ(3)+IPDQ(5) CONTAINING
C                 ESTIMATES OF THE AUTOREGRESSIVE PARAMETERS.
C        PMAS   -OUTPUT VECTOR OF LENGTH IPDQ(4) CONTAINING ESTIMATES
C                 OF THE MOVING AVERAGE PARAMETERS
C        PMAC   -OUTPUT ESTIMATE OF THE TREND FACTOR
C        WNV    -OUTPUT ESTIMATE OF THE WHITE NOISE VARIANCE
C        FCST   -OUTPUT MATRIX OF DIMENSION 3 BY IPDQ(6)
C                 FCST(I,J) , FOR LEAD TIMES J=1,2,...,IPDQ(6),
C                 CONTAINS, WHEN
C                   I=1, THE WEIGHTS FOR THE WEIGHTED SUM OF SHOCKS
C                     THAT GIVES THE FORECAST ERROR
C                   I=2, THE FORECASTS
C                   I=3, THE CORRESPONDING DEVIATION FROM EACH
C                     FORSCAST FOR THE 100(1-ALPHA(2)) PERCENT
C                     PROBABILITY LIMITS
C        SIM    -OUTPUT MATRIX OF DIMENSION IPDQ(6) BY IPDQ(7)
C                 DEFINED ONLY WHEN IPDQ(7) IS POSITIVE, SIM(J,I)
C                 FOR LEAD TIMES J=1,2,...,IPDQ(6), CONTAINS THE
C                 RESULTS OF THE IPDQ(7) SIMULATIONS
C        IS     -FIRST DIMENSION OF SIM AS DIMENSIONED IN THE
C                 CALLING PROGRAM
C        WK     -WORK AREA,LENGTH DEFINED IN THE PROGRAMMING NOTES
C        IER    -ERROR PARAMETER.
C                 TERMINAL ERROR =128+N
C                   N=1, INDICATES A TERMINAL ERROR OCCURRED IN FTMAPS
C                     OR FTARPS
C                   N=2, INDICATES ONE OF THE INPUT PARAMETERS IPDQ(I)
C                     WAS OUT OF RANGE
C                   N=3, INDICATES ALPHA(I),I=1,2 WAS NOT IN THE
C                     EXCLUSIVE RANGE(0,1).
C                 WARNING SIGNAL =32+N
C                   N=4, INDICATES THE MODEL DID NOT PASS THE
C                     ALPHA(1) SIGNIFICANCE TEST. THE NEED FOR MORE
C                     PARAMETERS IS INDICATED
C  NOTE       1. THIS ROUTINE IS OBTAINED BY MODIFYING THE IMSL
C               ROUTINE FTSIMP
C             2. WK IS A WORK VECTOR OF LENGTH AT LEAST THE MAXIMUM OF
C                          A. 2*IPDQ(2)+42
C                          B. IPDQ(2)+P*(P+6)+Q+1
C                          C. IPDQ(2)+(((Q+3)*Q)/2)+6*Q+P36
C                     WHERE P,D,Q ARE NUMBER OF PARAMETERS OF THE ARIMA(P,D,Q)
C-----------------------------------------------------------------------
      DIMENSION          Z(1),IPDQ(1),ALPHA(1),DARPS(1),FCST(3,1),
     1                   SIM(IS,1),WK(1),PMAS(1)
       DIMENSION WMEAN(60)
      DOUBLE PRECISION   SEED
      DATA               I0/0/,I1/1/,I3/3/,I5/5/,I20/20/
C                                  SET OUTPUT PARAMETERS OF IPDQ
 1010 CONTINUE
      N=IPDQ(2)
      IP=IPDQ(3)
      IPP1=IP+1
      IQ=IPDQ(4)
      IQP1=IQ+1
      IF (IP.EQ.0.AND.IPDQ(5).EQ.0) IPDQ(5)=1 .
      ID=IPDQ(5)
      IK=MAX0(IP,IQ)+1
      NM1=N
C                                  ERROR CHECK ON PARAMETERS
      L=IP+IQ
      N2=IP
      DO 1020 I=1,N
      WK(I)=Z(I)
 1020 CONTINUE
      IER=0
        IF (IPDQ(5).EQ.0) GO TO 1050
 1030 CONTINUE
      N=N-1
C             DIFFERENCE  SERIES
      DO 1040 I=1,N
      WK(I)=WK(I+1)-WK(I)
 1040 CONTINUE
        IF (IPDQ(5).EQ.2.AND.NM1.EQ.N+1) GO TO 1030
 1050 CONTINUE
      N2=N+N
C                                  FIND COVARIANCES FOR FTARPS,FTMAP-
      CALL FTAUTO (WK,N,L,I0,I3,WBAR,WK(N+1),WK(N+2),WK,WK,WK)
C             STANDARD ERROR OF WBAR (STE)
      TMP=N
      STE=SQRT(WK(N+1)/(TMP-1.0))/SQRT(TMP)
C             RANDOMLY DRAW PROBABILITY AND FIND T STATISTIC
      ISET=IPDQ(7)
      DO 1060 I=1,ISET
      TP=1.0
      TEMP=RANF(0)
      IF (TEMP.GT.0.5) TP=-1.0
      IF (TEMP.GT.0.5) TEMP=1.0-TEMP
      TEMP=TEMP*2.0
      TMP=N-1
      CALL MDSTI (TEMP,TMP,X,IER)
      XA=X*TP
C             PROBABILISTIC SENSE OF WBAR
      WMEAN(I)=WBAR-XA*STE
 1060 CONTINUE
C                                  FIND DARPS,PMAS,PMAC,WNV
      J=N+L+2
      TEMP=1.0
        IF (IPDQ(3).EQ.0) GO TO 1070
      CALL FTARPS (WK(N+1),WBAR,IPDQ(3),IPDQ(4),DARPS,PMAC,WK(J),IER)
```

```
      TEMP=PHAC/WBAR
         IF (IER.GT.0) GO TO 1230
      CALL CHECK (IPDQ(3),DARPS(1),IER)
         IF (IER.GT.0) GO TO 1240
         IF (IPDQ(4).EQ.0) GO TO 1080
 1070 CONTINUE
      CALL FTMAPS (WK(N+1),DARPS,IPDQ(3),IPDQ(4),PMAS,WNV,WK(J),IER)
         IF (IER.GT.0) GO TO 1250
      IF (IPDQ(3).EQ.0) PHAC=WBAR
 1080 CONTINUE
      CALL CHECK (IPDQ(4),PMAS(1),IER)
         IF (IER.GT.0) GO TO 1260
C                                 FIND ONE-STEP FORECAST ERROR
      DO 1090 I=1,IR
      WK(N+I)=0.0
 1090 CONTINUE
         IF (ID.GT.0) GO TO 1110
      DO 1100 I=1,N
      WK(I)=WK(I)-WBAR
 1100 CONTINUE
 1110 CONTINUE
      DO 1150 I=IR,N
      NPI=N+I
      WK(NPI)=WK(I)
         IF (IP.EQ.0) GO TO 1130
      DO 1120 J=1,IP
      WK(NPI)=WK(NPI)-DARPS(J)*WK(I-J)
 1120 CONTINUE
         IF (IQ.EQ.0) GO TO 1150
 1130 CONTINUE
      DO 1140 J=1,IQ
      WK(NPI)=WK(NPI)+PMAS(J)*WK(NPI-J)
 1140 CONTINUE
 1150 CONTINUE
C                                 FIND COVARIANCE OF ERRORS
      CALL FTAUTO (WK(N+1),N,I20,I0,I5,WK(N2+21),WK(N2+22),WK(N2+23),
     1       WK(N2+1),WK,WK)
C                                 FIND ERROR SIGNIFICANCE STATISTIC
      T=0.0
      DO 1160 I=1,I20
      T=T+WK(N2+I)**2
 1160 CONTINUE
      XN=N-IPDQ(5)
      T=T*XN
      XM=I20-IP-IQ
      CALL MDCH (T,XM,CHI,IER)
C             MINIMUN SIGNIFICANCE LECEL TEST
         IF ((1.0-CHI).GE.ALPHA(1)) GO TO 1170
         GO TO 1270
C                                 FORECASTS
 1170 CONTINUE
      ALPHA(1)=1.0-CHI
CC          LINEAR DECAY OF THE PHAC WHEN ID GT 0
      IF (ID.GT.0) PHAC=PHAC*(1.0-(0.2*IPDQ(6)/IPDQ(2)-0.1))
         IF (IPDQ(3).EQ.0) GO TO 1190
      DO 1180 I=1,IP
      WK(I)=DARPS(I)
 1180 CONTINUF
 1190 CONTINUE
         IF (IPDQ(4).EQ.0) GO TO 1210
      DO 1200 I=1,IQ
      WK(N+I)=PMAS(I)
 1200 CONTINUE
 1210 CONTINUE
      CALL FTCAST (Z,WK,WK(N+1),PHAC,ALPHA(2),IPDQ(2),DARPS,FCST,WNV,
     1       IERX)
C                                 SIMULATIONS
         IF (IPDQ(7).LE.0) GO TO 1220
      M=IPDQ(7)
      J=IP+IO
      K=MAX0(J,1)
      DO 1220 I=1,M
      PHAC=TEMP*WMEAN(I)
      IF (ID.GT.0) PHAC=PHAC*(1.0-(0.2*IPDQ(6)/IPDQ(2)-0.1))
      CALL FTGEN1 (DARPS,PMAS,PHAC,Z(N-K+1),WNV,SEED,J,I0,IPDQ(6),SIM(1,
     1       I),WK)
 1220 CONTINUE
      RETURN
 1230 CONTINUE
      ALPHA(1)=-1.0
      RETURN
 1240 CONTINUE
      ALPHA(1)=-2.0
      RETURN
 1250 CONTINUE
      ALPHA(1)=-3.0
      RETURN
 1260 CONTINUE
      ALPHA(1)=-4.0
      RETURN
 1270 CONTINUE
      ALPHA(1)=-5.0
      RETURN
      END
```

```
      SUBROUTINE PLOT (Q,N,NV,IFCTR,IBEGIN,YLABEL)
C-----------------------------------------------------------------------
C   FUNCTION     - TO PLOT UP TO 9 CURVES IN ONE CHART
C       Q        -INPUT MATRIX OF DIMENSION N BY NV
C       N        -LENGTH OF EACH VARIABLE IN Q(I,J)
C       NV       - NUMBER OF VARIABLES IN Q(I,J) TO BE PLOTTED
C       IFCTR    -SCALING FACTOR
C       IBEGIN   -BEGINNING DATE OF THE RECORD (USED IN X-AXIS)
C       YLABEL   -LABEL TO BE USED IN THE Y-AXIS OF THE PLOT
C   NOTE        1.DIMENSIONS OF THE PARAMETERS IPLOT AND ISVAE (CONTROL
C                  THE PRINTING SPACE ) ARE NOT CHANGEABLE
C               2. DIMENSION OF Q AND THE FIRST IF STATEMENT SHOULD BE
C                  CHANGED ACCORDING TO THE SIZE OF THE MATRIX PLOTTED
C-----------------------------------------------------------------------
      DIMENSION IPLOT(121),LTR(10),ISAVE(121),SCALE(13),Q(250,9)
      DIMENSION YLABEL(8)
      DATA LTR/1H1,1H2,1H3,1H4,1H5,1H6,1H7,1HX,1H*/
      DATA IBLANK,IPER/1H ,1H./
      WRITE (6,1200) (YLABEL(J),J=1,8)
         IF (N.GT.250.OR.NV.GT.9) GO TO 1190
      TMP=10.**IFCTR
      WRITE (6,1210) TMP
      DO 1010 I=1,121
      ISAVE(I)=IBLANK
      IPLOT(I)=IBLANK
 1010 CONTINUE
      WRITE (6,1220)
```

```
      AMIN=999999.                                                      PLO 280
      AMAX=-999999.                                                     PLO 290
      DO 1020 J=1,NV                                                    PLO 300
      DO 1020 I=1,N                                                     PLO 310
      Q(I,J)=Q(I,J)/TMP                                                 PLO 320
      TEMP=Q(I,J)                                                       PLO 330
      IF (TEMP.GT.AMAX) AMAX=TEMP                                       PLO 340
      IF (TEMP.LT.AMIN) AMIN=TEMP                                       PLO 350
 1020 CONTINUE                                                          PLO 360
      IF (AMAX.LE.AMIN) AMAX=1.01*AMIN                                  PLO 370
      TMP=12.                                                           PLO 380
      RATIO=(AMAX-AMIN)/TMP                                             PLO 390
      IFMT=1                                                            PLO 400
      STEP=1.                                                           PLO 410
 1030 CONTINUE                                                          PLO 420
         IF (RATIO.LE..5) GO TO 1040                                    PLO 430
         IF (RATIO.GT.5.) GO TO 1050                                    PLO 440
      IF (RATIO.LE.2.AND.RATIO.GT.1.) STEP=2.*STEP                      PLO 450
      IF (RATIO.LE.4.AND.RATIO.GT.2.) STEP=4.*STEP                      PLO 460
      IF (RATIO.GT.4) STEP=5.*STEP                                      PLO 470
         GO TO 1060                                                     PLO 480
 1040 CONTINUE                                                          PLO 490
      STEP=STEP*.1                                                      PLO 500
      RATIO=RATIO*10.                                                   PLO 510
      IFMT=IFMT+1                                                       PLO 520
         GO TO 1030                                                     PLO 530
 1050 CONTINUE                                                          PLO 540
      STEP=STEP*10.                                                     PLO 550
      RATIO=RATIO*.1                                                    PLO 560
         GO TO 1030                                                     PLO 570
 1060 CONTINUE                                                          PLO 580
      ITP=AMIN/STEP                                                     PLO 590
      TMP=ITP                                                           PLO 600
      TMP=(TMP-.01)*STEP                                                PLO 610
      IF (AMIN.LT.TMP) ITP=ITP-1                                        PLO 620
      ITMP=AMAX/STEP                                                    PLO 630
      TMP=ITMP                                                          PLO 640
      TMP=(TMP+.01)*STEP                                                PLO 650
      IF (AMAX.GT.TMP) ITMP=ITMP+1                                      PLO 660
      IF (IFMT.GT.3) IFMT=3                                             PLO 670
      ITEMP=1                                                           PLO 680
      DO 1070 L=1,13                                                    PLO 690
      SCALE(L)=0.                                                       PLO 700
 1070 CONTINUE                                                          PLO 710
      LX=1                                                              PLO 720
      TEMP=ITP                                                          PLO 730
      TEMP=TEMP*STEP                                                    PLO 740
      ITMP=ITMP+1-ITP                                                   PLO 750
 1080 CONTINUE                                                          PLO 760
      SCALE(LX)=TEMP                                                    PLO 770
      LX=LX+ITEMP                                                       PLO 780
      TEMP=TEMP+STEP                                                    PLO 790
         IF (LX.LE.13) GO TO 1080                                       PLO 800
         GO TO (1090,1100,1110), IFMT                                   PLO 810
 1090 CONTINUE                                                          PLO 820
      WRITE (6,1230) SCALE                                              PLO 830
         GO TO 1120                                                     PLO 840
 1100 CONTINUE                                                          PLO 850
      WRITE (6,1240) SCALE                                              PLO 860
         GO TO 1120                                                     PLO 870
 1110 CONTINUE                                                          PLO 880
      WRITE (6,1250) SCALE                                              PLO 890
 1120 CONTINUE                                                          PLO 900
      STP=STEP*.1                                                       PLO 910
      TMP=ITP                                                           PLO 920
      STRT=TMP*STEP                                                     PLO 930
      ISTEP=1                                                           PLO 940
      LX=1                                                              PLO 950
      DO 1130 L=1,13                                                    PLO 960
      ISAVE(LX)=IPER                                                    PLO 970
      IPLOT(LX)=IPER                                                    PLO 980
      LX=LX+10                                                          PLO 990
 1130 CONTINUE                                                          PLO01000
      KDEX=IBEGIN                                                       PLO01010
      DO 1180 K=1,N                                                     PLO01020
         IF (K.GT.1.AND.K.LT.N) GO TO 1150                              PLO01030
      DO 1140 L=1,121                                                   PLO01040
      IPLOT(L)=IPER                                                     PLO01050
 1140 CONTINUE                                                          PLO01060
 1150 CONTINUE                                                          PLO01070
      DO 1160 I=1,NV                                                    PLO01080
      TEMP=Q(K,I)                                                       PLO01090
      ITMP=(TEMP-STRT)/STP+.5                                           PLO01100
      LX=ITMP*ISTEP+1                                                   PLO01110
      IF (LX.GT.121) LX=121                                             PLO01120
      IF (LX.LT.1) LX=1                                                 PLO01130
      IPLOT(LX)=LTR(I)                                                  PLO01140
 1160 CONTINUE                                                          PLO01150
      WRITE (6,1260) KDEX,IPLOT                                         PLO01160
      KDEX=1+KDEX                                                       PLO01170
      LAG=LAG+1                                                         PLO01180
      DO 1170 L=1,121                                                   PLO01190
      IPLOT(L)=ISAVE(L)                                                 PLO01200
 1170 CONTINUE                                                          PLO01210
 1180 CONTINUE                                                          PLO01220
      WRITE (6,1270)                                                    PLO01230
      RETURN                                                            PLO01240
 1190 CONTINUE                                                          PLO01250
      WRITE (6,1280)                                                    PLO01260
      RETURN                                                            PLO01270
 1200 FORMAT  (8A10)                                                    PLO01280
 1210 FORMAT   (10X,20HQUANTITIES SCALED BY ,F8.0)                      PLO01290
 1220 FORMAT   (1H0)                                                    PLO01300
 1230 FORMAT   (4X,F6.0,12F10.0)                                        PLO01310
 1240 FORMAT   (4X,F6.1,12F10.1)                                        PLO01320
 1250 FORMAT   (4X,F6.2,12F10.2)                                        PLO01330
 1260 FORMAT  (I8,2X,121A1)                                             PLO01340
 1270 FORMAT   (1HR)                                                    PLO01350
 1280 FORMAT   (1X,18HDIMENSION EXCEEDED)                               PLO01360
      END                                                               PLO01370-
```

```
      SUBROUTINE BMD01M
CBMD01M       PRINCIPAL COMPONENT ANALYSIS             AUGUST 26, 1965
      DIMENSION X(150,12),XMEAN(12),COV(12,12),VALU(12),SCALE(12),
     *C(150,12),Z(12,12),FMT(80),NAMES(12)
C     DIMENSION X(400,25),XMEAN(25),COV(25,25),VALU(25),SCALE(25),
C    1C(400,25),Z(25,25),FMT(120),NAMES(25)
      DIMENSION XX(150,12)
      COMMON X,C,Z,COV,XMEAN,NV,N,NAMES,SCALE,XX
      INTEGER A123,B123,C123,TODE,GCK,RNCR
      INTEGER FMT
      DATA A123,B123,C123/6HFINISH,6HPROBLM,3HYES/,SAVE/-10.E36/
C
  209 FORMAT(42H1BMD01M - COMPONENT ANALYSIS - VERSION OF
     115HAUGUST 26, 1965/
     240H HEALTH SCIENCES COMPUTING FACILITY,UCLA//
     314H PROBLEM CODE A6,/
     421H NUMBER OF VARIABLES I3,/
     517H NUMBER OF CASES I6,/
     627H NUMBER OF VARIABLES ADDED I4,/
     735H NUMBER OF TRANSGENERATION CARD(S) I4,/
     8 35H NUMBER OF VARIABLE FORMAT CARD(S) I3,///)
C
C     A123=(+6HFINISH)
C     B123=(+6HPROBLM)
C     C123=(+3HYES)
      NTAPE=5
   10 READ(5,901) TODE,NPROB,NV,N,RNCR,GCK,NADD,NVG,NLV,MTAPE,KVR
      IERROR=0
      IF (A123.EQ.TODE) 201,200
  202 WRITE(6,204)
  201 IF(NTAPE-5)12,12,11
   11 REWIND NTAPE
12    CONTINUE
      RETURN
  200 IF(B123.NE.TODE) GO TO 202
      CALL TPWD(MTAPE,NTAPE)
    9 IF((NV-1)*(NV-26)) 205,202,202
  205 IF((N-2)*(N-401)) 206,202,202
  206 IF((NV+NADD-1)*(NV+NADD-26)) 207,202,202
  207 IF(KVR.GT.0.AND.KVR.LE.10) GO TO 208
      KVR = 1
      WRITE(6,4000)
 4000 FORMAT(1H023X71HNUMBER OF VARIABLE FORMAT CARDS INCORRECTLY SPECIF
     1IED, ASSUMED TO BE 1. )
  204 FORMAT(45H0CONTROL CARDS INCORRECTLY ORDERED OR PUNCHED)
  208 WRITE(6,209) NPROB,NV,N,NADD,NVG,KVR
  901 FORMAT(2A6,I2,I3,2A3,I3,2I2,38X,2I2)
  211 NV1=NV+NADD
      CALL RDLBL(NLV,NV1,NAMES)
C  17 KVR=KVR*12
   17 KVR=KVR*8
      READ(5,942) (FMT(I),I=1,KVR)
      DO 13 I=1,N
      READ(NTAPE,FMT) (X(I,J),J=1,NV)
      WRITE (6,5551) (X(I,J),J=1,NV)
5551  FORMAT (1X,10F12.2)
      DO 13 J=1,NV
13    XX(I,J)=X(I,J)
   19 ON=N
      IF(NVG) 202,600,601
  601 CALL TRANS (X,NV,N,IERROR,NVG)
      IF(IERROR)10,600,600
  600 IF(NV) 202,202,212
  212 NV=NV1
      DO 21 J=1,NV
      XMEAN(J)=0.0
      DO 20 I=1,N
   20 XMEAN(J)=XMEAN(J)+X(I,J)
   21 XMEAN(J)=XMEAN(J)/ON
      DO 22 I=1,NV
      DO 22 J=1,NV
      COV(I,J)=0.0
      DO 22 K=1,N
   22 COV(I,J)=COV(I,J)+(X(K,I)-XMEAN(I))*(X(K,J)-XMEAN(J))
      DO 23 I=1,NV
   23 SCALE(I)=SQRT(COV(I,I))
      DO 24 I=1,NV
      DO 24 J=1,NV
   24 Z(I,J)=COV(I,J)/(SCALE(I)*SCALE(J))
      WRITE(6,923)
      WRITE(6,904)
      CALL PATTY2(Z,NV,NAMES,1)
      CALL EIGEN(VALU,NV,NV)
      WRITE(6,923)
      WRITE(6,907)
      WRITE(6,906) (VALU(I),I=1,NV)
      RANK=0.0
      DO 26 I=1,NV
26    RANK=RANK+VALU(I)
      SMALL=0.0
      DO 18 I=1,NV
      SMALL=SMALL+VALU(I)
   18 VALU(I)=SMALL/RANK
      WRITE(6,937)
      WRITE(6,938) (VALU(I),I=1,NV)
      WRITE(6,923)
      WRITE(6,908)
      CALL PATTY2(Z,NV,NAMES,0)
      DO 29 J=1,NV
      DO 29 I=1,N
   29 X(I,J)=(X(I,J)-XMEAN(J))/SCALE(J)
      ONN=N-1
      SQ=SQRT(ONN)
      DO 43 I=1,N
      DO 43 J=1,NV
      C(I,J)=0.0
      DO 42 K=1,NV
   42 C(I,J)=C(I,J)+X(I,K)*Z(K,J)
   43 C(I,J)=C(I,J)*80
      IF (GCK.EQ.C123) RETURN
      DO 51 J=1,NV
      XMEAN(J)=0.0
      DO 50 I=1,N
   50 XMEAN(J)=XMEAN(J)+C(I,J)
   51 XMEAN(J)=XMEAN(J)/ON
      DO 52 I=1,NV
      DO 52 J=1,NV
      COV(I,J)=0.0
      DO 525 K=1,N
  525 COV(I,J)=COV(I,J)+(C(K,I)-XMEAN(I))*(C(K,J)-XMEAN(J))
   52 COV(I,J)=COV(I,J)/ONN
      WRITE(6,923)
```

```
      WRITE(6,922)                                                      00000125
      CALL PATTY2(COV,NV,NAMES,1)                                       00000126
      RETURN
   97 IF(RNCR.NE.C123) GO TO 40                                         00000127
      WRITE(6,923)                                                      00000128
      WRITE(6,909)                                                      00000129
      WRITE(6,910)                                                      00000130
C     SMALL=-(10.0**36.0)                                               00000131
      SMALL = SAVE                                                      00000132
      DO 39 II=1,NV                                                     00000133
      WRITE(6,912)  II                                                  00000134
      DO 32 I=1,N                                                       00000135
      C(I,1)=0.0                                                        00000136
      C(I,2)=0.0                                                        00000137
      DO 31 K=1,NV                                                      00000138
   31 C(I,1)=C(I,1)+X(I,K)*Z(K,II)                                      00000139
   32 C(I,1)=C(I,1)*SQ                                                  00000140
      DO 39 I=1,N                                                       00000141
      RANK=SMALL                                                        00000142
      DO 38 J=1,N                                                       00000143
      IF(C(J,1)-RANK)38,38,36                                           00000144
   36 IF(C(J,2)-999.0)37,38,38                                          00000145
   37 RANK=C(J,1)                                                       00000146
      NJ=J                                                              00000147
   38 CONTINUE                                                          00000148
      C(NJ,2)=999.0                                                     00000149
      WRITE(6,911) RANK,NJ                                              00000150
   39 CONTINUE                                                          00000151
   40 GO TO 10                                                          00000152
  902 FORMAT(19H0COMPONENT ANALYSIS)                                    00000153
  903 FORMAT(12H PROBLEM NO.I4)                                         00000154
  904 FORMAT(31H0CORRELATION COEFFICIENT MATRIX)                        00000155
  907 FORMAT(12H0EIGENVALUES)                                           00000156
  908 FORMAT(13H0EIGENVECTORS)                                          00000157
  909 FORMAT(48H0RANK ORDER OF EACH STANDARDIZED CASE ORDERED BY)       00000158
  910 FORMAT(44H SIZE OF EACH PRINCIPAL COMPONENT SEPARATELY)           00000159
  911 FORMAT(F18.6,I10)                                                 00000160
  912 FORMAT(16H0  COMPONENT NO.I3,12H    CASE NO.)                     00000161
  922 FORMAT(25H0EIGEN VALUE CHECK MATRIX)                              00000162
  923 FORMAT(1H0)                                                       00000163
  937 FORMAT(40H0CUMULATIVE PROPORTION OF TOTAL VARIANCE)               00000164
  938 FORMAT(1H F11.2,7F15.2)                                           00000165
C 942 FORMAT(12A6)                                                      00000166
  942 FORMAT(8A10)                                                      00000167
  986 FORMAT(6F16.7)                                                    00000168
      END                                                               00000169


      FUNCTION ANUMBF(I)                                                00000170
      ENCODE(4,1,ANUMBF) I                                              00000171
    1 FORMAT(I4)                                                        00000172
      RETURN                                                            00000173
      END                                                               00000174


CEIGEN     SUBROUTINE EIGEN FOR BMD01M        JUNE 28, 1963             00000175
      SUBROUTINE EIGEN(VALU,N,M)                                        00000176
C                                                                       00000177
```

```
C     EIGENVALUES AND EIGENVECTORS OF A REAL SYMMETRIC MATRIX           00000178
C                                                                       00000179
      DIMENSION A(12,12),B(12,12),VALU(12),DIAG(12),SUPERD(24),
     1Q(24),VALL(12),S(24),C(25),D(25),IND(25),U(25),DUMMY(9000)
C    X             DUMMY(10000)                                         00000183
      DIMENSION X123(150,12),Y123(150,12),DUMY1(150,12),DUMY2(144)
C     DIMENSION X123(400,25),Y123(400,25),DUMY1(400,25),DUMY2(625)      00000185
      COMMON X123, Y123, A, B
      EQUIVALENCE (X123,DUMY1),(B,DUMY2),(SUPERD,DUMMY(13)),(TAU,BETA), 00000187
     1            (VALL,D,DUMMY(24)),(Q,S,DUMMY(36)),(IND,U),(II,MATCH),00000188
     2            (DIAG,DUMMY,Y123),(ANORM,ANORM2),(P,PRODS),(T,SMALLD) 00000189
C                                                                       00000190
C     CALCULATE NORM OF MATRIX                                          00000191
C                                                                       00000192
      IX=1                                                              00000193
      ANORM2=A(1,1)**2                                                  00000194
      DO6I=2,N                                                          00000195
      DO5J=1,IX                                                         00000196
    5 ANORM2=A(I,J)**2+A(J,I)**2+ANORM2                                 00000197
      ANORM2=A(I,I)**2+ANORM2                                           00000198
    6 IX=I                                                              00000199
      ANORM=SQRT(ANORM2)                                                00000200
C                                                                       00000201
C     GENERATE IDENTITY MATRIX                                          00000202
C                                                                       00000203
      IF(M.EQ.0) GO TO 45                                               00000204
      IX=1                                                              00000205
      B(1,1)=1.                                                         00000206
      DO20I=2,N                                                         00000207
      DO10J=1,IX                                                        00000208
   10 B(I,J)=B(J,I)=0.                                                  00000209
      B(I,I)=1.                                                         00000210
   20 IX=I                                                              00000211
C                                                                       00000212
C     PERFORM ROTATIONS TO REDUCE MATRIX TO JACOBI FORM                 00000213
C                                                                       00000214
   45 IEXIT=1                                                           00000215
      NN=N-2                                                            00000216
      IF (NN) 890, 170, 55                                              00000217
   55 DO 160 I=1,NN                                                     00000218
      II=I+2                                                            00000219
      DO 160 J=II,N                                                     00000220
      T1=A(I,I+1)                                                       00000221
      T2=A(I,J)                                                         00000222
      GO TO 900                                                         00000223
   90 DO 105 K=I,N                                                      00000224
      T2=COS*A(K,I+1)+SIN*A(K,J)                                        00000225
      A(K,J)=COS*A(K,J)-SIN*A(K,I+1)                                    00000226
  105 A(K,I+1)=T2                                                       00000227
      DO 125 K=I,N                                                      00000228
      T2=COS*A(I+1,K)+SIN*A(J,K)                                        00000229
      A(J,K)=COS*A(J,K)-SIN*A(I+1,K)                                    00000230
  125 A(I+1,K)=T2                                                       00000231
      IF(M.EQ.0) GO TO 160                                              00000232
      DO 150 K=1,N                                                      00000233
      T2=COS*B(K,I+1)+SIN*B(K,J)                                        00000234
      B(K,J)=COS*B(K,J)-SIN*B(K,I+1)                                    00000235
  150 B(K,I+1)=T2                                                       00000236
  160 CONTINUE                                                          00000237
C                                                                       00000238
C     MOVE JACOBI FORM ELEMENTS AND INITIALIZE EIGENVALUE BOUNDS        00000239
C                                                                       00000240
```

```
  170 DO 200 I=1,N
      DIAG(I)=A(I,I)
      VALU(I)=ANORM
  200 VALL(I)=-ANORM
      DO 230 I=2,N
      SUPERD(I-1)=A(I-1,I)
  230 Q(I-1)=(SUPERD(I-1))**2
C
C     DETERMINE SIGNS OF PRINCIPAL MINORS
C
      TAU=0.0
      I=1
  260 MATCH=0
      T2=0.0
      T1=1.0
      DO 450 J=1,N
      P=DIAG(J)-TAU
      IF(T2) 300, 330, 300
  300 IF(T1) 310, 370, 310
  310 T=P*T1-Q(J-1)*T2
      GO TO 410
  330 IF(T1) 335, 350, 350
  335 T1=-1.0
      T=-P
      GO TO 410
  350 T1=1.0
      T=P
      GO TO 410
  370 IF(Q(J-1)) 380, 350, 380
  380 IF(T2) 400, 390, 390
  390 T=-1.0
      GO TO 410
  400 T=1.0
C
C     COUNT AGREEMENTS IN SIGN
C
  410 IF(T1) 425, 420, 420
  420 IF(T) 440, 430, 430
  425 IF(T) 430, 440, 440
  430 MATCH=MATCH+1
  440 T2=T1
  450 T1=T
C
C     ESTABLISH TIGHTER BOUNDS ON EIGENVALUES
C
      DO 530 K=1,N
      IF (K-MATCH) 470, 470, 520
  470 IF(TAU-VALL(K)) 530, 530, 480
  480 VALL(K)=TAU
      GO TO 530
  520 IF(TAU-VALU(K)) 525, 530, 530
  525 VALU(K)=TAU
  530 CONTINUE
  540 IF(VALU(I)-VALL(I)-5.0E-8) 570, 570, 550
  550 IF(VALU(I)) 560, 580, 560
  560 IF(ABS(VALL(I)/VALU(I)-1.0)-5.0E-8) 570, 570, 580
  570 I=I+1
      IF(I-N) 540, 540, 590
  580 TAU=(VALL(I)+VALU(I))/2.0
      GO TO 260
C
C     JACOBI EIGENVECTORS BY ROTATIONAL TRIANGULARIZATION
C
  590 IF(M.EQ.0) GO TO 890
      IEXIT=2
      DO 610 I=1,N
      DO610J=1,I
  610 A(I,J)=A(J,I)=0.
      DO 850 I=1,N
      IF (I-1) 625, 625, 621
  621 IF (VALU(I-1)-VALU(I)-5.0E-7) 730, 730, 622
  622 IF (VALU(I-1)) 623, 625, 623
  623 IF(ABS(VALU(I)/VALU(I-1)-1.0)-5.0E-7) 730, 730, 625
  625 COS=1.0
      SIN=0.0
      DO 700 J=1,N
      IF(J.EQ.1) GO TO 680
      GO TO 900
  650 S(J-1)=SIN
      C(J-1)=COS
      D(J-1)=T1*COS+T2*SIN
  680 T1=(DIAG(J)-VALU(I))*COS-BETA*SIN
      T2=SUPERD(J)
  700 BETA=SUPERD(J)*COS
      D(N)=T1
      DO 725 J=1,N
  725 IND(J)=0
  730 SMALLD=ANORM
      DO 780 J=1,N
      IF (IND(J)-1) 750, 780, 780
  750 IF(ABS(SMALLD)-ABS(D(J))) 780, 780, 760
  760 SMALLD=D(J)
      NN=J
  780 CONTINUE
      IND(NN)=1
      PRODS=1.0
      IF(NN.EQ.1) GO TO 850
      DO 840 K=2,NN
      II=NN+1-K
      A(II+1,I)=C(II)*PRODS
  840 PRODS=-PRODS*S(II)
  850 A(1,I)=PRODS
C
C     FORM MATRIX PRODUCT OF ROTATION MATRIX WITH JACOBI VECTOR MATRIX
C
      DO 885 J=1,N
      DO 865 K=1,N
  865 U(K)=A(K,J)
      DO 885 I=1,N
      A(I,J)=0.0
      DO 885 K=1,N
  885 A(I,J)=B(I,K)*U(K)+A(I,J)
  890 GO TO 941
C
C     CALCULATE SINE AND COSINE OF ANGLE OF ROTATION
C
  900 IF (T2) 910, 940, 910
  910 T=SQRT(T1**2+T2**2)
      COS=T1/T
      SIN=T2/T
      GO TO (90,650), IEXIT
  940 GO TO (160,910), IEXIT
  941 RETURN
      END
```

```
CPATTY2       SUBROUTINE PATTY2 FOR BMD01M         JUNE 28, 1963            00000366
C SUBROUTINE PATTY2 FOR BM01M                     JUNE 28, 1963            00000367
        SUBROUTINE PATTY2(A,N,NAMES,JK)                                    00000368
        DIMENSION A(12,12),NAMES(12),NN(8)
        IT=1                                                               00000370
        KK=0                                                               00000371
        K1=IT                                                              00000372
        K2=MIN0(8,N)                                                       00000373
      5 KK=KK+8                                                            00000374
        IF(N-KK)3,3,4                                                      00000375
      4 IT=IT+1                                                            00000376
        GO TO 5                                                            00000377
      3 DO 50 JX=1,IT                                                      00000378
        LLL=K2-K1+1                                                        00000379
        LL=0                                                               00000380
        IF(JK)35,35,37                                                     00000381
     35 WRITE(6,350) (IG,IG=1,LLL)                                         00000382
        GO TO 45                                                           00000383
     37 DO 40 JJ=K1,K2                                                     00000384
        LL=LL+1                                                            00000385
     40 NN(LL)=NAMES(JJ)                                                   00000386
        WRITE(6,300) (NN(II),II=1,LLL)                                     00000387
     45 DO 10 I=1,N                                                        00000388
     10 WRITE(6,20) NAMES(I),(A(I,J),J=K1,K2)                              00000389
        K1=K2+1                                                            00000390
        K2=K1+7                                                            00000391
        K2=MIN0(K2,N)                                                      00000392
    300 FORMAT(1H013X,A6,7(8X,A6)/)                                        00000393
     20 FORMAT(1H A6,1X,8F14.4)                                            00000394
    350 FORMAT(1H017X,I2,7(12X,I2)/)                                       00000395
     50 CONTINUE                                                           00000396
        RETURN                                                             00000397
        END                                                                00000398

CRDLBL                                                                     00000399
C       SUBROUTINE TO READ IN LABELS CARDS, STORE THEM IN ARRAY,           00000400
C       AND SUBSTITUTE NUMBERS FOR UNLABELED VARIABLES                     00000401
C       NVAR IS TOTAL NUMBER OF VARIABLES                                  00000402
C       NLBVAR IS NUMBER OF LABELED VARIABLES EXPECTED                     00000403
C                                                                          00000404
        SUBROUTINE RDLBL(NLBVAR,NVAR,ARRAY)                                00000405
C       EQUIVALENCE INTEGER AND FLOATING NAMES SO THAT INTEGER SUBTRACTION00000406
C       MAY BE USED TO TEST ALPHABETIC EQUALITY                            00000407
        EQUIVALENCE(LABEL,ALABEL)                                          00000408
        DIMENSION ARRAY(1),IDUM(7),DUMY(7)                                 00000409
        TYPE INTEGER ALABEL                                                00000410
        ALABEL = 3HLAB                                                     00000411
C       NUMBER VARIABLES                                                   00000412
        DO 1 I=1,NVAR                                                      00000413
      1 ARRAY(I)=ANUMBF(I)                                                 00000414
C       IF NO LABELS, RETURN                                               00000415
        IF(NLBVAR) 9,9,2                                                   00000416
      2 N=0                                                                00000417
C       READ 1 LABELS CARD                                                 00000418
     20 READ(5,3) ITEST,(IDUM(J),DUMY(J),J=1,7)                            00000419
      3 FORMAT(A3,3X,7(I4,A6))                                             00000420
C       TEST FOR =LAB= IN FIRST 3 COLS.                                    00000421
        IF(ITEST-LABEL) 4,6,4                                              00000422
C       ERROR--PRINT MESSAGE AND QUIT                                      00000423
      4 WRITE(6,5)                                                         00000424
      5 FORMAT(36H0LABELS CARD NOT FOUND WHEN EXPECTED)                    00000425
        CALL EXIT                                                          00000426
C       EXAMINE 7 FIELDS                                                   00000427
      6 DO 8 J=1,7                                                         00000428
        K=IDUM(J)                                                          00000429
C       TEST INDEX.  IF 0, IGNORE.  IF ILLEGAL, PRINT MESSAGE AND          00000430
C       IGNORE EXCEPT TO COUNT                                             00000431
        IF(K) 11,8,10                                                      00000432
     10 IF(K-NVAR) 7,7,11                                                  00000433
     11 WRITE(6,12) K,DUMY(J)                                              00000434
     12 FORMAT(18H0LABELS CARD INDEX,I7,18H INCORRECT. LABEL ,A6,9H IGNORE00000435
       1D.)                                                                00000436
        GO TO 13                                                           00000437
C       MOVE LABEL TO ARRAY                                                00000438
      7 ARRAY(K)=DUMY(J)                                                   00000439
C       STEP NUMBER OF VARIABLES                                           00000440
     13 N=N+1                                                              00000441
C       TEST FOR END. IF END, RETURN. IF NOT, SCAN OTHER FIELDS.           00000442
        IF(N-NLBVAR) 8,9,9                                                 00000443
      8 CONTINUE                                                           00000444
        GO TO 20                                                           00000445
      9 RETURN                                                             00000446
        END                                                                00000447
```

```
CTRANS    SUBROUTINE TRANS FOR BMD01M   MAY 19,1964                        00000448
        SUBROUTINE TRANS(DATA,NVAR,NSAM,IERROR,NVG)                        00000449
        ASNF(XX)=ATAN(XX/SQRT(1.0-XX**2))                                  00000450
        DIMENSION COV(12,12),C123(150,12),Z123(12,12)
C       DIMENSION COV(25,25),C123(400,25),Z123(25,25)                      00000452
        DIMENSION DATA(150,12)
C       DIMENSION DATA(400,25)                                             00000454
        TYPE INTEGER D123,TODE                                             00000456
C       D123=(+6HTRNGEN)                                                   00000457
        DATA D123/6HTRNGEN/                                                00000458
        MARY=0                                                             00000459
        FN=NSAM                                                            00000460
        WRITE(6,1403)                                                      00000461
        WRITE(6,1400)                                                      00000462
        DO1000 J=1,NVG                                                     00000463
        READ(5,1100) TODE,NEWA,LCODE,LVA,BNEW                              00000464
        IF (D123.EQ.TODE) GO TO 201                                        00000465
        NVAR=-NVAR                                                         00000466
        GO TO 1111                                                         00000467
    201 WRITE(6,1402) J,NEWA,LCODE,LVA,BNEW                                00000468
        IF(LCODE*(15-LCODE)) 710,710,714                                   00000469
    710 WRITE(6,712)                                                       00000470
    712 FORMAT(29HERROR ON TRANSGENERATION CODE)                           00000471
    714 IF(LCODE-10)4,5,5                                                  00000472
      5 NEWB=BNEW                                                          00000473
      4 DO3I=1,NSAM                                                        00000474
        D=DATA(I,LVA)                                                      00000475
        GOTO(10,20,30,40,50,60,70,80,90,100,110,120,130,140),LCODE         00000476
     10 IF(D)99,7,8                                                        00000477
      7 DATA(I,NEWA)=0.0                                                   00000478
```

```
      GO TO 3
    8 DATA(I,NEWA)=SQRT(D)
      GO TO 3
   20 IF(D)99,11,12
   11 DATA(I,NEWA)=1.0
      GO TO 3
   12 DATA(I,NEWA)=SQRT(D)+SQRT(D+1.0)
      GO TO 3
   30 IF(D)99,99,14
   14 DATA(I,NEWA)=ALOG10(D)
      GO TO 3
   40 DATA(I,NEWA)=EXP(D)
      GO TO 3
   50 IF(D)99,16,17
   16 DATA(I,NEWA)=0.0
      GO TO 3
   17 IF(D-1.0)18,19,99
   19 DATA(I,NEWA)=3.14159265/2.0
   19 DATA(I,NEWA)=3.14159265358979/2.0
      GO TO 3
   18 A=SQRT(D)
      DATA(I,NEWA)=ASNF(A)
      GO TO 3
   60 A=D/(FN+1.0)
      B=A+1.0/(FN+1.0)
      IF(A)99,23,24
   23 IF(B)99,26,27
   26 DATA(I,NEWA)=0.0
      GO TO 3
   27 DATA(I,NEWA)= ASNF(SQRT(B))
      GO TO 3
   24 IF(B)99,28,29
   28 DATA(I,NEWA)=ASNF(SQRT(A))
      GO TO 3
   29 A=SQRT(A)
      B=SQRT(B)
      DATA(I,NEWA)=ASNF(A)+ASNF(B)
      GO TO 3
   70 IF(D)31,99,31
   31 DATA(I,NEWA)=1.0/D
      GO TO 3
   80 DATA(I,NEWA)=D+BNEW
      GO TO 3
   90 DATA(I,NEWA)=D*BNEW
      GO TO 3
  100 IF(D)99,32,33
   32 DATA(I,NEWA)=0.0
      GO TO 3
33    DATA(I,NEWA)=D**NEWB
      GO TO 3
  110 DATA(I,NEWA)=D+DATA(I,NEWB)
      GO TO 3
  120 DATA(I,NEWA)=D-DATA(I,NEWB)
      GO TO 3
  130 DATA(I,NEWA)=D*DATA(I,NEWB)
      GO TO 3
  140 IF(DATA(I,NEWB))34,99,34
   34 DATA(I,NEWA)=D/DATA(I,NEWB)
      GO TO 3
   99 IF(MARY)43,44,44
   44 MARY=-999
      IERROR=-999
      WRITE(6,1404) J
   43 WRITF(6,1405) I
    3 CONTINUE
      IF(IFRROR)42,1000,1000
 1000 CONTINUE
 1100 FORMAT(A6,I3,I2,I3,F6.0)
 1400 FORMAT(46H0CARD      K       TRANS    ORIG.   ORIG. VAR(J)/45H  NO.
     1VARIABLE    CODE     VAR(I)    OR CONSTANT)
 1401 FORMAT(41H0PROGRAM CANNOT CONTINUE FOR THIS PROBLEM)
 1402 FORMAT(2H  I2,I8,2I9,F15.5)
 1403 FORMAT(1H06X,23HTRANS GENERATOR CARD(S))
 1404 FORMAT(30H0THE INSTRUCTIONS INDICATED ON/25H TRANS GENERATOR CARD
     1NO.I2,4H RE-/29H SULTED IN THE VIOLATION OF A/31H RESTRICTION FOR
     2THIS TRANSFOR-/31H MATION. THE VIOLATION OCCURRED/28H FOR THE ITEM
     3S LISTED BELOW./)
 1405 FORMAT(10H ITEM NO. I3)
      IF(IERROR)42,1111,1111
   42 WRITE(6,1401)
 1111 RETURN
      END


      SUBROUTINE TPHD( NT1,NT2 )
      IF( NT1 ) 40,10,12
   10 NT1 = 5
   12 IF( NT1-NT2 ) 14,19,14
   14 IF( NT2.EQ.5 ) GO TO 18
   17 REWIND NT2
   19 IF( NT1-5 ) 18,24,18
   18 IF( NT1-6 ) 22,40,22
   22 REWIND NT1
   24 NT2 = NT1
   28 RETURN
   40 WRITE(6,49)
      STOP
   49 FORMAT( 25H ERROR ON TAPE ASSIGNMENT )
      END
```