SHORT TERM LOAD FORECASTING WITH SPECIAL APPLICATION TO THE KENYA POWER SYSTEM

Shadrach Otieno Orero
B.sc. (HONS) (ELECTR. ENG)

A thesis submitted in partial fulfillment for the degree of Master of Science in Electrical Engineering in the University of Nairobi.

1988

This thesis is my original work and has not been presented for a degree in any other University.

ORERO, S.O.

This thesis has been submitted for examination with my Knowledge as University Supervisor.

DR. J.M.Mbuthia

ACKNOWLEDGEMENTS:

The author wishes to acknowledge the many useful discussions with members of staff of the department of Electrical Engineering, University of Nairobi, especially my supervisor, Dr. J.M. Mbuthia who did much to ensure that this work was completed in good time.

I also wish to thank the Kenya Power and Lighting Company for giving me all the assistance and the information that I required for this work.

My thanks also go to my family members, Molly, Curie and Vicky for their patience and understanding while this work was going on.

CONTENTS:

CONTENTS:

LIST OF NOTATIONS AND ABBREVIATIONS.

NOTATIONS.

CAPITAL LETTERS.

B - Backward shift operator; $B^m Z_t = Z_{t-m}$, m is an integer

D - Degree of seasonal differencing

E - Expectation

F - Forward shift operator; $F^m Z_t = Z_{t+m}$, m is an integer

L - Load forecast lead time in hours

N - Number of original load series observations

P - Number of seasonal autoregressive parameters

Q - Number of moving average prarmeters

W - Differenced load series observations

Z - Load demand in megawatts

$\hat{Z}_t$ - Load forecast at time origin t

SMALL LETTERS.

a_t - Normally distributed residual

d - Degree of regular differencing

k - Lag of autocorrelation and partial autocorrelation functions

l - Load forecast lead time in hours

m - Integer

n - Number of observations in the differenced load series observations

p - Number of regular autoregressive parameters

q - Number of regular moving average parameters s - Seasonal period in hours

t - time in hours

GREEK LETTERS.

∇ - Differencing operator, $\nabla_m Z_t = Z_t - Z_{t-m}$, m is an integer

ϕ_i , i=1,..,p are autoregressive parameters in the model

θ_i , i=1,..,q are moving average parameters in the model

σ_a - Standard deviation

X^2 - Chi squared statistic

ABBREVIATIONS.

ACF -Autocorrelation function

ACCFS -Autocorrelation function coefficient

AR -Autoregressive

ARMA - Autoregressive moving average

ARIMA - Autoregressive Integrated Moving Average

G.P.O -General post office

IBM - International business machines

MA - Moving average

ND - Degree of regular differencing

NM - Number of regular moving average parameters

NO - Number

NSR - Degree of seasonal autoregressive parameters

K P& T - Kenya posts and telecommunications

PACF - Partial autocorrelation function

PACCFS - Partial autocorrelation function coefficients

PAS - Power apparatus and systems

SCADA- System control and data acquisition.

SE - Standard error

SPSS - Statistical Package for social scientists

V.H.F - Very high frequency

Vol. - Volume

CHAPTER 1

INTRODUCTION

Power system operation and control requires, among other things, an accurate knowledge of the total system load demand. The main objective of effective power control is to regulate the generated power so as to follow the fluctuating load and then to maintain the system frequency within an allowable range.

In the past resources were abundant, fuel supplies were cheap and load forecasting did not receive the attention it deserved. The daily economic operation activities such as load flow studies, generation plant load scheduling, unit commitment, system security and contingency analysis and short term maintenance planning all require short term load forecasts for a period ranging from about 1 hour to 24 hours.

In a mixed hydrothermal system like the Kenyan case under study, the preparation of thermal and geothermal plants require much time before bringing them on line and a load forecast several hours ahead is required before committing such units. Kenya at the moment is interconnected with the Ugandan power system and plans are underway to interconnect with neighbouring Tanzania. Short term load forecasts will be useful in the energy trade with these utilities.

The subject of short term load forecasting has received widespread attention for more than a decade now. This is due to the ever pressing need to use the available scarce resources as economically as possible and also due to the availability of cheap and powerful computers for the analysis of power system load characteristics.

The review paper by Abu-El-Magd and N.K.Sihna [1] gives an overview of the work that has been done in the area of short term load forecasting. All these procedures require modeling of the power system load demand characteristics and thereafter evaluating the model parameters to be used in the forecast algorithm for producing the desired load prediction.

The forecasting techniques can broadly be classified into two classes;

1) Methods involving past load data only;

2) Methods involving both past load data and weather variables;

Amongst the most widely used forecasting techniques are;

1. Regression based algorithms, where the relationship between the residuals of the load and the weather variables is modeled using the mathematical techniques of linear regression analysis [2].

Multiple regression models are based on explanatory variables, and for a given time series, the explanation variables are selected on the basis of the correlation analysis of the load series. For example a multiple regression model can be written as;

$$Z(t)=a_1 Z_1(t)+ a_2 Z_2(t)+ \text{-----} + a_n Z_n(t)$$

where Z_1, Z_2, -----, Z_n are the explanatory variables for the time series Z(t). Normally this model is used to relate weather variables to the weather sensitive load.

This approach requires a long off-line analysis, using a lot of load data and the accuracy of the results depends heavily on the model assumed at the beginning. It has been used extensively in medium and long term load forecasting.

2: State space models in which the load and weather variables are represented using state space formulation and the weather and load states are updated using Kalman filtering techniques [3].

The main reason for this approach is that the powerful Kalman filtering theory is used to obtain the optimum forecasts. This approach is well suited for on-line analysis. The identification of the model parameters is the main difficulty associated with this approach because the Kalman filtering theory assumes the model is exactly known before hand.

Here the state variables are considered to be the system load itself, the increment of the system load and the short term and long term load patterns, different models being developed for different time frames. For example, in daily load forecasting some periodical load pattern is contained and the effects of the weather conditions on the system load cannot be neglected. Thus the model can be written as;

$$\begin{bmatrix} X(k+1) \\ \Delta(k+1) \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & a(k) \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ \beta(k) & \gamma(k) \end{bmatrix} \cdot \begin{bmatrix} T(k) \\ H(k) \end{bmatrix} + \begin{bmatrix} v1(k) \\ v2(k) \end{bmatrix}$$

where X(k) is the daily peak load, $\Delta(k)$ is the load fluctuation because of weather conditions, temperature T(k) and humidity H(k). a(k), $\beta(k)$, $\gamma(k)$ can be estimated using past observations.

The main difficulty in using Kalman filtering theory to obtain the best estimates of the states arises from the fact that the noise covariances are unknown.

3: Time series approach formulated by Box and Jenkins in which the load is modeled as a stochastic process and the model parameters are estimated using the maximum likelihood principle [4]-[6]. This stochastic approach to the problem of filtering and forecasting was first presented by Wiener [4]. He imposed the following restriction on the filtering and prediction problems:

a) the process is stationary

b) the predictor index is the minimization of mean square error, and

c) the predictor is linear.

With the above assumptions one only needs the autocorrelation function of the process and the noise input, and the cross-correlation function of the two. This idea was extended by Box and Jenkins [4] for handling a class of non-stationary processes by a finite linear transformation.

The determination of the model order is done by examining the pattern of the sample autocorrelation function, as well as the partial autocorrelation functions [4]. Such correlation plots is for identifying possible underlying behaviour.

4. Methods based on spectral decomposition [7]-[8].

These methods divides the load demand into various components. For example, Farmer et al [8], divides the load into 3 components; a long term trend, a component varying with the day of the week, and a random component. If $Z_{wd}(t)$ denotes the load in the W^{th} week of the year, on the d^{th} day of that week, and the time of the day (t), then $Z_{wd}(t)$ is expressed in the form;

$$Z_{wd}(t) = A_w(t) + B_d(t) + X_{wd}(t)$$

where $A_w(t)$ represents a trend term which is updated weekly, $B_d(t)$ denotes a term dependent on the day of the week, and $X_{wd}(t)$ denotes the residual component.

The terms $A_w(t)$ and $B_d(t)$ are found by minimising the mean square error of the random component average over several weeks of past data. This method requires a lot of past load data for analysis and this means a lot of computer memory is required for effective analysis, which is its main drawback.

5: Christiaanse, [9], for instance used the general method exponential smoothing where the weekly variations in hourly load are described as a cyclic function of time with a period of one week. The model selected is of the form;

$$Z(t)= c+ \sum_{i=1}^{m} (ai \sin wi(t) + bi \cos wi(t))$$

that is, a constant c and a fourier series with "m" frequencies. Forecasts from the model for lead time L are in the form;

$$Z(t+L)=a(t)\ f(t+L)$$

where $f(t)= \begin{bmatrix} \sin w1(t) \\ \cos w1(t) \\ . \\ . \\ \sin wm(t) \\ \cos wm(t) \end{bmatrix}$

and a(t) is a row vector containing the estimates of the parameters. These parameters are estimated in such a way as to minimise the square of the residuals, using a weighted least squares criterion,using a smoothing constant between zero and one.

The problem with the exponential smoothing method is that the accuracy of the forecasts depends heavily on the smoothing constant and to some extent on the general form of the model chosen beforehand.

Observations from a naturally occurring phenomena such as load demand which depends on a number of inter-related variables such as economic factors, social behaviour of consumers, effects of weather and other unquantifiable factors, posses an inherent probabilistic structure. Deterministic models cannot be obtained for such systems and stochastic models have been employed widely to model such series.

In this study, the time series analysis system identification approach to stochastic model building has been chosen to analyse the load using past data only. The exclusion of weather variables is due to the fact that Kenya is basically a tropical country and the amount of installed equipment that is sensitive to weather (such as space heaters, air conditioners etc.) is minimal. The other factor is due to the non-uniformity of weather conditions in various parts of the country which would make it difficult to include them in a study of this magnitude. Inclusion of such variables would also involve a prediction of weather parameters as well and this could possibly lead to more errors because of the double forecasting process.

The main advantage of the time series approach to short term load forecasting are it's ease of understanding, implementation and the accuracy of it's results.

There are many approaches to time series analysis including the better known regression analysis. The general mixed autoregressive integrated moving average (ARIMA) modeling approach pioneered by Box, G.E and Jenkins [4] for any stochastic process has many advantages over the alternative modeling approaches.

The beauty of the ARIMA approach is that Box and Jenkins have laid a firm and rigorous mathematical basis for its analysis. In the case of seasonality or periodicity in the data for instance, the alternative approaches often require a seasonal adjustment of the time series prior to analysis. The ARIMA approach in contrast models the dependencies which define seasonality. Another case is in the treatment of growth and trend effects in the series; while other techniques require separate treatment of such terms, the ARIMA approach takes care of such adjustments in the modeling process by a linear transformation of the original series into a stationary time series. These advantages are brought out clearly in a study by Plosser [10].

This particular study is unique in the sense that every power system is different from each other. Every system has its own load demand characteristics and no load forecasting programme can be universally applied. A load forecasting programme can be developed for a system only after its load demand characteristics are modeled and understood.

In Kenya, at the moment there is no proper load forecasting procedure. Much depends on the experience of the System controller on duty who uses his own judgment and past experience. While such forecasts are sometimes accurate, most times they lead to gross errors and hence uneconomic operation of the system.

It can be seen that there is an urgent need for a short term load forecasting algorithm for the Kenya power utility. A prediction scheme which provides accurate estimates of the load demand a few hours a head satisfies the requirements of the control system for which this study is based.

This study sets out to develop a load forecasting programme for the Kenya power system based on a sound mathematical and scientific load forecasting technique. It makes use of the available and affordable personal computer to develop the programme. The quantity to be forecast is the total average hourly load demand in megawatts. New forecasts are to be computed each hour, immediately following the reading of the integrated load demand of the previous hour.

CHAPTER 2

MATHEMATICAL THEORY OF THE MODELING PROCESS.

2.0 INTRODUCTION

Time series analysis is essentially concerned with evaluating the properties of the probability model which generated the observed time series. In this analysis the evolving load demand is assumed to be a stochastic process which can be described by some of the moments of the generating process such as the mean, variance and autocovariance functions. At its basis, a time series process consists of a random shock or white noise inputs and a realization or observation outputs, which in this case are the hourly megawatt total system load demand.

2.1 THE AUTOCORRELATION FUNCTION (ACF).

A stationary stochastic process is fully determined by its mean, variance and autocorrelation function. If two processes have the same three variables, then they are the same process.

Since each distinct process has a unique autocorrelation function, the autocorrelation function can be estimated from a realized time series and that information used to determine the process structure which generated the realization.

For a Z_t time series process, the autocorrelation function is defined as

$$ACF(k) = covariance\,(Z_t, Z_{t+k}) \,/\, variance\,(Z_t).$$

Given a realization of the Z_t process, a finite time series of N observations, the autocorrelation function may be estimated from the formulae;

$$ACF(k)= \frac{\sum_{i=1}^{i=N}(Z_i - \bar{Z}) \times (Z_{i+k} - \bar{Z}) \times (N/N-k)}{\sum_{i=1}^{i=N}(Z_i - \bar{Z})^2} \qquad 2.1.1$$

where $\bar{Z}$ is the mean and ACF(k) is thus a measure of the inherent relationship or correlation between Z_t and Z_{t+k} observations [ref.4].

A plot of the autocorrelation function coefficients ACF(k) as a function of k or lag k is called the autocorrelation function of the stochastic process. It is a dimensionless quantity and less than or equal to one. As the value of k increases, confidence in the estimates of ACF(k) diminishes.

Calculation of the standard error (SE) of the estimates helps in giving confidence limits within which the estimates can be accepted. The standard error is estimated from the formulae;

$$SE\,[ACF(k)]= \sqrt{1/N(1+2\sum_{i=1}^{k} ACF(i)^2)} \qquad 2.1.2$$

Estimated values of ACF(k) which lie within plus or minus two standard deviation confidence limits are thus not statistically different from zero with 95% confidence [4].

2.1.1 AUTOREGRESSIVE (AR) MODELS.

A p^{th} order autoregressive process is written as;

$$Z_t = \phi_1 Z_{t-1} + \phi_2 Z_{t-2} + \ldots\ldots + \phi_p Z_{t-p} + a_t \qquad \text{\_\_\_\_\_\_2.1.1.1}$$

In this model, the current value of the process is expressed as a finite, linear, aggregate of previous values of the process and a shock a_t.

One of the most common autoregressive processes is that of order one, written as $Z_t = \phi_1 Z_{t-1} + a_t$ ___2.1.1.2

The ACF of this process is expected to decay exponentially from lag to lag.

Some operators used in simplifying the analysis of the models are;

1) The backward shift operator, B which is defined as $BZ_t = Z_{t-1}$, thus $B^m Z_t = Z_{t-m}$

2) The inverse operation is performed by a forward shift operator $F = B^{-1}$ given by $FZ_t = Z_{t+1}$, hence $F^m Z_t = Z_{t+m}$

3) The difference operator, V which can be written as $\nabla Z_t = Z_t - Z_{t-1}$

A general autoregressive operator of order p can be defined by $\phi(B) = 1 - \phi_1 B - \phi_2 B^2 - \ldots\ldots - \phi_p B^p$ ____2.1.1.3

hence equation 2.1.1.1 of the general AR model can be economically written as;

$$\phi(B) Z_t = a_t \qquad \text{\_\_\_\_\_\_\_\_\_2.1.1.4}$$

Thus an AR process can be considered as the output Z_t from a linear filter with a transfer function $\phi(B)$ when the input is white noise, a_t.

2.1.2 MOVING AVERAGE (MA) MODELS.

In practice, the time series analysis begins with an autocorrelation function estimated from the original or raw time series, Z_t. If the ACF indicates that the process is non-stationary, then the series must be differenced. The second stage of the analysis is an identification of a model for the stationary series based on the serial correlation patterns shown in the autocorrelation function.

One class of serial dependency is the q^{th} order moving average process. Here Z_t is linearly dependent on a finite number of q previous a's. Thus

$$Z_t = a_t - \theta_1 a_{t-1} - \theta_2 a_{t-2} - \ldots\ldots - \theta_q a_{t-q} \qquad \_\_\_\_2.1.2.1$$

A general moving average operator of order q can be defined by

$$\theta(B) = 1 - \theta_1 B - \theta_2 B^2 - \ldots\ldots\ldots - \theta_q B^q \qquad \_\_\_\_2.1.2.2$$

hence the moving average model can be written as

$$Z_t = \theta(B)\, a_t \qquad \_\_\_\_2.1.2.3$$

An MA process of order one is expected to have a non-zero value of ACF(1) while all the successive lags of the ACF(k) are expected to be zero.

In general, a q^{th} order MA process is expected to have non-zero values of ACF(1),....ACF(q). The values of ACF(q+1) and beyond are all expected to be zero, thus an MA process model identification is based on a count of the number of non-zero spikes in the first q lags of the ACF.

2.2 THE PARTIAL AUTOCORRELATION FUNCTION (PACF)

This is a useful statistic for model identification. It is a complementary tool to the autocorrelation function in the identification of time series models.

The lag k partial autocorrelation function, PACF(k) is a measure of the correlation between time series observations k units apart after correlation between intermediate lags have been removed or 'partialled' out. Unlike the autocorrelation function, the partial autocorrelation function cannot be estimated from a simple straight forward formulae.

It is usually estimated from the autocorrelation function since it is a function of the expected autocorrelation function [4].

The below formulae is used to estimate the partial autocorrelation function. The PACF at lag k is denoted by $\varnothing(kk)$ and if the ACF at lag k is denoted by r_k then;

$$\emptyset(kk) = r_1 \qquad \text{for } k=1$$

and

$$\emptyset(kk) = \frac{r_k - \sum_{j=1}^{k-1} \emptyset_{k-1,j} \times r_{k-j}}{1 - \sum_{j=1}^{k-1} \emptyset_{k-1,j} \times r_j} \qquad \text{for } k=2,3,\ldots,k. \qquad \text{———— 2.2.1.}$$

$$\text{where } \emptyset_{kj} = \emptyset_{k-1,j} - \emptyset_{(kk)} \times \emptyset_{k-1,k-j} \qquad \text{for } j=1,2,\ldots,k-1$$

From the above formulae it can be seen that once the ACF is obtained, the PACF can be derived by simple algebraic substitution.

An AR model of order one is expected to have a non-zero PACF(1), while PACF(2) and all successive lags are expected to be zero, While a qth order MA process has a decaying PACF, that is, all PACF(k) are expected to be non-zero [4].

In general, the PACF of a q^{th} order MA process is expected to decay to zero but at a rate determined by $\theta 1, \ldots\ldots, \theta q$ parameters. This means that moving average processes have decaying partial autocorrelation functions, while autoregressive processes have spiking partial autocorrelations.

2.3 MIXED MODELS (ARMA).

To achieve a better fit for certain actual time series, it is sometimes advantageous to include both AR and MA terms in the model.

The relationships in the autoregressive and moving average processes so far discussed place some limits on mixed models. Such relationships sometimes lead to parameter redundancies, because at times complex models are equivalent to simpler models with fewer parameters. Both the ACF and PACF of a mixed process are expected to decay. A general mixed ARMA model can be written as;

$$Z_t = \phi_1 Z_{t-1} + \ldots\ldots + \phi_p Z_{t-p} + a_t - \theta_1 a_{t-1} - \ldots - \theta_q a_{t-q} \qquad 2.3.1$$

The general equation for a mixed model can then take the the form $\phi(B) Z_t = \theta(B) a_t$ ——— 2.3.2

In practice it is frequently true that adequate representation of actually occurring stationary series or one that has been made stationary by transformation can be obtained with AR, MA or mixed models in which the order is not greater than two.

2.4 AUTOREGRESSIVE INTEGRATED MOVING AVERAGE (ARIMA) ALGEBRA

An observed time series, in our case the hourly load demand denoted as $Z_1, Z_2, \ldots\ldots, Z_{t-1}, Z_t$, can be described as a realization of a stochastic process.

At the heart of the generating process is a sequence of random shocks, a_t, which conveniently summarize the multitude of factors producing the variation in the load demand. For computational simplicity it is assumed that the random shocks are normally and independently distributed.

Many actual series exhibit nonstationary behaviour and d not vary about a fixed mean. In particular, although th general level about which fluctuations are occurring may be different, the behaviour of the series, when differences in level are allowed for, may be similar. Thus a general model which can represent non-stationary behaviour is of the form

$$\phi(B)W_t = \theta(B)a_t \qquad \text{\_\_\_2.4.1}$$

where $W_t = \nabla Z_t$ ___2.4.2

and W_t is the series that has been made stationary by taking the d^{th} difference of the Z_t process to make it stationary. The process defined by equations 2.4.1 and 2.4.2 is called an autoregressive integrated moving average (ARIMA) process. This process is defined by

$$W_t = \phi_1 W_{t-1} + \ldots\ldots + \phi_p W_{t-p} + a_t - \theta_1 a_{t-1} - \ldots - \theta_q a_{t-q} \qquad \text{\_\_2.4.3}$$

An ARIMA model has three structural parameters denoted as p,d,q which describe the relationship between the random shocks and the observed load series. The parameter p indicates an autoregressive relationship. For example a model where p=1, q=d=0 denoted as a (1,0,0) model is written as $Z_t = \phi_1 Z_{t-1} + a_t$. This is a model where the current observation Z_t is composed of a portion of the preceding observation, Z_{t-1}, and a random shock a_t. An ARIMA (2,0,0) model would be written as $Z_t = \phi_1 Z_{t-1} + \phi_2 Z_{t-2} + a_t$ showing that the parameter p denotes the number of past observations used to predict the current observation.

The structural parameter q denotes the number of moving average structures in the model. An ARIMA (0,0,1) model would thus be written as $Z_t = a_t - \theta_1 a_{t-1}$ and a (0,0,2) model would be written as $Z_t = a_t - \theta_1 a_{t-1} - \theta_1 a_{t-2}$

An ARIMA (0,0,q) model is one where the current observation, Z_t is composed of a current random shock a_t and a portion of the q-1 preceding random shocks, a_{t-1} through a_{t-q}

Finally the structural parameter d indicates that the time series observations have been differenced. Differencing amounts to subtracting the first observation from the second, second from third and so on. This is usually performed on a non-stationary time series to make it a stationary process. An ARIMA (0,1,0) model would be written as $Z_t - Z_{t-1} = a_t$. This means that the current observation, Z_t is equal to the preceding observation, Z_{t-1} plus the current shock a_t

Model identification refers to the empirical procedures by which the best set of parameters p,d,q are selected for a given load series.

2.5 SEASONAL MODELS.

Seasonality is defined as any periodic or cyclic behaviour in the time series. The ARIMA approach of analysis models dependencies which define seasonality. There also exist seasonal ARIMA structures denoted by P,D,Q.

P denotes the number of seasonal autoregressive parameters, Q the number of seasonal moving average parameters and D the degree of seasonal differencing.

If a series exhibits seasonal non-stationarity, to make it stationary it must be differenced with respect to the seasonal period. Seasonal autoregression is where the current observation depends upon the corresponding observation of the series for the preceding period or season. Seasonal moving average is when the current observation depends upon the random shock of the preceding period.

Similar rules of regular ARIMA (p,d,q) models also apply to seasonal time series analysis. Identification of a seasonal ARIMA structure proceeds from an examination of the ACF and PACF of the raw data. The only difference between seasonal and regular models is that for the seasonal processes, patterns of spiking and decay in the autocorrelations and partial autocorrelations appear at the seasonal lags.

Seasonal non-stationarity is indicated by an ACF that dies out slowly from seasonal lag to seasonal lag. Seasonal autoregression is indicated by an ACF that dies out exponentially from seasonal lag to seasonal lag while the ACF of a seasonal moving process spikes at the seasonal lags.

Most time series with seasonal ARIMA behaviour also exhibit regular behaviour as well. A powerful model can be realized by incorporating regular and seasonal structures multiplicatively, an example of such a model of two moving average parameters can be written as;

$$(1-B)(1-B^{s})Z_t=(1-\Theta_1 B)(1-\Theta_2 B^{s})a_t \qquad \text{_____ 2.5.1}$$

Simplifying the equation and using the del (∇) operator we have

$$\nabla_1 \nabla_s Z_t=(1-\Theta_1 B)(1-\Theta_2 B^{s})a_t . \qquad \text{_____2.5.2}$$

The general ARIMA model which includes seasonality is denoted as (p,d,q) x (P,D,Q)s which makes it s multiplicative nature explicit,where s is the seasonal period length.

CHAPTER 3

MODEL DEVELOPMENT

3.0 INTRODUCTION.

Having developed the theory behind ARIMA models, the problem of building a model for the load time series is now addressed.

The model building strategy is based on three procedures of identification, estimation and diagnosis. The main aim is to construct a model which is statistically adequate as well as parsimonious (having the minimum number of parameters).

The model building process is summarised by the block diagram below:

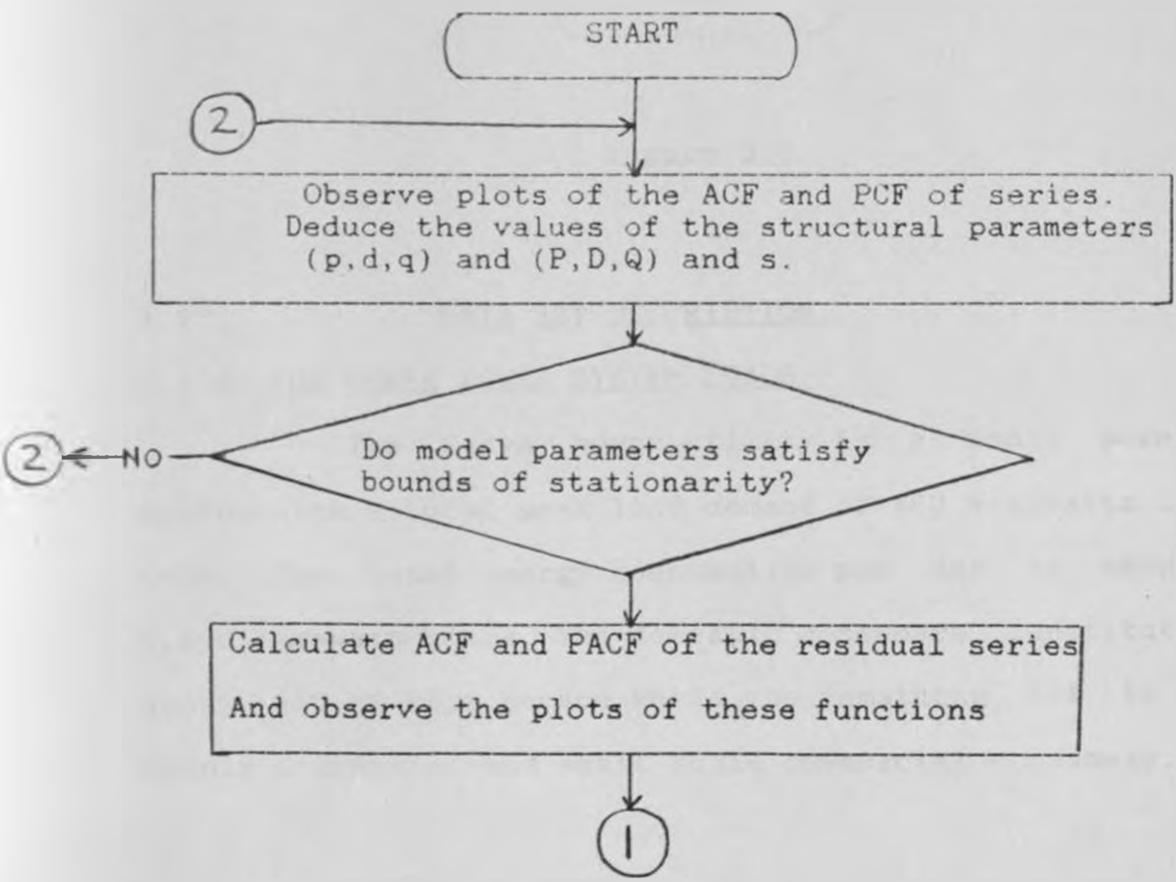


Figure 3.0.

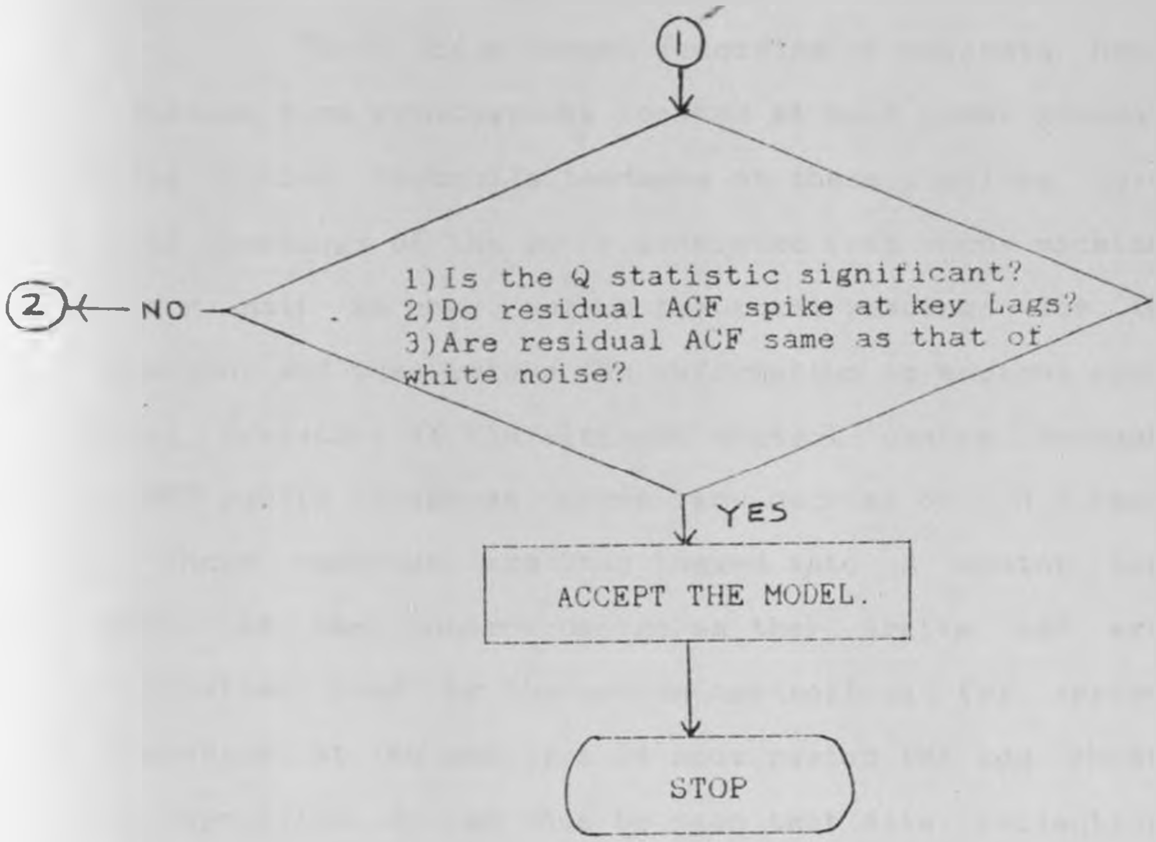


Figure 3.0

3.1 DATA SET DESCRIPTION.

3.1.0 THE KENYA POWER SYSTEM LOADS.

The Kenya power utility is a small power system with a total peak load demand of 460 megawatts to date. The total energy consumption per day is about 8,500 megawatt hours. The domestic consumers constitute about 35% of this demand while the remaining 65% is mainly industrial and small scale commercial consumers.

3.1.1 DATA RECORDING.

There is a manual recording of megawatt hour readings from printographs located at each power generating station. Control attendants at these stations take load readings of the power generated from each machine every half an hour, sum up the total readings for the station, and then relays the information to another control assistant at the national control centre through K.P&T public telephone, power line carrier or V.H.F radio.

These readings are then logged into a master log sheet at the control centre as they arrive and are thereafter used by the system controllers for system operation. At the end of a 24 hour period the log sheet is kept filed. It can thus be seen that data collection and retrieval is a tedious and difficult process.

3.1.2 DATA ANALYSIS

Some readings of the half-hourly load readings for 1987 and part of 1988 was collected at the Kenya power and lighting utility national control centre. The half hourly load readings are averaged to obtain the hourly readings which are then entered into a load data file for computer analysis.

Six weeks of data were used for model development. Six weeks of data were analysed for four different periods in the year for a comparison to see whether the load model structure and parameters change significantly within the year.

Data for periods as long as twelve weeks were analysed for some of the periods to observe the behaviour of the model structure and parameters as the size of the data base increased.

3.2 MODEL IDENTIFICATION AND STRUCTURE DETERMINATION.

The key to model identification is the human pattern recognition of the autocorrelation and partial autocorrelation functions of the various forms of the load time series observations.

The estimated ACF and PACF will indicate whether the series is stationary or not, the existence of any seasonal patterns and whether the series is a moving average, autoregressive, mixed ARMA or just white noise process. The general ARIMA model can be denoted as;

$(p,d,q) \times (P,D,Q)_s$.

Where p denotes the number of regular autoregressive parameters, d the degree of regular differencing, q the number of regular moving average parameters, P the number of seasonal autoregressive parameters, D the degree of seasonal differencing, Q the number of seasonal moving average parameters and s the seasonal period. The values of the parameters p,d,q as well as P,D,Q and s can also be obtained from these plots.

The analysis begins by looking at the plots of these functions which are obtained from the programs which have been written specifically for evaluating the autocorrelation and partial autocorrelation functions whose block diagram is shown in figure 3.2.0 below.

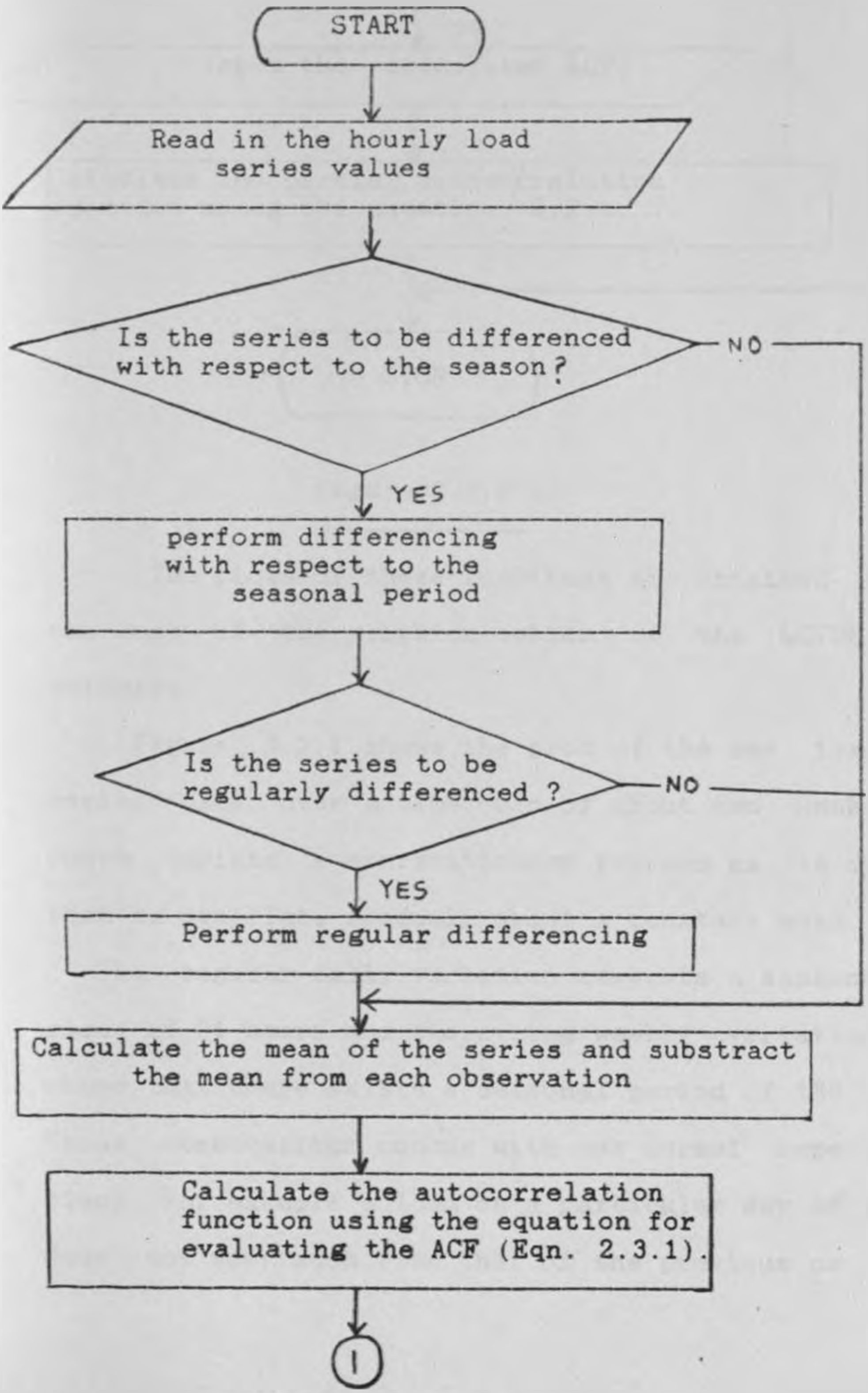


Figure 3.2.0.

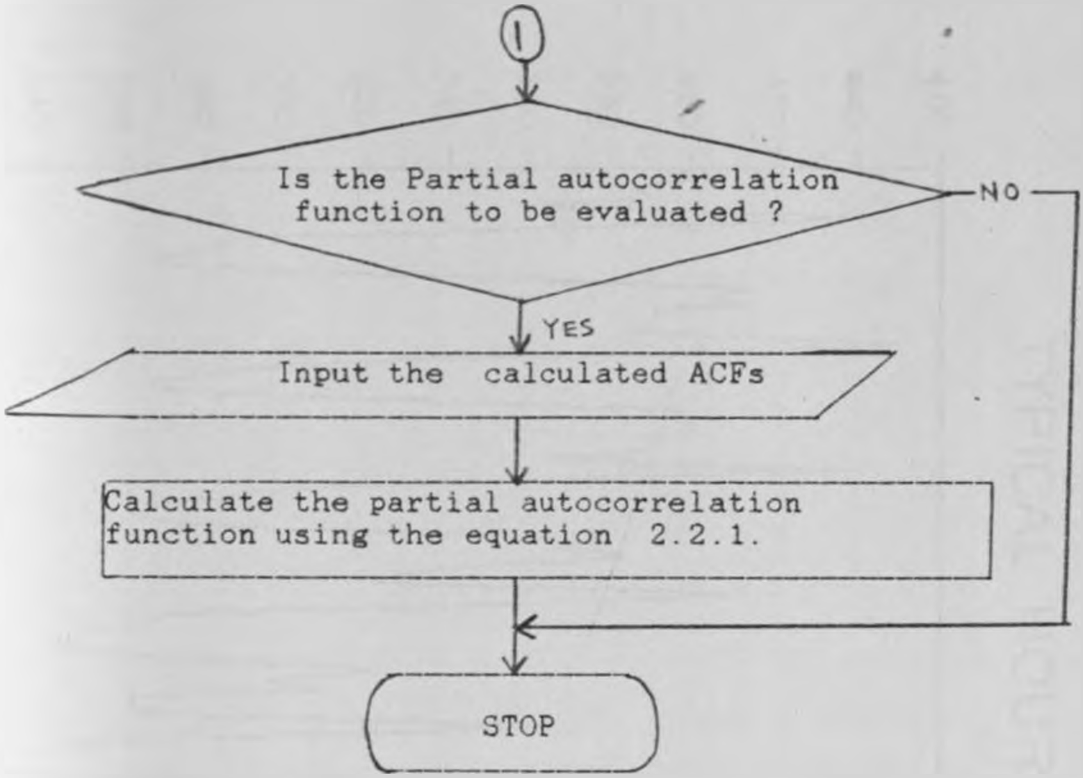


Figure 3.2.0.

The plots of these functions are obtained through the use of the graphics option of the LOTUS 1-2-3 software.

Figure 3.2.1 shows the plot of the raw load time series data over a time span of about two weeks. The curve depicts a non-stationary process as it does not seem to oscillate randomly about a constant mean.

The regular daily variation suggests a seasonal period of 24 hours and the strong weekly variation also shows that there exists a seasonal period of 168 hours. These observations concur with our normal expectations since, for example a load on a particular day of the week does not vary much from that of the previous or coming

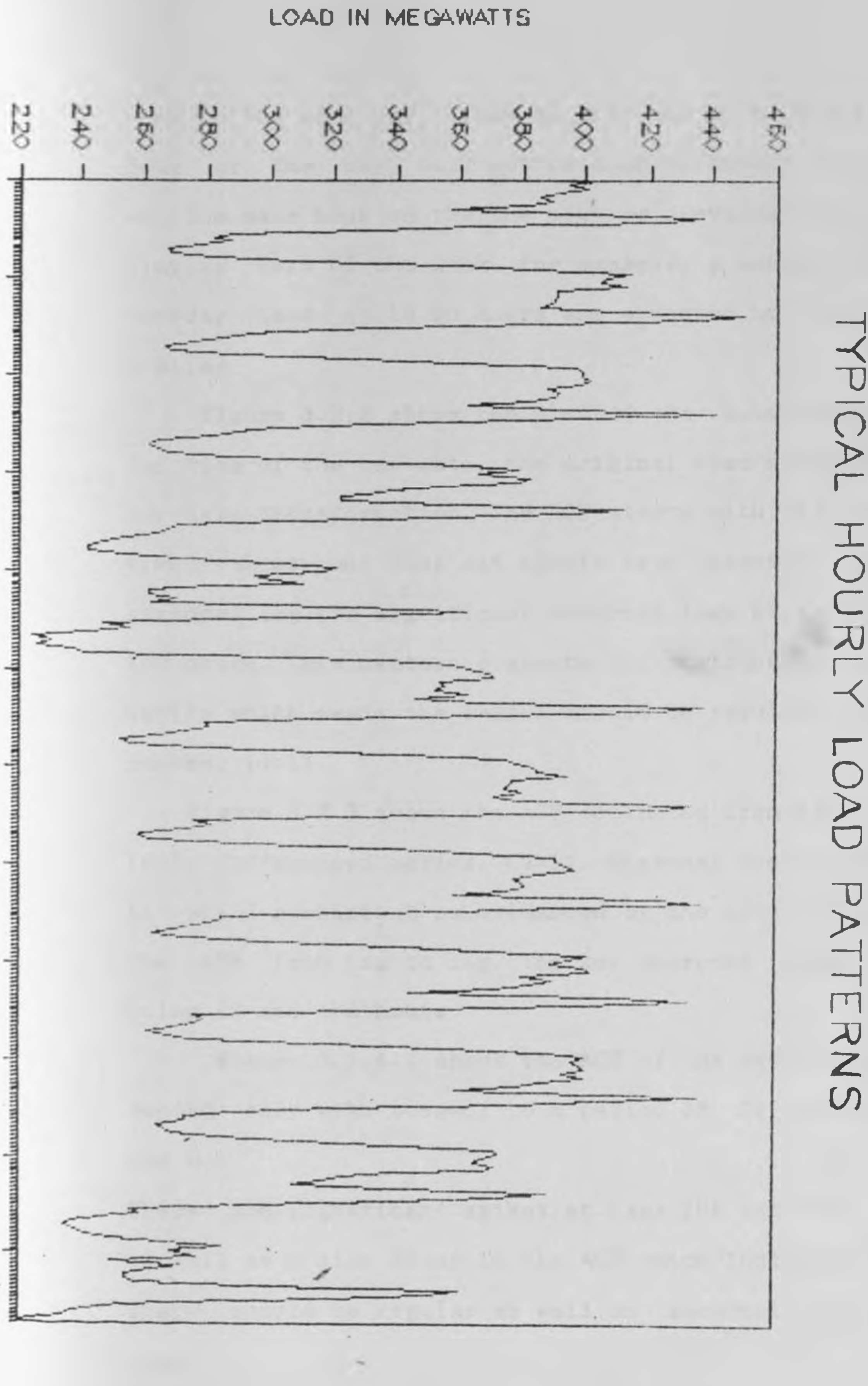

TYPICAL HOURLY LOAD PATTERNS
LOAD IN MEGAWATTS
220
240
260
280
300
320
340
360
380
400
420
440
460
1
25
49
73
97
121
145
169
193
217
241
265

Figure 3.2.1

week on the same day, likewise the load at a particular hour of the day will not be much different from that of the same hour on the the next or previous day, for similar days of the week, for example, a monday and a tuesday load at 18.00 hours are expected to be quite similar.

Figure 3.2.2 shows the plot of the autocorrelation function of the raw data, the original load series before any data transformation. The ACF starts with high positive values and dies out slowly from seasonal lag to seasonal lag, the significant seasonal lags being 24 and 168 hours. This pattern suggests non-stationarity in the series which means the series should be regularly differenced, (d=1).

Figure 3.2.3 shows the ACF estimated from the regularly differenced series, (d=1). Seasonal non-stationarity is still persistent as evidenced by the slow decay of the ACF from lag to lag. The key seasonal lags still being 24 and 168 hours.

Figure 3.2.4.1 shows the ACF of the series differenced only with respect to a period of 24 hours, s=24 and D=1.

There are significant spikes at lags 168 and 336 hours as well as a slow decay in the ACF which indicates that there should be regular as well as seasonal differencing.

ACCFS

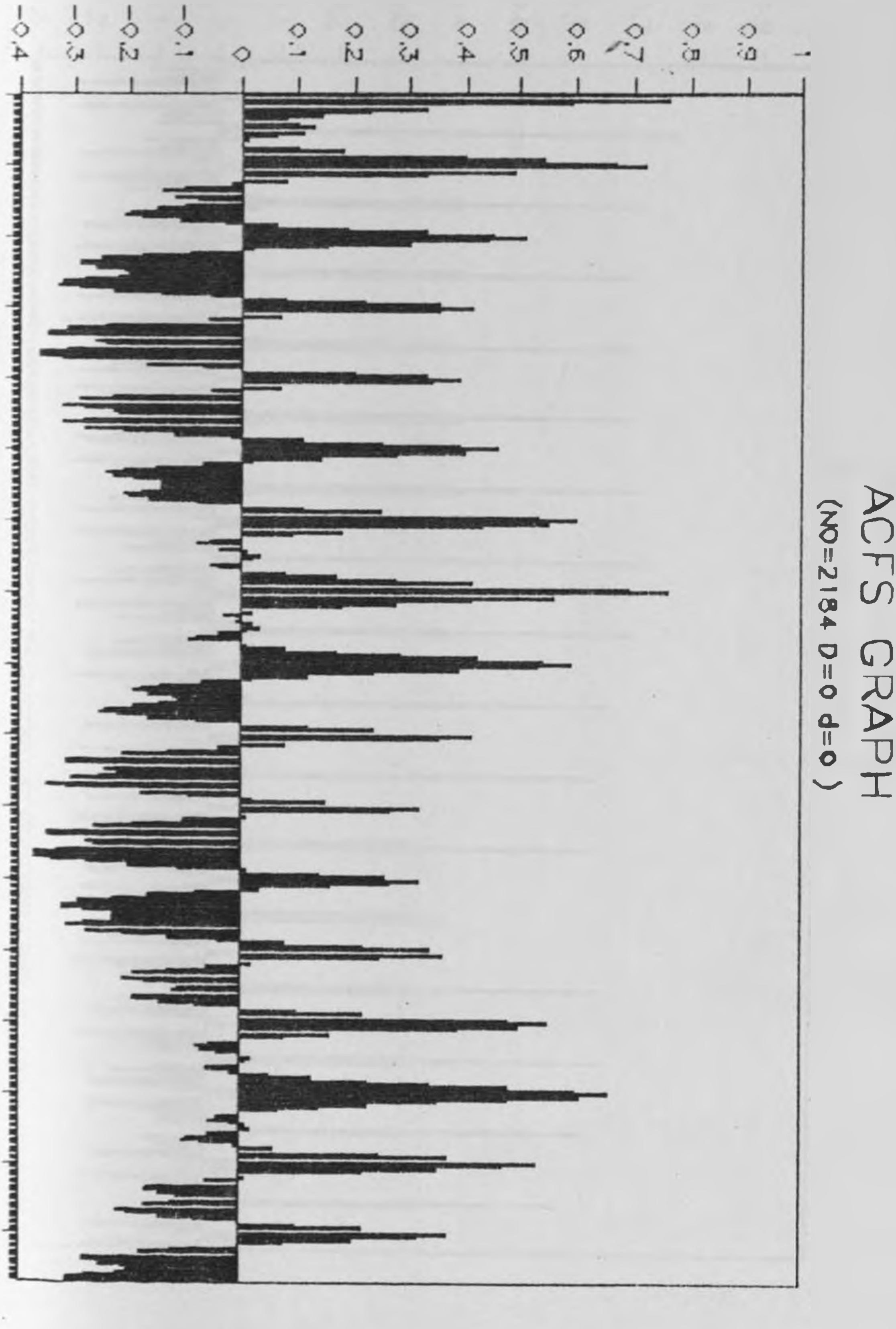


Figure 3.2.2.

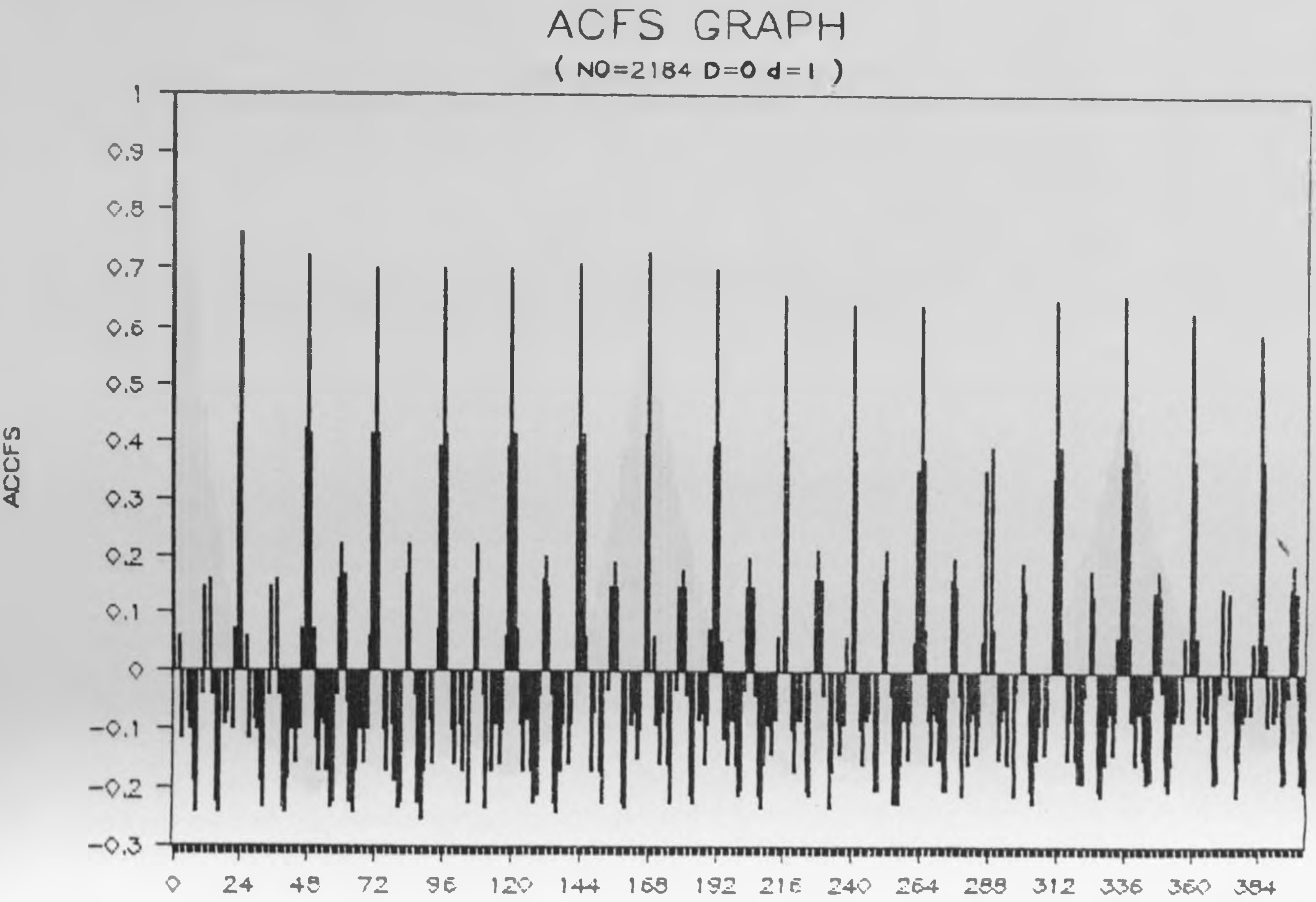

ACFS GRAPH
(NO=2184 D=0 d=1)
ACCFS
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
-0.1
-0.2
-0.3
0 24 48 72 96 120 144 168 192 216 240 264 288 312 336 360 384

Figure 3.2.3.

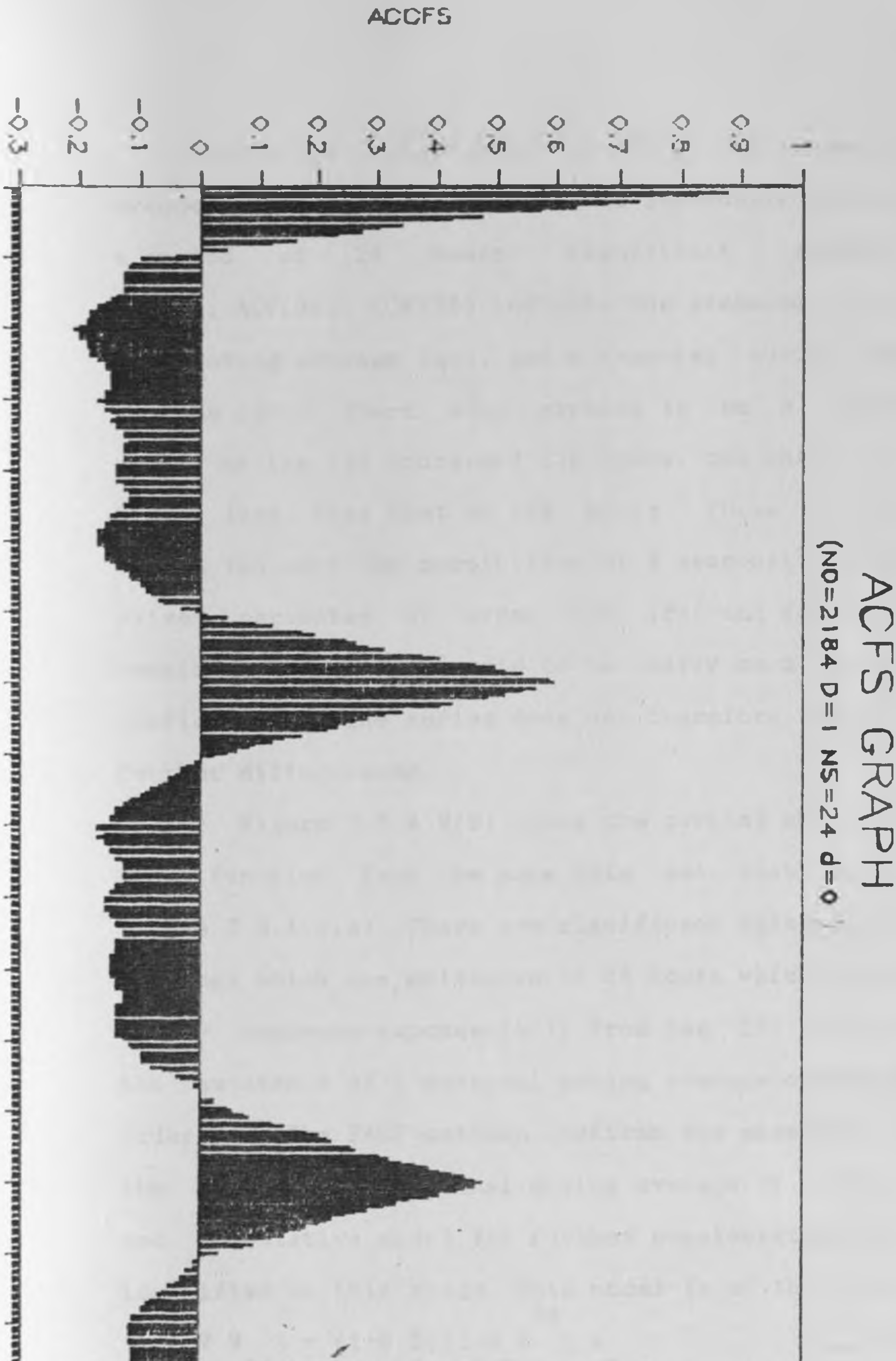

ACFS GRAPH
(NO=2184 D=1 NS=24 d=0)
ACCFS
-0.3 -0.2 -0.1 0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1
0 24 48 72 96 120 144 168 192 216 240 264 288 312 336 360 384

Figure 3.2.4.1.

Figure 3.2.4.2(a) shows the ACF of the series differenced regularly (d=1) as well as seasonally (D=1), with a period of 24 hours. Significant spikes at ACF(1), ACF(24), ACF(25) indicate the presence of a regular moving average (q=1) and a seasonal moving average process (Q=1). There also appears to be a significant spike at lag 168 hours and 336 hours, the value at 336 being less than that at 168 hours. These additional spikes indicate the possibility of a seasonal autoregressive parameter of order one, (P=1 and s=168). The remaining lags can be said to be nearly zero with 95% confidence and the series does not therefore require any further differencing.

Figure 3.2.4.2(b) shows the partial autocorrelation function from the same data set that generated figure 3.2.4.2(a). There are significant spikes at seasonal lags which are multiples of 24 hours which progressively decrease exponentially from lag 24, confirming the existence of a seasonal moving average operator of order one. The PACF pattern confirms the assertion that the process is a seasonal moving average of order one and a tentative model for further consideration can be identified at this stage. This model is of the form;

$$\nabla \nabla_{24} Z_t = (1-\Theta_1 B)(1-\Theta_2 B^{24}) a_t \qquad \text{\_\_\_\_3.2.1}$$

Figure 3.2.5.1 shows the ACF of the series differenced only with respect to a period of 168 hours, (D=1 and

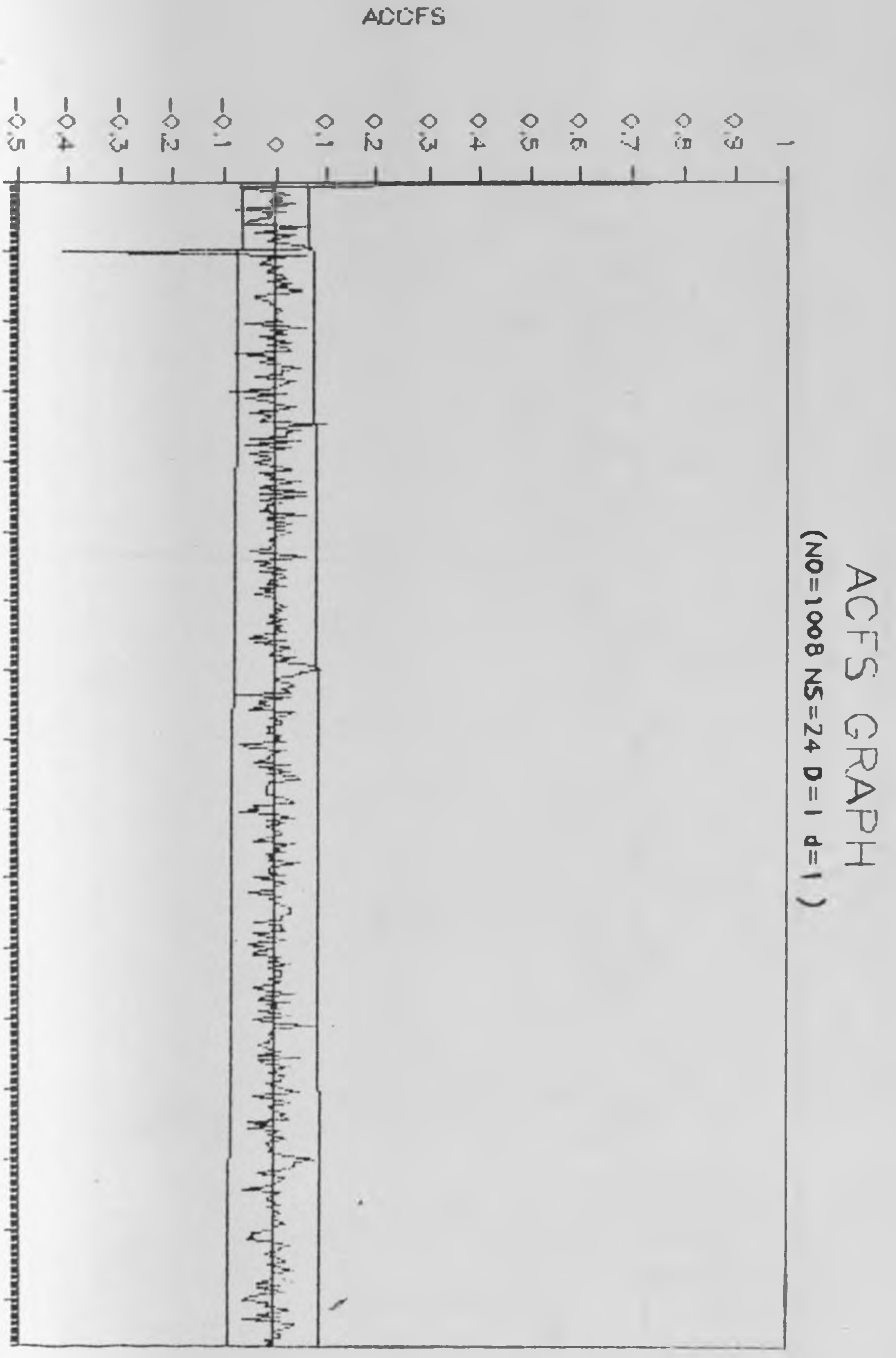


Figure 3.2.4.2 (a)

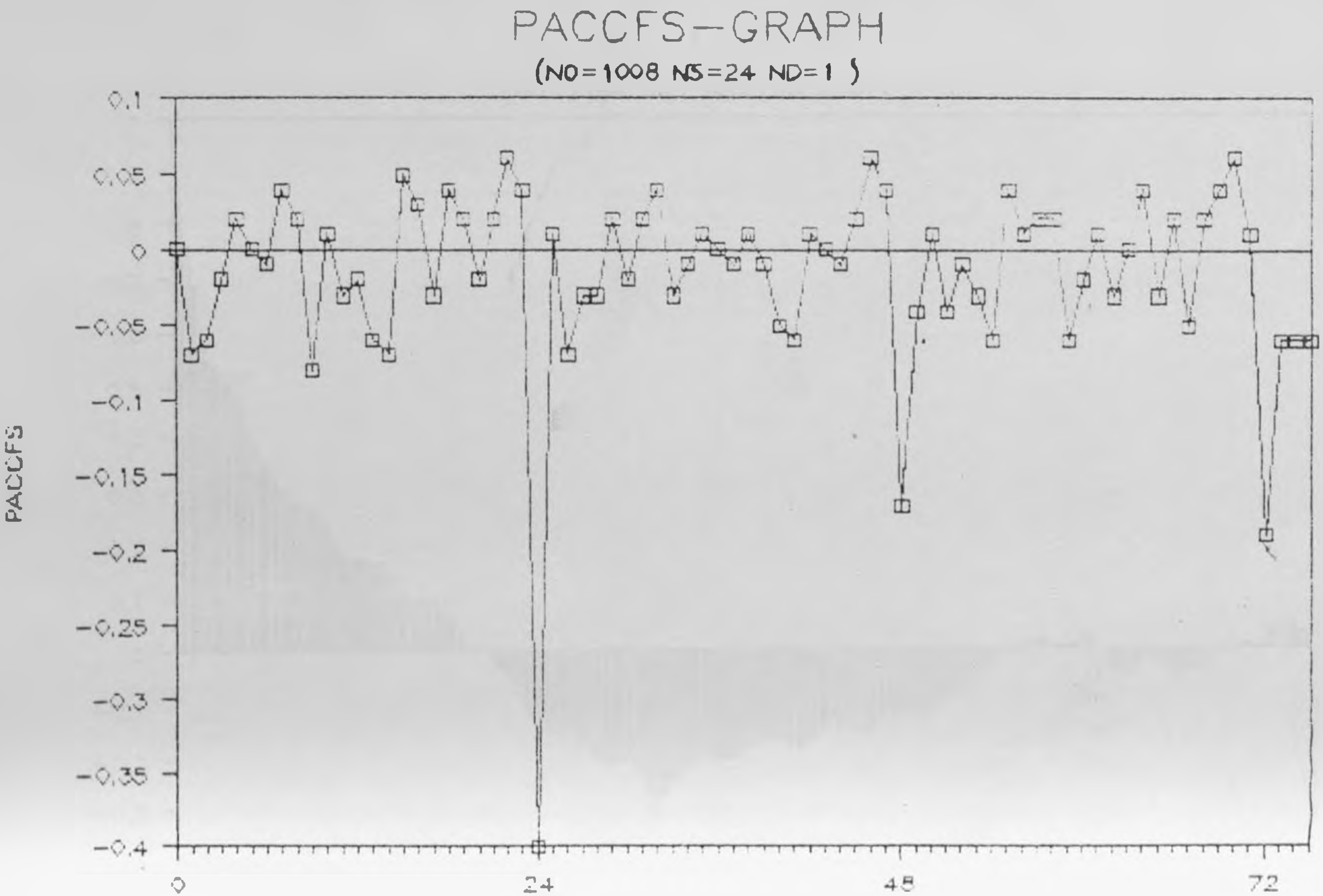


Figure 3.2.4.2. (b)

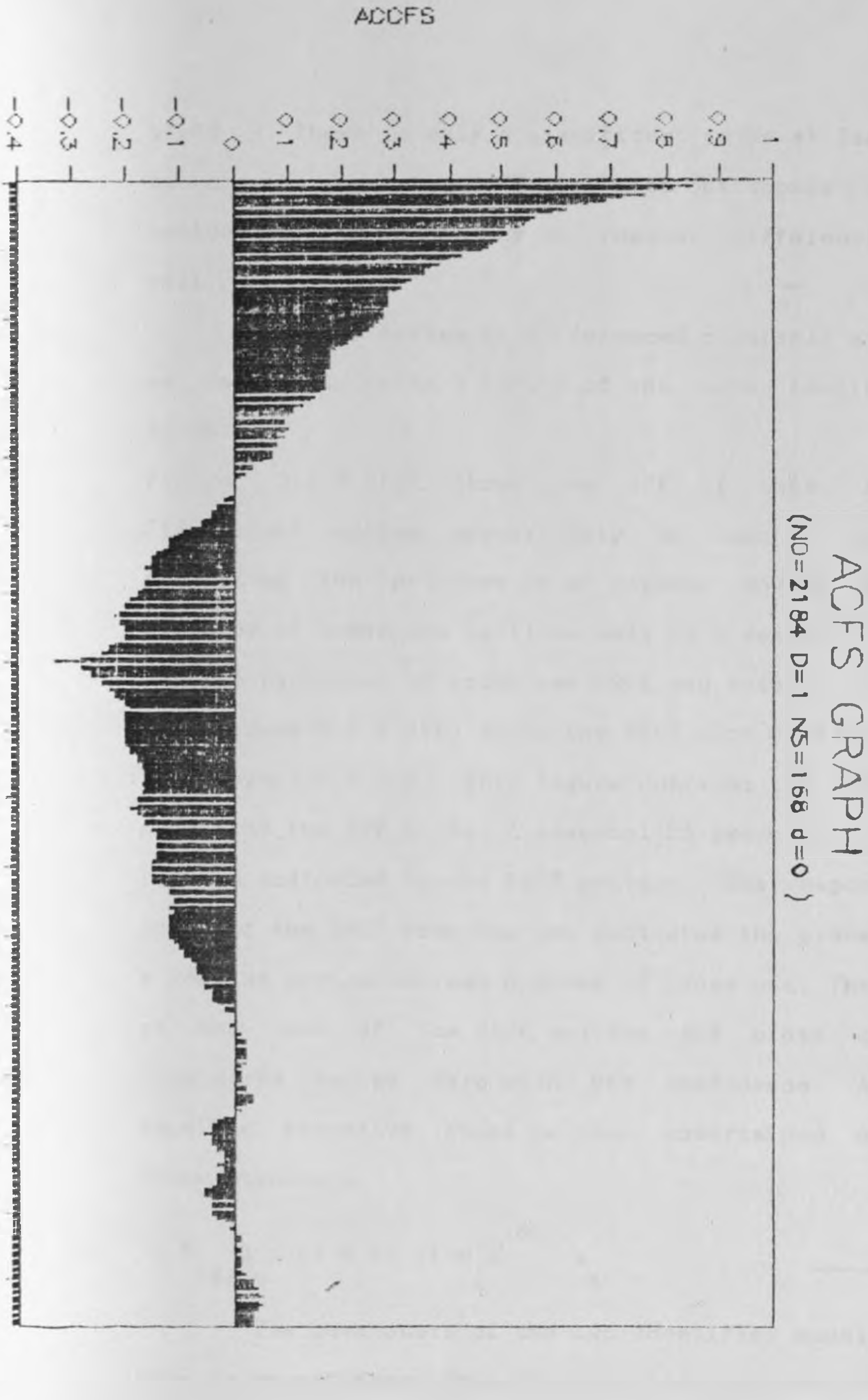


Figure 3.2.5.1.

$s=168$). There is only a significant spike at lag 168 hours, otherwise the ACF also dies out slowly to zero indicating the necessity of regular differencing as well.

The load series is differenced regularly as well as seasonally with a period of one week, ($d=1, D=1$ and $s=168$).

Figure 3.2.5.2(a) shows the ACF of this series. Significant spikes appear only at lags 1 and 168 suggesting the presence of a regular moving average operator of order one ($q=1$) as well as a seasonal moving average parameter of order one ($Q=1$ and $s=168$).

Figure 3.2.5.2(b) shows the PACF plot corresponding to figure 3.2.5.2(a). This figure confirms the deductions made from the ACF plots. A seasonal MA process of order one is indicated by the PACF pattern. The exponential decay of the PACF from lag one indicates the presence of a regular moving average process of order one. The rest of the lags of the PACF and the ACF plots can be considered to be zero with 95% confidence. Another possible tentative model is thus entertained of the below structure;

$$\nabla \nabla_{168} Z_t = (1-\theta_1 B)(1-\theta_2 B^{168}) a_t \qquad \text{\_\_\_\_\_3.2.2}$$

The parameters of the two identified models then have to be estimated from the data and diagnostic checks

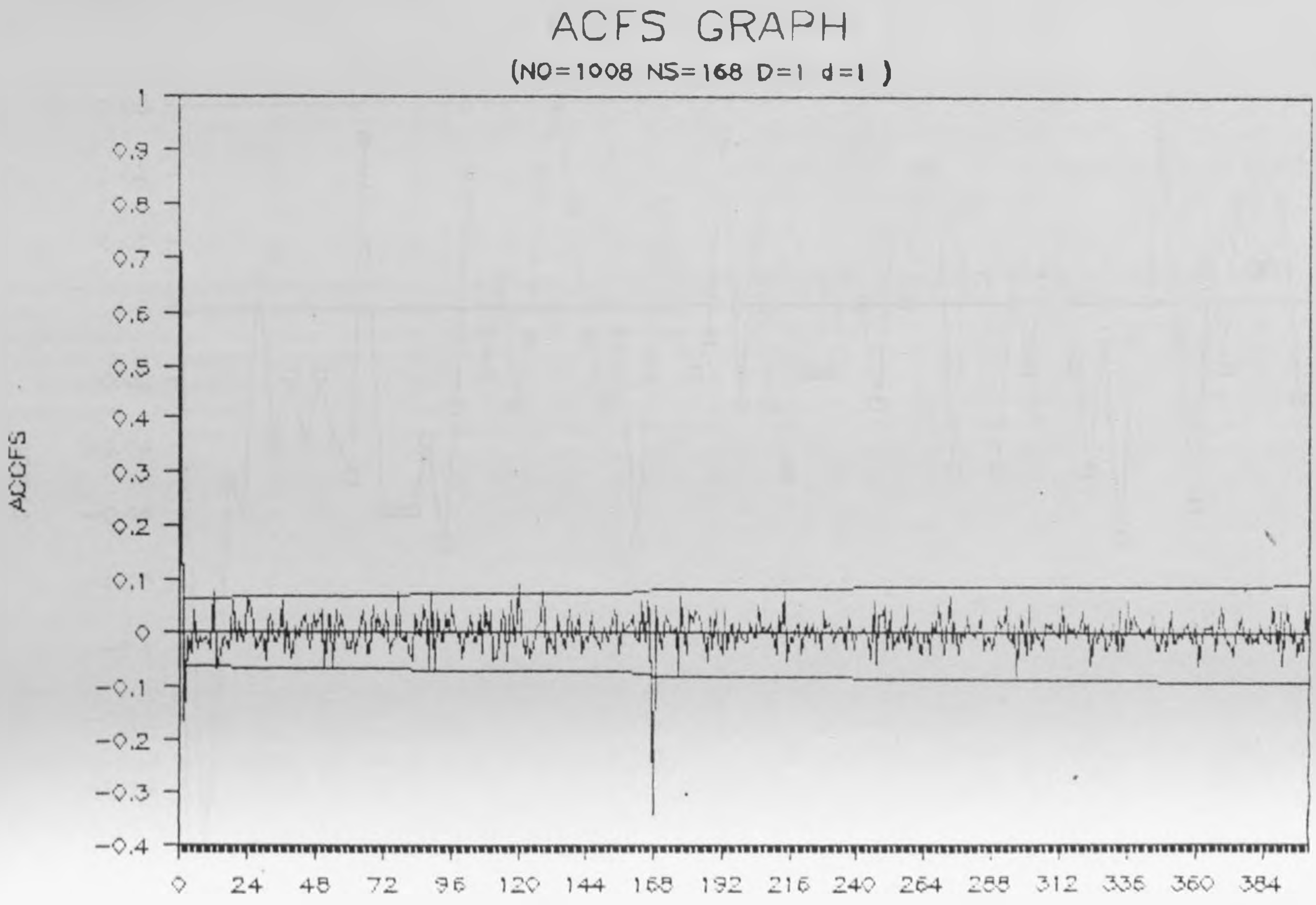


Figure 3.2.5.2 (a)

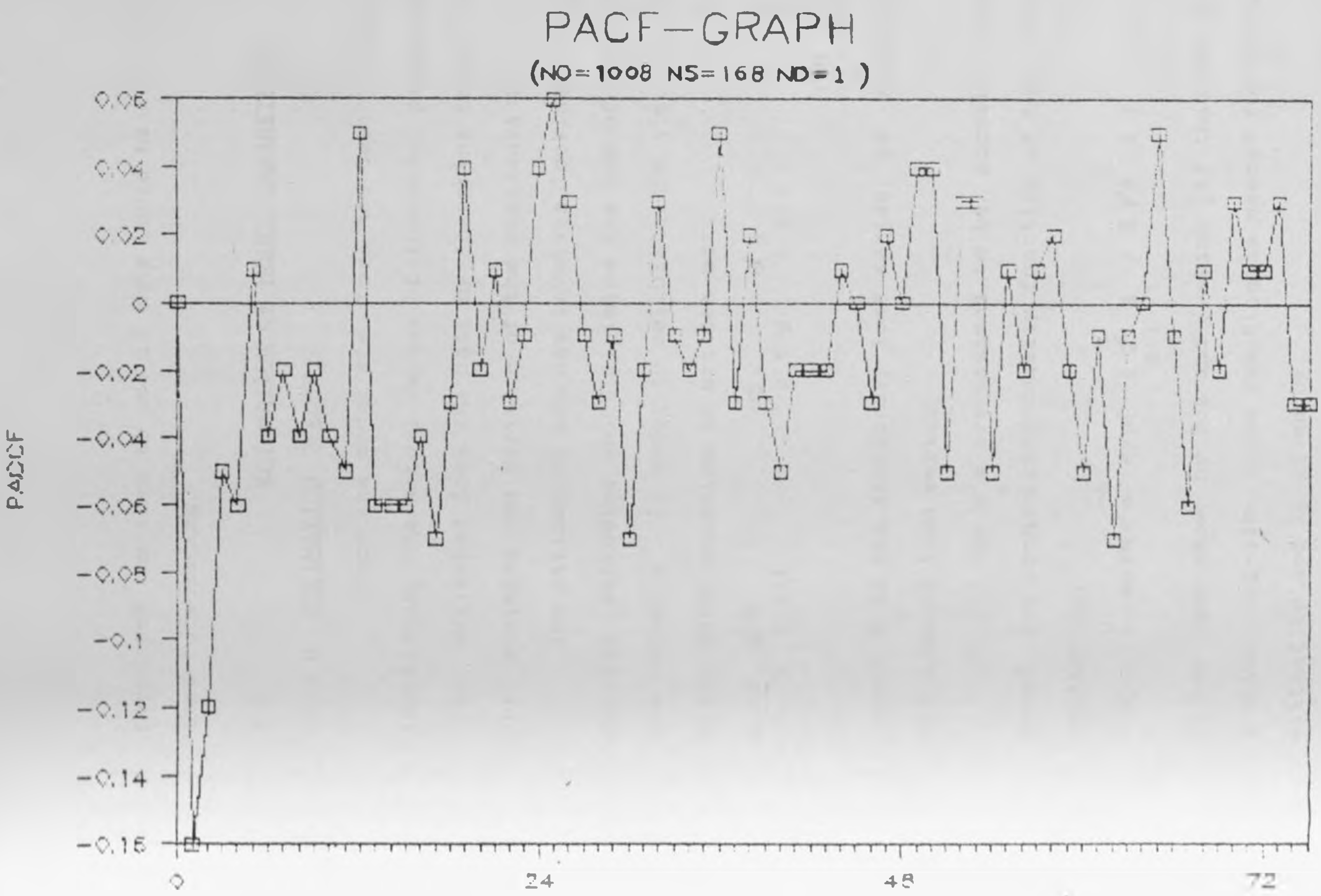


Figure 3.2.5.2 (b)

performed on them to see if they could be adequate for load forecasting.

3.3 ESTIMATION OF MODEL PARAMETERS.

3.3.0 ESTIMATION THEORY.

Once the model structure has been tentatively identified, the actual values of the model parameters are then estimated from the data by searching those values that minimise the variance of the residuals (a_t).

The estimation process involves looking for least squares estimates which minimise the sum of squares of the noises a_t. In order to calculate the a_t's, the general mixed ARIMA equation is written as:

$$a_t = W_t - \phi_1 W_{t-1} - \dots\dots\dots - \phi_p W_{t-p} + \theta_1 a_{t-1} + \dots\dots\dots + \theta_q a_{t-q} \qquad \text{\_\_\_\_\_\_3.0.1}$$

where W_t is the stationary load series, or appropriately differenced load series.

Since the a_t's are assumed to be normally distributed, the probability density function of the a_t's can be written as;

$$P(a1,\dots,an) \propto \sigma a^{-n} \exp\{ -(\sum_{t=1}^{n} a_t^2 / 2\,\sigma a^2) \} \qquad \text{\_\_\_\_\_\_3.0.2}$$

It has been shown by Box and Jenkins [4] that the unconditional log-likelihood function is needed for parameter estimation and is given by;

$$L(\phi,\theta,\sigma a) = f(\phi,\theta) - n \operatorname{Ln} \sigma a - S(\phi,\theta)/2\sigma a^2 \qquad \text{\_\_\_\_\_\_3.0.3}$$

where $f(\phi,\theta)$ is a function of ϕ and θ. The unconditional sum of squares function is given by;

$$S(\phi,\theta)=\sum_{t=1}^{n}[a_t]^2 \qquad 3.0.4$$

where $[a_t]=E[a_t|\phi,\theta,w]$ denotes the expectation of a_t conditional on ϕ,θ and w.

Normally $f(\phi,\theta)$ is of importance only when the number of terms in the series (n) is small. When n is large $S(\phi,\theta)$ dominates the log-likelihood function and it follows that the parameter estimates which minimise the sum of squares will be a close approximation to the maximum likelihood estimates.

In order to calculate the unconditional sum of squares it is necessary to estimate the values of $W_o, W_1, ...W_{-Q}$ of the series which occurred before the first observation of the series was made.
This enables the starting off of the difference equation 3.0.1 .

To facilitate this backward estimation ,the forward form of the general model equation is introduced where all B's are replaced by F's, $(F=B^{-1})$.

$$\phi(F) = \theta(F)\ e_t \qquad \text{or}$$

$$W_t - W_{t+1} -......- W_{t+p} = e_t - \theta e_{t+1} -.....\theta q\ e_{t+q} \qquad 3.0.5$$

where $\{e_t\}$ is a sequence of independently distributed normal random variables. This process is a stationary representation where W_t is expressed in terms of W's and e's and this model and the backward model of equation 3.0.1 have identical probability structures.

With the forward shift operator, it is possible to use equation 3.0.5 to estimate W's which occurred prior to the first observation. To calculate the unconditional sum of squares $S(\phi,\theta)$ for any given set of parameters ϕ and θ, equation 3.0.5 is first used to estimate the W's prior to the start of the series, then these initial values are used with equation 3.0.1 to estimate the $[a_t$'s$]$ and finally the $[a_t$'s$]$ are summed to obtain $S(\phi,\theta)$.

3.3.1 ESTIMATION METHOD.

Model estimation is an optimization process that requires a suitable software package. The general ARIMA model is non-linear in it's parameters, so standard software regression packages such as SPSS cannot be used. There are several mathematical optimization methods which can be used for non-linear least squares estimation.

The method to be used in any optimization problem will depend on the nature of the problem. If the problem is formulated mathematically in an analytical form, the method chosen will depend on whether;

1) it is a static or dynamic optimization process
2) the performance function is constrained or not
3) the objective function is linear or non-linear

4) the function is single variable or multi-variable.

In this work static optimization will be used. In principle, static systems are those whose parameters do not change with time, however, systems whose parameters vary slightly within a reasonable range of time will also be considered static. It is also a non-linear formulated problem, with more than one parameter to be estimated.

The methods which have been successfully used include the Marquardt algorithm, conjugate gradient method of Fletcher and Reeves, Hookes and Jeeves optimization method, Descent method of Fletcher and Powell among others [1]. All these methods and their algorithms are fully discussed in any standard optimization mathematics text [11].

In this work, the method of Hookes and Jeeves [12] has been chosen because it is easier to understand and program and its computer memory requirements are less, a factor that is of considerable importance since this work is being developed for use on a personal computer. It also meets the conditions stipulated above for solving a non-linear, multivariable, least squares formulated optimization problem.

The model estimation program using Hookes and Jeeves optimization algorithm was developed in standard Fortran 77 language and is flexible enough to estimate for any number of model parameters, although as the number of parameters increase naturally the computation time also goes up.

This estimation program also generates the residual series $\{a_t\}$ corresponding to the optimum parameter values. The generated residual series $\{a_t\}$ is used as the input to another program which performs diagnostic checks to test for model adequacy, by calculating the autocorrelation function of the model residuals as well as evaluating the X^2 statistic for the fitted models.

3.3.1.1 HOOKES AND JEEVES ALGORITHM.

This algorithm finds the minimum of a multivariable, unconstrained function. The procedure is based on the direct search method proposed by Hookes and Jeeves [12]. The algorithm proceeds as follows;

1) A base point is picked using the autocorrelation function coefficients as estimates and the objective function evaluated.

2) Local searches are made in each direction of steps Xi, for each parameter value and then evaluating the objective function to see if a lower function value is obtained.

3) If there is no function decrease, the step size is reduced and searches are made from the previous best point.

4) If the value of the objective function has decreased, a "temporary head", $X_{i,o}^{k+1}$, is located using the two previous base points X_i^{k+1} and X_i^{k};

$$X_{i,o}^{k+1} = X_i^{k+1} + a(X_i^{(k+1)} - X_i^{(k)})$$

where i is the variable index = 1,2,3,...,N

o denotes the temporary head

k is a stage index (a stage is the end of N searches)

a is the acceleration factor, $a \geq 1$.

5) If the temporary head results in a lower function value, a new local search is performed about the temporary head, a new head is located and the value of the function, F is checked. This process continues so long as F decreases.

6) If the temporary head does not result in a lower function value, a search is made from the previous best point.

7) The procedure terminates when a convergence criterion is satisfied (e.g when change in F is less than a convergence factor).

For example in the evaluation of the optimum model parameters of equation 3.2.2 (pp36) using 1008 hours of load data, initial parameter estimates of 0.2 and 0.4 obtained from the ACF were used as the base point. These converged to optimum values of 0.06 and 0.85 respectively after 165 iterations in 1.5 minutes, with a change in value of the function being less than 0.000001.

Figure 3.3.1.1 Below is the block diagram for Hookes algorithm

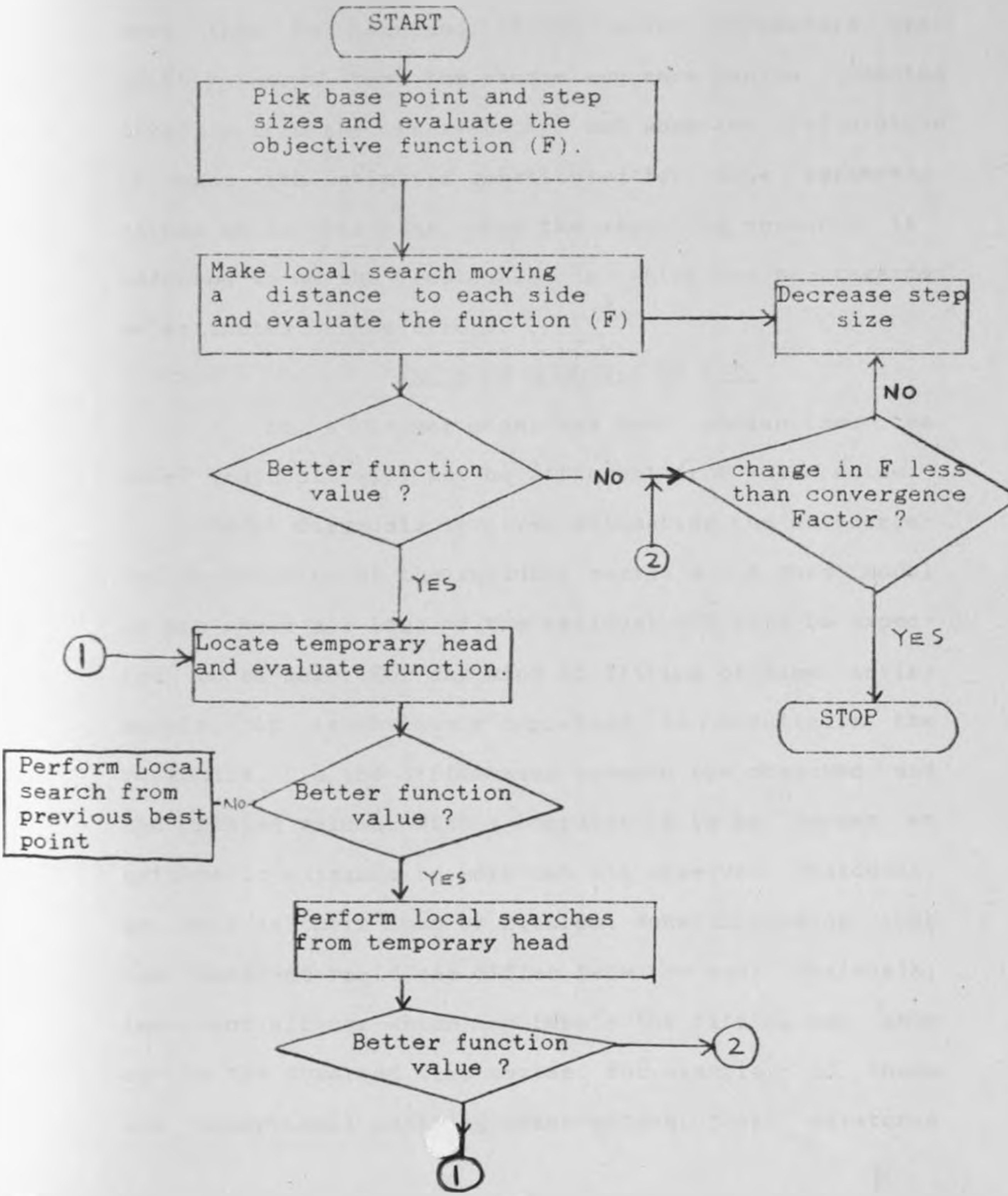


Fig. 3.3.1.1.

3.3.2 PARAMETER ESTIMATES AND DIAGNOSTIC CHECKS.

Once a model has been satisfactorily identified and its optimal parameters obtained, the adequacy of fit must then be assessed. If the model parameters are exactly known, then the random sequence can be computed directly from the observations, but when the calculation is made with estimates substituted for true parameter values as in this case, then the resulting sequence is referred to as the 'residuals', a_t which can be regarded as estimates of the errors.

3.3.2.1 TESTS OF GOODNESS OF FIT.

If a proper model has been chosen, then the model residuals will not be different from white noise.

Model diagnosis involves estimating the autocorrelation function of the residual series a_t. A good model is one where all lags of the residual ACF will be expected to be zero. For any kind of fitting of time series models, it is obviously important to scrutinize the residuals, i.e the differences between the observed and the fitted values. With a computer it is no longer an arithmetic nuisance to work out all observed residuals, as well as their sums of squares. Notwithstanding that the observed residuals differ from the real residuals, important effects which may impair the fitting may show up in the observed time series. For example, if there are exceptional outlying observations, their existance will be revealed by a large residual term.

One of the tests of goodness of fit is the Portmanteau lack of fit test. This uses the Q statistic to test whether the entire residual ACF is different from that expected of a white noise process [4].

If a fitted model is correct, then

$$Q=n\sum_{i=1}^{k}[ACF(i)]^2 \qquad \_\_\_\_\_\_\_3.3.2.1$$

is approximately distributed as X^2 (Chi squared) distribution with (k-P-Q-p-q) degrees of freedom, where n is the number of observations used to fit the model and ACF(i) is the autocorrelation function of the model residuals at lag i.

Another test for goodness of fit is the evaluation of the autocorrelation function of the residual series and their 95% confidence limits. The sequence a_t is white noise with 95% confidence if

$$[ACF(k)] \leq 2\sqrt{1/N(1+2\sum_{i=1}^{k}[ACF(i)]^2)} \qquad \_\_\_\_\_\_3.3.2.2$$

Estimates of the residual ACF which lie within plus or minus two standard error of confidence are thus not statistically different from zero at a 0.95 level of confidence.

3.3.2.2 FITTED MODELS.

The models $\nabla\nabla_{168} Z_t=(1-0.06B)(1-0.85B^{168})\, a_t$ ____3.3.2.2.1

and $\nabla\nabla_{24} Z_t=(1-0.2B)(1-0.9B^{24})\, a_t$ _____3.3.2.2.2

were tested for goodness of fit after having been tentatively identified and their parameters estimated using Hookes and Jeeves method.

Figure 3.3.2.1 shows the ACF of the residuals from the model of equation 3.3.2.2.1 and figure 3.3.2.2 shows the ACF of the residuals of the model of equation 3.3.2.2.2. The chi-squared statistic (Q) was also evaluated for each model as well as the residual variance. Residual variance is obtained by dividing the minimum sum of squares function by the number of observations,n.

The below table shows some typical values of these parameters.

Model Type	Q statistic For n=840	Percentage points on distribution 2.5%	5%	Residual Variance	Degrees of Freedom
$\nabla\nabla_{168} Z_t = (1-\theta_1 B)(1-\theta_2 B^{168})a_t$	23.7	31.5	28.9	143.1	18
$\nabla\nabla_{24} Z_t = (1-\theta_1 B)(1-\theta_2 B^{24})a_t$	27.2	31.5	28.9	179.5	18

Table 3.3.2.1

Both these two models passed the diagnostic checks using the Q statistic criterion. A careful scrutiny of the autocorrelation function of the model residuals and observation of the 95% confidence lines over a time span

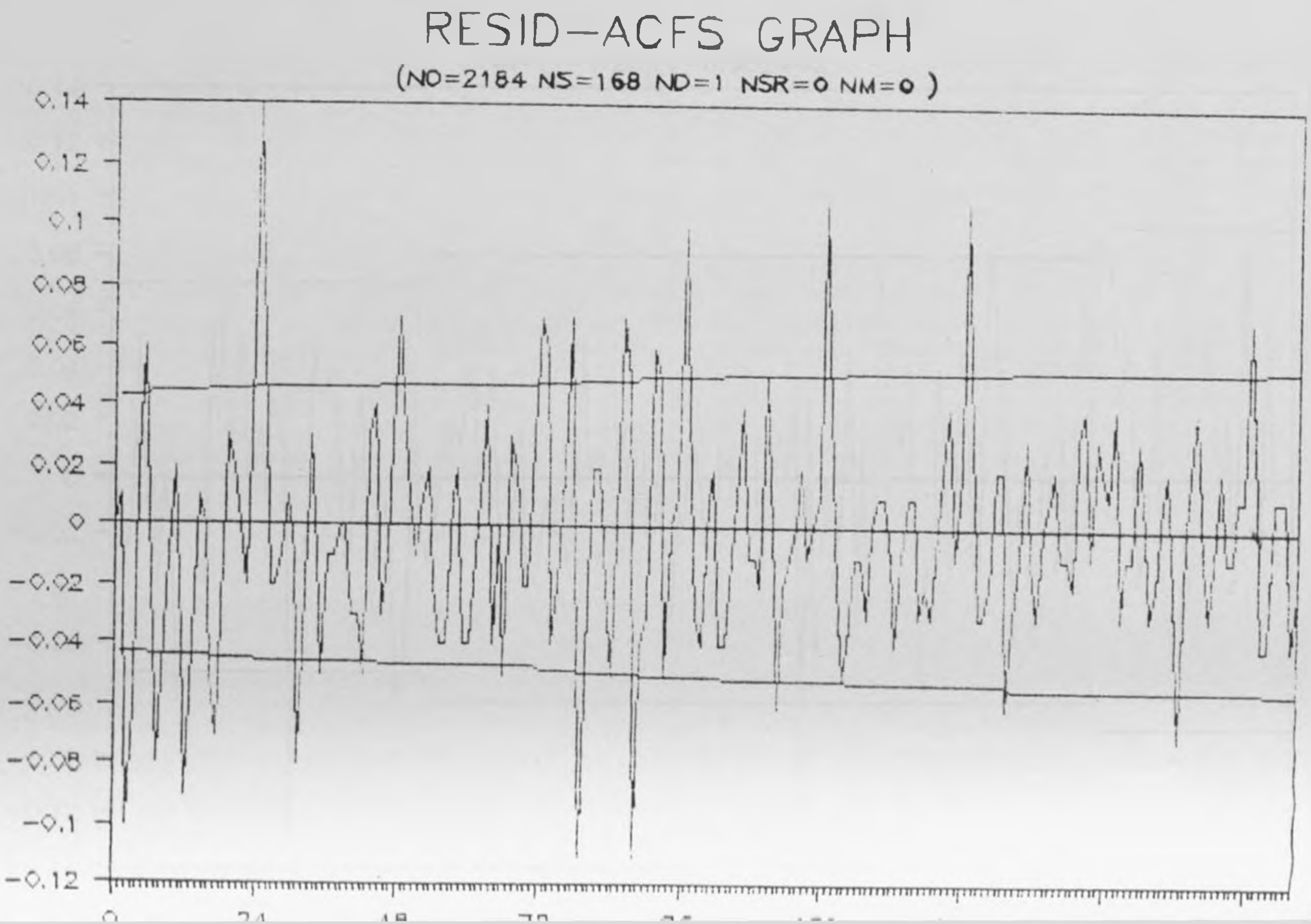

RESID-ACFS GRAPH
(NO=2184 NS=168 ND=1 NSR=0 NM=0)
ACCFS
0.14
0.12
0.1
0.08
0.06
0.04
0.02
0
-0.02
-0.04
-0.06
-0.08
-0.1
-0.12

Figure 3.3.2.1

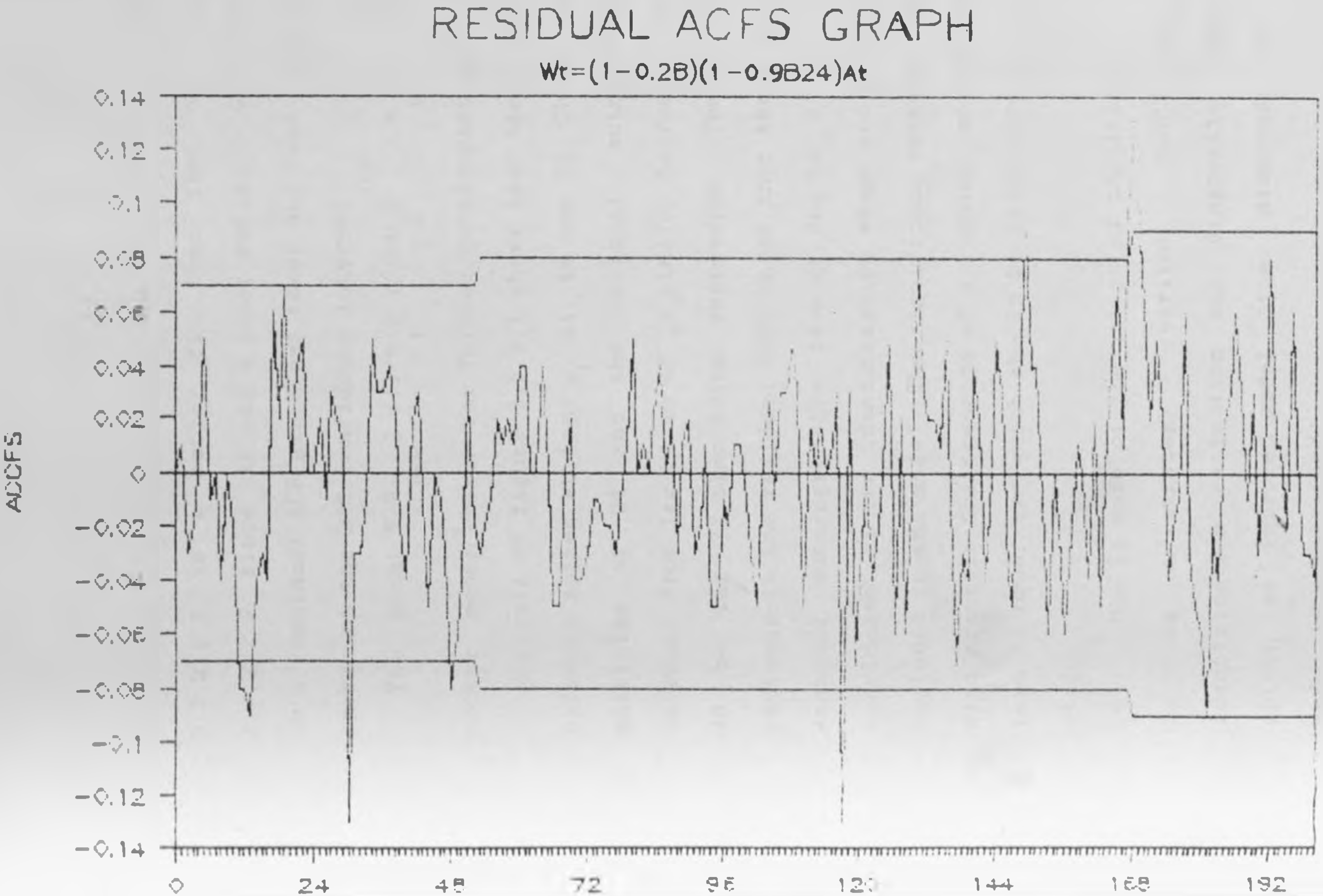

RESIDUAL ACFS GRAPH
Wt=(1-0.2B)(1-0.9B24)At
ACFS
0.14
0.12
0.1
0.08
0.06
0.04
0.02
0
-0.02
-0.04
-0.06
-0.08
-0.1
-0.12
-0.14
0
24
48
72
96
120
144
168
192

Figure 3.2.2.2

of about 200 lags reveals that the model of equation 3.3.2.2.1 is a better fit than that of equation 3.3.2.2.2 since it has a lower residual variance and it's residual ACF graph has fewer and less significant spikes at the 95% confidence interval.

The model $\nabla\nabla_{168}Z_t=(1-\theta_1 B)(1-\theta_2 B^{168})a_t$ is thus the better model chosen for further development and analysis.

Analysis of figure 3.3.2.1 shows that there are significant spikes at lags 2, 24, 48 and 72. These spikes are multiples of 24 and the residual autocorrelations decrease from lag 24 to 48 ,a similar decrease from lag 48 to lag 72 also being registered. These evolving patterns in the residual ACFS shows that there exists a seasonal autoregressive term of degree 24 which was overlooked at the identification stage and must now be included in the model. The significant spike at lag two also reveals the presence of a regular moving average term of order two which should be taken care of at this stage.

Use is made of the notion of the iterative model building strategy earlier mentioned of identification, estimation and diagnostic checks, to obtain an improved model. After diagnosis the below model is entertained;

$$(1-\phi_3 B^{24})\nabla\nabla_{168}Z_t=(1-\theta_1 B-\theta_4 B^{2})(1-\theta_2 B^{168})a_t \qquad \text{\_\_\_\_ 3.3.2.2.3}$$

The parameters of this new model were then evaluated using the parameter estimation program.

Figure 3.3.2.3 shows the autocorrelation function estimated from the model residuals of the model of equation 3.3.2.2.3.There are no significant spikes at the early or seasonal lags. The Q statistic for this model is (20.4), which is not significant at the 0.05 level,as can be seen from table F in appendix. Scrutiny of the residual ACFs graph also shows that there is no significant departure from zero.

The final model chosen is thus:

$$(1-\phi_3 B^{24})\nabla\nabla_{168} Z_t = (1-\theta_1 B-\theta_4 B^2)(1-\theta_2 B^{168})a_t . \qquad \_\_\_\_\_\_3.3.2.2.4$$

The model of equation 3.3.2.2.4 is the selected objective function for the Hookes and Jeeves optimization program. This model is then incorporated in the subroutine LEAST which is used by the main program for evaluating the optimum model parameters.

Five weeks of load data (840 hours) for different periods of the year was used in the evaluation of the model parameters as summarized in table 3.3.2.2.

It can be seen from the table that the model parameters vary within expected limits as the year is spanned indicating the fact that the model chosen and its parameters is quite a good representation of the load process. The average values for the optimum model

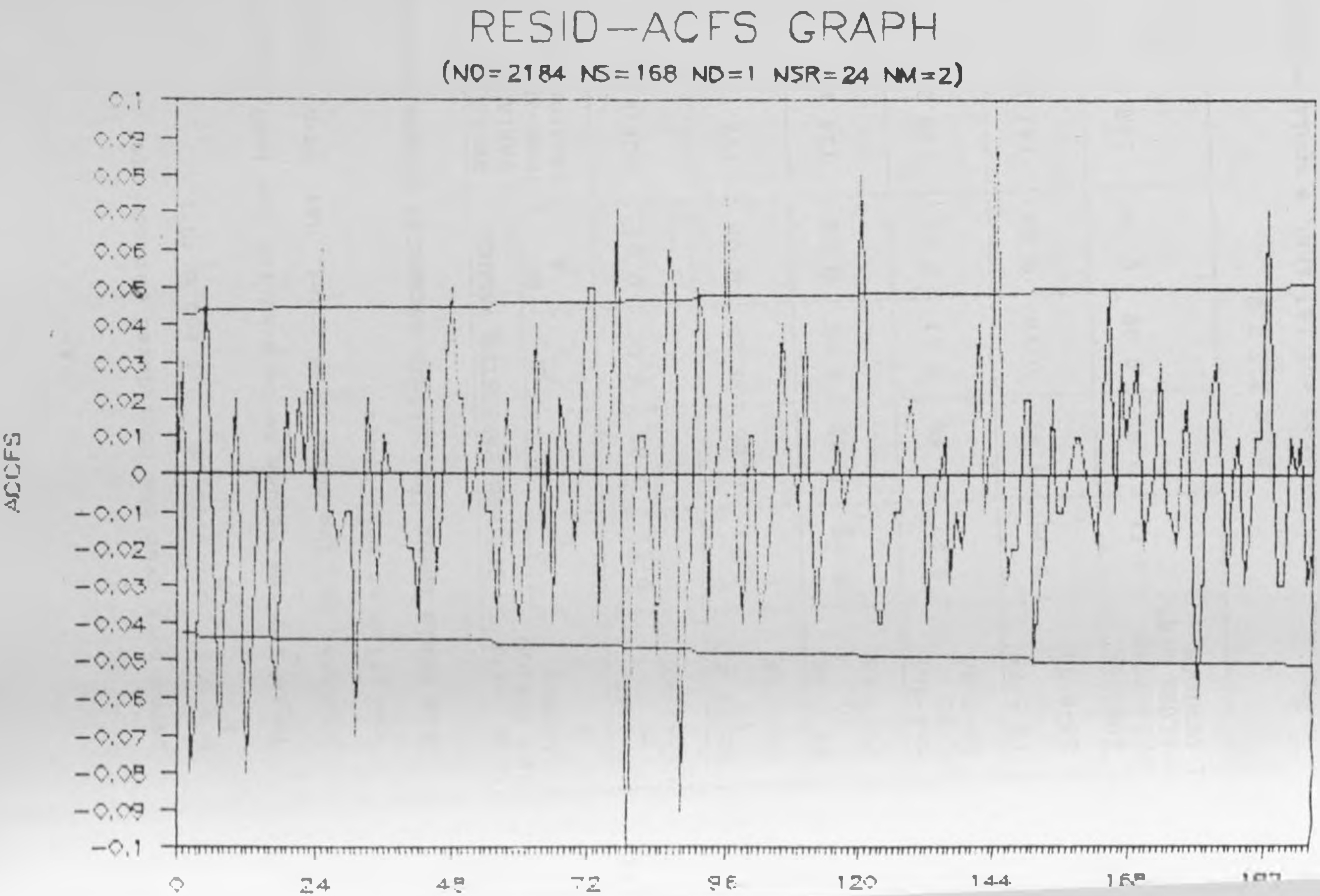

RESID—ACFS GRAPH
(NO=2184 NS=168 ND=1 NSR=24 NM=2)
ACCFS
0.1
0.09
0.08
0.07
0.06
0.05
0.04
0.03
0.02
0.01
0
−0.01
−0.02
−0.03
−0.04
−0.05
−0.06
−0.07
−0.08
−0.09
−0.1
0
24
48
72
96
120
144
168

Figure 3.3.2.3

parameters over the considered periods are:

$\theta_1 = 0.1$, $\theta_2 = 0.85$, $\phi_3 = 0.1$ and $\theta_1 = 0.1$

These are the values to be used for the load forecasting process as the power load model has been completely specified at this stage.

This table shows the optimum parameter values Evaluated.

LOAD DATA SET PERIOD (N=840)	OPTIMUM PARAMETER VALUES θ_1	θ_2	ϕ_3	θ_4	RESIDUAL VARIANCE (Sum of Squares /N)	CHI-SQUARED X^2 statistic (k=200)
1-1-87 TO 15-2-87	0.11	0.85	0.00	0.01	202.9	225.8
25-11-87 TO 10-1-88	0.03	0.85	0.23	0.05	110.1	195.4
24-2-88 TO 22-3-88	0.11	0.83	0.05	0.06	124.48	208.9
6-4-88 TO 10-5-88	0.27	0.89	0.01	0.07	168.3	176.6
18-5-88 TO 28-6-88	0.05	0.87	0.18	0.08	141.9	163.9
AVERAGE PARAMETER VALUES OVER PERIODS	0.11	0.85	0.09	0.06	149.5	194.1

Table 3.3.2.2

It has been pointed out [4] that a small variation in the value of the model parameters does not affect the accuracy of the load forecasting process as long as the right form of the model has been identified. For example a ten percent variation in the value of the parameters does not affect the forecasts appreciably, as shown in table 3.3.2.3 below.

ESTIMATED MODEL PARAMETERS FOR FORECASTING.	MEAN ABSOLUTE PERCENTAGE ERROR (24 HR) LEAD TIME FOR		
	OPTIMUM VALUE	10% INCREASE IN OPTIMUM PARAMETER VALUE	10% DECREASE IN OPTIMUM PARAMETER VALUE
$\theta 1$	3.6	3.59	3.62
$\theta 2$	3.6	3.84	3.45
$\phi 3$	3.6	3.60	3.61
$\theta 4$	3.6	3.60	3.61

Table 3.3.2.3.

The parameters could also be updated on-line if there becomes available faster algorithms for parameter estimation and on-line data acquisition methods. The parameters can be updated off-line periodically, to see whether they change appreciably as time goes on.

CHAPTER 4

FORECASTING.

4.0 INTRODUCTION.

One of the important aspects of a modeling process is to put the identified model to use. In this case, the identified power load model is to be used for load prediction. A good model is one that provides accurate forecasts of the load and the forecasting abilities of these models can be investigated by comparing actual load values with the forecasted values.

4.1 THE FORECASTING ALGORITHM.

The estimation stage of the model identification yields the model

$$(1-0.1B^{24})\nabla_1\nabla_{168}Z_t=(1-0.1B-0.1B^2)(1-0.85B^{168})a_t \qquad \text{4.1.1}$$

as the best amongst the ones considered and since it passed diagnostic checks, it is the final model to be used for forecasting.

Expanding the polynomial equation for the model and solving for Z_t, the load value at time t, we have

$$(1-0.1B^{24})(1-B)(1-B^{168})Z_t=(1-0.1B-0.1B^2)(1-0.85B^{168})a_t \qquad \text{4.1.2}$$

and further expansion gives

$$Z_t=Z_{t-1}+Z_{t-168}-Z_{t-169}+0.1(Z_{t-24}-Z_{t-25}+Z_{t-193}-Z_{t-192})+a_t$$

$$-0.1a_{t-1}-0.1a_{t-2}-0.85a_{t-168}+0.085a_{t-169}+0.085a_{t-170} \qquad \text{4.1.3}$$

Starting at a time t, the load forecast L hours ahead is given by

$$\hat{Z}(L)=Z_{t+L-1}+Z_{t+L-168}-Z_{t+L-169}+0.1(Z_{t+L-24}-Z_{t+L-25}+Z_{t+L-193}$$

$$+Z_{t+L-192})+a_{t+L}-0.1a_{t+L-1}-0.1a_{t+L-2}-0.85a_{t+L-168}$$

$$+0.085a_{t+L-169}+0.085a_{t+L-170}. \qquad \text{\_\_\_\_\_\_\_\_\_\_4.1.4}$$

where a_t is the one step ahead forecast error.thus

$$a_{t+L}=Z_{t+L}-\hat{Z}_{t+L-1}(1).$$

For example the forecast for a lead time of one hour would be given by

$$\hat{Z}_t(1)=Z_t+Z_{t-167}-Z_{t-168}-0.1(Z_{t-23}-Z_{t-24}-Z_{t-192}+Z_{t-191})-$$

$$0.1(Z_t-\hat{Z}_{t-1}(1))-0.1(Z_{t-1}-\hat{Z}_{t-2}(1))-0.85(Z_{t-167}-\hat{Z}_{t-168}(1))$$

$$+0.085(Z_{t-168}-\hat{Z}_{t-169}(1))+0.085(Z_{t-169}-\hat{Z}_{t-170}(1)). \qquad \text{\_\_\_4.1.5}$$

Forecast equations for other lead times can be similarly obtained by substituting the appropriate value of lead time L and time origin t into equation 4.1.4.

It can be seen from the forecast algorithm that at least 169 observations are required to start up the forecasting process because of the regular and seasonal differencing operations performed on the series (d=D=1 and s=168).

The seasonal moving operator also means that at least 170 one step ahead forecasting errors are required before an accurate forecast can be obtained. This in essence means that a load forecasting data base of at least 339 hours of load observations are required to obtain any meaningful load forecast. It was found that the load forecasting process stabilizes with about three weeks of hourly load data. Three weeks of load data (504 hours) was chosen as the size of the load forecasting data bank, since it is also important to store only the minimum amount of historical data at any time to minimize the computation time of the forecasts.

4.2 FORECASTING ERROR MEASUREMENT CRITERIA.

The forecasting error is the difference between the actual load and the predicted value.

Denoting the error by e, various measures can be defined;

1) Average error (e) = $1/n\ (e1 + e2 + \cdots + en)$ ____4.2.1

where n is the number of forecasts made.

2) Mean absolute error (MAE)=$1/n(|e1|+|e2|+\cdots+|en|)$__4.2.2

3) Mean square error (MSE)=$1/n(e1^2+e2^2+\cdots+en^2)$ _____4.2.3

4) Root mean square error(RMSE)=SQRT(MSE) _____4.2.4

5) Percentage error=(error (e)/actual load)x100 ___4.2.5

A good forecasting system should give small errors and a good error measurement criterion should be chosen. Some of these measures of the error are not quite suitable for this particular case, for instance the average of the errors (e) can give misleading results since large positive and negative errors can possibly cancel each other out.

One of the best measures of the forecasting error is the mean absolute error (MAE). Root mean square error is also an acceptable criterion. In this study mean absolute error measurement criterion is used for analysis of the forecasting results.

4.3 TREATMENT OF ANOMALOUS LOAD PATTERNS.

During some days of the year,the load model fails to describe the normal load. Such anomalous load patterns occur during public holidays such as christmas, easter, labour day, independence day and others that may be declared from time to time. These abnormal loads could also occur due to unforeseen circumstances like industrial strikes, power blackouts and other emergencies.

The large errors in a normal forecast of the holiday loads tends to distort the model temporarily causing further errors in the following normal days. Holiday loads are considerably lower than normal weekday loads but are mostly like neighbouring sunday loads.

Figure 4.3.1 shows a plot of typical sunday loads compared with typical holiday loads. The graphs clearly show that sunday loads do not differ appreciably from holiday loads.

An analysis of the load patterns has shown that loads of holidays are not necessarily the same,for exam-

TYPICAL SUNDAY & HOLIDAY LOAD CURVES

(+ –SUN., □– SUN., △ –HOLIDAY)

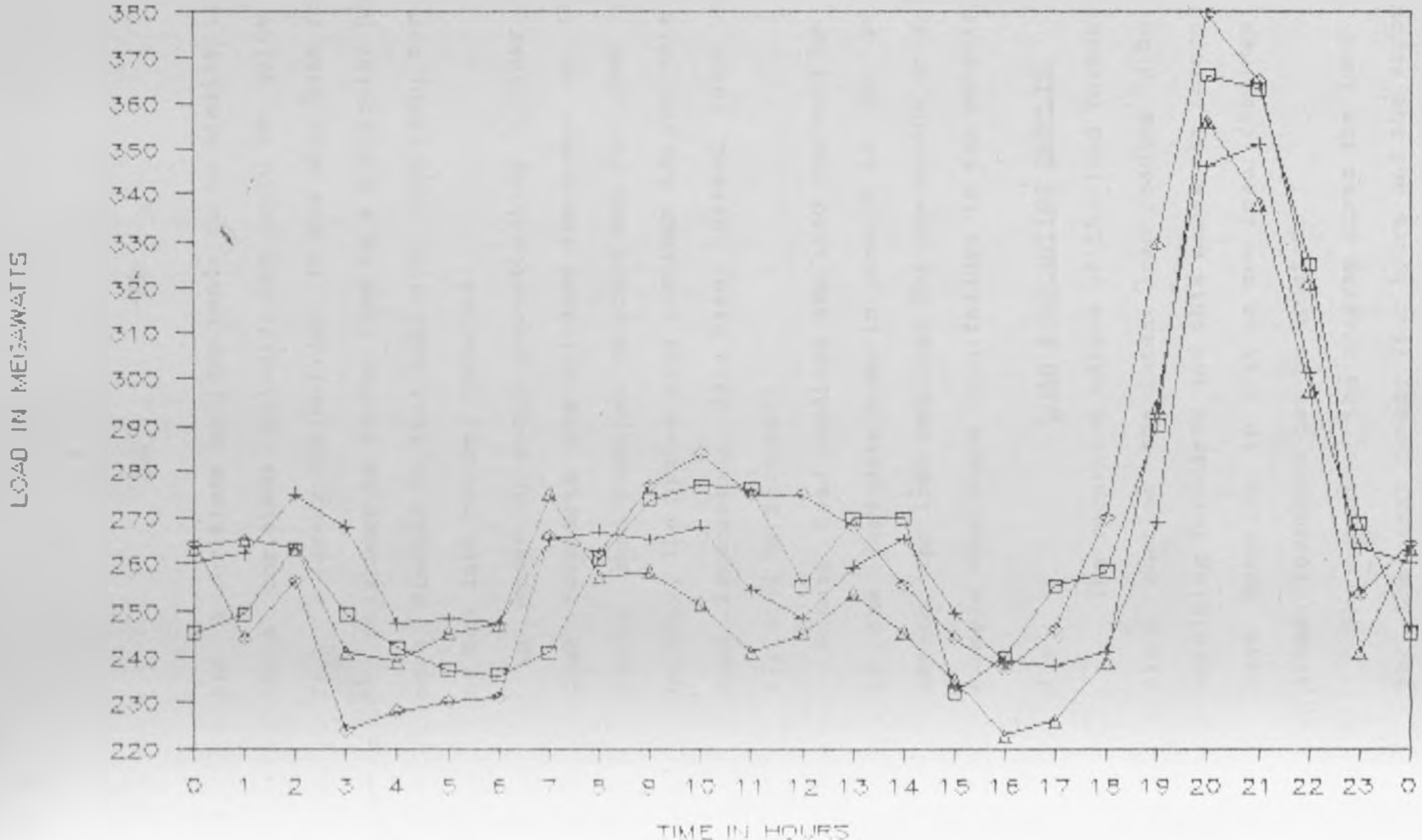


Figure 4.3.1

ple a christmas holiday tends to be similar to a previous christmas holiday, but could be quite different from a labour day holiday. It has also been found that the neighbouring sunday load to a particular holiday is more similar to that holidays load than, say the load of the last holiday observed.

In order to avoid overestimating holiday loads.the load forecasts for holidays are taken as the latest sunday load readings recorded and for that particular holiday, the system load readings are not entered in the load forecasting data base, instead these values are replaced by forecasts.

Actual load readings are also replaced by forecasts in the load data base in case it is not possible to obtain the load readings for one reason or another for example when there is a failure in the metering system.

4.4 LOAD FORECASTING RESULTS.

The recursive nature of the load forecasting algorithm enables the latest load reading to be used for obtaining forecasts and this makes the forecasting process adaptive in that as new load readings are obtained, forecasts can be updated.

The output of the program gives the load prediction for the next twenty four hours and the errors in the previous hour's forecasts.

One of the advantages of this model is that its inherent structure allows the forecasting of weekend loads with the same degree of accuracy as for weekdays.

For purposes of testing the adequacy of the models for forecasting, forecasts were evaluated for the year 1988 with the data being provided by the Kenya power utility. Forecasts for 1 hour, 2 hours, 4 hours, 12 hours and 24 hours lead times were evaluated over a number of three week periods, and the average mean absolute (MAE) error evaluated for each interval. Results of such forecast are shown in table 4.4.1.

Figure 4.4.1 (a) shows a typical forecasting result for 24 hour lead time forecasts over a period of six days, starting from a wednesday through to the next monday. Each forecast was evaluated at midnight.

Figure 4.4.1(b) shows the 24 hour lead time percentage forecasting errors over a two week period, the forecasting origin being midnight of each day.

Figure 4.4.2 (a) shows the forecasting results of one hour lead time forecasts over a six day period running from a wednesday through to monday, while figure 4.4.2 (b) shows the one hour lead time percentage forecasting errors over a two week period.

Figures 4.4.3, 4.4.4 and 4.4.5 show typical 24 hour lead time load forecasts for a typical weekday, saturday and sunday loads respectively.

AVERAGE MEAN ABSOLUTE PERCENTAGE FORECASTING ERRORS.

PERIOD DURING WHICH FORECASTS WERE MADE	LEAD TIME OF THE FORECASTS				
	1HR (MAE)	2HRS (MAE)	4HRS (MAE)	12HRS (MAE)	24HRS (MAE)
29th June to 2nd August 1988	2.87	3.18	3.50	3.52	3.60
18th May to 28th June 1988	2.84	3.10	3.53	3.52	3.71
24th Feb.to 29th March 88	2.82	2.97	3.30	3.94	4.36
THE AVERAGE MEAN ABSOLUTE ERROR FOR ALL THE PERIODS.	2.84	3.08	3.44	3.66	3.89

AVERAGE LOAD OVER THE WHOLE PERIOD = 328.8 MW

Table 4.4.1

From the above table it can be seen that all the mean absolute forecasting errors are well below five percent and it can also be observed that the errors slightly increase with an increase in lead time.

The errors for one step ahead load forecasts average to 2.8 percent showing the usefulness of updating the forecasts every hour to obtain a better forecast for the next hour.

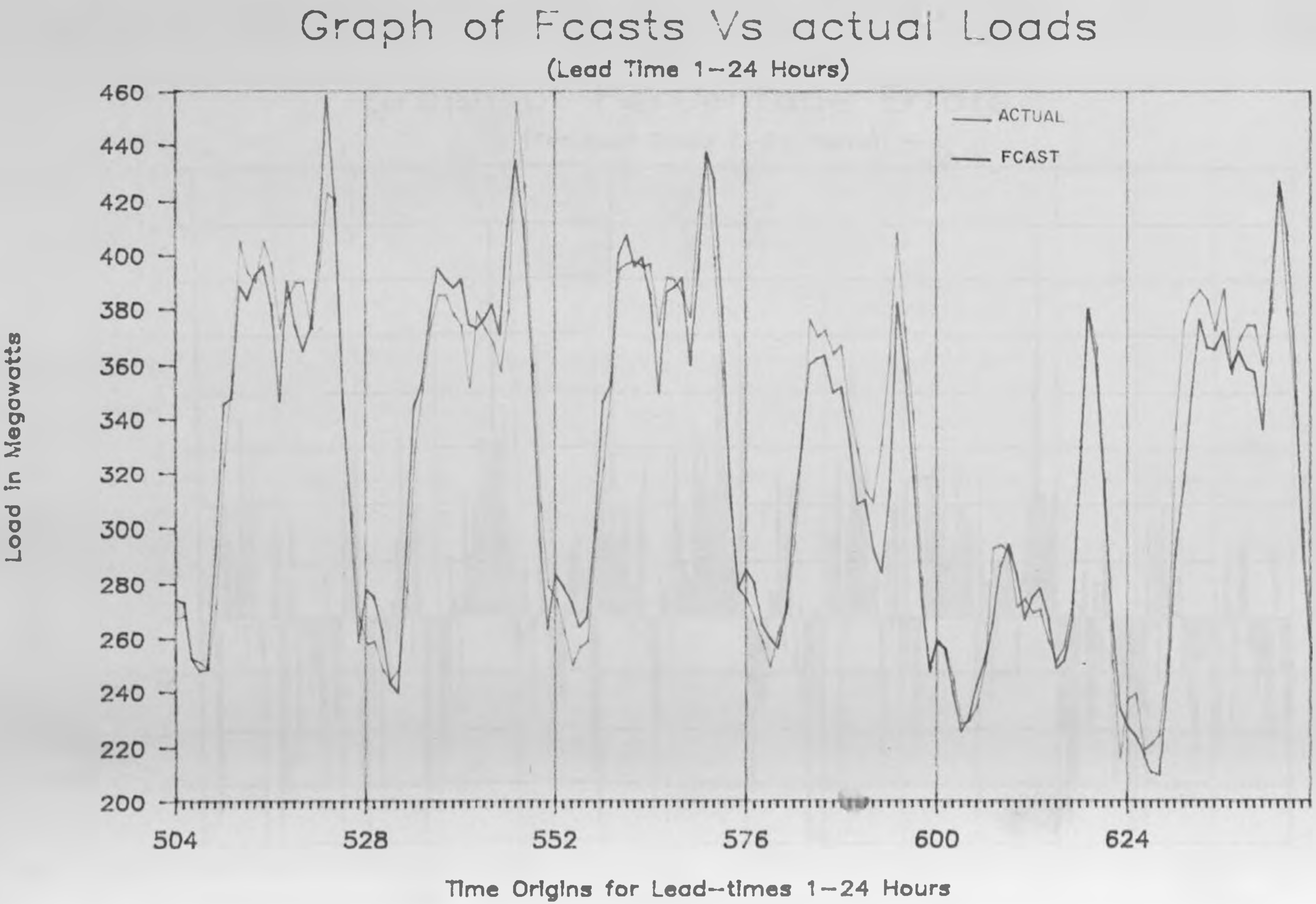

Graph of Fcasts Vs actual Loads
(Lead Time 1-24 Hours)
ACTUAL
FCAST
Load in Megawatts
460
440
420
400
380
360
340
320
300
280
260
240
220
200
504
528
552
576
600
624
Time Origins for Lead-times 1-24 Hours

Figure 4.4.1(a).

Graph of Percentage Errors

(For Lead Times 1-24 Hours)

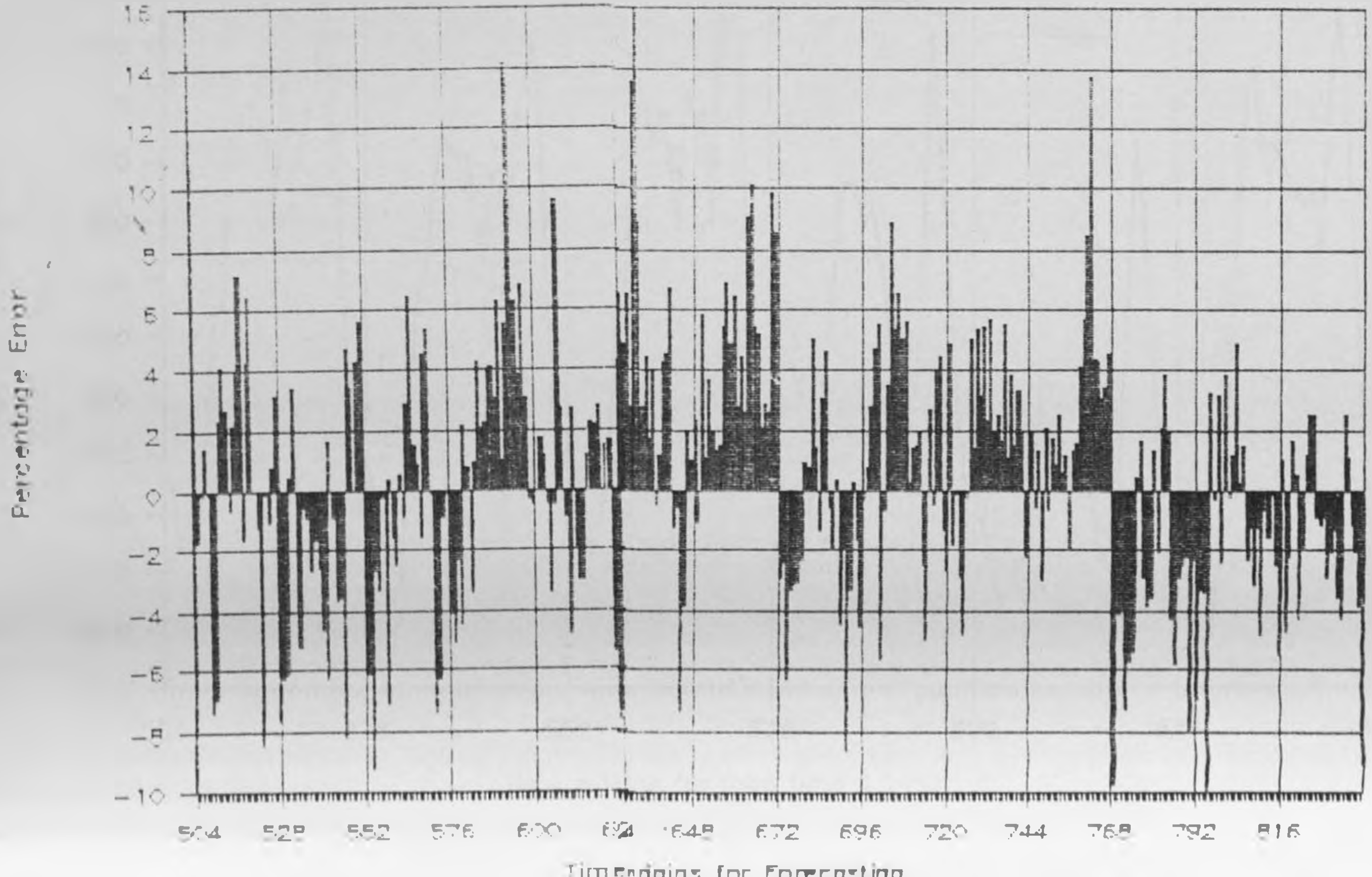


Figure 4.4.1 (b)

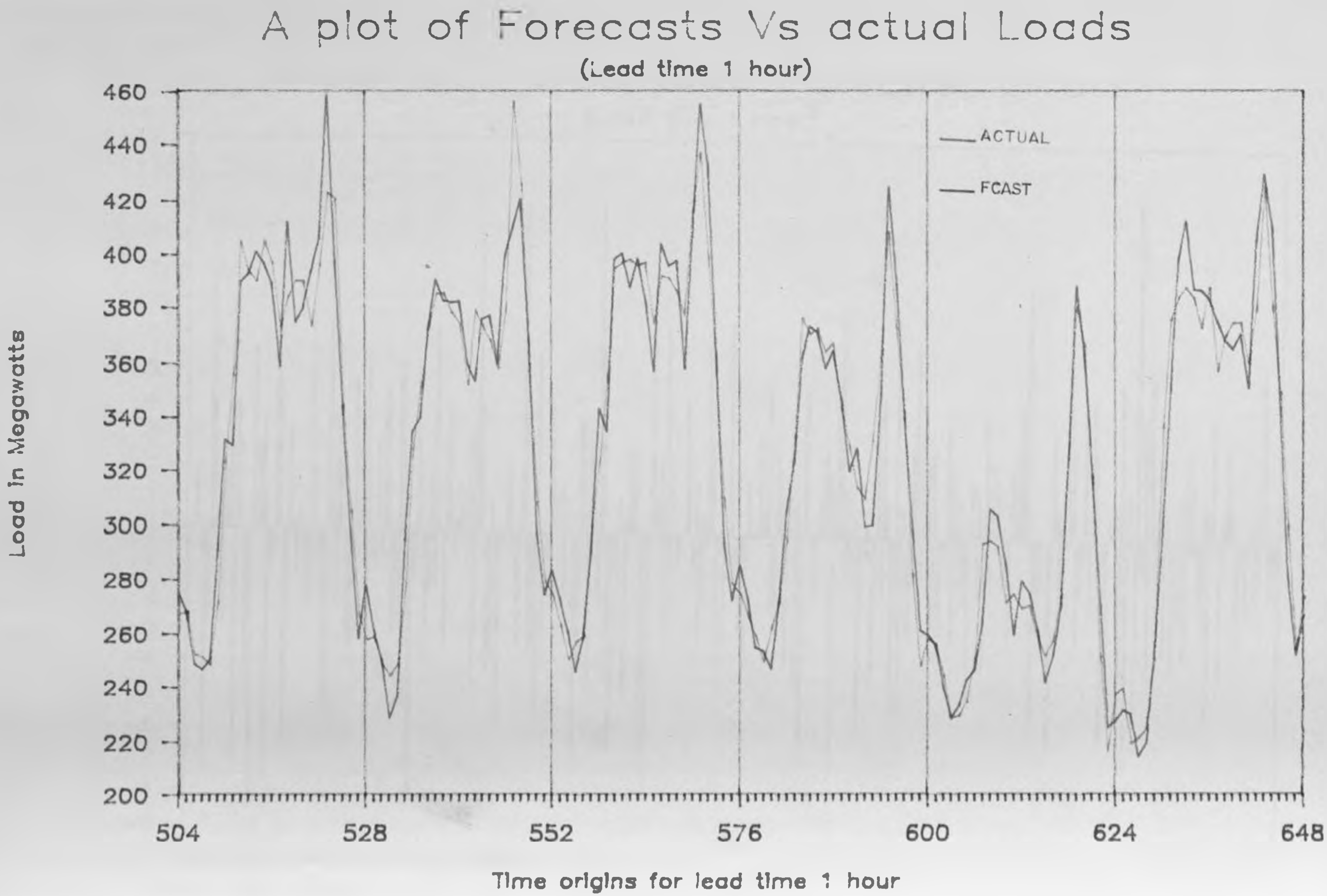

A plot of Forecasts Vs actual Loads
(Lead time 1 hour)
Load in Megawatts
460
440
420
400
380
360
340
320
300
280
260
240
220
200
ACTUAL
FCAST
504
528
552
576
600
624
648
Time origins for lead time 1 hour

Figure 4.4.2(a).

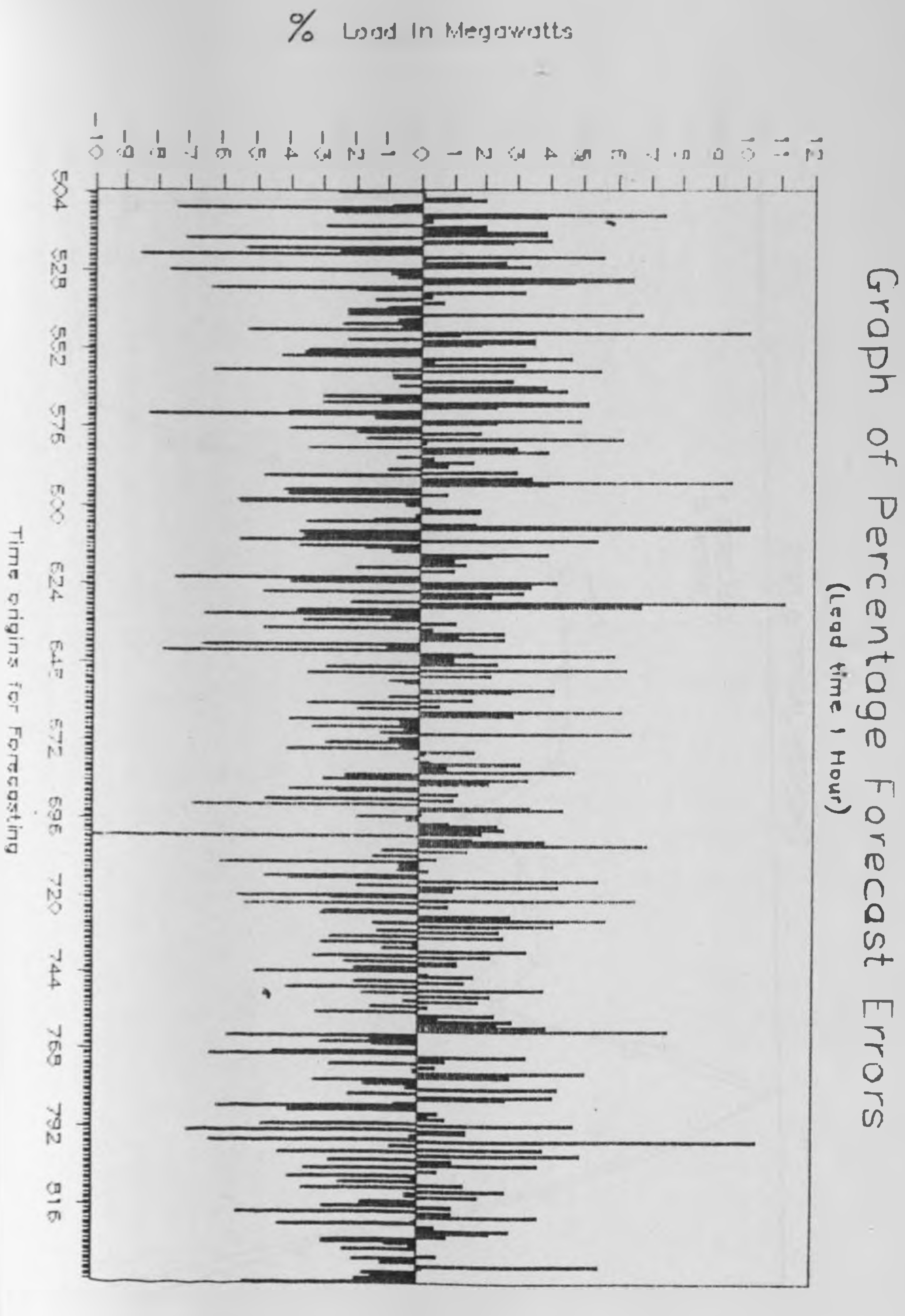

Graph of Percentage Forecast Errors
(Lead time 1 Hour)
% Load in Megawatts
Time origins for Forecasting
504
528
552
576
600
624
648
672
696
720
744
768
792
816

Figure 4.4.2 (b)

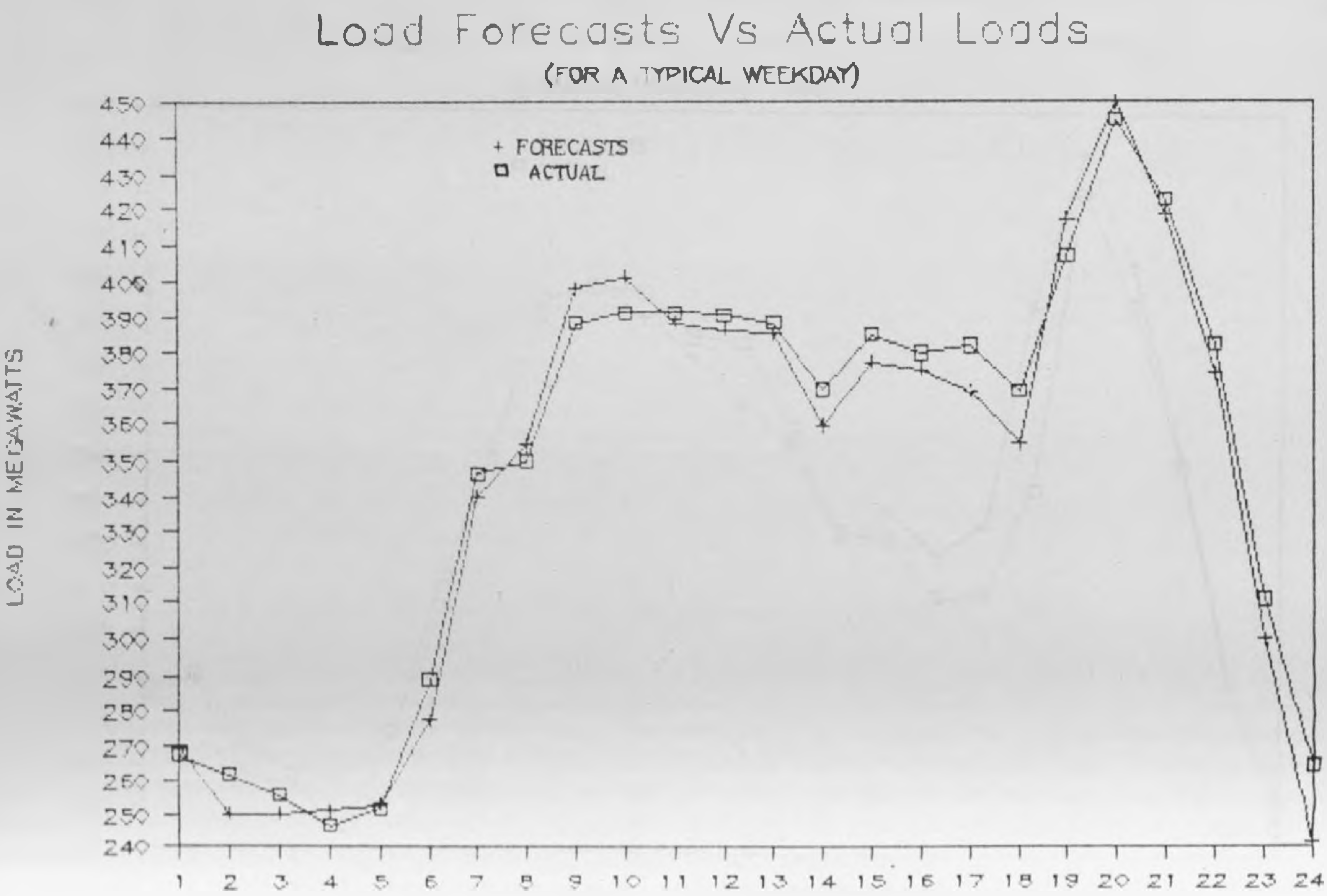

Load Forecasts Vs Actual Loads
(FOR A TYPICAL WEEKDAY)
+ FORECASTS
□ ACTUAL
LOAD IN MEGAWATTS
450
440
430
420
410
400
390
380
370
360
350
340
330
320
310
300
290
280
270
260
250
240
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24

Figure 4.4.3

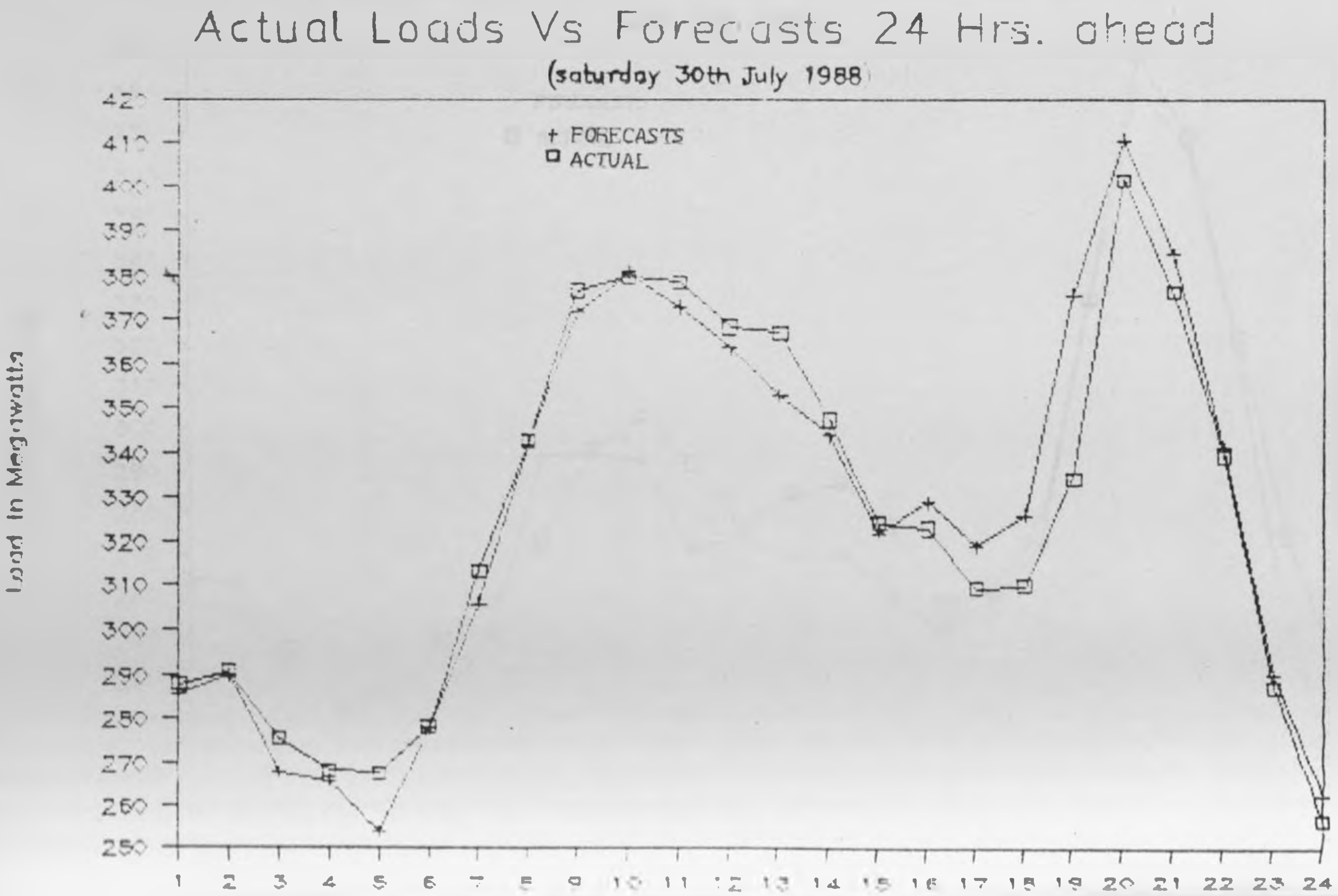

Actual Loads Vs Forecasts 24 Hrs. ahead
(saturday 30th July 1988)
+ FORECASTS
□ ACTUAL
Load in Megawatts
420
410
400
390
380
370
360
350
340
330
320
310
300
290
280
270
260
250
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24

Figure 4.4.4

Forecasts for a Typical sunday Load
(24th July 1988)

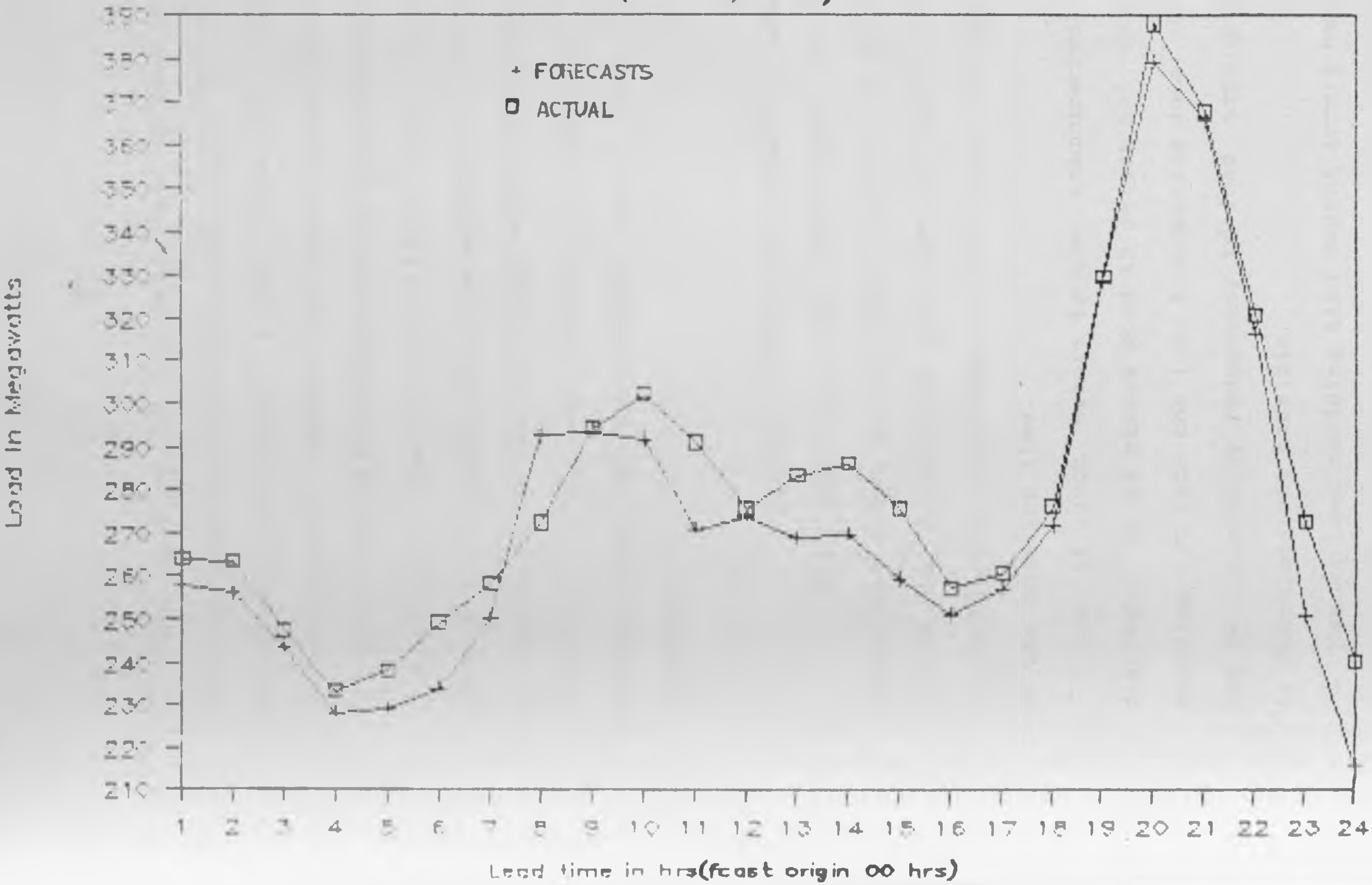


Figure 4.4.5

The average 24 hour lead time forecast error in terms of megawatts of actual load is 12.8 MW, while that of one hour lead time is 9.3 MW. These errors are quite acceptable when compared with other works done elsewhere on short term load forecasting as cited in some of the literature in the reference [2],[5],[9],[13]-[15]. The table 3.3.2.3 (pp55) shows the effect of a 10% variation in the values of the individual model parameters on the accuracy of the load forecast, as tested on the data between june and august 1988, for 24 hour lead time forecast.

4.5 SOURCES OF ERROR IN FORECASTED LOAD VALUES.

The model for load forecasting is developed assuming a normally distributed process. There are certain errors that distort the model due to human intervention and unforeseen emergencies which cannot be taken care of at the modeling time.

One of such errors is the inaccuracies in load readings. It is always good to detect bad data before entering it into the load forecasting data base. This can be achieved by redundancy in load measurement, which is sometimes not possible.

Manual Load shedding also causes errors because this means the power system cannot meet the load demand and it is then difficult to gauge the actual load demand.

System frequency management can be another source of error. Sometimes the system frequency deviates from the required normal value and this could lead to errors in knowing the actual load demand, for example in the Kenyan power system every 0.1 hertz change in system frequency is equivalent to about 5 megawatts change in electricity demand.

The frequency deviation occurs mostly where utilities are interconnected and there is no proper co-ordination between the various control centers.

The load management program of ripple control of domestic hot water heating systems, irrigation pumps in farms and street lighting control also create errors in forecasting because some of these installations are manually operated by the power system controllers who normally put them on at their discretion. An automatic switch on and off system which responds to the system load demand levels would alleviate this problem.

4.6 AREAS OF POSSIBLE IMPROVEMENT.

One of the improvements that could be made to this forecasting process is to model individual major load buses and then using optimal control techniques to combine the models for forecasting. These can only be done subject to the availability of hourly load readings for these nodes. This will be possible for the Kenya system once the SCADA system of control which is currently under installation is implemented.

Another area that needs serious and thorough study is the treatment of holidays and special days load forecasts. This study ignores the loads for holidays and instead replaces them in the load data base by their forecasts. A comparative analysis found that the loads on a particular holiday are quite similar to the loads on a neighbouring sunday. It was also discovered that particular holidays are similar from year to year if they occur on corresponding days such as the easter holiday, for example. With good data management techniques, a solution to such holiday problems can be sought.

Erroneous load readings is one of the greatest sources of forecasting errors and if many could also lead to a poor model choice. A lot of work is currently going on, on ways and means of bad data detection and correction for power system state estimation. Some of these techniques could be put to use in the load forecasting problem. At the rudimentary level,there should be a redundant load measurement system that checks the loads before they are used for forecasting. The effect of the use of statistical tests to detect outliers in load readings before forecasting as an improvement to load measurements can also be a subject of further investigation.

CHAPTER 5

5.0 CONCLUSIONS.

This study set out to develop a mathematical model that describes the daily pattern of electricity consumption in Kenya. The model was then to be used for forecasting the load demand for a period ranging from one hour to twenty four hours ahead.

Using time series analysis techniques, an auto-regressive integrated moving average model of four parameters has been developed. The model parameters were estimated using Hookes and Jeeves optimization techniques. Six weeks of system hourly load data was found adequate for model development.

Several tests were done at different times of the year to verify that the model predicts accurately and consistently. A continuous three week load forecast was done for each of the testing periods for different lead times and forecasting origins.

It has also been pointed out that the model development is based on the stochastic nature of the load process and no weather variables have been included because weather inputs could lead to double forecasting errors if there are no adequate weather records. In any case, in Kenya there is very little weather sensitive equipments installed because of the country's location within the tropics.

The load forecasting process is hourly adaptive with low computer memory requirements since the program only needs to store three weeks of hourly load data to be able to make a forecast. The hourly updating of the load forecasts minimizes the effects of unconsidered weather variables.

This studys forecasts gives mean absolute percentage errors of less than five percent for 24 hour lead time forecasts and less than three percent error for one hour lead time. These results are quite accurate compared with other results obtained elsewhere for short term load forecasts. These results should go along way in enhancing the efficient monitoring, operation and control of the Kenyan power system.

REFERENCES:

1. M.A.Abu-El-Magd and N.K.Sihna, "Short term load demand modeling and forecasting" A review, IEEE Trans. on systems, man and cybernetics; vol. SMC-12, no 3 pp 370-382 May-June 1982.

2. R.Campo and P.Ruiz, "Adaptive weather sensitive short term load forecasting", IEEE trans. on power systems vol. PWRS-2 no 3 August 1987.

3. J.Toyoda, M.Chen, Y.Inouye, "An application of state estimation to short term load forecasting", IEEE Trans. PAS 89 no 7 Sep.\Oct. 1970 pp 1678-1688.

4. G.E.Box and G.M.Jenkins, Time Series Analysis, Forecasting And Control; San Fransisco: Holden-day, 1970.

5. M.T.Hagan and Suzanne Behr, "Time series approach to short term load forecasting", IEEE Trans.on power systems vol. PWRS-2 no 3 August 1987.

6. M.Hagan and R.Klein, "Identification techniques of Box and Jenkins applied to the problem of short term load forecasting", IEEE power Engineering society meeting July 1977 paper no A77 618-2.

7. P.D.Mathewman and H.Nicholson, "Techniques of load prediction in the electricity supply industry", Proceedings of IEE vol. 115 no 10 1968.

8. E.D.Farmer and M.J.Potton, " Development of on-line load prediction techniques with results from trials in the South-West region of the CEGB", Proc. IEE Vol. 115 1968.

9. W.R.Christiaanse, "Short term load forecasting using general exponential smoothing", IEEE Trans. PAS 90 March-April pp 900 1972.

10. M.Plosser, "Short term load forecasting and seasonal adjustment", Journal of American statistical association 1979 pp 15.

11. D.J Wilde and C.S Beightler. Foundations of optimization Prentice Hall, N.J. 1969.

12. J.C.Kuester and J.H.Mize, "Optimization Techniques With Fortran, Mcgraw Hill, 1976.

13. M.A.Abu-El-Magd and N.K.Sihna, "Univariate and multivariate time series techniques for modeling and forecasting short term load demand", IFAC symposium on theory and application of digital control, India, Jan. 1982, paper no 5-4, pp 329.

14. M.S.Abou-Hussein and M.S.Kandil, "An accurate model for short term load forecasting", IEEE Trans. PAS 1981 pp 4158

15. B.Krogh, E.S.DeLlinas and D.Lesser,"Design and implementation of an on line load forecasting algorithm", IEEE Trans. PAS pp 3280 1982.

BIBLIOGRAPHY:

1. K.Srinivasan and R.Provonost, "Short term load forecasting using correlation models", IEEE trans. PAS 94 no 5, 1975.

2. P.C Gupta and K.Yamada, "Adaptive short term load forecasting of hourly loads using weather information", IEEE Trans. PAS 91 no 5, Sept\Oct. 1972.

3. P.Stoica, "Tests for Whiteness", IEEE Trans. Automatic control vol.22 no 6 pp 992-993 1977.

4. D.C.Hamilton and D.G.Watts, "Interpreting partial autocorrelation functions of seasonal time series models Biometrika 1978 vol. 65 pp 135-140.

5. H.Akaike, "A new look at stochastic model identification", IEEE Trans. Automatic control vol. AC 19 pp 716-732, 1974.

6. J.Murdoch, Statistical Tables For Science And Engineering, Mc Millan, pp 17.

7. J.A.Greenwood and Hartley, Guide To Tables in Mathematical Statistics, Princeton University press 1962, pp 140.

8. S.M.Simpson, Time Series Computations In Fortran And Fap, Addision Wesley.

9. W.Gilchrist, statistical Forecasting, John Wiley and sons, 1976.

APPENDIX.

The load modeling and forecasting procedure has been implemented as five computer programs,all written in standard Fortran 77 Language and developed on an IBM personal computer.

The programs TEST is for model identification and it evaluates the autocorrelation functions of various transformations of the series such as differrencing ,natural logarithmic transformation as well as calculating the 95% confidence intervals.

The programme PCF is also a model identification programme which takes the autocorrelations as it's input and evaluates the partial autocorrelations up to the desired lags.

Programmes SHADD and DIAGN are for parameter estimation and diagnostic checks. These give the values of the optimum model parameters,their residual variance,the autocorrelation function of the model residuals as well as the Chi-squared statistic (Q).

The final programme FCAST is the load forecasting programme that allows load prediction for a lead time of one to 24 hours,as well as giving the values of the mean absolute percentage errors of the forecasts and the average load values over the forecast period.

The next couple of pages contain a user manual and a print out of the program listings.

LOAD FORECASTING PROGRAMS USER MANUAL.

The complete short term load forecasting algorithm is implemented as five computer programs which are for model identification, parameter estimation, diagnostic checks and load forecasting.

These programs are coded in standard Fortran 77 language and are compiled using the RMFORT compiler that requires a maths co-processor 8087 to run. A necessary editor that is used is the Turbo Pascal editor which is used for fortran file creation as well as the creation of data files. Other editors such as the Wordstar can also be used for data file creation

All the programs are contained in a single 3.5" or three 5.25" diskettes and they can all run on an IBM compatible machine which has an 8087 maths co-processor. Once the diskette is inserted, all that one needs to type is the command README and a summary of each of the programs and the directories where they are available will be given. One is then able to chose the particular program of interest.

PROGRAM TEST.

The compiled version is TEST.EXE. It calculates the autocorrelation functions to the desired lag, and for this particular case the lag is limited to 500 but this could be increased by dimensioning the ACOR(J) array. The input data required for the program is the past hourly load series observations which has to be entered through the keyboard into a data file named LDATA. This program handles upto 2200 hours of load data, which can also be increased by proper dimensioning of the X array.

The screen inputs required for this program are;

1) The number of load series observations
2) The maximum lag of autocorrelation required
3) The seasonal period
4) The degree of regular differencing

All the above inputs are through appropriate screen prompts. The results are obtained from a file ACFS which lists the autocorrelation function coefficients versus their lags in a tabular form as well as their 95% confidence intervals.

PROGRAM PCF.

The compiled version is PCF.EXE. Execution of PCF always follows that of TEST programs since it obtains it's input from the output of TEST. The only input required here is the maximum lag of the partial autocorrelation desired. In this program it is limited to 75, but it's value could be increased by dimensioning the array P accordingly if there is sufficient memory.

The output of the partial autocorrelation coefficients versus the lags in tabular form are obtained from the file PCFOUT.

PROGRAM SHADD.

The compiled version is SHADD.EXE. It's execution requires the use of past hourly load data, to be entered in a 'LDATA' file before execution of the program. The array size for this particular program allows upto 2200 hours of load data for model parameter estimation.

Other inputs for the program which are entered on the screen through prompts are;

1) Number of model parameters to be evaluated
2) Initial parameter estimates
3) Initial step sizes for changing parameter estimates
4) Number of hours of load series observations
5) Maximum lag of autocorrelation required for diagnostic checks.
6) Seasonal period
7) order of AR and MA parameters
8) Choice of whether series is to be Log transformed or not.
9) Convergence criteria.

The results obtained from a file PESTIM gives the optimum parameter values, the minimum sum of squares function, the residual variance, the number of iterations as well as generating the residual series, a_t corresponding to the optimum parameter values. The a_t series are used as the input to the program DIAGN for evaluating the ACF of the residuals for checking for goodness of fit. The objective function for this model parameter evaluation program is developed in the subroutine LEAST1.

PROGRAM DIAGN.

The compiled version is DIAGN.EXE. It is run only after executing the program SHADD which generates the residual series a_t, which it writes into a file RESID which is then used by the DIAGN program to evaluate the autocorrelation functions of the residuals. It also the X^2 statistics as well as the 95% limits of the ACFS. All these results are obtained from a file ACFS.

PROGRAM FCAST.

This is the program that generates the required hourly load forecasts. To start off the forecasting process, a load data base of the most recent 504 hours (3 weeks of hourly load data) is entered into a data file, 'LDATA'. This data file can be created by the turbo editor. Once this has been done, a decision can then be made of how often the forecasts are required.

A policy of generating a forecast either once a day (every 24 hours) or every one hour is recommended to avoid a lot of confusion. If possible load forecasts should be done every one hour, since the hourly updating makes the forecasting process more accurate.

Before one performs a forecast, there will be a screen LOGO of invitation to a forecasting session asking for the following information to be entered on the screen;

1) The last time a forecast was made (hour,day,year)

2) The present hour and date

3) The number of hours that have elapsed since the last forecast was made.

4) If the forecast is done hourly, then the load reading of the previous hour is required, otherwise if the forecast is done once a day., then the latest 24 hours of hourly load values will be required before a forecast is made since this is a time series process.

The forecasts are then generated and the results obtained from the file FCOUT which lists the lead times versus the Forecasted load values. The error in forecasting the previous hours load is also generated. Once the forecasting process is started, the load data file is updated automatically, through a recursive process as the forecasting process continues.

GRAPHS.

All the outputs of the above programs are in a tabular form which makes it quite easy for graph plotting. The LOTUS 123 utility is used for generating and plotting the graphs.

The 123.EXE option is used to invoke the worksheet menu. The "import " option on the menu is used to transfer the results file, (whose data is to be plotted) into the worksheet. The graph can then be easily generated and viewed on the screen using the graph commands of the lotus software. The graphs can be plotted on paper using the "printgraph" option of the software, through either a text printer or a graphic plotter.

```
      PROGRAMME TEST
C     {creating storage space for arrays:Array X stores the load series
C     observation ,array ACOR is for storing the  autocorrelations,array
C     DFX is for holding the differenced Load observs.and Y is for
C     standard error evaluations}

      DIMENSION X(2200),ACOR(500),DFX(2200),Y(500)
      WRITE(*,1)
1     FORMAT(1X,' THIS PROGRAMME IS FOR MODEL IDENTIFICATION AND IT CALCU
     LATES THE AUTOCORRELATIONFUNCTION COEFFICIENTS. /)
      WRITE(*,*) INPUT The Number of Load series observations
      READ(*,*)LX
      WRITE(*,*)'INPUT The maximum lag of autocorrelation required
      READ(*,*)MXLAG
      WRITE(*,*) INPUT The seasonal period
      READ(*,*)NS
      WRITE(*,*) INPUT The degree of regular differencing
      READ(*,*)ND
      IF (ND.GT.2) THEN
      WRITE(*,*)'FOR THIS PROGRAMME DEGREE OF NON-SEASONAL DIFFERENCING
     & IS UPTO ORDER 2 ONLY'
      ELSE
      WRITE(*,2)
2     FORMAT(1X,' THE INPUT DATA FOR THE LOAD-SERIES (Zt) IS THROUGH "LDA
     TA" FILE WHOSE CONTENTS  CAN BE AMMENDED APPROPRIATELY USING THE '
     Turbo pascal editor' /)
      WRITE(*,3)
3     FORMAT(1X,' WAIT FOR COMPUTATIONS AND THEN OBTAIN THE RESULTS FROM
     THE FILE 'ACFS' AFTER THEWORD 'press a key to continue' or 'A ' A
     PPEARS ON THE SCREEN. )

C     {              =========opening input data files================}

      OPEN(5,FILE='LDATA',STATUS='OLD')
      OPEN(6,FILE='PCFIN',STATUS='OLD')
C      {  LX=No of observations of load data time series
C         MXLAG=maximum lag of autocorrelation function coeffs.
C         NS=degree of seasonal differencing (D)
C         ND=Degree of non-seasonal differencing (d)  }


      READ(5,*)(X(I),I=1,LX)
      CLOSE(5,STATUS='KEEP')
C     ============== performing natural logarithmic  data transfomation
      DO 10 I=1,LX
      X(I)=ALOG(X(I))
10    CONTINUE
C       =======opening output results file==========
      OPEN(7,FILE='ACFS',STATUS='OLD')
      JMAX=MXLAG+2
      IF (NS.GE.1) THEN
         CALL DIFF(X,LX,LS,NS,DFX)
           IF (ND.GE.1) THEN
             CALL DIFF(DFX,LS,LD,ND,DFX)
              IF (ND.EQ.2) THEN
                CALL DIFF(DFX,LD,LB,ND,DFX)
                CALL REMAV(DFX,LB,XAVG)
                CALL FORAC(DFX,LB,MXLAG,ACOR(J))
              ELSE
               CALL REMAV(DFX,LD,XAVG)
               CALL FORAC(DFX,LD,MXLAG,ACOR(J))
```

```
          ENDIF
        ELSE
         CALL REMAV(DFX,LS,XAVG)
         CALL FORAC(DFX,LS,MXLAG,ACOR(J))
        ENDIF
     ELSE
       CALL REMAV(X,LX,XAVG)
       CALL FORAC(X,LX,MXLAG,ACOR(J))
     ENDIF
c                            {========outputting results=============}
     WRITE(7,51)LX,MXLAG,NS,XAVG,ND
c    {===========caculating confidence limits of standard errors======}
     SUM=0.0
     DO 20 J=2,JMAX-1
      SUM=SUM+(ACOR(J)/ACOR(1))**2
       Y(J)=SUM
20   CONTINUE
     WRITE(7,*)'the autocorrelation coefficients for various lags are:'
     WRITE(7,*)'======================================================'
     WRITE(7,*)'LAG      ACCF       SE          +2SE        -2SE'
     WRITE(7,*)'====     ======    ======       =======     ======='
     DO 50 J=2,JMAX-1
      SE=SQRT((1+2*Y(J))/LX)
       WRITE(7,100)J-1,ACOR(J)/ACOR(1),SE,2*SE,-2*SE
50   CONTINUE

c    {==inputs to pcf for calculating partial auto-corr. coefficients}

     WRITE(6,*)MXLAG,NS,ND
     WRITE(6,*)(ACOR(J)/ACOR(1),J=2,JMAX-1)
c                          {=======outputting results=============}

100  FORMAT(1X,I3,6X,F5.2,7X,F5.3,7X,F5.3,8X,F5.3)
51   FORMAT(1X,'NO OF OBSERVS.IS  ', I4,'  MAX. LAG OF ACF IS', I3,/1X
    &, 'SEASONAL PERIOD IS  ', I3  , ' AVERAGE IS', F8.2,'  DEGREE OF
    &REGULAR DIFFERENCING', I3 )
     CLOSE(7,STATUS='KEEP')
     CLOSE(6,STATUS='KEEP')
     ENDIF
     END
c              {=====START OF SUBROUTINE PROGRAMMES====}

c    =====This SUB. evaluates autocorrelation coefficients=========

     SUBROUTINE FORAC(X,LX,MXLAG,ACOR)
c    {=====X,LX,ACOR,MXLAG are as defined in main programme===}
c    {========================* is just a dummy variable====}
     DIMENSION X(*),ACOR(*)
     JMAX=MXLAG+2
     DO 20 J=1,JMAX
      SUM=0.0
      NMAX=LX-J+2
      DO 10 I=1,NMAX
        K=J+I-1
         SUM=SUM+X(I)*X(K-1)
10      CONTINUE
       ACOR(J)=SUM
20   CONTINUE
     RETURN
     END
c    {=====This SUB. is for evaluating and removing the average====}

     SUBROUTINE REMAV(X,LX,XAVG)
```

```
      DIMENSION X(*)
      SUM=0.0
      DO 10 I=1,LX
       SUM=SUM+X(I)
10    CONTINUE
      XAVG=SUM/LX
      DO 20 I=1,LX
       X(I)=X(I)-XAVG
20    CONTINUE
      RETURN
      END
c     ====for perfoming seasonal & non-seasonal differencing of series
      SUBROUTINE DIFF(X,LX,LS,NS,DFX)
      DIMENSION X(*),DFX(*)
      LS=LX-NS
      DO 10 I=1,LS
      DFX(I)=X(I+NS)-X(I)
10    CONTINUE
      END
      END
```

```
      PROGRAMME pcf
c     {array P holds the partial autocorrelation coefficients;array R
c      is for the autocorrelations}
c                                {=======program starts============}
      DIMENSION P(75,75),R(400)
      WRITE(*,2)
2     FORMAT(1X,'THIS PROGRAMME CALCULATES THE PARTIAL AUTOCORRELATION
     ^COEFFS. USED IN MODELLING')
      WRITE(*,3)
3     FORMAT(1X,'This programme is to be run ONLY after executing the pr
     ^ogramme (TEST) as it obt-ains its input from the output of TEST'/)
      WRITE(*,*)'INPUT (MXLAG), MAXIMUM LAG OF PARTIAL AUTOCORRELATIONS'
      READ(*,*)MXLAG
      OPEN(5,FILE='PCFIN',STATUS='OLD')
      READ(5,*)LX,NS,ND
      IF(MXLAG.GT.75) THEN
      WRITE(*,*)'MAXIMUM LAG SHOULD NOT EXCEED 75, TRY AGAIN PLEASE'
      ELSE
      WRITE(*,4)
4     FORMAT(1X,'WAIT FOR A WHILE FOR COMPUTATIONS THEN OBTAIN THE RESUL
     ^TS FROM THE FILE "PCFOUT"AFTER THE WORD "Press key to continue" or
     ^ " A>" APPEARS ON SCREEN')

C         {=== file pcfin contains the autocorrelation functions which
c          is the input data file for this program  }
c         {=====LX,NS,ND are output of program TEST=======}

C         {LX=No of load time series observations
C         MXLAG=maximum lag of partial autocorrelation function coeffs.
C         NS=degree of seasonal differencing (D)
C         ND=degree of non-seasonal differencing }

c     {====inputs of autocorr.coeffs (R(I)) for partial autocorrelation
c         calculations }

      READ(5,*)(R(I),I=1,MXLAG)
      CLOSE(5,STATUS='KEEP')
c                                         {====pcfout is a results file==}
      OPEN(7,FILE='PCFOUT',STATUS='OLD')

c     {=======partial autocorrelation function coeffs. evaluation====}

c              {=========MAIN BODY OF PROGRAMME=====}
      P(1,1)=R(1)
      DO 10 I=2,MXLAG
       SUMA=0.0
       SUMB=0.0
       DO 20 J=1,I-1
         SUMA=SUMA+P(I-1,J)*R(I-J)
          SUMB=SUMB+P(I-1,J)*R(J)
           SUMAT=R(I)-SUMA
           SUMBT=1-SUMB
            P(I,I)=SUMAT/SUMBT
          DO 30 K=1,I-1
             P(I,K)=P(I-1,K)-P(I,I)*P(I-1,I-K)
30         CONTINUE
20      CONTINUE
10    CONTINUE
c    { ================output of the results=================}
      WRITE(7,120)NS
      WRITE(7,130)ND
```

```
      WRITE(7,50)MXLAG
      WRITE(7,55)
c     ==============evaluating confidence limits of the estimates
c     ==============SE is the standard Error====================
      SE=SQRT(1.0/LX)
      WRITE(7,*)' LAG      PACCF       SE'
      WRITE(7,*)'=====     ======      ===='
c     {=====P(I,I) is the partial autocorrelation at lag I====}
      DO 60 I=1,MXLAG
       WRITE(7,100)I,P(I,I),SE
60    CONTINUE
c     { ================ outputting results=====================}
100   FORMAT(1X,I3,6X,F5.2,7X,F5.3)
120   FORMAT(1X,'SEASONAL PERIOD OF ORIGINAL SERIES IS :'  ,  I3)
130   FORMAT(1X,'DEGREE OF NON-SEASONAL DIFFERENCING IS :'  ,  I3)
50    FORMAT(1X,'MAXIMUM LAG OF PARTIAL AUTOCORRELATION COEFFS IS ', I3)
55    FORMAT(1X,'THE PARTIAL AUTOCRRELATION COEFFICIENTS FOR VARIOUS LAG
     &S ARE:   ')
      CLOSE(7,STATUS='KEEP')
      ENDIF
      END
      END
```

```
      PROGRAME SHADD
c     This program is for model estimation using Hookes and Jeeves
c     Non-linear optimization method.
      dimension EPS(10),RK(10),Q(10),QQ(10),W(10)
      dimension X(2200),D(2200),E(2200),A(2200)
      write(*,21)
21    Format(1X,10X,'THIS PROGRAMME IS FOR ESTIMATION OF MODEL PARAMETER
     ^S'/,10X,'=====================================================')
      Write(*,*)'INPUT N,The No of model parameters'
      Read(*,*)NSTAGE
      Write(*,*)'INPUT THE INITIAL PARAMETER ESTIMATES'
      Read(*,*)(RK(II),II=1,NSTAGE)
      Write(*,*)'INPUT INITIAL STEP SIZE INCREMENTS FOR THE PARAMETERS '
      Read(*,*)(EPS(JJ),JJ=1,NSTAGE)
      Write(*,*)'INPUT LX(no of Load values),MXLAG(Lag of ACFS required)
     ^,NS(seasonal period)'
      Read(*,*)LX,MXLAG,NS
      Write(*,*)'INPUT NSR(Order of AR parameter),NM(Order of 2nd MA Par
     ^ameter) '
      Read(*,*)NSR,NM
      write(*,*)'IF LOAD SERIES IS LOGGED ENTER (1),OTHERWISE ENTER (0)'
      Read(*,*)LG
c     ===========================LDATA file contains input load series
      Open(5,FILE='LDATA',STATUS='OLD')
      Read(5,*)(X(J),J=1,LX)
      Close(5,STATUS='KEEP')
c     ===================Performing natural logarithmic transformation
      If (LG.EQ.1)THEN
      do 11 I=1,LX
      X(I)=ALOG(X(I))
11    continue
       else
        do 12 I=1,LX
         X(I)=X(I)
12    continue
      endif
      open(7,file='PESTIM',status='OLD')
      open(6,file='RESID',status='OLD')
c     { IPRINT =1 means intermediate results are printed,and when=0 then
c       only final results printed.
c       MAXK is the number of function evaluatons before the program is
c       terminated.
c       NKAT is the number of times the step size is decreased before
c       programe terminates.
c       ALPHA is the acceleration factor.
c       BETA is the fraction by which the step size is decreased.
c       EPSY is the difference in function values before final result
c       is accepted.
c       ND is the degree of regular differencing. }

c      ================inputting the programme constants
      IPRINT=0
      MAXK=500
      NKAT=20
      ALPHA=1.0
      BETA=0.5
      EPSY=0.0000001
      ND=1

c     ======================START OF MAIN PROGRAMME==================

      QD=0.0
```

```
      Write(7,10)
10    Format(1X,10X,'HOOKES & JEEVES OPTIMIZATION ROUTINE')
      Write(7,20) ALPHA,BETA,MAXK,NKAT
20    Format(//,2X,'INPUT PARAMETERS',/,2X,'ALPHA=',E5.2,4X,'BETA=',F5.2
     ^,4X,'ITMAX=',I4,4X,'NKAT=',I3)
      Write(7,30)NSTAGE
30    Format(/,2X,'NUMBER OF VARIABLES=',I3)
      Write(7,40)
40    Format(/,2X,'INITIAL STEP SIZES')
      do 45 I=1,Nstage
        write(7,50)I,Eps(I)
50        Format(/,2X,'EPS(',I2,')=',F8.4)
45    Continue
      Write(7,61)Epsy
61    Format(//,2X,'ERROR IN FUNCTION VALUES FOR CONVERGENCE=',E16.8)
      Call DIFF (X,LX,LS,NS,X)
      Call DIFF (X,LS,LD,ND,X)
      KFLAG=0
      Do 601 I=1,Nstage
        Q(I)=RK(I)
         W(I)=0.0
601   Continue
      Kat=0
      Kk1=0
70    Kcount=0
      Wbest=W(Nstage)
      Call LEAST1(LX,X,NS,ND,E,D,A,RK,NSTAGE,SUM,NSR,LD,M,NM)
      Kk1=Kk1+1
      BO=SUM
      IF(KK1.EQ.1)QD=SUM
      IF(KK1.EQ.1)GO TO 201
      IF(BO.GT.QD)KFLAG=1
      IF(BO.LT.QD)QD=BO
C
C     ===================Establishing Pattern Direction ===============
201   DO 55 I=1,NSTAGE
      QQ(I)=RK(I)
      TSRK=RK(I)
      RK(I)=RK(I)+EPS(I)
      CALL LEAST1(LX,X,NS,ND,E,D,A,RK,NSTAGE,SUM,NSR,LD,M,NM)
      KK1=KK1+1
      W(I)=SUM
      IF(W(I).LT.QD)GO TO 58
      RK(I)=RK(I)-2.0*EPS(I)
      CALL LEAST1(LX,X,NS,ND,E,D,A,RK,NSTAGE,SUM,NSR,LD,M,NM)
      KK1=KK1+1
      W(I)=SUM
      IF(W(I).LT.QD)GO TO 58
      RK(I)=TSRK
      IF(I.EQ.1)GO TO 513
      W(I)=W(I-1)
      GO TO 613
513   W(I)=BO
613   CONTINUE
      KCOUNT=1+KCOUNT
      GO TO 55
58    QD=W(I)
      QQ(I)=RK(I)
55    CONTINUE
      IF (IPRINT)60,65,60
60    WRITE(7,100)KK1
C
C     ===============Recording Response & Location ===================
```

```
      WRITE(7,102)
      WRITE(7,207)(RK(I),I=1,NSTAGE),QD
C
C                    =====Tests to determine termination of programme
65    IF(KK1.GT.MAXK)GO TO 94
      IF(KAT.GE.NKAT)GO TO 94
      IF(ABS(W(NSTAGE)-WBEST).LE.EPSY)GO TO 94
C
C     ==============================Reduction of step size test
      IF (KCOUNT.GE.NSTAGE)GO TO 28
C
C     ========================acceleration in pattern direction
      DO 26 I=1,NSTAGE
      RK(I)=RK(I)+ALPHA*(RK(I)-Q(I))
26    CONTINUE
      DO 25 I=1,NSTAGE
      Q(I)=QQ(I)
25    CONTINUE
      GO TO 70
C
C     =================Reduction of step sizes =======================
28    KAT=KAT+1
      IF(KFLAG.EQ.1)GO TO 202
      GO TO 204
202   KFLAG=0
      DO 203 I=1,NSTAGE
      RK(I)=Q(I)
203   CONTINUE
204   DO 80 I=1,NSTAGE
      EPS(I)=EPS(I)*BETA
80    CONTINUE
      IF(IPRINT)85,70,85
85    WRITE(7,101)KAT
      GO TO 70
94    WRITE(7,460)(EPS(I),I=1,NSTAGE)
      WRITE(7,462)QD,LX,M,NS.NSR,ND,QD/LD,NM
      DO 104 I=1,NSTAGE
104   WRITE(7,103)I,RK(I)
      WRITE(7,100)KK1
100   FORMAT(//,2X,'NO. OF FUNCTION EVALUATIONS=',I8)
101   FORMAT(/,2X,'STEP SIZE REDUCED',I2,2X,'TIMES')
102   FORMAT(//,2X,'END OF EACH PATTERN SEARCH'/)
103   FORMAT(/,2X,'FINAL OPTIMUM PARAMETER ESTIMATE P(',I2,')=',F5.2)
207   FORMAT(//,2X,'VARIABLES AND SUM',3X,9E12.4//)
460   FORMAT(1X,'FINAL STEP SIZES ARE',5F20.8//)
C461   FORMAT(1X,'THE FINAL OPTIMUM PARAMETER ESTIMATES ARE',5F18.4/)
462   FORMAT(1X,'THE MINIMUM SUM OF SQUARES OF ERROR FUNCTION S(01,..0 n
     ^)',F20.4/,1X,'NO OF LOAD SERIES OBSERVATIONS (HRS)=',I5/,1X,'MODEL
     ^ TYPE=',I3,2X,'SEASONAL PERIOD=',I3,2X,'DEGREE OF AR PARAMETER=',I
     ^3,/,1X,'DEGREE OF NON-SEASONAL DIFFERENCING=',I3,2X,'RESIDUAL VARI
     ^ANCE=',F10.3,1X,'ORDER OF 2nd MA PARAMETER=',I3)
c
c     =============ouputting the residual series (at)================
      WRITE(6,*)Ld,MXLAG,0,0
```

```
      WRITE(6,*)(A(I),I=ns+2,Lx)
      CLOSE(6,STATUS='KEEP')
      CLOSE(7,STATUS='KEEP')
      END
c
c     ============The function describing subroutine================

      SUBROUTINE LEAST1(LX,X,NS,ND,E,D,A,AKE,NSTAGE,SUM,NSR,LD,M,NM)
      DIMENSION AKE(*),E(LX),A(LX),X(LD),D(LX)
c     { AKE is the vector for the parameter variables
c       E is the error in load series vector
c       A is the residual load series vector
c       X is the original load series vector
c       D is the differenced load series vector
c       LX is the number of load series observations
c       NS is the degree of seasonal differencing
c       NSR is the order of autoregressive operator and NM is the order
c       of moving average operator    }
      P1=AKE(1)
      P2=AKE(2)
      P3=AKE(3)
      P4=AKE(4)
      J=LX-NSR
      E(J)=X(J-NS-1)
      DO 10 J=LX-NSR-1,LX-NSR-NS-NM,-1
      E(J)=X(J-NS-1)-P3*X(J+NSR-NS-1)+P1*E(J+ND)
10    CONTINUE
      DO 40 J=LX-NSR-NS-NM-1,NS+2,-1
      E(J)=X(J-NS-1)-P3*X(J+NSR-NS-1)+P1*E(J+ND)+P4*E(J+NM)+P2*E(J+NS)
     ^-P1*P2*E(J+NS+1)-P2*P4*E(J+NM+NS)
40    CONTINUE
      DO 45 J=NS+1,1,-1
      E(J)=0.0
45    CONTINUE
      DO 50 J=1,NS+1
      D(J)=P3*D(J+NSR)-P1*E(J+ND)-P2*E(J+NS)-P4*E(J+NM)+P1*P2*E(J+NS+ND1
     ^)+P2*P4*E(J+NS+NM)
50    CONTINUE
      do 55 J=NS+2,LX
      D(J)=X(J-NS-1)
55    CONTINUE
      DO 60 I=1,NSR+1
      A(I)=0.0
60    CONTINUE
      DO 65 I=NSR+2,NS+NM+1
      A(I)=D(I)-P3*D(I-NSR)+P1*A(I-ND)
65    CONTINUE
      DO 80 I=NS+NM+2,LX
      A(I)=D(I)-P3*D(I-NSR)+P1*A(I-ND)+P4*A(I-NM)+P2*A(I-NS)-P1*P2*A(I-
     ^NS-1)-P2*P4*A(I-NM-NS)
80    CONTINUE
      SUMN=0.0
      DO 85 J=NS+NM+2,LX
      SUMN=SUMN+A(J)**2
85    CONTINUE
      SUM=SUMN
      RETURN
      END
c
c     ===============This subroutine performs differencing operations

      SUBROUTINE DIFF(X,LX,LS,NS,X)
c     { LX is the number of original observations before differencing
```

```
C       LS is the number of observations atfer differencing
C       NS is the degree of differencing
C       X is the series vector before and after differencing operations}
      Dimension X(1)
      LS=LX-NS
      DO 10 I=1,LS
        X(I)=X(I+NS)-X(I)
10    Continue
      Return
      End
      END
```

```
      PROGRAMME DIAGN
      DIMENSION X(2200),ACOR(400),DFX(2200),Y(400)
      WRITE(*,*)'THIS PROGRAMME PERFORMS DIAGNOSTIC CHECKS BY CALCULATIN
     *G THE AUTOCORRELATION    FUNCTIONS OF THE RESIDUAL SERIES(at)'
      WRITE(*,*)'
      WRITE(*,*)'This programme is run ONLY after running the programme
     *SHADD (FOR PARAM. ESTIM) Since it obtains its input from the outpu
     *t of SHADD'
      WRITE(*,*)'WAIT FOR A WHILE THEN OBTAIN RESULTS FROM A FILE "ACFS"
     *AFTER THE WORD== press a key to continue OR A)==appears on screen
C        {===============START OF MAIN PROGRAMME=========}

       OPEN(8,FILE='RESID',STATUS='OLD')
       READ(8,*)LX,MXLAG,NS,ND
       IF (ND.GT.2) THEN
       WRITE(*,*)'FOR THIS PROGRAMME DEGREE OF NON-SEASONAL DIFFERENCING
     * IS UPTO ORDER 2 ONLY'
      ELSE
      READ(8,*) (X(I),I=1,LX)
      CLOSE(8,STATUS='KEEP')
      OPEN(7,FILE='ACFS',STATUS='OLD')
      JMAX=MXLAG+2
      IF (NS.GE.1) THEN
       CALL DIFF(X,LX,LS,NS,DFX)
         IF (ND.GE.1) THEN
          CALL DIFF(DFX,LS,LD,ND,DFX)
           IF (ND.EQ.2) THEN
              CALL DIFF(DFX,LD,LB,ND,DFX)
              CALL REMAV(DFX,LB,XAVG)
              CALL FORAC(DFX,LB,MXLAG,ACOR(J))
          ELSE
           CALL REMAV(DFX,LD,XAVG)
           CALL FORAC(DFX,LD,MXLAG,ACOR(J))
          ENDIF
        ELSE
         CALL REMAV(DFX,LS,XAVG)
         CALL FORAC(DFX,LS,MXLAG,ACOR(J))
        ENDIF
      ELSE
      CALL REMAV(X,LX,XAVG)
      CALL FORAC(X,LX,MXLAG,ACOR(J))
      ENDIF
      SUM=0.0
       DO 20 J=2,JMAX-1
       SUM=SUM+(ACOR(J)/ACOR(1))**2
      Y(J)=SUM
 20   CONTINUE
      XS=SUM*LX
      WRITE(7,51)LX,MXLAG,NS,XAVG,ND,XS
      WRITE(7,*)'LAG      RES-ACF       +2SE        -2SE
      WRITE(7,*)'====     ========     =======     ======
      DO 50 J=2,JMAX-1
      SE=SQRT((1+2*Y(J))/LX)
      WRITE(7,100)J-1,ACOR(J)/ACOR(1),2*SE,-2*SE
 50   CONTINUE
 100  FORMAT(1X,I3,6X,F5.2,8X,F5.2,7X,F5.2)
 51   FORMAT(1X,'NO OF OBSERVATIONS=',I5,3X,'MAX. LAG OF ACF =',I3,/1X
     *,'SEASONAL PERIOD =',I3,3X,'AVERAGE OF RESIDUALS=',F8.2,/1X,'DEGRE
     *E OF REGULAR DIFFERENCING',I3,2X,'CHI**2 STATISTIC=',F6.2,/ )
      CLOSE(7,STATUS='KEEP')
      CLOSE(8,STATUS='KEEP')
      ENDIF
```

```
      END
      SUBROUTINE FORAC(X,LX,MXLAG,ACOR)
      DIMENSION X(*),ACOR(*)
      JMAX=MXLAG+2
      DO 20 J=1,JMAX
       SUM=0.0
       NMAX=LX-J+2
        DO 10 I=1,NMAX
         K=J+I-1
          SUM=SUM+X(I)*X(K-1)
10      CONTINUE
       ACOR(J)=SUM
20    CONTINUE
      RETURN
      END
      SUBROUTINE REMAV(X,LX,XAVG)
      DIMENSION X(*)
      SUM=0.0
      DO 10 I=1,LX
      SUM=SUM+X(I)
10    CONTINUE
      XAVG=SUM/LX
      DO 20 I=1,LX
      X(I)=X(I)-XAVG
20    CONTINUE
      RETURN
      END
      SUBROUTINE DIFF(X,LX,LS,NS,DFX)
      DIMENSION X(*),DFX(*)
      LS=LX-NS
      DO 10 I=1,LS
      DFX(I)=X(I+NS)-X(I)
10    CONTINUE
      END
      END
```

```
      program FCAST
c     This program generates Load forecasts for the next 24 hours
c     -----------------------------------------------------------------
c     array x and y store the past hourly load values, while array xf
c     stores the load forecasts, the lead times being 24 hours.
c     =================START OF THE MAIN ROUTINE=====================
      dimension x(550),y(550),xf(550,25),Hrs(24)
c     The datehr file stores the date and the hour the forecast was last
c     made.
      open(7,file='datehr',status='old')
      read(7,*)time,kday,kmonth,kyear
      close(7)
      write(*,190)time,kday,kmonth,kyear
 190  format(1x,'The last time a Forecast was made was at',1x,f5.2,2x,'h
     ^ours on',i3,'-',i3,'-',i4)
      write(*,*)'Enter the present hour <e.g 16.00>
      read(*,*)time
      write(*,*)'Enter todays date,Thus<day,month,year>  '
      read(*,*)kday,kmonth,kyear
      write(*,195)
 195  format(5x,'***** The Load Forecast MUST be done At Least ONCE Ever
     ^y Day *****')
      write(*,200)
 200  format(15X,'==================================================='
     ^/15x,':',5x,'WELCOME TO THIS FORECASTING SESSION',11x,':'/15X,':',
     ^15x,'COURTESY OF ORERO',19x,':'/15X,'============================
     ^======================= /)
c     (For this algorithm forecasts  have to begin at time origin greater
c     than 339 due to the nature of the model.
c     The lead time varies from 1 hr to a maximum of 24 hours.'
c     The Accurate Forecasts can only be obtained for forecast origins
c     beyond (3*NS=504)  Since there is some basic minimum starting
c     data required.
c                     }

      write(*,*)'How many HOURS have elapsed since the last Forecast was
     ^ made ?<e.g 4>  '
      read(*,*)km
      if (km.eq.1)then
      write(*,*)'ENTER The Latest hours average Load reading in MW
      else
        if (km.gt.1)then
         write(*,204)km
        else
           if (km.lt.1) go to 300
        endif
 204  format(5x,'ENTER The Latest',2x,i3,2x,'hours,of hourly load readin
     ^gs (in MW)'/5x,'Starting with earliest one recorded.')
      endif
      read(*,*)(hrs(ii),ii=1,km)
      write(*,206)
 206  format(5x,'ARE YOU SURE ALL YOUR READINGS ARE CORRECT?'/5x,'If cor
     ^rect then enter 1,otherwise enter 0   ')
      read(*,*)ikt
      if (ikt.gt.1.or.ikt.lt.1) then
      write(*,*)'WRONG DATA ENTRY START THE FORECASTING PROCESS A FRESH
     ^AND BE MORE ACCURATE'
      else
      write(*,207)
 207  format(/1x,'!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
     ^!!!!!!!!!!!!!!!!!!!!!!'/1x,'WAIT PLEASE,FORECASTS ARE BEING GENER
```

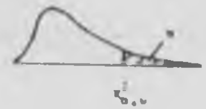
```
     ^ATED,THEN OBTAIN RESULTS FROM FILE FCOUT'/1x,'********************
     ^*************************************************************')
      open(7,file='datehr',status='old')
      write(7,*)time,kday,kmonth,kyear
      close(7,status='keep')

        LX=504
        L=24
        LT=504
       open(5,file='data',status='old')
         read(5,*)(y(it),it=1,Lx)
         close(5,status='keep')
        open(6,file='!data',status='old')
         do 210 it=1,Lx-km
         x(it)=y(it+km)
         write(6,205)x(it)
 210     continue
         do 230 ii=1,km
         write(6,205)hrs(ii)
 230     continue
 205     format(20x,f5.1)
        close(6,status='keep')
       open(6,file='!data',status='old')
        read(6,*)(x(it),it=1,Lx)
        close(6,status='keep')
      open(5,file='data',status='old')
        do 220 it=1,lx
        write(5,205)x(it)
 220    continue
        close(5,status='keep')
      open(9,file='fcout',status='old')

c     { inputting the model constants p1,p2,p3,p4 and nsr,ns }
c     {ns=seasonal period and nsr=degree of seasonal autoregression}
      p1=0.3
      p2=0.8
      p3=0.1
      p4=0.1
      ns=168
      nsr=24

c     {Starting the forecasting process}

      It=ns+1
        Xf(It,1)=x(It)+x(It+1-ns)-x(It-ns)+p3*(x(It+1-nsr)-x(It-nsr))

      It=ns+2
        Xf(It,1)=x(It)+x(It+1-ns)-x(It-ns)-p1*(x(It)-xf(It-1,1))
     ^   +p3*(x(It+1-nsr)-x(It-nsr))

      DO 10 IT=NS+3,NS+NSR+3
        XF(IT,1)=X(IT)+X(IT+1-NS)-X(IT-NS)-P1*(X(IT)-XF(IT-1,1))
     ^   +P3*(X(IT+1-NSR)-X(IT-NSR))-P4*(X(IT-1)-XF(IT-2,1))
 10   CONTINUE

      DO 20 IT=NS+NSR+4,NS+NS
       XF(IT,1)=X(IT)+X(IT+1-NS)-X(IT-NS)-P1*(X(IT)-XF(IT-1,1))
     ^  +P3*(X(IT+1-NSR)-X(IT-NSR))-P4*(X(IT-1)-XF(IT-2,1))
     ^      +P3*(X(IT-NS-NSR)-X(IT+1-NS-NSR))
 20   CONTINUE

      DO 30 IT=NS+NS+1,NS+NS+2
```

```
      XF(IT,1)=X(IT)+X(IT+1-NS)-X(IT-NS)-P1*(X(IT)-XF(IT-1,1))
     ^    +P3*(X(IT+1-NSR)-X(IT-NSR))-P4*(X(IT-1)-XF(IT-2,1))
     ^   +P3*(X(IT-NS-NSR)-X(IT+1-NS-NSR))-P2*(X(IT+1-NS)-XF(IT-NS,1))
30    CONTINUE

      DO 40 IT=NS+NS+3,LT
      XF(IT,1)=X(IT)+X(IT+1-NS)-X(IT-NS)-P1*(X(IT)-XF(IT-1,1))-
     &   P2*(X(IT+1-NS)-XF(IT-NS,1))+P3*(X(IT+1-NSR)-X(IT-NSR)+X(IT
     &   -NSR-NS)-X(IT+1-NSR-NS))-P4*(X(IT-1)-XF(IT-2,1))+P1*P2*(X(IT
     &     -NS)-XF(IT-NS-1,1))+P2*P4*(X(IT-NS-1)-XF(IT-NS-2,1))

      K=2
        XF(IT,K)=XF(IT,K-1)+X(IT+K-NS)-X(IT+K-NS-1)-P2*(X(IT+K-NS)
     &   -XF(IT+K-NS-1,1))+P3*(X(IT+K-NSR)-X(IT+K-NSR-1)+X(IT+K-NSR
     &   -NS-1)-X(IT+K-NSR-NS))-P4*(X(IT+K-2)-XF(IT+K-3,1))+P1*P2*(X(
     & IT+K-NS-1)-XF(IT+K-NS-2,1))+P2*P4*(X(IT+K-NS-2)-XF(IT+K-NS-3,1))

      DO 35 K=3,L
      XF(IT,K)=XF(IT,K-1)+X(IT+K-NS)-X(IT+K-NS-1)-P2*(X(IT+K-NS)-X
     & F(IT+K-NS-1,1))+P3*(X(IT+K-NSR)-X(IT+K-NSR-1)+X(IT+K-NSR-NS-
     & 1)-X(IT+K-NSR-NS))+P1*P2*(X(IT+K-NS-1)-XF(IT+K-NS-2,1))+P2*P4*
     &   (X(IT+K-NS-2)-XF(IT+K-NS-3,1))
35      continue
40    continue
C      {calculating the mean Load over Forecasting period}

      sum=0.0
      do 65 m=1,lx
       sum=sum+x(m)
65    continue
      average=sum/lx
C                 {outputting the forecast results  }

      error=(x(LX)-XF(LX-1,1))*100/X(LX)

      write(9,120)
120   format(1x,'Time origin',4x,'lead time',3x,'Time(Hrs)',4x,'Load fo
     ^recast(MW)'/1x,'===========',4x,'=========',3x,'=======',6x,'=====
     ^==========')

      I=Lt
      do 160 J=1,L
       timeTotal=time+J
       if(timeTotal.GE.24)then
       timesum=timeTotal-24
       else
       timeSum=timeTotal
       endif
       write(9,100)time,J,timeSum,XF(I,J)
160    continue
      write(9,110)average,error
110   format(2I,/1x,'Average Load=',F5.1,2x,'MW',3x,'One step Ahead Fore
     ^casting Error(at)=',f5.1,'%'/1x,'**********************************
     ^*************************************')
100   format(4x,f5.2,10x,i3,8x,f5.2,10x, f5.1)

      Close(9,Status='keep')
      endif
300   End
```

Table 8

PERCENTAGE POINTS OF THE χ^2 DISTRIBUTION CHI SQUARE

Table of $\chi^2_{\alpha;\nu}$ — the 100 α percentage point of the χ^2 distribution for ν degrees of freedom



α =	.995	.99	.98	.975	.95	.90	.80	.75	.70	.50	.30	.25	.20	.10	.05	.025	.02	.01	.005	.001	= α
ν = 1	$0^{4}393$	$0^{3}157$	$0^{3}628$	$0^{3}982$	.00393	.0158	.0642	.102	.148	.455	1.074	1.323	1.642	2.706	3.841	5.024	5.412	6.635	7.879	10.827	ν = 1
2	.0100	.0201	.0404	.0506	.103	.211	.446	.575	.713	1.386	2.408	2.773	3.219	4.605	5.991	7.378	7.824	9.210	10.597	13.815	2
3	.0717	.115	.185	.216	.352	.584	1.005	1.213	1.424	2.366	3.665	4.108	4.642	6.251	7.815	9.348	9.837	11.345	12.838	16.268	3
4	.207	.297	.429	.484	.711	1.064	1.649	1.923	2.195	3.357	4.878	5.385	5.989	7.779	9.488	11.143	11.668	13.277	14.860	18.467	4
5	.412	.554	.752	.831	1.145	1.610	2.343	2.675	3.000	4.351	6.064	6.626	7.289	9.236	11.070	12.832	13.388	15.086	16.750	20.517	5
6	.676	.872	1.134	1.237	1.635	2.204	3.070	3.455	3.828	5.348	7.231	7.841	8.558	10.645	12.592	14.449	15.033	16.812	18.548	22.457	6
7	.989	1.239	1.564	1.690	2.167	2.833	3.822	4.255	4.671	6.346	8.383	9.037	9.803	12.017	14.067	16.013	16.622	18.475	20.278	24.322	7
8	1.344	1.646	2.032	2.180	2.733	3.490	4.594	5.071	5.527	7.344	9.524	10.219	11.030	13.362	15.507	17.535	18.168	20.090	21.955	26.125	8
9	1.735	2.088	2.532	2.700	3.325	4.168	5.380	5.899	6.393	8.343	10.656	11.389	12.242	14.684	16.919	19.023	19.679	21.666	23.589	27.877	9
10	2.156	2.558	3.059	3.247	3.940	4.865	6.179	6.737	7.267	9.342	11.781	12.549	13.442	15.987	18.307	20.483	21.161	23.209	25.188	29.588	10
11	2.603	3.053	3.609	3.816	4.575	5.578	6.989	7.584	8.148	10.341	12.899	13.701	14.631	17.275	19.675	21.920	22.618	24.725	26.757	31.264	11
12	3.074	3.571	4.178	4.404	5.226	6.304	7.807	8.438	9.034	11.340	14.011	14.845	15.812	18.549	21.026	23.337	24.054	26.217	28.300	32.909	12
13	3.565	4.107	4.765	5.009	5.892	7.042	8.634	9.299	9.926	12.340	15.119	15.984	16.985	19.812	22.362	24.736	25.472	27.688	29.819	34.528	13
14	4.075	4.660	5.368	5.629	6.571	7.790	9.467	10.165	10.821	13.339	16.222	17.117	18.151	21.064	23.685	26.119	26.873	29.141	31.319	36.123	14
15	4.601	5.229	5.985	6.262	7.261	8.547	10.307	11.036	11.721	14.339	17.322	18.245	19.311	22.307	24.996	27.488	28.259	30.578	32.801	37.697	15
16	5.142	5.812	6.614	6.908	7.962	9.312	11.152	11.912	12.624	15.338	18.418	19.369	20.465	23.542	26.296	28.845	29.633	32.000	34.267	39.252	16
17	5.697	6.408	7.255	7.564	8.672	10.085	12.002	12.792	13.531	16.338	19.511	20.489	21.615	24.769	27.587	30.191	30.995	33.409	35.718	40.790	17
18	6.265	7.015	7.906	8.231	9.390	10.865	12.857	13.675	14.440	17.338	20.601	21.605	22.760	25.989	28.869	31.526	32.346	34.805	37.156	42.312	18
19	6.844	7.633	8.567	8.907	10.117	11.651	13.716	14.562	15.352	18.338	21.689	22.718	23.900	27.204	30.144	32.852	33.687	36.191	38.582	43.820	19
20	7.434	8.260	9.237	9.591	10.851	12.443	14.578	15.452	16.266	19.337	22.775	23.828	25.038	28.412	31.410	34.170	35.020	37.566	39.997	45.315	20
21	8.034	8.897	9.915	10.283	11.591	13.240	15.445	16.344	17.182	20.337	23.858	24.935	26.171	29.615	32.671	35.479	36.343	38.932	41.401	46.797	21
22	8.643	9.542	10.600	10.982	12.338	14.041	16.314	17.240	18.101	21.337	24.939	26.039	27.301	30.813	33.924	36.781	37.659	40.289	42.796	48.268	22
23	9.260	10.196	11.293	11.688	13.091	14.848	17.187	18.137	19.021	22.337	26.018	27.141	28.429	32.007	35.172	38.076	38.968	41.638	44.181	49.728	23
24	9.886	10.856	11.992	12.401	13.848	15.659	18.062	19.037	19.943	23.337	27.096	28.241	29.553	33.196	36.415	39.364	40.270	42.980	45.558	51.179	24
25	10.520	11.524	12.697	13.120	14.611	16.473	18.940	19.939	20.867	24.337	28.172	29.339	30.675	34.382	37.652	40.646	41.566	44.314	46.928	52.620	25
26	11.160	12.198	13.409	13.844	15.379	17.292	19.820	20.843	21.792	25.336	29.246	30.434	31.795	35.563	38.885	41.923	42.856	45.642	48.290	54.052	26
27	11.808	12.879	14.125	14.573	16.151	18.114	20.703	21.749	22.719	26.336	30.319	31.528	32.912	36.741	40.113	43.194	44.140	46.963	49.645	55.476	27
28	12.461	13.565	14.847	15.308	16.928	18.939	21.588	22.657	23.647	27.336	31.391	32.620	34.027	37.916	41.337	44.461	45.419	48.278	50.993	56.893	28
29	13.121	14.256	15.574	16.047	17.708	19.768	22.475	23.567	24.577	28.336	32.461	33.711	35.139	39.087	42.557	45.722	46.693	49.588	52.336	58.302	29
30	13.787	14.953	16.306	16.791	18.493	20.599	23.364	24.478	25.508	29.336	33.530	34.800	36.250	40.256	43.773	46.979	47.962	50.892	53.672	59.703	30
40	20.706	22.164	23.838	24.433	26.509	29.051	32.345	33.660	34.872	39.335	44.165	45.616	47.269	51.805	55.759	59.342	60.436	63.691	66.766	73.402	40
50	27.991	29.707	31.664	32.357	34.764	37.689	41.449	42.942	44.313	49.335	54.723	56.334	58.164	63.167	67.505	71.420	72.613	76.154	79.490	86.661	50
60	35.535	37.485	39.699	40.482	43.188	46.459	50.641	52.294	53.809	59.335	65.227	66.981	68.972	74.397	79.082	83.298	84.580	88.379	91.952	99.607	60
70	43.275	45.442	47.893	48.758	51.739	55.329	59.898	61.698	63.346	69.334	75.689	77.577	79.715	85.527	90.531	95.023	96.388	100.425	104.215	112.317	70
80	51.171	53.539	56.213	57.153	60.391	64.278	69.207	71.145	72.915	79.334	86.120	88.130	90.405	96.578	101.880	106.629	108.069	112.329	116.321	124.839	80
90	59.196	61.754	64.634	65.646	69.126	73.291	78.558	80.625	82.511	89.334	98.524	98.650	101.054	107.565	113.145	118.136	119.648	124.116	128.299	137.208	90
100	67.327	70.065	73.142	74.222	77.929	82.358	87.945	90.133	92.129	99.334	106.906	109.141	111.667	118.498	124.342	129.561	131.142	135.807	140.170	149.449	100

For values of $\nu > 30$ approximate values for χ^2 may be obtained from the expression $\nu\left[1-\frac{2}{9\nu}+\frac{x}{\sigma}\sqrt{\frac{2}{9\nu}}\right]^3$, where $\frac{x}{\sigma}$ is the normal deviate cutting off the corresponding tails of a normal distribution. If $\frac{x}{\sigma}$ is taken at the 0.02 level, so that 0.01 of the normal distribution is in each tail, the expression yields χ^2 at the 0.99 and 0.01 points. For very large values of ν it is sufficiently accurate to compute $\sqrt{2\chi^2}$, the distribution of which is approximately normal around a mean of $\sqrt{2\nu-1}$ and with a standard deviation of 1. This table is taken by consent from Statistical Tables for Biological, Agricultural, and Medical Research, by R. A. Fisher and F. Yates, published by Oliver and Boyd, Edinburgh, and from Table 8 of Biometrika Tables for Statisticians, Vol. 1, by permission of the Biometrika Trustees.