ISBN Number 0 947447 79 2
WRC 200/1/90

A STOCHASTIC DAILY CLIMATE MODEL FOR SOUTH AFRICAN CONDITIONS

BY

ANABELA DE GUSMÃO BRANDÃO

AND

WALTER ZUCCHINI

REPORT DESCRIBING A RESEARCH PROJECT CARRIED OUT BY THE DEPARTMENT OF MATHEMATICAL STATISTICS, UNIVERSITY OF CAPE TOWN, UNDER CONTRACT TO THE WATER RESEARCH COMMISSION.

DEPARTMENT OF MATHEMATICAL STATISTICS
UNIVERSITY OF CAPE TOWN
SEPTEMBER 1990

EXECUTIVE SUMMARY

A STOCHASTIC DAILY CLIMATE MODEL FOR SOUTH AFRICAN CONDITIONS

by

Anabela de Gusmão Brandão and Walter Zucchini

Motivation

Effective water resources management is essential in a country like South Africa which is particularly prone to the adverse effects of drought. This will only become feasible when the risk associated with drought occurrences can be reliably assessed.

Present methods of assessing the risk of adverse weather conditions are based on rainfall and streamflow only and do not take account of the many other climatic factors such as evaporation, humidity, wind run, temperature etc. Such factors play an important role in establishing drought conditions, especially in the agricultural sector.

Methods, such as those based on the Palmer drought index, are purely descriptive and are designed to quantify what has happened in the past rather than what is likely to happen in the future. These methods are therefore of limited use for planning purposes.

This project arose from a need to develop reliable methods to generate artificial climate sequences over any period of the year and thereby enable water resources and agricultural planners to assess the probable consequences of decisions whose outcomes depend on climate factors. For example, sequences generated by a suitable model could be used as the input to plant yield models associated with crops such as maize, wheat and sugar cane, and thereby provide the probability distribution of yield under alternative options regarding, for example, planting date, cultivar and irrigation strategy.

The climate model to be developed in the course of this project was seen as a logical extension of the daily rainfall model which was developed in a previous Water Research Commission project (WRC Report No. 91/1/84 – 91/3/84). The latter model has been used by a number of institutions involved in Forestry, Agriculture, Nature Conservation, as well as by individual researchers at a number of universities and the South African Museum. It is offered as one of the data products available from the Computing Centre

ACKNOWLEDGEMENTS

This report resulted from a research project financed by the Water Research Commission and entitled "Research on the development of a stochastic daily climate model for South African conditions".

The steering Committee responsible for the project consisted of the following members:

Dr G C Green	Water Research Commission (Chairman)
Mr P W Weideman	Water Research Commission (Secretary)
Dr P C M Reid	Water Research Commission
Prof T J Stewart	University of Cape Town
Mr S van Biljon	Dept of Water Affairs
Mr J Myburgh	Dept of Agricultural Development
Dr A L du Pisani	Dept of Environmental Affairs
Mr D B Versfeld	FORESTEK, CSIR
Mr S D Lynch	University of Natal

The financial support of the WRC and the contributions of the Steering Committee are acknowledged with thanks.

We wish to express our thanks to the individuals whose advice and cooperation contributed to the completion of this project, especially the staff of the Department of Mathematical Statistics at the University of Cape Town. In particular we wish to thank Professor Theo Stewart for participating in the running of the project during 1989 and 1990.

Thanks are due to the Weather Bureau, the Computing Centre for Water Research and the Winter Rainfall Region, Department of Agriculture and Water Affairs for their cooperation in providing the historical records used in this report.

We wish to thank the Steering Committee for their guidance and support.

Finally we wish to thank Mrs M I Cousins for typing the report, and especially, for making such a fine job of typesetting the mathematical formulae.

of Water Research. The climate model, incorporating several additional variables, would therefore supplement the rainfall model.

Objectives

The objective of this project was to develop a stochastic model for the simultaneous description of climate variables at fixed locations on a daily basis. The variables to be analysed were rainfall, sunshine duration, maximum and minimum temperature, maximum and minimum relative humidity, evaporation and wind run.

Once a suitable model was identified the object was to develop methodology to estimate the relevant parameters from a given historical record and then to develop algorithms to generate artificial daily climate sequences at the given site.

A further objective of the project was that the technicalities of the methodology developed should be transparent to the user, that is, the results should be accessible to users with limited or no knowledge of statistics.

Summary of results

We investigated the properties of the only daily climate model (Model 1) that has been described in the literature. A number of limitations of this model were identified and four alternative models were constructed, Models 2, 3, 4 and 5. (Model 2 was designed as a prototype for the subsequent models and is described in the report for the sake of completeness rather than as a suitable model in its own right.)

The new models, which vary in complexity, are designed to form a compatible family. This allows one to select a model of appropriate complexity for the particular historical record that is available. In general the simpler models outperform more complex models when the historical record is short (as is presently the case at almost all sites in South Africa) whereas the latter can be expected to become increasingly applicable as more data becomes available. Furthermore the compatibility property allows one to model the different climate variables using components from any one of Models 3, 4 and 5 and then to combine these into a single multivariate daily climate model.

To fulfil its purpose a daily climate model must incorporate all the important properties exhibited by climate variables. These include the seasonal cyclical behaviour of climate, its short–term persistence, the interrelationships between the different variables (for example between rainfall and humidity) and the boundedness of some of the variables (for example the upper and lower limits of maximum and minimum relative humidity). In addition the behaviour of each of the variables on wet days is different to that on dry days. For example, on average, the maximum temperature on dry days is higher than it is on wet days. All these properties have to be preserved, not only qualitatively, but also quantitatively by the climate model.

The results of this project confirm that it is indeed possible to construct models that preserve the above properties. This is in spite of the fact that the historical records which are presently available in South Africa are extremely short for the purpose of modelling a process of the complexity of daily climate. (The length of the records available to us ranged from 6 to 12 years.) An additional factor which reduces the *effective* length of the records for this type of modelling is the average number of *rainy* days which, in many parts of South Africa, is quite small.

The models were calibrated at six sites, namely Elsenburg (South Western Cape), Kakamas (Northern Cape), Middelburg (Eastern Central Cape), Nelspruit (Eastern Transvaal), Cedara (Natal) and Hoopstad (Orange Free State) which, within the constraints of the data available to us, were selected to represent as wide a variety of climate types as possible. Extensive validation tests were carried out and our results show that, on the whole, the models perform remarkably well.

There is no clear–cut answer regarding which model will perform best. As mentioned, one would expect the simpler models to outperform the more complex alternatives when the data records are short. For some sites Model 1 preserves the properties of some of the climate variables better than the more complex alternative models. At other sites the opposite was found to be the case. We therefore recommend that, at new sites, each of the models be applied and tested before a final selection is made.

A major theoretical obstacle that had to be overcome in the course of the project was that of developing methods which could accommodate records with missing observations, with invalid recordings and with outlying observations. Although we had access to some of the best historical records that are available in South Africa, there were considerable gaps and imperfections in these records (amounting to between 1% and 13% of the total record lengths for the records which we examined). As climate variables are both serially correlated and cross–correlated it is not possible to simply ignore missing values. Methods had to be developed to incorporate the estimation of missing values as part of the parameter estimation procedure.

A substantial portion of the research effort in this project was directed to deriving the mathematical theory for the climate models which were developed. This material, which is rather technical and thus is not accessible to the general user, includes the development of estimation methods both for the individual climate variables as individual time series models and then for the multivariate series which combines these models so as to synchronise the various climate variables.

The second major component of the project was the preparation of computer programs to implement the theory. In order to make the software accessible to as wide a variety of users as possible it was decided at the outset that all programs would be such that they could be implemented on micro–computers. Secondly, it was decided that no use should be made of licensed software packages which may not be available to some potential users. Thus the programs which are listed in this report are self–contained and are coded in ANSI FORTRAN 77, (the HUGE attribute in programs 6 and 8 is an extension to the full ANSI standard but this can be omitted without any problems on a mainframe) a language for which compilers are generally available. This includes the programs to estimate the parameters and to generate the required artificial climate sequences.

An objective of the project was that the results of the project should be accessible to users with limited or no knowledge of statistics. This objective has been mostly met, but with the following qualification. The programs that have been developed to generate the climate sequences are accessible by any user who can operate a micro–computer. Such a person would not have to know anything about programming but merely how to run an existing program. We envisage that most users of the methodology will only be interested in making use of the generating program.

Some training is required to apply the methodology at a new site, that is, to estimate the model parameters from a given historical record. We estimate that, with instruction, it would take a competent programmer between two to three weeks to learn how to make efficient use of the estimation software provided. Most of the training would be concerned with methods for preparing the data for estimation. This aspect of the methodology simply cannot be automated since it requires judgement.

Thus we must distinguish between two types of users; those who wish to calibrate the model for a new site and those who wish to use the model for a site that has already been calibrated. The former task requires some training but the latter does not. This issue is discussed in the recommendations below.

We believe that the main objectives of the project have been met. We have demonstrated that the models which were investigated in the course of this project meet all the requirements that can be reasonably expected of models for a phenomenon of the complexity of daily climate.

Recommendations

Quality of historical records

The main obstacle to the application of the techniques described in this report on a large scale is the lack of suitable historical records. This refers to both the quantity and the quality of available data. The records which were used for this report represent some of the best available in South Africa. Nevertheless, for the purpose of modelling daily climate, they are barely adequate. Although there is little that can be done to increase the length of records except to wait for more data to be collected, it should be possible to improve the quality of historical records. In particular it would be useful if some measure of the reliability of the observations were also recorded on a regular basis. As we have repeatedly pointed out in the body of the report, one of the problems which we encountered was that of identifying incorrect observations. This task would be considerably simplified if one had some index of reliability associated with (ideally) each recording or set of recordings.

Transfer of technology

For the methods developed in this project to realise their full potential it will be necessary to calibrate the models at many more sites. As was pointed out, no special training is required to use the programs for generating climate sequences once the parameters of the model have been estimated. However, some training is required to use the programs to prepare the data for estimation and to carry out the estimation for a new site.

We recommend that the Computing Centre for Water Research (CCWR) be approached to acquire the expertise to implement the estimation techniques and with the help of users, gradually build up a data base of estimates of the model parameters for as many sites as possible in South Africa. The CCWR already offer a similar data product, namely the parameter estimates of a daily rainfall model for 2550 sites in South Africa. These arose from a previous Water Research Commission project (Zucchini and Adamson (1984)). The

CCWR also offer the artificial rainfall generating program which can be applied to any of these sites. Thus the programs developed in the course of this project constitute a logical extension of a service that the CCWR already offer.

LIST OF CONTENTS

LIST OF DISPLAYS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Climate is a critical factor in determining the variety and abundance of vegetation and animal life that can exist in a region. It imposes limits on agricultural and other human activities that are economically feasible. Thus it is not surprising that various aspects of climate, such as precipitation, temperature, solar radiation, humidity, wind speed and others, are recorded on a regular basis throughout the world.

The purpose of measuring these climate variables is to extend our knowledge of the behaviour patterns of climate and thereby, among other things, to identify those activities which are feasible and to determine how these may be most profitably carried out. For this, one has to take account of the fact that both the climate process and human requirements, such as demand for water, are dynamic processes which are stochastic rather than deterministic in character. For example, the annual rainfall in most regions of South Africa varies considerably from year to year and it is obviously inadequate to base water–related decisions solely on the average annual rainfall; the entire distribution of annual rainfall needs to be considered.

Statistical theory provides an ideal framework for expressing our knowledge about the properties of climate. Firstly, it provides a means of quantifying our knowledge in a precise manner. Secondly being designed to describe stochastic phenomena, the theory provides a conceptual framework which accommodates notions such as uncertainty and risk, thereby providing a convenient basis for rational decision making in the face of uncertainty. Thirdly, statistical methodology provides an effective means of synthesising and analysing the information contained in large data sets such as daily climate records. In particular the theory enables one to quantitatively distinguish the systematic patterns in climate (such as the seasonal cycles) from the random fluctuations about these patterns and to express this information in terms of a statistical model.

As well as providing a concise description of the patterns that exist in the different components of climate, a statistical model can be used to generate artificial climate sequences which preserve the properties of real climate sequences, that is, artificial sequences that are indistinguishable from real climate sequences. Among other things, artificial climate sequences are useful as inputs to crop growth models which can then be used to determine the distribution of yield, the risk of crop failure due to adverse climate, optimal planting dates, the potential profitability of irrigation, and so on. For such purposes artificial climate sequences generated by a good stochastic model are more useful than the original historical record. Firstly, they are free of the typical imperfections which are especially prevalent in historical climate records, for example, incorrect recordings and missing observations. Secondly, the historical records presently available in South Africa are mostly quite short and thus only reflect a small fraction of the different climate sequences that could occur.

It is sometimes argued that artificial sequences generated by a stochastic model constitute no more than complicated extrapolations of the historical record. However, a model contains more than the information that can be extracted from a single historical record. It contains our knowledge (in the form of model assumptions) about the behaviour of climate derived from theory and from observations at other locations. For example, it is reasonable to assume that certain average properties of climate variables vary smoothly with time. Such assumptions give the model a structure which may not be evident in a single short historical record.

The main objective of this project has been to develop a stochastic daily climate model for South African conditions. The time resolution was taken as one day because climate data commonly available are recorded on a daily basis. The variables included in the model are rainfall, maximum and minimum temperature, maximum and minimum relative humidity, evaporation, wind run and sunshine. In fact the models that have been developed can be used to model a subset of the set of variables in cases where some of the above set are not available. Alternatively, it is possible to augment this list if measurements on additional variables are available.

As already mentioned the model needs to preserve the important properties of daily climate sequences. These were identified as being:

a) **Seasonality.** Each of the climate variables exhibits seasonal behaviour, that is, the recordings fluctuate about a curve which has a cyclical pattern with a period of one year. The shape of the curve is approximately sinusoidal which suggests that it can be parsimoniously approximated by a truncated form of its Fourier representation.

b) **Wet/dry day effect.** The probability distribution of the climate variables on wet days is different from their distribution on dry days. For example, the maximum relative humidity is generally higher on wet days than on dry days; and the opposite is true for the number of sunshine hours. Thus it is necessary to treat dry days and wet days differently in the model.

c) **Autocorrelation.** The individual variables exhibit short–term persistence over and above that attributable to seasonality. Generally there is a positive correlation between readings on successive days. This type of persistence needs to be incorporated into the model.

d) **Cross–correlation.** Apart from the wet/dry day effect already mentioned, the variables are cross–correlated. For example, there is a positive correlation between minimum temperature and maximum temperature on the same day. To preserve this property it is not possible to model the climate variables separately — they have to be modelled jointly.

e) **Boundedness.** The values of some of the variables are bounded, for example, relative humidity lies in the range 0% to 100%. Other variables are bounded with respect to others, for example, the minimum temperature on any one day must not be higher than the maximum temperature on the same day. To preserve this type of property the variables have to be transformed.

f) **Non–normality.** The probability distribution of climate variables does not follow the normal distribution. This is problematic because there is practically no other multivariate distribution available that is both sufficiently flexible and mathematically tractable to deal with a phenomenon as complex as climate. It is therefore necessary to transform the variables to achieve normality.

Taken together these properties indicate that we are dealing with a multivariate time series which is non–stationary and which contains a number of variables with special properties. In particular rainfall has the property that it is partly discrete (there is a non–zero probability that it does not rain) and partly continuous (the rainfall depth on rainy days is a continuous random variable). There is no standard statistical model which can be applied directly to such a multivariate time series. A special model has to be constructed for the daily climate process.

Although there is an extensive literature on the modelling of daily rainfall sequences, apart from Richardson (1981), very little work has been reported on models which describe the joint probability distribution of several components of climate. Richardson proposed separating the observations into two sequences; one for observations which occurred on wet days and one for those which occurred on dry days. For each of these two sequences the seasonal mean and standard deviation of each variable is estimated separately and then a new time series (of residuals) is obtained by deseasonalising the original observations using the appropriate means and standard deviations, depending on whether the observations occurred on wet days or on dry days. The (multivariate) time series computed in this way has mean zero and variance unity and is modelled using a single multivariate autoregressive model.

Richardson's model (Model 1) is probably the simplest structure that has the potential of preserving properties (a) to (d), outlined above, in a sufficiently flexible form. To accommodate (e) and (f) it is necessary to make suitable transformations of the variables at the very start of the modelling procedure. Such transformations are required for all the models which were considered.

At the start of the project Model 1 was fitted to six years of record (1979–1984) at Elsenburg. The model was found to fit some aspects of the historical records quite well but performed poorly on certain other aspects. In particular the annual standard deviation for wind run, maximum and minimum humidity were systematically underestimated. The (lagged) cross–correlations between some of the variables (e.g. maximum temperature and minimum temperature) were not preserved by the model. However the most noticeable deficiency was found to be that the model did not preserve the serial correlation structure of many of the variables. This was attributed to the lack of flexibility of Model 1 in this respect. In particular the model is based on the assumption that the serial correlation function does not depend on the wet/dry status of the days in question. In fact the correlation between variables on two successive days depends on whether the two days are both wet, both dry, wet followed by dry or dry followed by wet. It was therefore decided to develop a model which incorporates additional flexibility in its autocorrelation function, that is, a model which allows for the serial correlations between variables on successive days to depend on their wet/dry status.

In developing a model for a process as complex as daily climate there are two conflicting objectives. On the one hand it is desirable to construct a model that is as flexible as possible so that it can accommodate as many of the special features of the process as possible. On the other hand additional flexibility can only be achieved by increasing the number of unknown parameters in the model. These parameters have to be estimated from the historical record. Now, for a record of given length, increasing the number of parameters that has to be estimated decreases the precision of the estimates, on average. Put differently, if a model has too many parameters it becomes too specific to the particular historical record that is available and less representative of the population of typical climate sequences that could arise. The appropriate complexity of a model depends on the length of the historical record. In general, simpler models which depend on only a small number of parameters will outperform more complex models if the historical record is short, but the reverse is true if the record is large. A second issue is that some of the climate variables are more appropriately modelled by simpler structures than others. It is therefore not always optimal to use the same model for all the components of climate.

The strategy that we adopted to circumvent the above difficulties was to develop a *family of models* of varying degrees of complexity ranging from the simplest feasible model to more complex alternatives. This allows one to select the particular model from the family which is most appropriate for the historical record that is available. In addition the family which was developed is such that the individual models within the family are compatible in the following sense. One can use different submodels for each of the individual climate variables and then combine these into a multivariate model at the last stage of the modelling procedure. Thus, for example, it is possible to fit a simple model to wind run but a more complex model to minimum temperature. This compatibility feature of the family thus allows for additional flexibility.

Three compatible models were developed which we will refer to as Models 3, 4 and 5 (Model 2 was developed as a prototype to the others and is included in the report for the sake of completeness). Models 3 and 4 are two alternative relatively simple models whereas Model 5 is more general than each of them. Thus one would expect Models 3 and 4 to be suitable for short data records (as are presently available in South Africa) and Model 5 to become preferable as the historical data base increases in length. The method of maximum likelihood was used to estimate the parameters. Since the likelihoood equations are extremely complex, it was necessary to develop numerical methods to carry out the estimation. This involved deriving the first and second derivatives of the likelihood function (given in Chapter 3) and the development of procedures to compute initial estimates of the parameters.

One of the major problems which we encountered in applying the estimation procedures was the presence of missing observations and also of outliers, mainly in the form of incorrect readings (e.g. outside the admissible range of values). It is necessary to filter the data in order to remove such outliers before attempting to estimate the parameters — this introduces additional gaps in the record. Thus the estimation procedure that was developed had to be able to cope with the problem of missing values.

Some aspects of the lack of fit of Model 1 which were identified at the start of the project were later found to be attributed, at least in part, to outlying observations. In fact, a conclusion of this project is that in many respects Model 1 outperforms the more sophisticated models developed here. All the models considered here are strongly influenced by outlying observations. This fact makes it necessary to pay special attention to the quality of the historical record before attempting to fit a model.

In order to objectively determine which member of the family of models is most appropriate in a given situation a model selection criterion is used. The Akaike's Information Criterion (AIC) is proposed for this purpose.

Six sites were selected to evaluate the performance of the models considered in this report. The choice of the sites was, of course, constrained by the availability of suitable historical records. Within this constraint we attempted to represent, as well as possible, the various climate regions of South Africa. The sites chosen were:

Elsenburg — South West Cape

Kakamas — Northern Cape

Middelburg — Eastern Central Cape

Nelspruit — Eastern Transvaal

Cedara — Natal

Hoopstad — Orange Free State.

The aim of model validation is to establish that the models preserve the important properties of the historical records, at least to an appropriate degree, so that the generated sequences can be regarded as representative of the population of sequences which could occur.

An objective of this project was that the technicalities of the methodology should be transparent to the user, that is, the results should be accessible to users with limited or no knowledge of statistics. Here one must distinguish between the person who fits the model to observations at a new site and the person who uses the fitted model to generate artificial climate sequences. A limited amount of training is required to apply the estimation techniques using the software that is provided, but once the parameters of the model have been estimated at a particular site, the software provided to generate artificial climate sequences is accessible to anyone who can run a computer program.

In order to make the methodology accessible to as wide a variety of users as possible, it was decided at the outset of the project that all the software developed would have to be such that it could be implemented on micro computers. Secondly, it was decided that no use should be made of licensed software which may not be available to some potential users. The programs listed at the back of this report are self-contained — no additional software is required either to estimate the parameters or to generate artificial climate sequences.

This report is structured as follows:

The preliminary statistical analysis of the data is described in Chapter 2. This includes a description of the data, the types of difficulties encountered in detecting and dealing with faults in the data and the statistics computed to identify the structures present in the climate sequence.

Chapter 3 gives a theoretical description of the five climate models which were investigated and of the methods used to estimate the model parameters. The contents of the chapter are technical and very detailed and thus rather demanding of the reader. Fortunately it is not necessary to absorb all this detail in order to understand the remainder of the report. We recommend that the reader who is not concerned with the mathematical development of the models simply skip over this chapter. Details on the implementation of the models to the historical records are given in Chapter 4. The algorithms for implementing the theory are listed in Chapter 5. These include algorithms for generating artificial climate sequences. These algorithms are intended to bridge the gap between the formulae given in Chapter 3 and the FORTRAN programs given in Appendix D. Extensive tests were performed on the fitted models in order to assess their performance in preserving the important properties of climate sequences. The results of this model validation investigation are summarized in Chapter 6. A summary of the findings of the study and the main conclusions are given in Chapter 7.

There are 5 Appendices: Appendix A explains the choice of the Fourier approximation, L , Appendix B describes the properties of the Fourier series approximation, Appendix C gives an algorithm, known as the Cholesky decomposition, which rewrites a matrix as a product of a triangular matrix with its transpose. This is needed to generate normal random numbers with a covariance matrix Σ . Appendix D gives information on where a list of the ANSI FORTRAN 77 programs used in this study can be obtained. Appendix E describes the EM algorithm, a very general iterative method for maximum likelihood estimation in incomplete data sets. The EM algorithm is used in this study to estimate and fill missing values in the climate data sets.

CHAPTER 2

THE DATA SET AND PRELIMINARY ANALYSIS

It is to be expected that data records collected over a long period of time will contain gaps, and usually the number of gaps increases in proportion with the size of the data set. The data sets considered in this study are no exception to this.

Gaps occur for two reasons. Firstly, a high proportion of the observations are missing. Although missing observations are relatively easy to detect, they lead to complications in the analysis. In particular, the multivariate time series models considered here require simultaneous observations of all the variables. Furthermore, the serial correlation structure in the series does not allow one to simply discard observations as one would do if the observations were serially independently distributed.

Secondly, some of the readings are incorrect (or incorrectly recorded). These are often quite difficult to detect, especially if the values fall within the feasible range of the variable under consideration. This problem is particularly difficult to deal with satisfactorily.

This chapter describes the general format of the data sets used, some of the problems encountered and the method used to overcome them. Finally, some preliminary analyses performed for initial model identification are discussed.

The data set

(a) Format

The climate variables of interest are:

- rainfall (mm)
- maximum temperature ($^\circ C$)
- minimum temperature ($^\circ C$)
- A pan evaporation (mm)
- sunshine duration (hours)
- windrun (km/day)
- maximum humidity (%)
- minimum humidity (%)

Not all stations have records for evaporation, however, as it is easily derived from other climate variables, it is not essential to include it in the models. Whenever readings are available, evaporation is kept purely to demonstrate the elasticity of the models in that variables can be omitted or incorporated without the model structure changing. The number of variables is simply increased.

The unit of measurement for each variable is shown above in brackets following the variable name.

Three properties of the time series (discussed later) determine how the final data set for parameter estimation must be constructed. Firstly, simultaneous observations for all variables are required as one is dealing with a multivariate time series. As data collection of some variables (humidity, for example) has only been started recently, only years for which measurements are available for all variables simultaneously can be used. The only exception is rainfall as it is modelled independently of the other variables.

Secondly, continuous data is required because of the seasonality and serial–correlation structure in the time series. Large gaps in the data caused by shutting down a station for a long period of time and then reopening it at a later stage, cannot be treated as missing values. Only the sequence previous to or following the closing (depending on which period is longer) may be used.

Finally, records should begin on 1 January and end on 31 December. This restriction simplifies the algorithms and the programming. However, it is not necessary to waste data in order to meet this requirement if only a few months are missing in a year. For example, if the original available record starts on 1.2.1978, then one should code the days 1.1.1978–31.1.1978 as missing and then regard the record as starting on 1.1.1978.

(b) Quality

There is always a possibility of readings being recorded incorrectly. The type of recording error which can easily be identified is when the value recorded lies outside the permissible range, for example a recorded value for sunshine duration of 25 hours. In addition, a variable may have values which, although within a feasible region, are nevertheless incorrect. This cannot be established with any certainty. The fact that one is clearly dealing with data sets that are not "clean" and that preliminary work done showed the models to be extremely sensitive to "unclean" data, a thorough procedure to detect possible errors in the records is necessary.

From model assumptions, the residual time series obtained after fitting Model 1 (any other model can be used) to the climate series, is normally distributed with mean zero and standard deviation of unity. Therefore, 99.7% of the residual values should lie within the interval (-3, 3). In the present case, it is almost impossible to tell whether large residuals reflect model misfit or poor data, therefore large residuals were examined for possible occurrences of outliers. Preceding and succeeding values can give an indication whether or not these values should be considered as outliers. Observations across the variables at these times also show what patterns to expect. For example, Barry and Charley (1968) state that evaporation can be expressed by:

- duration of sunshine
- mean air temperature
- mean air humidity
- mean wind speed.

Thus, one would expect to see an increase of evaporation with an increase of sunshine duration.

Outliers are treated as missing values.

(c) Treatment of leap years

Whenever a leap year occurs, the value observed on the 29 February is added to the value observed on 28 February for the variables

- rainfall and
- evaporation.

For the variables

- maximum temperature
- minimum temperature
- sunshine duration
- windrun
- maximum humidity, and
- minimum himidity,

the mean of the observed values of the 29 February and 28 February replaces the observed value of 28 February.

If the 28 February has a missing value then it is replaced by the observed value of 29 February.

Distinctive features of the time series useful for model identification

The station chosen for preliminary analyses was Elsenburg in the Cape Province (Latitude 33°51′ , Longitude 18°50′). Any station could have been chosen for this purpose as they all display similar features.

(a) Seasonality

A simple moving average smooth was used to filter the series. This is given by

$$M_t = \frac{1}{2L+1} \sum_{\ell=-L}^{L} m_{t+\ell}$$

where m_t is the mean of the time series at time t , i.e.

$$m_t = \frac{1}{I} \sum_{i=1}^{I} x_{i,t}$$

where $x_{i,t}$ is the observation made at time t of the ith year,

$i = 1, 2, \ldots, I$, I being the number of years for which data is available,

$t = 1, 2, \ldots, 365$

and L is the lag.

Lags of 10, 25, 50 and 100 days were applied.

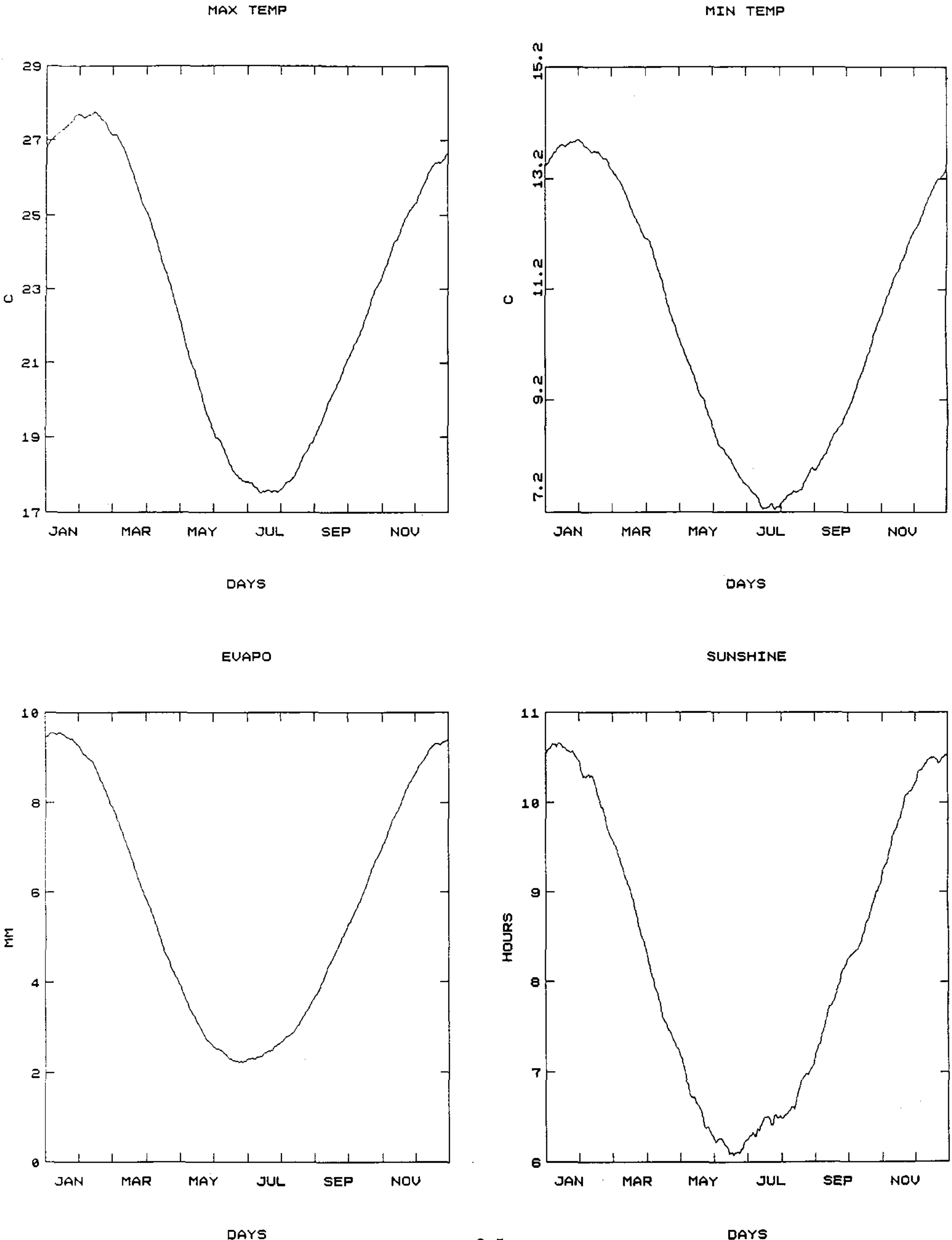
Note that in the above equation, because m_t is cyclic one has that

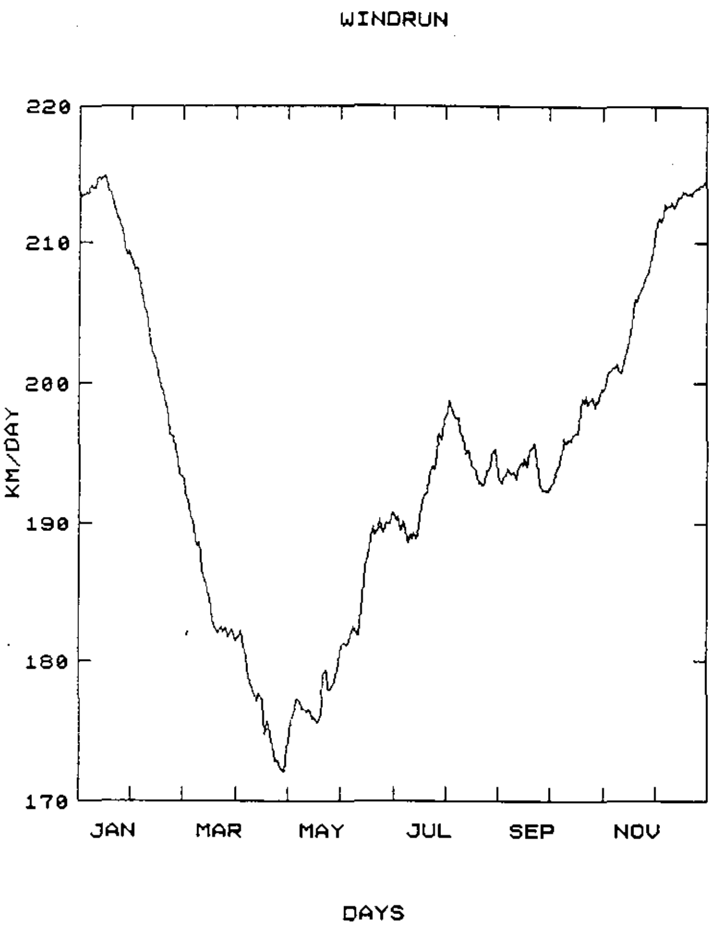
$$m_{366} = m_1$$
$$m_{367} = m_2$$
$$\vdots$$
$$m_0 = m_{365}$$
$$m_{-1} = m_{364}$$

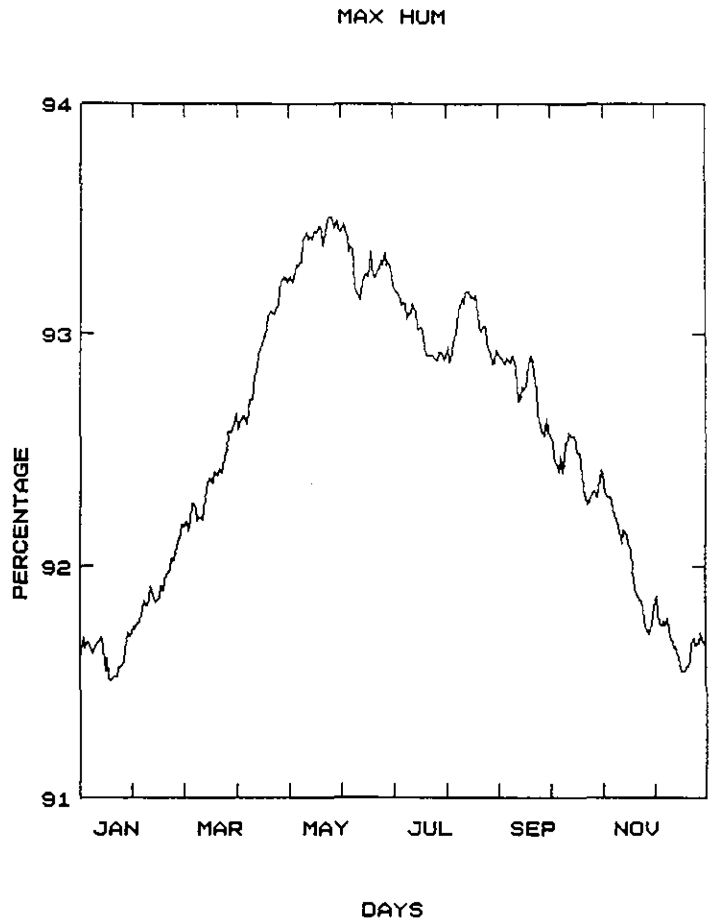
and so on.

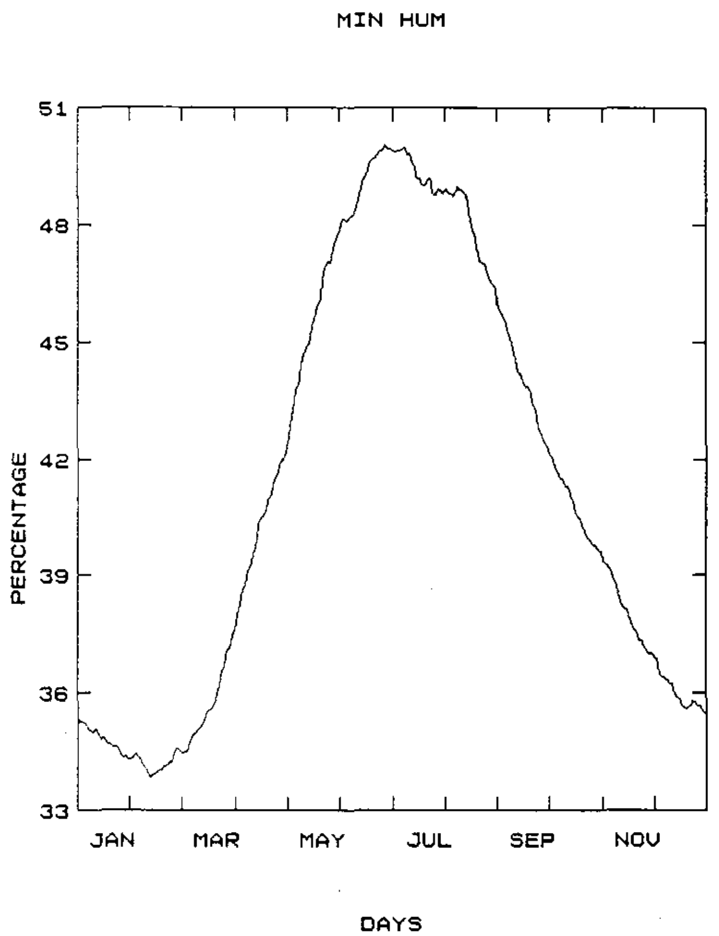
Figure 2.1 shows the smooth plots for the various variables.

FIGURE 2.1 Simple moving average smooth (lag=50)for all variables of Elsenburg




WINDRUN
KM/DAY
220
210
200
190
180
170
JAN
MAR
MAY
JUL
SEP
NOV
DAYS


MAX HUM
PERCENTAGE
94
93
92
91
JAN
MAR
MAY
JUL
SEP
NOV
DAYS


MIN HUM
PERCENTAGE
51
48
45
42
39
36
33
JAN
MAR
MAY
JUL
SEP
NOV
DAYS

From the smooth plots it can be concluded that each time series of the variables is seasonal, has a cyclic period of one year and has a sinusoidal shape.

(b) Autocorrelation

Table 2.1 shows the autocorrelation for each variable up to lags of three.

The following abbreviations for each variable will be adopted in the annotation of tables and figures:

max temp — maximum temperature

min temp — minimum temperature

evapo — evaporation

sunshine — sunshine duration

max hum — maximum humidity

min hum — minimum humidity

From Table 2.1 it can be seen that the variables are autocorrelated, i.e. there is a short–term persistence within each variable.

TABLE 2.1 Autocorrelation coefficients

Variable	Lag 1	Lag2	Lag3
rainfall	0.23	0.08	0.05
max temp	0.77	0.59	0.51
min temp	0.69	0.54	0.48
evapo	0.76	0.70	0.66
sunshine	0.52	0.30	0.22
windrun	0.38	0.10	0.02
max hum	0.32	0.15	0.08
min hum	0.53	0.33	0.27

(c) Cross–correlation

Intuitively, one would expect climate variables to be related to each other in some way, for example one would expect the amount of evaporation to be related to the temperature. In fact, as already mentioned, evaporation can be expressed approximately in terms of the other variables. The interdependence among the variables was determined by computing the lag cross–correlation coefficients of the time series. These cross–correlation coefficients are shown in Table 2.2. These results confirm that the variables are indeed interdependent.

TABLE 2.2 Cross–correlation coefficients between variables

Variables	Lag cross–correlation				
	$R_2(j,i)$	$R_1(j,i)$	$R_0(i,j)$	$R_1(i,j)$	$R_2(i,j)$
max temp – min temp	0.42	0.44	0.52	0.72	0.73
max temp – evapo	0.65	0.74	0.78	0.64	0.56
max temp – sunshine	0.52	0.65	0.63	0.34	0.24
max temp – wind	-0.07	-0.17	-0.14	0.07	0.11
max temp – max hum	-0.12	-0.21	-0.27	-0.19	-0.08
max temp – min hum	-0.42	-0.57	-0.71	-0.39	-0.26
min temp – evapo	0.65	0.61	0.51	0.50	0.48
min temp – sunshine	0.43	0.25	0.06	0.17	0.23
min temp – windrun	-0.01	0.22	0.26	0.14	0.07
min temp – max hum	-0.17	-0.18	-0.18	-0.05	-0.05
min temp – min hum	-0.44	-0.32	-0.07	-0.13	-0.20
evapo – sunshine	0.50	0.59	0.75	0.50	0.39
evapo – windrun	0.09	0.04	0.12	0.12	0.13
evapo – max hum	-0.12	-0.18	-0.29	-0.30	-0.17
evapo – min hum	-0.37	-0.46	-0.61	-0.47	-0.37
sunshine – windrun	0.03	-0.18	-0.20	-0.02	0.08
sunshine – max hum	0.00	-0.06	-0.19	-0.24	-0.15
sunshine – min hum	-0.16	-0.33	-0.70	-0.50	-0.32
windrun – max hum	-0.05	-0.08	-0.10	-0.12	-0.06
windrun – min hum	-0.12	-0.06	0.18	0.17	-0.01
max hum – min hum	0.16	0.32	0.32	0.17	0.07

(d) Time series observations differ depending on the wet or dry status of the day

It is known that on days that rain occurs, a marked change also occurs in other climatic variables, for example, temperature and sunshine duration are more likely to be below normal on rainy days than on dry days. Humidity, on the other hand will be above average on a rainy day rather than on a dry day. This property of the climate variables was investigated to determine whether the difference was significantly distinct.

The observations of all variables were found to be significantly different depending on whether rain had or had not occurred in that time period. Figure 2.2 shows the mean time series of each variable conditioned on the wet or dry status of the day. Table 2.3 shows a comparison of the mean for each variable conditioned on the wet or dry status of the day.

TABLE 2.3 Mean for conditioned time series

Variable	Dry state	Wet state
max temp	24.14	18.13
min temp	10.43	10.91
evapo	6.71	2.76
sunshine	9.73	4.15
windrun	177.7	245.8
max hum	91.9	94.3
min hum	36.8	55.1

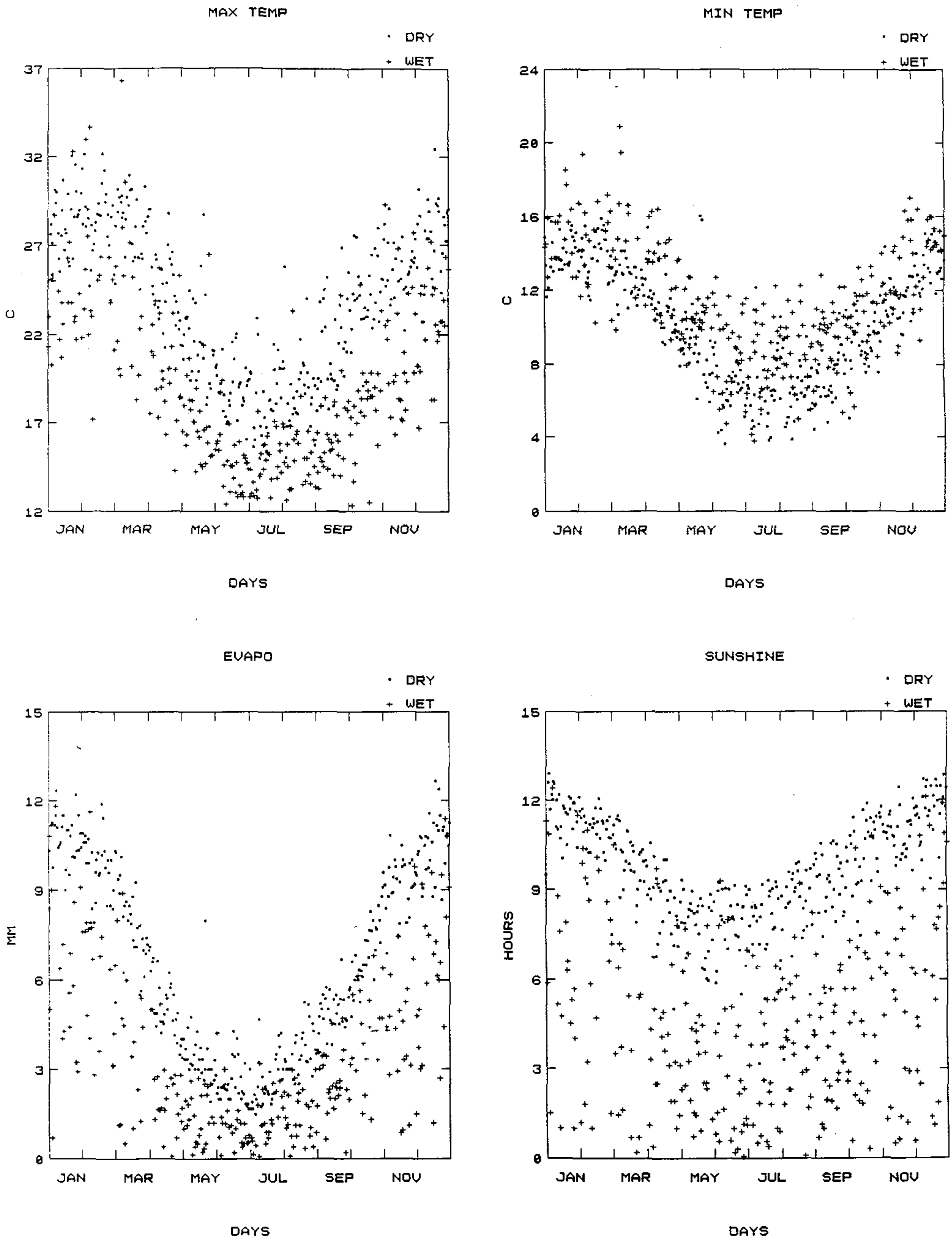
Having concluded that climatic variables vary depending on whether rain or no rain has occurred, it remains to examine whether the amount of rainfall is related in any way to the observations of the climate variables. Figure 2.3 shows the graphs of rainfall versus each climate variable. From these plots it is concluded that there is no visible pattern to the values of the climate variables in relation to the amount of rainfall.

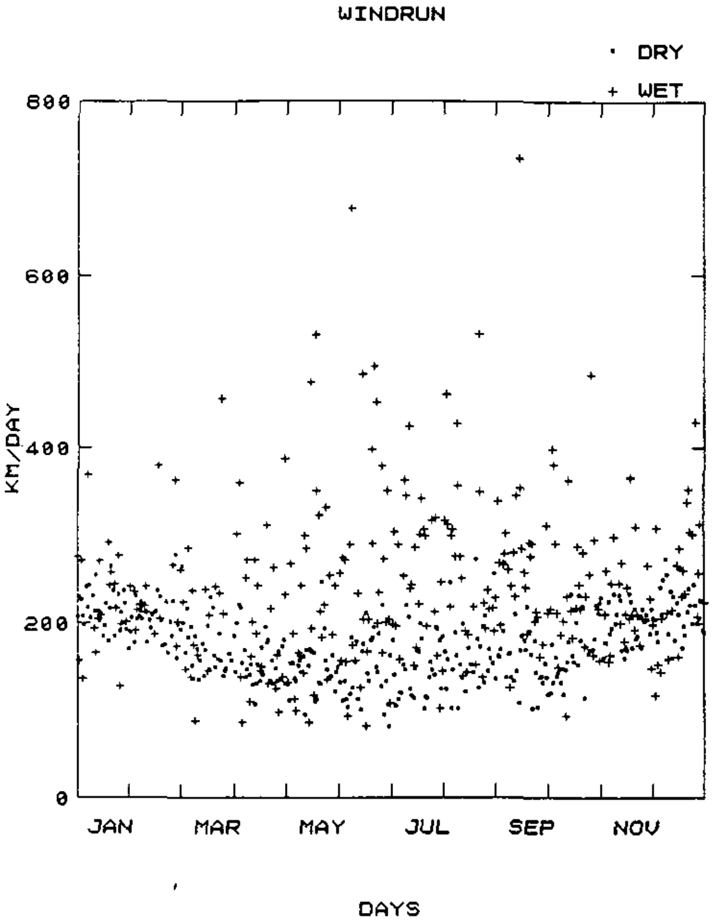
(e) Rainfall is a "strange" variable

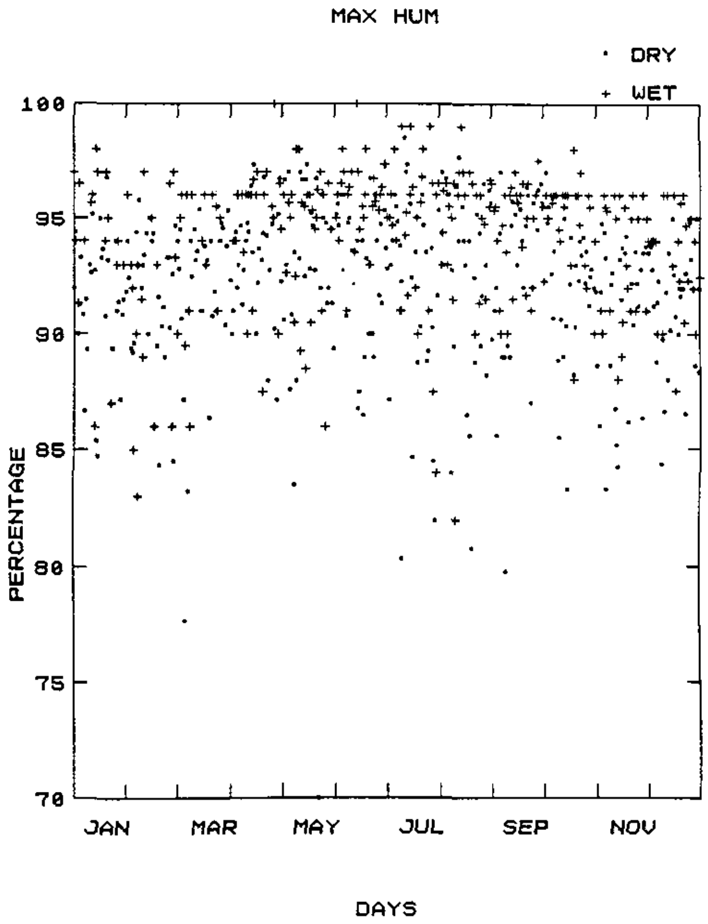
The rainfall variable is somewhat unusual from a statistical point of view in the sense that it exhibits different properties from those of the other climatic variables. The distribution of rainfall is both discrete and continuous. The occurrence or non-occurrence of rainfall is considered as discrete, while on the times that it does rain, the depth of rainfall has a continuous distribution.

Another distinctive feature of rainfall is that especially in a country like South Africa, the proportion of rainy days is relatively small.

FIGURE 2.2 Mean time series conditioned on status of day




WINDRUN
• DRY
+ WET
KM/DAY
800
600
400
200
0
JAN
MAR
MAY
JUL
SEP
NOV
DAYS


MAX HUM
• DRY
+ WET
PERCENTAGE
100
95
90
85
80
75
70
JAN
MAR
MAY
JUL
SEP
NOV
DAYS

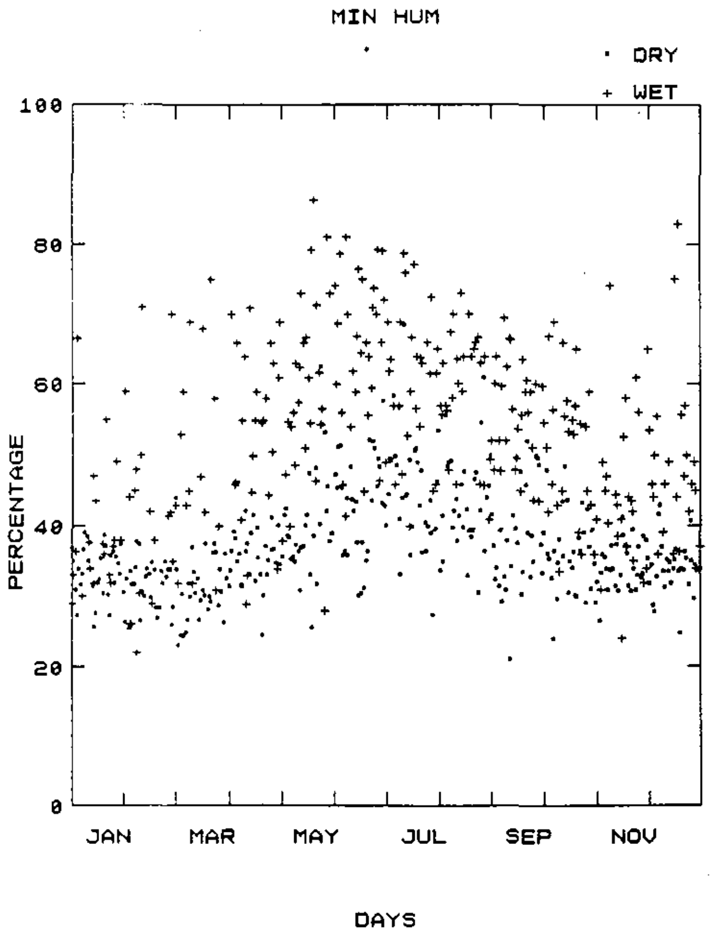
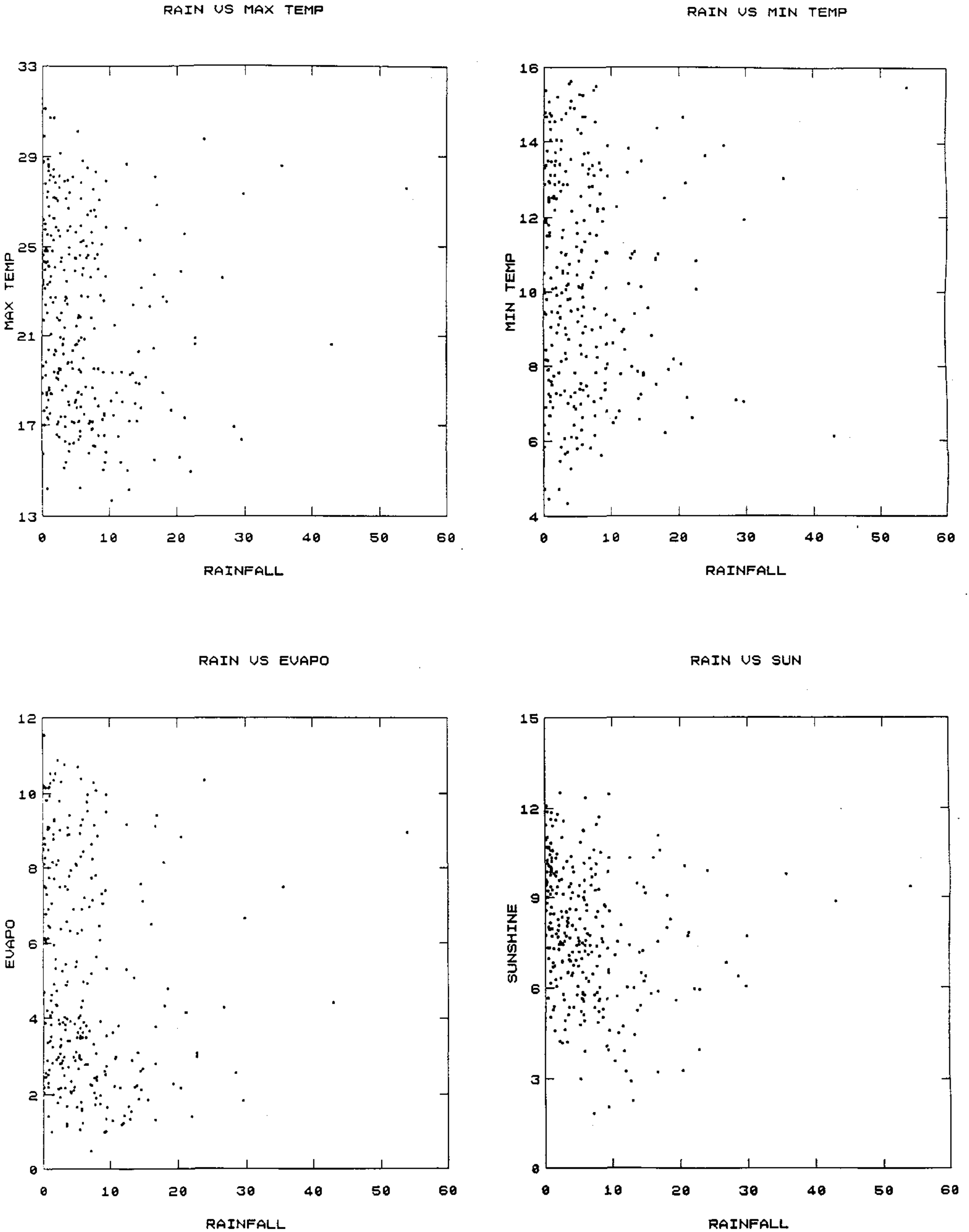
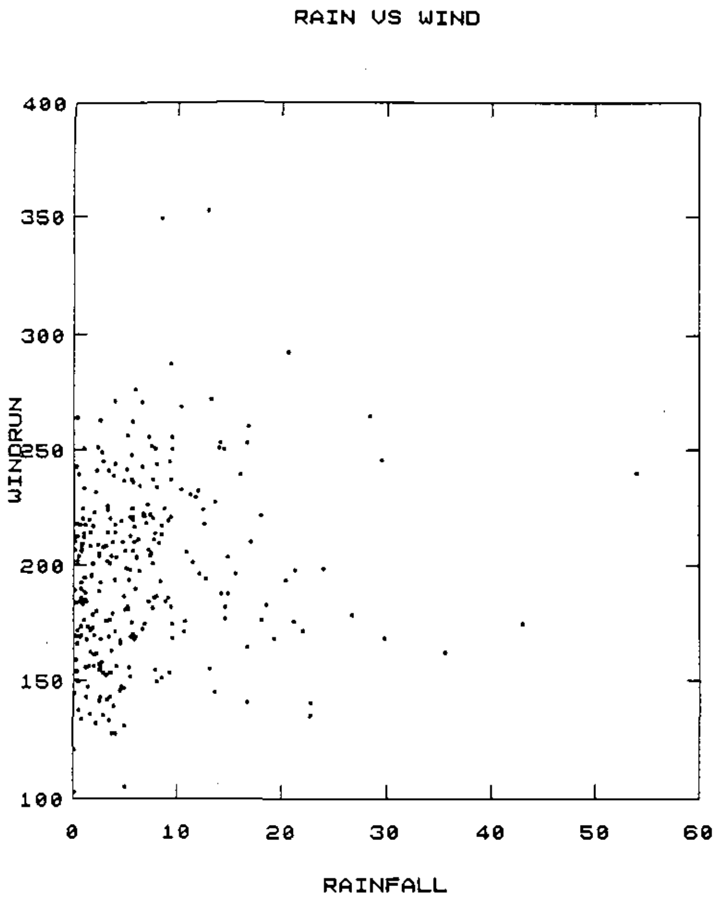
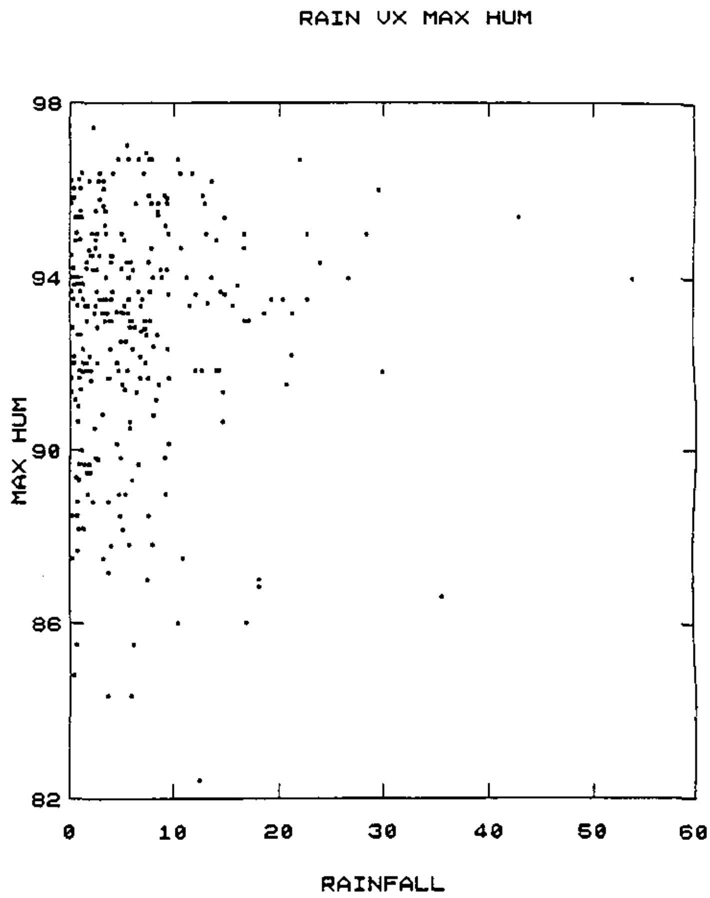
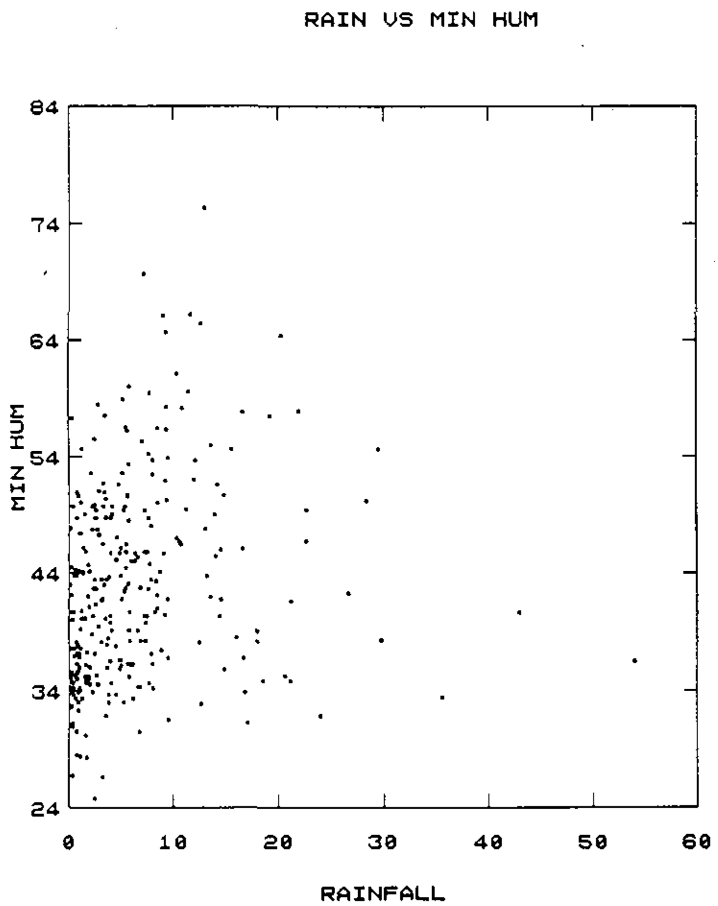

MIN HUM
• DRY
+ WET
PERCENTAGE
100
80
60
40
20
0
JAN
MAR
MAY
JUL
SEP
NOV
DAYS

FIGURE 2.3 Daily mean rainfall versus daily mean climate for all variables




RAIN VS WIND
WINDRUN
400
350
300
250
200
150
100
0
10
20
30
40
50
60
RAINFALL


RAIN VX MAX HUM
MAX HUM
98
94
90
86
82
0
10
20
30
40
50
60
RAINFALL


RAIN VS MIN HUM
MIN HUM
84
74
64
54
44
34
24
0
10
20
30
40
50
60
RAINFALL

Concluding remarks

The above preliminary analysis establishes a number of facts. Firstly, the individual climate variables exhibit seasonal fluctuations and these fluctuations appear to follow a sinusoidal pattern. This would suggest that the mean function of each variable could be parsimoniously modelled by means of a truncated Fourier series. Secondly, the individual variables are serially correlated (even after this seasonal fluctuation has been taken into account). In other words, the individual climate variables constitute time series and have to be modelled as such. This preliminary analysis would suggest that an autoregressive model might be suitable to describe the autocorrelation structure of the variables. Here one has to keep in mind that the number of parameters in the final model must be kept to a minimum in order to avoid the usual statistical problems associated with estimating a large number of parameters. An autoregressive model is ideal in this respect.

Finally, the variables are cross–correlated, that is, they do not vary independently of each other. It follows that it is not possible to model climate by separately modelling its component variables; a multivariate time series model is required.

Seeing that the variable rainfall has some extra properties that have to be taken into consideration and that the remaining climatic variables differ depending on the state of the rainfall variable, it is proposed that the rainfall variable should be determined independently of the other variables and then to condition the other variables for a given day on whether the day was wet or dry.

As no pattern was found between different precipitation amounts and the climate observations, it was decided to consider a non–rainy day as one with a precipitation amount of zero and a rainy day as one with a rainfall depth greater than zero.

CHAPTER 3

THE MODELS

The preliminary analysis described in Chapter 2 established that sequences of climate variables exhibit a number of distinctive features. In particular the distribution of each climate variable varies seasonally, the variables are serially correlated, they are dependent, and finally the distributions of the variables depend on the wet or dry status of the day under consideration. Any useful model for the simultaneous discription of climate sequences must of course preserve all these properties.

The models considered here are constructed in two stages. One begins by constructing a model for the rainfall process. This provides synthetic sequences of wet and dry days. The remaining variables are then modelled according to the wet or dry status of each day. Thus the joint distribution of all the variables other than rainfall changes not only with season but also with changes in wet or dry status.

The rainfall component of the five models to be discussed is common to all five models and is thus described first. The first of the five models is due to Richardson (1981), the remaining four are new.

The rainfall model

Several models have been proposed for simulating daily precipitation. (Gabriel and Neumann, 1962; Richardson, 1981; Roldan and Woolhiser, 1982; Stern and Coe, 1984; Zucchini and Adamson, 1984.) Most precipitation models are specified by a discrete occurrence process describing the sequence of wet and dry days, and a continuous distribution function for the amount of precipitation of days with rain. The parameters of the model are allowed to vary seasonally.

A model to describe the occurrence of wet and dry sequences of days

A first–order Markov chain is used to describe the occurrence of wet and dry days. By this one assumes that the state of day t depends on the state of the previous day, $t-1$. This does not imply that the state at time t is independent of the state on day $t-2, t-3$, etc ..., but rather that the information given by $t-1$ is equivalent to all the information given by $t-1, t-2$, etc .... One also assumes that, except for the seasonality, the process is stationary.

A first–order Markov chain has been found to be an adequate model for precipitation occurrence in many different regions. (Gabriel and Neumann, 1962; Caskey, 1963; Weiss, 1964; Hopkins and Robillard, 1964; Haan *et al*, 1976; Smith and Schreiber, 1973; Woolhiser and Prengram, 1979; Richardson, 1981; Roldan and Woolhiser, 1982; Zucchini and Adamson, 1984.) The order of the Markov chain may of course be increased, but this has to be done at the cost of increasing complexity and the number of parameters in the model. A further problem arises if one attempts to increase the order of the Markov chain in arid areas, namely the estimation of the probability that a rain day follows two or more consecutive rain days. In arid areas there are relatively few runs of three or more consecutive rain days and thus there is hardly any data on which to base estimates of this conditional probability. (Note that this has to be estimated for each day of the year.) Finally, it was demonstrated in Zucchini and Adamson (1984) that a first order Markov chain provides an adequate description of the occurrence of wet and dry sequences of days in the complete range of South African conditions.

(a) Notation and preliminaries

The day will be used as the time unit. That is, the year is divided into $NT(=365)$ equal intervals, denoted by $t = 1, 2, \ldots, NT$. A day with total rainfall greater that 0 mm is considered as a wet day.

The following notation will be used:

R represents the occurrence of rain (i.e. wet day).

$\overline{R}$ represents the non–occurrence of rain (i.e. dry day).

For $t = 1, 2, \ldots, NT$

$NR(t)$ is the number of times it was wet in period t .

$N\overline{R}(t)$ is the number of times it was dry in period t .

$N\overline{R}R(t)$ is the number of times it was dry in period $t-1$ and wet in period t .

$N\overline{R}\overline{R}(t)$ is the number of times it was dry in period $t-1$ and dry in period t .

$NRR(t)$ is the number of times it was wet in period $t-1$ and wet in period t .

$ND(t) = N\overline{R}R(t) + N\overline{R}\overline{R}(t)$ is the number of times that it was dry in period $t-1$ and there was an observation (wet or dry) in period t .

$NW(t) = NRR(t) + NR\overline{R}(t)$ is the number of times that it was wet in period $t-1$ and there was an observation (wet or dry) in period t .

$\pi_{R/R}(t)$ the probability that period t is wet given that period $t-1$ is wet.

$\pi_{\overline{R}/R}(t)$ the probability that period t is dry given that period $t-1$ is wet.

$\pi_{R/\overline{R}}(t)$ the probability that period t is wet given that period $t-1$ is dry.

$\pi_{\overline{R}/\overline{R}}(t)$ the probability that period t is dry given that period $t-1$ is dry.

Then $\pi_{R/R}(t)+\pi_{\overline{R}/R}(t)=1$

$\pi_{\overline{R}/\overline{R}}(t)+\pi_{R/\overline{R}}(t)=1$.

Therefore the transition probabilities are fully defined given $\pi_{R/R}(t), \pi_{R/\overline{R}}(t)$ and the wet or dry state on day $t-1$, and one only needs to estimate these two probabilities.

From elementary probability theory we have

$$NRR(t) \sim B(NW(t),\ \pi_{R/R}(t))$$
$$N\overline{R}R(t) \sim B(ND(t),\ \pi_{R/\overline{R}}(t)), \qquad t=1,2,\ldots,NT$$

where $B(N,\pi)$ denotes the binomial distribution with parameters N and π .

(b) Estimation

The functions $\pi_{R/R}(t)$ and $\pi_{R/\overline{R}}(t)$ are estimated using the same method but different data. To simplify the notation in what follows, one makes use of the following generic names:

$$\text{Let} \qquad M(t) \sim B(MM(t),\ \pi(t)), \qquad t=1,2,\ldots,NT.$$

First we note that the binomial distribution belongs to the exponential family. Therefore we have a set of independent random variables $M(t), t=1,2,\ldots,NT$, each with a distribution from the exponential family; each $M(t)$ depends on a single parameter $\pi(t)$ and the distributions of all $M(t)$, $t=1,2,\ldots,NT$, are of the same form (i.e. all binomial). Thus the properties of a generalized linear model are satisfied, and estimates of $\pi(t)$ may be obtained by using the theory for estimation for generalized linear models. (Dobson, 1983.)

The probabilities $\pi(t)$ are assumed to be functions of linear combinations of parameters $\gamma_1,\gamma_2,\ldots,\gamma_L$, $L<NT$. That is

$$g(\pi(t))=\lambda(t,L)$$

where g is the link function and $\lambda(t, L)$ is a linear combination of the γ_i .

To ensure that the estimated values of $\pi(t)$ are restricted to the interval $[0,1]$, one uses the logit link function, given by

$$g(\pi(t)) = \log\left(\frac{\pi(t)}{1-\pi(t)}\right) = \lambda(t, L).$$

To obtain the linear combination of the γ_i, $\lambda(t, L)$, we look at some of the properties of $\pi(t)$, namely that it is a smooth, periodic and approximately sinusoidal shaped function. Transforming $\pi(t)$, using the logistic transformation, to a logit $\lambda(t)$ given by

$$\lambda(t) = \log\left(\frac{\pi(t)}{1-\pi(t)}\right) ,$$

one obtains a representation which still has the same properties as $\pi(t)$, and thus we can approximate $\lambda(t)$ by the first few terms of its Fourier representation. This approximation has been used by Stern and Coe (1984) and Zucchini and Adamson (1984).

The exact Fourier representation of $\lambda(t)$ is given by

$$\lambda(t) = \sum_{i=1}^{NT} \gamma_i \varphi_i(t), \qquad t = 1, 2, \ldots, NT$$

where

$$\varphi_i(t) = \begin{cases} \cos(\omega(t-1)i/2) & i = 2, 4, \ldots \\ \sin(\omega(t-1)(i-1)/2) & i = 3, 5, \ldots \end{cases}$$

$$\varphi_1(t) = 1 \; ; \qquad t = 1, 2, \ldots, NT,$$

and

$$\omega = \frac{2\pi}{NT}.$$

Define the function $\lambda(t, L)$ by

$$\lambda(t, L) = \sum_{i=1}^{L} \gamma_i \varphi_i(t), \qquad t = 1, 2, \ldots, NT; \quad L \leq NT$$

where $\varphi_i(t)$ is defined as before and L is the order of the Fourier series approximation. One is thus making the following approximation:

For some $L < NT$

$$\lambda(t, L) \approx \lambda(t) \; , \qquad t = 1, 2, \ldots, NT.$$

A procedure to choose the order of the Fourier series approximation (i.e. the value of L) will be discussed later. Generally this approximation is accurate for small values of L . The number of parameters, L , is always chosen to be an odd number. The reason for this choice is given in Appendix A. An outline of the properties of the Fourier representation are given in Appendix B.

The log–likelihood function of the observed values as a function of the probabilities $\pi(t)$, is given by

$$\ell(\pi(t);\ M(t)) = \sum_{t=1}^{NT} \left[M(t)\log\left(\frac{\pi(t)}{1-\pi(t)}\right) + MM(t)\log(1-\pi(t)) + \log\binom{MM(t)}{M(t)} \right].$$

Therefore, the log–likelihood function of the observed values as a function of the parameters $\gamma_1, \gamma_2, \ldots, \gamma_L$ is given by

$$\ell(\gamma; M(t)) = \sum_{t=1}^{NT} \left[M(t)\lambda(t,L) - MM(t)\log(1+e^{\lambda(t,L)}) + \log\binom{MM(t)}{M(t)} \right].$$

The score vector U with respect to $\gamma_1, \gamma_2, \ldots, \gamma_L$ has elements given by

$$\begin{aligned} U_j = \frac{\partial \ell(\gamma; M(t))}{\partial \gamma_j} &= \sum_{t=1}^{NT} \left[M(t) - MM(t)\,\frac{e^{\lambda(t,L)}}{1+e^{\lambda(t,L)}} \right] \varphi_j(t) \\ &= \sum_{t=1}^{NT} [M(t) - MM(t)\pi(t)]\varphi_j(t) \end{aligned}$$

since $\mathrm{Var}(M(t)) = MM(t)\,\dfrac{e^{\lambda(t,L)}}{(1+e^{\lambda(t,L)})^2}$ and

$$E(M(t)) = \frac{MM(t)e^{\lambda(t,L)}}{(1+e^{\lambda(t,L)})} \qquad \text{and so}$$

$$\frac{\partial E(M(t))}{\partial \lambda(t,L)} = \frac{MM(t)e^{\lambda(t,L)}}{(1+e^{\lambda(t,L)})^2} = \mathrm{Var}(M(t)).$$

Similarly, the information matrix $\mathbf{I}_{L\times L}$ has elements given by

$$\mathbf{I}_{jk} = \sum_{t=1}^{NT} \varphi_j(t)\varphi_k(t)MM(t)\,\frac{e^{\lambda(t,L)}}{(1+e^{\lambda(t,L)})^2}.$$

Since $\dfrac{e^{\lambda(t,L)}}{(1+e^{\lambda(t,L)})^2} = \pi(t)(1-\pi(t))$ it follows that

$$\mathbf{I}_{jk} = \sum_{t=1}^{NT} \varphi_j(t)\varphi_k(t)MM(t)\pi(t)(1-\pi(t)).$$

The maximum likelihood estimates for $\gamma_1, \gamma_2, \ldots, \gamma_L$ are then obtained by solving the iterative equation

$$\mathbf{I}^{(m-1)}\hat{\gamma}^{(m)} = \mathbf{I}^{(m-1)}\hat{\gamma}^{(m-1)} + U^{(m-1)}$$

where m indicates the mth approximation and $\hat{\gamma}$ is the vector of estimates.

Some initial approximation $\gamma^{(0)}$ is used to evaluate $\mathbf{I}^{(0)}$ and $U^{(0)}$, then the iterative equation is solved to give $\gamma^{(1)}$ which in turn is used to obtain better approximations for $\mathbf{I}$ and U , and so on until adequate convergence is achieved. When the difference between successive approximations $\gamma^{(m)}$ and $\gamma^{(m-1)}$ is sufficiently small, $\gamma^{(m)}$ is taken as the maximum likelihood estimate vector.

(c) Model selection

Whenever a model is fitted to observed data, two types of discrepancy arise. The discrepancy due to approximation (the fewer the number of parameters fitted, the higher the value of this discrepancy) and the discrepancy due to estimation (the more parameters fitted, the higher the value of this discrepancy). When choosing the number of parameters to be fitted, one attempts to minimize the combined effect arising from the two discrepancies.

Selection of the number of parameters, L , may be done by using the criterion of the Kullbach–Leibler measure of discrepancy. (Linhart and Zucchini, 1982; Zucchini and Adamson, 1984.)

Under the assumption that for some L_0, $\lambda(t)$ is exactly fitted by L_0 parameters, i.e.

$$\lambda(t) = \lambda(t, L_0), \quad L_0 < NT,$$

the above method leads to the Akaike Information Criterion where

$$AIC = -\ell(\gamma; M(t)) + L$$

where $\ell(\gamma; M(t))$ is the log–likelihood function given before.

Each value of L leads to a different approximating model. The criterion is computed for $L = 1, 3, 5, \ldots$ and the model which leads to the smallest value of the criterion is selected.

The AIC criterion is much easier to compute than the full Kullbach–Leibler discrepancy and leads to almost identical results if the discrepancy due to aproximation is small (which it is in this application).

The distribution of rainfall on days when rain occurs

Several models have been proposed for the distribution of precipitation amounts given the occurrence of a wet day. These include the exponential (Todorovic and Woolhiser, 1975; Richardson, 1981); gamma (Ison *et al.*, 1981; Buishand, 1977; Stern and Coe, 1984); two–parameter gamma (Buishand, 1978); three–parameter mixed exponential (Woolhiser and Pegram, 1979); kappa (Mielke, 1973); lognormal and Weibull (Zucchini and Adamson, 1984).

Woolhiser and Roldan (1982b) found that out of the exponential, gamma and mixed exponential distributions, the latter fitted the model of precipitation amounts best. Zucchini and Adamson (1984) found that for stations in South Africa, the lognormal distribution did not fit some stations, while the Weibull seemed to provide better fits.

It is known that the distribution of precipitation depths when rain occurs is positively skewed (i.e. smaller amounts occurring more frequently than the larger amounts) and that it exhibits the same seasonal variability as found with the probabilities $\pi(t)$. To account for this seasonality, the simplest solution is to fit a family of distributions and then to allow the parameters to change over the year, where these parameters are expressed in terms of its Fourier series approximation.

The method of modelling precipitation amounts is adopted from Zucchini and Adamson (1984). Here one does not fit any model initially, the first two moment functions of the distribution are fitted instead. These are then used to estimate the parameters (by the method of moments) to any desired two–parameter model. Different families can be fitted to a single record, e.g. one for the rainy season and a second for the dry season.

(a) Notation

The year is divided into NT equal intervals denoted by $t = 1, 2 \ldots, NT$.

$M(t)$ represents the number of times that it rained in period t .

$R(i,t)$ represents the rainfall depth on the ith year that it rained in period t , where $i = 1, 2, \ldots, M(t)$.

C represents the coefficient of variation which we assume to be constant for all t (Zucchini and Adamson, 1984).

$\mu(t)$ represents the mean rainfall per rainy day in period $t = 1, 2, \ldots, NT$.

(b) Estimating the mean and coefficient of variation

As observed before $\mu(t)$ can be approximated by its truncated Fourier series representation and thus reducing the number of parameters to be estimated. That is, we make the approximation:

$$\mu(t, L) \approx \mu(t), \qquad t = 1, 2, \ldots, NT; \quad L < NT$$

where $\mu(t)$ is defined as

$$\mu(t) = \sum_{i=1}^{NT} \mu_i \varphi_i(t) \qquad t = 1, 2, \ldots, NT$$

and

$$\mu(t, L) = \sum_{i=1}^{L} \mu_i \varphi_i(t) \qquad t = 1, 2, \ldots, NT; \quad L \leq NT$$

and $\varphi_i(t)$ is defined as before.

Define $m(t)$ to be the observed means for each period, i.e.

$$m(t) = \frac{1}{M(t)} \sum_{i=1}^{M(t)} R(i, t), \qquad t = 1, 2, \ldots, NT; \quad i = 1, 2, \ldots, M(t); \quad M(t) > 0$$

where $m(t)$ is not defined when $M(t) = 0$, i.e. it never rained in period t .

We use the method of least squares on $m(t)$ to estimate $\mu_1, \mu_2, \ldots \mu_L$, that is, minimize

$$\sum_{t=1}^{NT} (m(t) - \mu(t, L))^2 \tag{1}$$

with respect to the μ_i, $i = 1, 2, \ldots, L$. Approximations to the least squares estimators when some of the $M(t) = 0$, something which occurs often in arid regions, are given by

$$\widehat{\mu}_i = K(i) \sum_{\substack{t=1 \\ M(t)>0}}^{NT} m(t)\varphi_i(t) \tag{2}$$

where

$$K(i) = \sum_{\substack{t=1 \\ M(t)>0}}^{NT} \varphi_i(t)^2 \qquad i = 1, 2, \ldots, L.$$

The $m(t)$ in (1) are given the same weight and so periods which had very little rainfall have a large influence in the estimates of $\mu(t)$. To overcome this difficulty, the following criterion is used instead:

Minimize

$$S(\mu) = \sum_{t=1}^{NT} \sum_{i=1}^{M(t)} (R(i,t) - \mu(t,L))^2 \tag{3}$$

with respect to μ_i, $i = 1, 2, \ldots, L$.

By adding and subtracting $m(t)$ inside the squared term of (3), $S(\mu)$ can be rewritten as

$$S(\mu) = S + \sum_{t=1}^{NT} M(t)(m(t) - \mu(t,L))^2 \tag{4}$$

where

$$S = \sum_{t=1}^{NT} \sum_{i=1}^{M(t)} (R(i,t) - m(t))^2$$

and $m(t)$ is defined as before if $M(t) \neq 0$ and $m(t) = 0$ if $M(t) = 0$.

To minimize (4) set its partial derivatives equal to zero:

$$\frac{\partial S(\mu)}{\partial \mu_i} = -2 \sum_{t=1}^{NT} M(t)(m(t) - \mu(t,L))\varphi_i(t), \qquad i = 1, 2, \ldots, L.$$

These L equations can be solved using the Newton–Raphson iteration method. For this, we need the second partial derivatives:

$$\frac{\partial^2 S(\mu)}{\partial \mu_i \partial \mu_j} = 2 \sum_{t=1}^{NT} M(t)\varphi_i(t)\varphi_j(t), \qquad i,j = 1, 2, \ldots, L.$$

Denote the ith element of the vector $f^{(k)}$ by

$$f_i^{(k)} = \sum_{t=1}^{NT} M(t)(m(t) - \mu^{(k)}(t,L))\varphi_i(t), \qquad i = 1, 2, \ldots, L \tag{5}$$

and the (i,j)th element of the matrix $F^{(K)}$ by

$$F_{ij}^{(k)} = \sum_{t=1}^{NT} M(t)\varphi_i(t)\varphi_j(t), \qquad i,j = 1, 2, \ldots, L \tag{6}$$

where k denotes the kth iteration.

Then an algorithm to estimate μ_i, $i = 1, 2, \ldots, L$ is given by:

Step 1: Obtain initial estimates $\mu_1^{(0)}, \ldots, \mu_L^{(0)}$ using (2) and compute $\mu^{(0)}(t,L)$.

Step 2: Compute $f^{(k)}$ using (5) and $F^{(k)}$ using (6).

Step 3: Compute the vector $\delta^{(k)}$ which is the solution to the system of L linear equations given by

$$F^{(k)}\delta^{(k)} = f^{(k)}$$

Step 4: Set $\mu^{(k+1)} = \mu^{(k)} - \delta^{(k)}$.

Step 5: Test for convergence, e.g. if the elements of $f^{(k)}$ are sufficiently close to zero. If the convergence criterion is met, stop, otherwise increase k by 1 and go to Step 2.

Note that $F^{(k)}$ is symmetric. This fact can be used to reduce the number of computations performed.

An estimator of C is given by:

$$\widehat{C} = \left[\frac{\left[\sum_{t=1}^{NT} \sum_{i=1}^{M(t)} (R(i,t) - \widehat{\mu}(t))^2\right]}{\left[\sum_{t=1}^{NT} M(t)\widehat{\mu}(t)^2\right]} \right]^{\frac{1}{2}} .$$

(c) Selecting the number of parameters

$$\Delta(L) = \sum_{t=1}^{NT} (\mu(t) - E(\widehat{\mu}(t,L)))^2, \qquad L = 1,3,5,\ldots$$

would be a suitable discrepancy on which to base the selection, except that some $M(t)$ are zero and so only approximately unbiased estimators are available. The reliability of this criterion is therefore difficult to determine.

If one is prepared to make distributional assumptions, then selection criteria are relatively easy to derive, for example based on the Kullbach–Leibler discrepancy.

A reasonable procedure is to select L for a parametric family of models and then use the same L in the estimation of $\mu(t)$.

(d) Fitting the Weibull family

Zucchini and Adamson (1984) found the Weibull family to fit the rainfall depth models for stations in South Africa and so this family was used to model the observed rainfall amounts on days that rain was recorded.

Having estimated the mean value function $\mu(t)$ and the coefficient of variation, C , one can apply the method of moments to estimate the parameter functions of the Weibull distribution.

Denote the scale parameter by $\alpha(t), \quad t = 1, 2, \ldots, NT$ and the shape parameter by β.

Now

$$C = \left\{ \frac{\Gamma(1 + 2/\beta)}{\Gamma(1 + 1/\beta)^2} - 1 \right\}^{\frac{1}{2}}$$

To obtain β as a function of C a rational function approximation has to be derived as no closed expression of this function is available.

The following approximation has been obtained from Zucchini and Adamson (1984):

$$\widehat{\beta} = \frac{339.5410 + 148.445\widehat{C} + 192.7492\widehat{C}^2 + 22.4401\widehat{C}^3}{1 + 257.1162\widehat{C} + 287.8362\widehat{C}^2 + 157.2230\widehat{C}^3}.$$

Using the relationship

$$\mu(t) = \alpha(t)\Gamma(1 + 1/\beta) \qquad t = 1, 2, \ldots, NT$$

we obtain the estimator

$$\widehat{\alpha}(t) = \frac{\widehat{\mu}(t)}{\Gamma(1 + 1/\widehat{\beta})} \qquad t = 1, 2, \ldots, NT.$$

MODEL FOR CLIMATE SEQUENCES

Little attention has been given to stochastic modelling of climatic variables such as maximum and minimum temperature, evaporation, sunshine duration, windrun, and maximum and minimum humidity. Recently, though, there have been some models proposed to stochastically simulate possible sequences of maximum and minimum temperature and solar radiation. (Goh and Tan, 1977; Nicks and Harp, 1980; Richardson, 1981; Larsen and Pense, 1982.) Bruhn *et al* (1980) looked at minimum relative humidity as well.

Variables such as temperature, evaporation, sunshine duration, windrun and humidity are not as difficult to model statistically as precipitation because there is not a high proportion of zero observations and the distributions of these variables are not as skewed as the rainfall distribution.

In the models that follow, because the cross-correlation between the variables is non zero, the variables are considered to reflect a continuous multivariate stochastic process with the parameters conditioned on the wet or dry status of the day.

Model 1: Multivariate model for climate data proposed by Richardson (1981)

The approach taken here to model the climate variables is the method suggested by Richardson (1981). The weather variables evaporation, windrun, maximum and minimum humidity have been added to the multivariate process.

(a) Notation

Partition the year into $NT(=365)$ equal intervals, denoted by $t=1,2,\ldots,NT$.

NV is the number of variables.

NY is the number of years observed.

W represents the occurrence of rain.

D represents the non–occurrence of rain.

$Y_{i,t}$ is the precipitation amount on period t of year i , $i=1,2,\ldots,NY$.

$S_{i,t}$ is the generic name for the observation at period t of the ith year.

μ_t^D is the generic name for the mean for a dry day on period t (i.e. $Y_{i,t}=0$) .

μ_t^W is the generic name for the mean for a wet day on period t (i.e. $Y_{i,t}>0$) .

σ_t^D is the generic name for the standard deviation for a dry day on period t .

σ_t^W is the generic name for the standard deviation for a wet day on period t .

$\chi_{i,t}$ is the generic name for the residual component at period t and year i .

$\rho_0(j,k)$ is the lag 0 cross–correlation coefficient between variables j and k .

$\rho_1(j,k)$ is the lag 1 cross-correlation coefficient between variables j and k .

$\rho_1(j)$ is the lag 1 serial correlation for variable j .

(b) The model and assumptions

Each variable is modelled in the same way. The procedure given below to model $S_{i,t}$ is carried out once for each variable to be included in the multivariate model.

The distribution of $S_{i,t}$ is seasonal and so its parameters, e.g. the mean and standard deviation, are allowed to vary seasonally. As was the case with the parameter functions of the rainfall model, it can be reasonably assumed that the parameter functions of the climate variables are smooth, periodic and sinusoidal in shape. This would again lead one to expect that they can be accurately approximated by the first few terms of their Fourier representation.

The truncated Fourier representations for the daily means and standard deviations for wet days and for dry days are given by:

$$\left.\begin{aligned} \mu_t^W &= \sum_{i=1}^{L} \alpha_i^W \varphi_i(t) \\ \sigma_t^W &= \sum_{i=1}^{L} \xi_i^W \varphi_i(t) \end{aligned}\right\} \qquad \text{if} \quad Y_{i,t} > 0$$

$$\left.\begin{aligned} \mu_t^D &= \sum_{i=1}^{L} \alpha_i^D \varphi_i(t) \\ \sigma_t^D &= \sum_{i=1}^{L} \xi_i^D \varphi_i(t) \end{aligned}\right\} \qquad \text{if} \quad Y_{i,t} = 0$$

$$t = 1, 2, \ldots, NT$$

where

$$\varphi_i(t) = \begin{cases} \cos(\omega(t-1)i/2) \ , & i = 2, 4, \ldots, L-1 \\ \sin(\omega(t-1)(i-1)/2) \ , & i = 3, 5, \ldots, L \end{cases}$$

$$\varphi_1(t) = 1 \ ,$$

$$\omega = 2\pi/NT \qquad \text{and}$$

α_i^W, α_i^D, ξ_i^W, ξ_i^D are the coefficients of the respective Fourier series and, L is the order of the Fourier series approximation, i.e. we assume that for some $L < NT$

$$\mu_t = \sum_{i=1}^{L} \alpha_i \varphi_i(t) \approx \sum_{i=1}^{NT} \alpha_i \varphi_i(t)$$

and

$$\sigma_t = \sum_{i=1}^{L} \xi_i \varphi_i(t) \approx \sum_{i=1}^{NT} \xi_i \varphi_i(t)$$

where the above two equations hold for both wet and dry days. (Whenever W or D is omitted it means that the equation applies for both.)

The number of parameters, L, does not have to be the same in all instances, i.e. the number of parameters for the means of wet days can differ from that for dry days. The same applies for the standard deviations. To avoid complicating the notation, it will be assumed in what follows that L refers to the number of parameters of the particular parameter function under consideration.

The estimation of the Fourier coefficients will be discussed later.

The approach used by Richardson (1981) is to determine the daily means and standard deviations of each variable conditioned on the wet or dry status of each day where Fourier series is used to smooth their seasonality. The time series $S_{i,t}$ is then reduced to a time series of residual elements by removing the periodic means and standard deviations. This residual time series is given by the equations:

$$\chi_{i,t} = \begin{cases} \dfrac{S_{i,t} - \mu_t^D}{\sigma_t^D} & \text{if} \quad Y_{i,t} = 0 \\ \dfrac{S_{i,t} - \mu_t^W}{\sigma_t^W} & \text{if} \quad Y_{i,t} > 0. \end{cases}$$

This standardization leads to a residual series for each variable that is stationary in the mean and standard deviation with mean zero and standard deviation of unity.

The serial correlation and cross–correlation coefficients are then calculated to describe the time dependence and the interdependence (respectively) of the residual series.

The model proposed for generating residual series for each variable is the weakly stationary process suggested by Matalas (1967) given by

$$\chi_{i,t} = A\ \chi_{i,t-1} + B\ \epsilon_{i,t} \tag{7}$$

where $\epsilon_{i,t}$ is a $(NV \times 1)$ matrix of independent random components that are normally distributed with mean zero and a variance of unity, i.e.

$$\epsilon_{i,t} \sim NID(0,1).$$

A and B are $(NV \times NV)$ matrices whose elements are defined in such a way that the sequence generated will have the desired serial correlation and cross–correlation coefficients.

This model is based on the assumption that the residuals of the variables are normally distributed and that the serial correlation of each variable may be described by a first–order linear autoregressive model.

(c) Estimation

Firstly, a method for estimating the matrices A and B will be considered.

From the properties of the distribution of $\epsilon_{i,t}$ and $\chi_{i,t}$ we have that

$$E(\epsilon_{i,t}) = 0$$

and

$$E(\chi_{i,t}) = E(\chi_{i,t-1}) = 0.$$

Postmultiplying (7) through by $\chi_{i,t-1}^T$, the transpose of $\chi_{i,t-1}$, and taking expectations we have

$$E[\chi_{i,t}\chi_{i,t-1}^T] = AE[\chi_{i,t-1}\chi_{i,t-1}^T] + BE[\epsilon_{i,t}\chi_{i,t-1}^T] \tag{8}$$

Define

$$M_0 = E[\chi_{i,t-1}\chi_{i,t-1}^T]$$

and

$$M_1 = E[\chi_{i,t}\chi_{i,t-1}^T].$$

M_0 is an $(NV \times NV)$ matrix whose elements are the lag 0 cross–correlation coefficients and M_1 is an $(NV \times NV)$ matrix whose elements are the lag 1 cross–correlation coefficients.

The matrices may be written as

$$M_0 = \begin{pmatrix} 1 & \rho_0(1,2) & \dots & \rho_0(1,NV) \\ \rho_0(2,1) & 1 & \dots & \rho_0(2,NV) \\ \vdots & & \dots & \vdots \\ \rho_0(NV,1) & \dots & \dots & 1 \end{pmatrix}$$

and

$$M_1 = \begin{pmatrix} \rho_1(1) & \rho_1(1,2) & \dots & \rho_1(1,NV) \\ \rho_1(2,1) & \rho_1(2) & \dots & \rho_1(2,NV) \\ \vdots & & \dots & \vdots \\ \rho_1(NV,1) & \rho_1(NV,2) & \dots & \rho_1(NV) \end{pmatrix}$$

where $\rho_0(j,k)$ is the lag 0 cross–correlation coefficient between variables j and k, $\rho_1(j,k)$ is the cross–correlation coefficient between variables j and k with variable k lagged 1 day in relation to variable j, and $\rho_1(j)$ is the lag 1 serial correlation for variable j. We can thus rewrite (8) as

$$M_1 = AM_0 \quad \text{since} \quad E[\epsilon_{i,t}\epsilon_{i,t-1}^T] = 0.$$

Since M_0 is a variance covariance matrix, it is non–singular, and therefore its inverse exists.

The matrix A is given by

$$A = M_1 M_0^{-1}.$$

Postmultiplying (7) through by $\chi_{i,t'}^T$ and taking expectations one gets

$$M_0 = AM_1^T + BB^T$$

since $E[\epsilon_{i,t}\epsilon_{i,t}^T] = I$, the identity matrix.

Therefore, the matrix B is given by the solution to

$$BB^T = M_0 - M_1 M_0^{-1} M_1^T.$$

The Cholesky decomposition (Appendix C) can be used to solve for B .

Now, we will discuss the method to obtain parameter estimates for the coefficients of the truncated Fourier series.

The functions μ_t and σ_t are estimated using the same method but different data sets. The theory will thus be discussed for the mean function μ_t only.

Let $\overline{S}_t$ be the daily mean vector for $S_{i,t}$ and assume that it is given by the linear model

$$\overline{S}_t = x_t^T\beta + e_t, \qquad t = 1, 2, \ldots, NT$$

with

$$e_t \sim NID(0, \sigma_t^2).$$

This is a special case of a generalized linear model because the elements $\overline{S}_t$ are independent with distributions $N(\mu_t, \sigma_t^2)$ where

$$\mu_t = x_t^T\beta.$$

Also the normal distribution is a member of the exponential family (provided the σ_t^2 are regarded as known). In this case the link function, g , is the identity function, i.e.

$$g(\mu_t) = \mu_t = \sum_{i=1}^{L} \alpha_i\varphi_i(t) = \eta_t$$

where $\sum_{i=1}^{L} \alpha_i\varphi_i(t)$ represents the truncated Fourier series of the mean function μ_t , and $\varphi_i(t)$ is defined as before.

The log–likelihood function of the observed mean values as a function of the mean function μ_t is given by:

$$\ell(\mu_t; \overline{S}_t) = \left[\sum_{t=1}^{NT} -\frac{\overline{S}_t^2}{2\sigma_t^2} + \frac{\overline{S}_t\mu_t}{\sigma_t^2} - \frac{\mu_t^2}{2\sigma_t^2} - \frac{1}{2}\log(2\pi\sigma_t^2)\right].$$

Therefore, the log–likelihood function of the observed values as a function of the parameters $\alpha_1, \alpha_2, \ldots, \alpha_L$ is given by

$$\ell(\alpha;\overline{S}_t) = \sum_{t=1}^{NT}\Bigg[-\frac{\overline{S}_t^2}{2\sigma_t^2} + \frac{\overline{S}_t \sum_{i=1}^{L} \alpha_i\varphi_i(t)}{\sigma_t^2} - \frac{\sum_{i=1}^{L} \alpha_i\varphi_i(t)}{2\sigma_t^2} - \frac{1}{2}\log(2\pi\sigma_t^2)\Bigg]$$

The score vector U with respect to $\alpha_1, \alpha_2, \ldots, \alpha_L$ has elements given by

$$U_j = \frac{\partial \ell(\alpha;\overline{S}_t)}{\partial \alpha_j} = \sum_{t=1}^{NT}\left[\frac{(\overline{S}_t - \mu_t)}{\sigma_t^2}\ \varphi_j(t)1\right]$$

since

$$E(\overline{S}_t) = \mu_t\ ,$$

$$\mathrm{Var}(\overline{S}_t) = \sigma_t^2\ , \qquad \text{and}$$

$$\partial\mu_t/\partial\eta_t = 1.$$

Similarly, the information matrix $\mathbf{I}_{L\times L}$ has elements given by

$$\mathbf{I}_{jk} = \sum_{t=1}^{NT}\frac{\varphi_j(t)\varphi_k(t)}{\sigma_t^2}\ 1^2.$$

The maximum likelihood estimates for $\alpha_1, \alpha_2, \ldots, \alpha_L$ are then obtained by solving the iterative equation

$$\mathbf{I}^{(m-1)}\widehat{\alpha}^{(m)} = \mathbf{I}^{(m-1)}\widehat{\alpha}^{(m-1)} + U^{(m-1)}$$

where m indicates the mth approximation and $\widehat{\alpha}$ is the vector of estimates.

When the difference between successive approximations $\widehat{\alpha}^{(m)}$ and $\widehat{\alpha}^{(m-1)}$ is sufficiently small, $\widehat{\alpha}^{(m)}$ is taken as the maximum likelihood estimate vector.

(d) Model selection

The order of the Fourier series approximation, L , for the conditioned mean function and for the conditioned standard deviation function is selected by Akaike Information Criterion, AIC, where

$$\mathrm{AIC} = -\ell(\alpha;\theta) + L$$

where $\ell(\alpha;\theta)$ is the log–likelihood function of the model. Each value of L leads to a different approximating model. The criterion is computed for $\alpha = 1, 3, 5, \ldots$ and the model which leads to the smallest value of the criterion is selected.

Model 2: Multivariate model for climate data

Although Model 1 appeared to be satisfactory in many respects, it performed poorly in some respects. In particular the annual standard deviation for windrun, maximum and minimum humidity were systematically underestimated. The (lagged) cross–correlations between some of the variables (e.g. maximum temperature and minimum temperature) were not preserved by the model. However the most noticeable deficiency was found to be that the model did not preserve the serial correlation structure of many of the variables. This was attributed to the lack of flexibility of Model 1 in this respect. In particular the model is based on the assumption that the serial correlation function does not depend on the wet/dry status of the days in question. In fact the correlation between variables on two successive days depends on whether the two days are both wet, both dry, wet followed by dry or dry followed by wet. It was therefore decided to develop a model which incorporates additional flexibility in its autocorrelation function, that is, a model which allows for the serial correlations between variables on successive days to depend on their wet/dry status.

Model 2 was developed as a prototype to Models 3, 4 and 5. It attempts to deal with the mentioned deficiency in Model 1.

(a) Notation

Partition the year into $NT(=365)$ equal intervals, denoted by $t = 1, 2, \ldots NT$.

NV is the number of variables.

NY is the number of years observed.

W represents the occurrence of rain.

D represents the non–occurrence of rain.

$N(D)$ is the set of time periods t such that period t was dry.

$N(W)$ is the set of time periods t such that period t was wet.

$Y_{i,t}$ is the precipitation amount on period t of year i, $i = 1, 2, \ldots, NY$.

$S_{i,t}$ is the generic name for the observation at time t of the ith year.

μ_t^D is the generic name for the mean for a dry day on period t (i.e. $Y_{i,t} = 0$) .

μ_t^W is the generic name for the mean for a wet day on period t (i.e. $Y_{i,t} > 0$) .

σ^D is the generic name for the standard deviation for a dry day.

σ^W is the generic name for the standard deviation for a wet day.

θ^D is the coefficient of the AR(1) process, given a dry day.

θ^W is the coefficient of the AR(1) process given a wet day.

$C(D)$ denotes the number of elements in the set $N(D)$.

$C(W)$ denotes the number of elements in the set $N(W)$.

Then $T = C(D) + C(W)$.

Since all variables are modelled in the same way, the representation will be given for modelling one variable. The same procedure is then repeated for each of the remaining climate variables.

(b) Model and assumptions

The Model under consideration is given by:

$$\chi_{i,t} = \begin{cases} \dfrac{S_{i,t} - \mu_t^D}{\sigma^D} & \text{if } Y_{i,t} = 0 \\[2ex] \dfrac{S_{i,t} - \mu_t^W}{\sigma^W} & \text{if } Y_{i,t} > 0 \end{cases}$$

where $i = 1, 2, \ldots, NY$ and $t = 1, 2, \ldots, NT$.

That is, the residual time series $\chi_{i,t}$ is obtained by removing the periodic mean and the standard deviation from the observed time series $S_{i,t}$. The resulting time series thus has a mean of zero and unit variance.

Assume $\chi_{i,t}$ is generated by an autoregressive process of order p (AR(p)) defined as

$$\chi_{i,t} = \theta_1 \chi_{i,t-1} + \theta_2 \chi_{i,t-2} + \cdots + \theta_p \chi_{i,t-p} + e_{i,t}$$

where $\{e_{i,t}\}$ is a set of independent, normally distributed variables with mean zero and variance of unity, i.e.

$$e_{i,t} \sim NID(0,1).$$

That is, $\chi_{i,t}$ is regressed on past values of $\chi_{i,t}$ instead of on independent variables as on the classical multiple regression.

The assumption that $\chi_{i,t}$ is described by an autoregressive process can be substantiated by arguments put forward by Cochrane and Orcutt (1949). The sources of autocorrelation in the error term can be:

(i) When modelling climatic variables, errors in modelling arise from faulty descriptions of these variables. Since these variables are themselves autocorrelated, this type of error will be autocorrelated.

(ii) Error terms may arise from omitting variables from the analysis because these variables are either not available or their importance is not realised or because the influence they have is so small that it is not convenient to insert them. As already indicated these variables are autocorrelated and, therefore, one may expect the resulting error terms to be also autocorrelated.

An autoregressive process of order 1, AR(1), sometimes called the Markov process, was chosen to describe $\chi_{i,t}$. The rèason for this choice will be discussed later. To simplify the formula, the theory will only be shown for an AR(1) process from now on. The order of the process can be increased to any order desired, but this has to be done at the cost of increasing both the complexity and the number of parameters to be estimated.

The form of $\chi_{i,t}$ is thus given by:

$$\chi_{i,t} = \theta\chi_{i,t-1} + e_{i,t}$$

where $e_{i,t} \sim NID(0,1) \qquad i = 1,2,\ldots,NY;\ t = 1,2,\ldots,NT$.

The model that incorporates the different wet and dry sequences is given by:

$$e_{i,t} = \frac{S_{i,t} - \mu_t^D}{\sigma^D} - \theta^D \frac{S_{i,t-1} - \mu_{t-1}^D}{\sigma^D} \qquad \text{for a dry sequence}$$

or

$$e_{i,t} = \frac{S_{i,t} - \mu_t^W}{\sigma^W} - \theta^W \frac{S_{i,t-1} - \mu_{t-1}^W}{\sigma^W} \qquad \text{for a wet sequence.}$$

The seasonal mean function, μ_t , is approximated by its truncated Fourier series representation, i.e.

$$\mu_t^D = \sum_{i=1}^{L} \alpha_i^D \varphi_i(t) \qquad \text{if } t \text{ dry}$$

and

$$\mu_t^W = \sum_{i=1}^{L} \alpha_i^W \varphi_i(t) \qquad \text{if } t \text{ wet}$$

where $\varphi_i(t)$ is defined as before and L is the order of the Fourier series representation.

(c) Estimation

The procedure to estimate the parameters $\alpha_j^D, \alpha_j^W, \theta^D, \theta^W, \sigma^D$ and σ^W; $j = 1,2,\ldots,L$ is now discussed.

Since $e_{i,t} \sim NID(0,1)$, the density function of $e_{i,t}$ is given by

$$f(e_{i,t}) = \frac{1}{\sqrt{2\pi}} \exp(-\frac{1}{2}e_{i,t}^2)$$

Therefore the joint likelihood function, conditioned on the wet and dry status of the day is given by

$$\begin{aligned} L(\psi) &= L(\alpha_j^D, \alpha_j^W, \theta^D, \theta^W, \sigma^D, \sigma^W; e_{i,t}) \\ &= \prod_{t \in N(D)} f(e_{i,t}|D) \prod_{t \in N(W)} f(e_{i,t}|W) \end{aligned}$$

where $f(e_{i,t}|D)$ represents the density function of $e_{i,t}$ given that a dry day has been observed, and $f(e_{i,t}|W)$ represents the density function of $e_{i,t}$ given that a wet day has been observed.

Substituting the density function, one obtains

$$L(\psi) = \left(\frac{1}{\sqrt{2\pi}}\right)^T \exp\left\{-\frac{1}{2}\left[\sum_{t \in N(DD)} (e_{i,t}|D)^2 + \sum_{t \in N(WW)} (e_{i,t}|W)^2\right]\right\}.$$

Making the following transformation

$$e_{i,t} = \frac{S_{i,t} - \mu_t}{\sigma} - \theta\frac{S_{i,t-1} - \mu_{t-1}}{\sigma}$$

where the Jacobian of the transformation is given by

$$\left|\frac{\partial e_{i,t}}{\partial S_{i,p}}\right|, \quad i = 1,2,\ldots,NY; \; t = 1,2,\ldots,NT; \; p = 1,2,\ldots,NT$$

$$= \left|\frac{\partial e_k}{\partial S_n}\right|, \quad k = 1,2,\ldots,T; \; n = 1,2,\ldots,T$$

$$= \begin{vmatrix} 1/\sigma & 0 & \ldots & \ldots & 0 \\ -\theta/\sigma & 1/\sigma & 0 & \ldots & 0 \\ \vdots & & & & \vdots \\ 0 & \ldots & \ldots & -\theta/\sigma & 1/\sigma \end{vmatrix} = \prod_{k=1}^{T} 1/\sigma = \prod_{i=1}^{NY}\prod_{t=1}^{NT} 1/\sigma$$

since we are dealing with a triangular matrix. Taking into account the dry and wet status of the day, the Jacobian can be rewritten as

$$\left|\frac{\partial e_{i,t}}{\partial S_{i,p}}\right| = \prod_{t\in N(D)} \frac{1}{\sigma^D} \prod_{t\in N(W)} \frac{1}{\sigma^W},$$

then the joint likelihood function is given by

$$L(\psi) = \left(\frac{1}{\sqrt{2\pi}}\right)^T \left(\frac{1}{\sigma^D}\right)^{C(D)} \left(\frac{1}{\sigma^W}\right)^{C(W)}$$
$$\exp\left\{-\frac{1}{2}\left[\sum_{t\in N(D)} \left(\frac{S_{i,t}-\mu_t^D}{\sigma^D} - \theta^D \frac{S_{i,t-1}-\mu_{t-1}^D}{\sigma^D}\right)^2\right.\right.$$
$$\left.\left.+ \sum_{t\in N(W)} \left(\frac{S_{i,t}-\mu_t^W}{\sigma^W} - \theta^W \frac{S_{i,t-1}-\mu_{t-1}^W}{\sigma^W}\right)^2\right]\right\}$$

and the log likelihood function is given by

$$\ell(\psi) = -\frac{T}{2}\log(2\pi) - C(D)\log(\sigma^D) - C(W)\log(\sigma^W)$$
$$-\frac{1}{2}\left[\sum_{t\in N(D)} \left(\frac{S_{i,t}-\mu_t^D}{\sigma^D} - \theta^D \frac{S_{i,t-1}-\mu_{t-1}^D}{\sigma^D}\right)^2\right.$$
$$\left.+ \sum_{t\in N(W)} \left(\frac{S_{i,t}-\mu_t^W}{\sigma^W} - \theta^W \frac{S_{i,t-1}-\mu_{t-1}^W}{\sigma^W}\right)^2\right].$$

Maximum likelihood estimates can be obtained by minimising $\ell(\psi)$. That is, the first partial derivatives with respect to the parameters are set to zero.

The first partial derivatives with respect to the parameters are given by

$$\frac{\partial \ell(\psi)}{\partial \theta^D} = \sum_{t\in N(D)} \left(\frac{S_{i,t}-\mu_t^D}{\sigma^D} - \theta^D \frac{S_{i,t-1}-\mu_{t-1}^D}{\sigma^D}\right)$$
$$\left(\frac{S_{i,t-1}-\mu_{t-1}^D}{\sigma^D}\right)$$

$$\frac{\partial \ell(\psi)}{\partial \theta^W} = \sum_{t\in N(W)} \left(\frac{S_{i,t}-\mu_t^W}{\sigma^W} - \theta^W \frac{S_{i,t-1}-\mu_{t-1}^W}{\sigma^W}\right)$$
$$\left(\frac{S_{i,t-1}-\mu_{t-1}^W}{\sigma^W}\right)$$

$$\frac{\partial \ell(\psi)}{\partial \alpha_j^D} = -\left[\sum_{t\in N(D)} \left(\frac{S_{i,t}-\mu_t^D}{\sigma^D} - \theta^D \frac{S_{i,t-1}-\mu_{t-1}^D}{\sigma^D}\right)\right.$$
$$\left.\left(\frac{-\varphi_j(t)}{\sigma^D} + \frac{\theta^D \varphi_j(t-1)}{\sigma^D}\right)\right]$$

$$\frac{\partial \ell(\psi)}{\partial \alpha_j^W} = -\left[\sum_{t\in N(W)} \left(\frac{S_{i,t}-\mu_t^W}{\sigma^W} - \theta^W \ \frac{S_{i,t-1}-\mu_{t-1}^W}{\sigma^W}\right)\right.$$
$$\left.\left(\frac{-\varphi_j(t)}{\sigma^W} + \frac{\theta^W \varphi_j(t-1)}{\sigma^W}\right)\right]$$

$$\frac{\partial \ell(\psi)}{\partial \sigma^D} = -\frac{C(D)}{\sigma^D} - \sum_{t\in N(D)} \left(\frac{S_{i,t}-\mu_t^D}{\sigma^D} - \theta^D \ \frac{S_{i,t-1}-\mu_{t-1}^D}{\sigma^D}\right)$$
$$\left(-\frac{(S_{i,t}-\mu_t^D)}{(\sigma^D)^2} + \theta^D \frac{S_{i,t-1}-\mu_{t-1}^D}{(\sigma^D)^2}\right)$$

$$\frac{\partial \ell(\psi)}{\partial \sigma^W} = -\frac{C(W)}{\sigma^W} - \sum_{t\in N(W)} \left(\frac{S_{i,t}-\mu_t^W}{\sigma^W} - \theta^W \ \frac{S_{i,t-1}-\mu_{t-1}^W}{\sigma^W}\right)$$
$$\left(-\frac{(S_{i,t}-\mu_t^W)}{(\sigma^W)^2} + \theta^D \frac{S_{i,t-1}-\mu_{t-1}^W}{(\sigma^W)^2}\right)$$

The parameter estimates are given by

$$\widehat{\theta}^D = \frac{\sum\limits_{t\in N(D)} (S_{i,t}-\widehat{\mu}_t^D)(S_{i,t-1}-\widehat{\mu}_{t-1}^D)}{\sum\limits_{t\in N(D)} (S_{i,t-1}-\widehat{\mu}_{t-1}^D)^2} \quad (1)$$

$$\widehat{\theta}^W = \frac{\sum\limits_{t\in N(W)} (S_{i,t}-\widehat{\mu}_t^W)(S_{i,t-1}-\widehat{\mu}_{t-1}^W)}{\sum\limits_{t\in N(W)} (S_{i,t-1}-\widehat{\mu}_{t-1}^W)^2} \quad (2)$$

$$\widehat{\sigma}^D = \left(\frac{1}{C(D)} \sum_{t\in N(D)} (S_{i,t}-\widehat{\mu}_t^D - \widehat{\theta}^D(S_{i,t-1}-\widehat{\mu}_{t-1}^D))^2\right)^{\frac{1}{2}} \quad (3)$$

$$\widehat{\sigma}^W = \left(\frac{1}{C(W)} \sum_{t\in N(W)} (S_{i,t}-\widehat{\mu}_t^W - \widehat{\theta}^W(S_{i,t-1}-\widehat{\mu}_{t-1}^W))^2\right)^{\frac{1}{2}} \quad (4)$$

$$\widehat{\alpha}_j^D = \left\{\sum_{t\in N(D)} (\varphi_j(t) - \widehat{\theta}^D \varphi_j(t-1))^2\right\}^{-1} [A - M] \quad (5)$$

where

$$A = \sum_{t\in N(D)} (S_{i,t} - \widehat{\theta}^D S_{i,t-1})(\varphi_j(t) - \widehat{\theta}^D \varphi_j(t-1))$$

and

$$M = \sum_{t \in N(D)} \left[\left(\sum_{\substack{k=1 \\ k \neq j}}^{L} \widehat{\alpha}_k^D \varphi_k(t) \right) - \widehat{\theta}^D \left(\sum_{\substack{k=1 \\ k \neq j}}^{L} \widehat{\alpha}_k^D \varphi_j(t-1) \right) \right]$$
$$[\varphi_j(t) - \widehat{\theta}^D \varphi_j(t-1)]$$

$$\widehat{\alpha}_j^W = \left\{ \sum_{t \in N(W)} (\varphi_j(t) - \widehat{\theta}^W \varphi_j(t-1))^2 \right\}^{-1} [A_2 - M_2] \qquad (6)$$

where

$$A_2 = \sum_{t \in N(W)} (S_{i,t} - \widehat{\theta}^W S_{i,t-1})(\varphi_j(t) - \widehat{\theta}^W \varphi_j(t-1))$$

and

$$M_2 = \sum_{t \in N(W)} \left[\left(\sum_{\substack{k=1 \\ k \neq j}}^{L} \widehat{\alpha}_k^W \varphi_k(t) \right) - \widehat{\theta}^W \left(\sum_{\substack{k=1 \\ k \neq j}}^{L} \widehat{\alpha}_k^W \varphi_k(t-1) \right) \right]$$
$$[\varphi_j(t) - \widehat{\theta}^W \varphi_j(t-1)]$$

These equations cannot be solved explicitly and therefore have to be solved iteratively. Note that

μ_t is a function of the α_j where the α_j are functions of θ

θ is a function of μ_t and

σ is a function of μ_t and θ .

The following algorithm can be performed to estimate the parameters.

Algorithm

Step 1: Estimate initial μ_t by approximating by it Fourier series transformation and estimating the parameters α_i by the method mentioned in the previous models.

Step 2: Estimate θ using (1) and (2)

Step 3: Estimate σ using (3) and (4)

Step 4: Estimate μ_t using (5) and (6)

Step 5: Test for convergence of all parameters, i.e. when the percentage change in parameter estimates is sufficiently small. If convergence is not met, return to Step 2.

To model the multivariate time series the covariance matrix of the residuals of the different climate variables is needed.

The cross-correlation matrix, $\widehat{\Sigma}$, has elements R_{jk} , where

$$R_{jk} = \frac{\frac{1}{T}\sum_{i=1}^{NY}\sum_{t=1}^{NT} e_{i,t}^{(j)} e_{i,t}^{(k)} - \frac{1}{T^2}\sum_{i=1}^{NY}\sum_{t=1}^{NT} e_{i,t}^{(j)} \sum_{i=1}^{NY}\sum_{t=1}^{NT} e_{i,t}^{(k)}}{\left[\frac{1}{T}\sum_{i=1}^{NY}\sum_{t=1}^{NT}(e_{i,t}^{(j)})^2 - \frac{1}{T^2}\left(\sum_{i=1}^{NY}\sum_{t=1}^{NT} e_{i,t}^{(j)}\right)^2\right]^{\frac{1}{2}} \left[\frac{1}{T}\sum_{i=1}^{NY}\sum_{t=1}^{NT}(e_{i,t}^{(k)})^2 - \frac{1}{T^2}\left(\sum_{i=1}^{NY}\sum_{t=1}^{NT} e_{i,t}^{(k)}\right)^2\right]^{\frac{1}{2}}}$$

where $e_{i,t}^{(j)}$ denotes the residual time series of variable j, $j = 1, 2, \ldots, NV$ and $e_{i,t}^{(k)}$ denotes the residual time series of variable k, $k = 1, 2, \ldots, NV$.

(d) **Model Selection**

The order of the autoregressive process is selected in the same way as in the previous models, as is the order of the Fourier series approximation.

A major problem was encountered in Model 2, in that a high proportion of information is discarded. The reason why this problem occurs is explained by means of an example.

Suppose the following sequence has been observed

day t	1	2	3	4	5	6	7	8	9	10	11	12
status of day	D	D	W	D	W	W	D	W	D	D	D	W

By definition, $N(D)$ and $N(W)$ represent the sets of time periods t such that t was dry or wet respectively. Thus, $N(D)$ consists of the elements

$$\{1, 2, 4, 7, 9, 10, 11\},$$

and $N(W)$ of

$$\{3, 5, 6, 8, 12\}.$$

In Model 2, one is only interested in conditioning the parameters into dry and wet sequences, therefore one is restricting the status of time period $t-1$ to be the same as that of t . Thus, given that day t was dry, the model is given by

$$e_{i,t} = \frac{S_{i,t} - \mu_t^D}{\sigma^D} - \theta^D \frac{S_{i,t-1} - \mu_{t-1}^D}{\sigma^D}$$

and one does not consider the case of

$$e_{i,t} = \frac{S_{i,t} - \mu_t^D}{\sigma^D} - \theta \frac{S_{i,t-1} - \mu_{t-1}^W}{\sigma^W}.$$

Similarly for when t is wet.

Therefore, when constructing $N(D)$ and $N(W)$, only sequences of at least two dry (or two wet) consecutive days can be used. In this case, $N(D)$ has elements

$$\{2, 10, 11\}$$

and $N(W)$

$$\{6\}.$$

Thus, a high proportion of the observations are discarded. This led to the development of Model 3 and therefore Model 2 is of no further interest.

Model 3: Multivariate model for climate data

The two previous models condition the parameters of the model on the wet or dry status of the day. When generating climate sequences these models represent conditions in which a wet day follows a wet day and a dry day follows a dry day but fail to explain the relationship between conditions such as a wet day following a dry day or a dry day following a wet day.

To generate representative climate sequences these sequences must be related to the sequences of rain and no–rain days. To achieve this relationship, the parameters of the model must be conditioned on the four possible sequences in the rainfall variable:

1. a dry day follows a dry day,
2. a wet day follows a wet day,
3. a wet day follows a dry day,
4. a dry day follows a wet day.

(a) **Notation**

Partition the year into $NT(=365)$ equal intervals, denoted by $t = 1, 2, \ldots, NT$

NV is the number of variables.

NY is the number of years observed.

W represents the occurrence of rain.

D represents the non–occurrence of rain.

DD represents the sequence when day $t-1$ was dry and day t was dry.

WW represents the sequence when day $t-1$ was wet and day t was wet.

DW represents the sequence when day $t-1$ was dry and day t was wet.

WD represents the sequence when day $t-1$ was wet and day t was dry.

T represents the total number of observations, i.e. $NYNT$.

$N(DD)$ is the set of time periods t such that period t was dry and period $t-1$ was dry, $t = 1, 2, \ldots, T$.

$N(WW)$ is the set of time periods t such that period t was wet and period $t-1$ was wet.

$N(DW)$ is the set of time periods t such that period t was wet and period $t-1$ was dry.

$N(WD)$ is the set of time periods t such that period t was dry and period $t-1$ was wet.

$Y_{i,t}$ is the precipitation amount of period t of year i, $i = 1,2,\ldots,NY$.

$S_{i,t}$ is the generic name for the observation at time t of the ith year.

μ_t^D is the generic name for the mean for a dry day on period t.

μ_t^W is the generic name for the mean for a wet day on period t.

σ^{DD} is the generic name for the standard deviation given sequence DD.

σ^{WW} is the generic name for the standard deviation given sequence WW.

σ^{DW} is the generic name for the standard deviation given sequence DW.

σ^{WD} is the generic name for the standard deviation given sequence WD.

θ^{DD} is the coefficient of the AR(1) process given sequence DD.

θ^{WW} is the coefficient of the AR(1) process given sequence WW

θ^{DW} is the coefficient of the AR(1) process given sequence DW.

θ^{WD} is the coefficient of the AR(1) process given sequence WD.

$C(DD)$ is the number of elements in the set $N(DD)$.

$C(WW)$ is the number of elements in the set $N(WW)$.

$C(DW)$ is the number of elements in the set $N(DW)$.

$C(WD)$ is the number of elements in the set $N(WD)$.

Then $T = C(DD) + C(WW) + C(DW) + C(WD)$.

(b) Model and assumptions

The time series $S_{i,t}$ is reduced to a time series of residual elements, $\chi_{i,t}$, by removing the periodic means and the standard deviation for the appropriate sequence, i.e.

$$\chi_{i,t} = \frac{S_{i,t} - \mu_t}{\sigma}$$

This results in a time series with zero mean and standard deviation of unity, which is assumed to follow an AR(1) process, i.e.

$$\chi_{i,t} = \theta\, \chi_{i,t-1} + e_{i,t}$$

where $e_{i,t} \sim NID(0,1) \qquad i = 1,2,\ldots NY; \quad t = 1,2,\ldots,NT$.

The model that incorporates the different wet–dry sequences is then given by:

$$e_{i,t} = \frac{S_{i,t} - \mu_t^{DD}}{\sigma^{DD}} - \theta^{DD}\frac{S_{i,t-1} - \mu_{t-1}^{DD}}{\sigma^{DD}} \quad \text{if day } t-1 \text{ was dry and day } t \text{ was dry.}$$

$$e_{i,t} = \frac{S_{i,t} - \mu_t^{W}}{\sigma^{WW}} - \theta^{WW}\frac{S_{i,t-1} - \mu_{t-1}^{W}}{\sigma^{WW}} \quad \text{if day } t-1 \text{ was wet and day } t \text{ was wet.}$$

$$e_{i,t} = \frac{S_{i,t} - \mu_t^{W}}{\sigma^{DW}} - \theta^{DW}\frac{S_{i,t-1} - \mu_{t-1}^{D}}{\sigma^{DW}} \quad \text{if day } t-1 \text{ was dry and day } t \text{ was wet.}$$

$$e_{i,t} = \frac{S_{i,t} - \mu_t^{D}}{\sigma^{WD}} - \theta^{WD}\frac{S_{i,t-1} - \mu_{t-1}^{W}}{\sigma^{WD}} \quad \text{is day } t-1 \text{ was wet and day } t \text{ was dry.}$$

The parameter μ_t for this model, and for the models following, was only conditioned on the rain and no–rain status of the day. This was done to simplify the model, otherwise for each sequence one would have two equations. For example, if sequence DD has occurred, then

$$e_{i,t} = \frac{S_{i,t} - \mu_t^{D}}{\sigma^{DD}} - \theta^{DD}\frac{S_{i,t-1} - \mu_{t-1}^{D}}{\sigma^{DD}} \quad \text{if the sequence } DDD \text{ was observed}$$

or

$$e_{i,t} = \frac{S_{i,t} - \mu_t^{DD}}{\sigma^{DD}} - \theta^{DD}\frac{S_{i,t-1} - \mu_{t-1}^{WD}}{\sigma^{DD}} \quad \text{if the sequence } WDD \text{ was observed}$$

This not only increases the number of parameters to be estimated but in addition we are no longer assuming that all the information we need of previous values of the model is given by the value of the previous day. The state of the second previous day is also required.

As already discussed, it is reasonable to appproximate the mean function μ_t by its truncated Fourier representation, i.e.

$$\mu_t^D = \sum_{i=1}^{L} \alpha_i^D \varphi_i(t) \qquad \text{if } t \text{ dry}$$

and

$$\mu_t^W = \sum_{i=1}^{L} \alpha_t^W \varphi_i(t) \qquad \text{if } t \text{ wet}$$

where $\varphi_i(t)$ is defined as before and L is the order of the Fourier series approximation.

(c) Estimation

The procedure to estimate the parameters $\alpha_j^D, \alpha_j^W, \theta^{DD}, \theta^{WW}, \theta^{WD}, \theta^{DW}, \sigma^{DD}, \sigma^{WW}, \sigma^{DW}, \sigma^{WD}$; $j = 1, 2, \ldots, L$ is now discussed.

Maximum likelihood estimates can be obtained by observing that since $e_{i,t} \sim NID(0,1)$, then the density function of $e_{i,t}$ is given by

$$f(e_{i,t}) = \frac{1}{\sqrt{2\pi}} \exp(-\frac{1}{2} e_{i,t}^2)$$

Therefore the joint likelihood function conditioned on the four different sequences is given by

$$L(\psi) = L(\alpha_j^D, \alpha_j^W, \theta^{DD}, \theta^{WW}, \theta^{WD}, \theta^{DW}\sigma^{DD}, \sigma^{WW}, \sigma^{DW}\sigma^{WD};\ e_{i,t})$$
$$= \prod_{t\in N(DD)} f(e_{i,t}|DD) \prod_{t\in N(WW)} f(e_{i,t}|WW) \prod_{t\in N(DW)} f(e_{i,t}|DW) \prod_{t\in N(WD)} f(e_{i,t}|WD)$$

where $f(e_{i,t}|DD)$ represents the density function of $e_{i,t}$ given that the sequence DD has been observed, and similarly for the other density functions.

$$= \left(\frac{1}{\sqrt{2\pi}}\right)^T \exp\left\{-\frac{1}{2}\left[\sum_{t\in N(DD)} (e_{i,t}|DD)^2 + \sum_{t\in N(WW)} (e_{i,t}|WW)^2 + \sum_{t\in N(DW)} (e_{i,t}|DW)^2 + \sum_{t\in N(WD)} (e_{i,t}|WD)^2\right]\right\}$$

One now makes the following transformation

$$e_{i,t} = \frac{S_{i,t} - \mu_t}{\sigma} - \theta\ \frac{S_{i,t-1} - \mu_{t-1}}{\sigma}.$$

The Jacobian of the transformation is given by

$$\left|\frac{\partial e_{i,t}}{\partial S_{i,p}}\right| \ , i = 1, 2, \ldots, NY; \quad t = 1, 2, \ldots, NT; \quad p = 1, 2, \ldots, NT.$$
$$= \left|\frac{\partial e_k}{\partial S_n}\right| \ , k = 1, 2, \ldots, T; \quad n = 1, 2, \ldots, T.$$
$$= \begin{vmatrix} 1/\sigma & 0 & \ldots & \ldots & 0 \\ -\theta/\sigma & 1/\sigma & 0 & \ldots & 0 \\ \vdots & & & & \\ 0 & \ldots & \ldots & -\theta/\sigma & 1/\sigma \end{vmatrix} = \prod_{k=1}^{T} 1/\sigma = \prod_{i=1}^{NY}\prod_{t=1}^{NT} 1/\sigma$$

since we are dealing with a triangular matrix.

Taking into account the conditional sequences imposed on $e_{i,t}$, the Jacobian is given by

$$\left|\frac{\partial e_{i,t}}{\partial S_{i,p}}\right| = \prod_{t\in N(DD)} \frac{1}{\sigma^{DD}} \prod_{t\in N(WW)} \frac{1}{\sigma^{WW}} \prod_{t\in N(DW)} \frac{1}{\sigma^{DW}} \prod_{t\in N(WD)} \frac{1}{\sigma^{WD}}.$$

Then the joint probability density function of $S_{i,t}$ is given by

$$\begin{aligned} L(\psi) = &\left(\frac{1}{\sqrt{2\pi}}\right)^{T} \left(\frac{1}{\sigma^{DD}}\right)^{C(DD)} \left(\frac{1}{\sigma^{WW}}\right)^{C(WW)} \left(\frac{1}{\sigma^{DW}}\right)^{C(DW)} \left(\frac{1}{\sigma^{WD}}\right)^{C(WD)} \\ &\exp\left\{-\frac{1}{2}\left[\sum_{t\in N(DD)} \left(\frac{S_{i,t}-\mu_t^{DD}}{\sigma^{DD}} - \theta^{DD}\frac{S_{i,t-1}-\mu_{t-1}^{DD}}{\sigma^{DD}}\right)^2\right.\right. \\ &+ \sum_{t\in N(WW)} \left(\frac{S_{i,t}-\mu_t^{W}}{\sigma^{WW}} - \theta^{WW}\frac{S_{i,t-1}-\mu_{t-1}^{W}}{\sigma^{WW}}\right)^2 \\ &+ \sum_{t\in N(DW)} \left(\frac{S_{i,t}-\mu_t^{W}}{\sigma^{DW}} - \theta^{DW}\frac{S_{i,t-1}-\mu_{t-1}^{D}}{\sigma^{DW}}\right)^2 \\ &\left.\left.+ \sum_{t\in N(WD)} \left(\frac{S_{i,t}-\mu_t^{D}}{\sigma^{WD}} - \theta^{WD}\frac{S_{i,t-1}-\mu_{t-1}^{W}}{\sigma^{WD}}\right)^2\right]\right\} \end{aligned}$$

and the log likelihood is given by:

$$\begin{aligned} \ell(\psi) = &\frac{-T}{2}\log(2\pi) - C(DD)\log(\sigma^{DD}) - C(WW)\log(\sigma^{WW}) \\ &- C(DW)\log(\sigma^{DW}) - C(WD)\log(\sigma^{WD}) \\ &-\frac{1}{2}\left[\sum_{t\in N(DD)} \left(\frac{S_{i,t}-\mu_t^{DD}}{\sigma^{DD}} - \theta^{DD}\frac{S_{i,t-1}-\mu_{t-1}^{DD}}{\sigma^{DD}}\right)^2\right. \\ &+ \sum_{t\in N(WW)} \left(\frac{S_{i,t}-\mu_t^{W}}{\sigma^{WW}} - \theta^{WW}\frac{S_{i,t-1}-\mu_{t-1}^{W}}{\sigma^{WW}}\right)^2 \\ &+ \sum_{t\in N(DW)} \left(\frac{S_{i,t}-\mu_t^{W}}{\sigma^{DW}} - \theta^{DW}\frac{S_{i,t-1}-\mu_{t-1}^{D}}{\sigma^{DW}}\right)^2 \\ &\left.+ \sum_{t\in N(WD)} \left(\frac{S_{i,t}-\mu_t^{D}}{\sigma^{WD}} - \theta^{WD}\frac{S_{i,t-1}-\mu_{t-1}^{W}}{\sigma^{WD}}\right)^2\right]. \end{aligned}$$

Maximum likelihood estimates can be obtained by minimising $\ell(\psi)$ and this is achieved by setting its partial derivatives with respect to the parameters equal to zero.

The first partial derivatives with respect to the parameteers are given by:

$$\frac{\partial\ell(\psi)}{\partial\theta^{DD}} = \sum_{t\in N(DD)} \left(\frac{S_{i,t}-\mu_t^{DD}}{\sigma^{DD}} - \theta^{DD}\frac{S_{i,t-1}-\mu_{t-1}^{DD}}{\sigma^{DD}}\right)\left(\frac{S_{i,t-1}-\mu_{t-1}^{D}}{\sigma^{DD}}\right)$$

$$\frac{\partial\ell(\psi)}{\partial\theta^{WW}} = \sum_{t\in N(WW)} \left(\frac{S_{i,t}-\mu_t^{W}}{\sigma^{WW}} - \theta^{WW}\frac{S_{i,t-1}-\mu_{t-1}^{W}}{\sigma^{WW}}\right)\left(\frac{S_{i,t-1}-\mu_{t-1}^{W}}{\sigma^{WW}}\right)$$

$$\frac{\partial\ell(\psi)}{\partial\theta^{DW}} = \sum_{t\in N(DW)} \left(\frac{S_{i,t}-\mu_t^{W}}{\sigma^{DW}} - \theta^{DW}\frac{S_{i,t-1}-\mu_{t-1}^{D}}{\sigma^{DW}}\right)\left(\frac{S_{i,t-1}-\mu_{t-1}^{D}}{\sigma^{DW}}\right)$$

$$\frac{\partial\ell(\psi)}{\partial\theta^{WD}} = \sum_{t\in N(WD)} \left(\frac{S_{i,t}-\mu_t^{D}}{\sigma^{WD}} - \theta^{WD}\frac{S_{i,t-1}-\mu_{t-1}^{W}}{\sigma^{WD}}\right)\left(\frac{S_{i,t-1}-\mu_{t-1}^{W}}{\sigma^{WD}}\right)$$

$$\begin{aligned}\frac{\partial\ell(\psi)}{\partial\alpha_j^{D}} = -\Bigg[&\sum_{t\in N(DD)} \left(\frac{S_{i,t}-\mu_t^{DD}}{\sigma^{DD}} - \theta^{DD}\frac{S_{i,t-1}-\mu_{t-1}^{DD}}{\sigma^{DD}}\right)\\ &\left(\frac{-\varphi_j(t)}{\sigma^{DD}} + \frac{\theta^{DD}\varphi_j(t-1)}{\sigma^{DD}}\right)\\ &+\sum_{t\in N(DW)} \left(\frac{S_{i,t}-\mu_t^{W}}{\sigma^{DW}} - \theta^{DW}\frac{S_{i,t-1}-\mu_{t-1}^{D}}{\sigma^{DW}}\right)\left(\frac{\theta^{DW}\varphi_j(t-1)}{\sigma^{DW}}\right)\\ &+\sum_{t\in N(WD)} \left(\frac{S_{i,t}-\mu_t^{D}}{\sigma^{WD}} - \theta^{WD}\frac{S_{i,t-1}-\mu_{t-1}^{W}}{\sigma^{WD}}\right)\left(\frac{-\varphi_j(t)}{\sigma^{WD}}\right)\Bigg]\end{aligned}$$

$$\begin{aligned}\frac{\partial\ell(\psi)}{\partial\alpha_j^{W}} = -\Bigg[&\sum_{t\in N(WW)} \left(\frac{S_{i,t}-\mu_t^{W}}{\sigma^{WW}} - \theta^{WW}\frac{S_{i,t-1}-\mu_{t-1}^{W}}{\sigma^{WW}}\right)\\ &\left(\frac{-\varphi_j(t)}{\sigma^{WW}} + \frac{\theta^{WW}\varphi_j(t-1)}{\sigma^{WW}}\right)\\ &+\sum_{t\in N(DW)} \left(\frac{S_{i,t}-\mu_t^{W}}{\sigma^{DW}} - \theta^{DW}\frac{S_{i,t-1}-\mu_{t-1}^{D}}{\sigma^{DW}}\right)\left(\frac{-\varphi_j(t)}{\sigma^{DW}}\right)\\ &+\sum_{t\in N(WD)} \left(\frac{S_{i,t}-\mu_t^{D}}{\sigma^{WD}} - \theta^{WD}\frac{S_{i,t-1}-\mu_{t-1}^{W}}{\sigma^{WD}}\right)\left(\frac{\theta^{WD}\varphi_j(t-1)}{\sigma^{WD}}\right)\Bigg]\end{aligned}$$

$$\begin{aligned}\frac{\partial\ell(\psi)}{\partial\sigma^{DD}} = -\frac{C(DD)}{\sigma^{DD}} - &\sum_{t\in N(DD)} \left(\frac{S_{i,t}-\mu_t^{DD}}{\sigma^{DD}} - \theta^{DD}\frac{S_{i,t-1}-\mu_{t-1}^{DD}}{\sigma^{DD}}\right)\\ &\left(-\frac{(S_{i,t}-\mu_t^{D})}{(\sigma^{DD})^2} + \theta^{DD}\frac{S_{i,t-1}-\mu_{t-1}^{D}}{(\sigma^{DD})^2}\right)\end{aligned}$$

$$\frac{\partial \ell(\psi)}{\partial \sigma^{WW}} = -\frac{C(WW)}{\sigma^{WW}} - \sum_{t \in N(WW)} \left(\frac{S_{i,t} - \mu_t^W}{\sigma^{WW}} - \theta^{WW} \frac{S_{i,t-1} - \mu_{t-1}^W}{\sigma^{WW}} \right)$$

$$\left(-\frac{(S_{i,t} - \mu_t^W)}{(\sigma^{WW})^2} + \theta^{WW} \frac{S_{i,t-1} - \mu_{t-1}^W}{(\sigma^{WW})^2} \right)$$

$$\frac{\partial \ell(\psi)}{\partial \sigma^{DW}} = -\frac{C(DW)}{\sigma^{DW}} - \sum_{t \in N(DW)} \left(\frac{S_{i,t} - \mu_t^W}{\sigma^{DW}} - \theta^{DW} \frac{S_{i,t-1} - \mu_{t-1}^D}{\sigma^{DW}} \right)$$

$$\left(-\frac{(S_{i,t} - \mu_t^W)}{(\sigma^{DW})^2} + \theta^{DW} \frac{S_{i,t-1} - \mu_{t-1}^D}{(\sigma^{DW})^2} \right)$$

$$\frac{\partial \ell(\psi)}{\partial \sigma^{WD}} = -\frac{C(WD)}{\sigma^{WD}} - \sum_{t \in N(WD)} \left(\frac{S_{i,t} - \mu_t^D}{\sigma^{WD}} - \theta^{WD} \frac{S_{i,t-1} - \mu_{t-1}^W}{\sigma^{WD}} \right)$$

$$\left(-\frac{(S_{i,t} - \mu_t^D)}{(\sigma^{WD})^2} + \theta^{WD} \frac{S_{i,t-1} - \mu_{t-1}^W}{(\sigma^{WD})^2} \right)$$

The parameter estimates are given by:

$$\widehat{\theta}^{DD} = \frac{\sum\limits_{t\in N(DD)} (S_{i,t} - \widehat{\mu}_t^D)(S_{i,t-1} - \widehat{\mu}_{t-1}^D)}{\sum\limits_{t\in N(DD)} (S_{i,t-1} - \widehat{\mu}_{t-1}^D)^2}$$

$$\widehat{\theta}^{WW} = \frac{\sum\limits_{t\in N(WW)} (S_{i,t} - \widehat{\mu}_t^W)(S_{i,t-1} - \widehat{\mu}_{t-1}^W)}{\sum\limits_{t\in N(WW)} (S_{i,t-1} - \widehat{\mu}_{t-1}^W)^2}$$

$$\widehat{\theta}^{DW} = \frac{\sum\limits_{t\in N(DW)} (S_{i,t} - \widehat{\mu}_t^W)(S_{i,t-1} - \widehat{\mu}_{t-1}^D)}{\sum\limits_{t\in N(DW)} (S_{i,t-1} - \widehat{\mu}_{t-1}^D)^2}$$

$$\widehat{\theta}^{WD} = \frac{\sum\limits_{t\in N(WD)} (S_{i,t} - \widehat{\mu}_t^D)(S_{i,t-1} - \widehat{\mu}_{t-1}^W)}{\sum\limits_{t\in N(WD)} (S_{i,t-1} - \widehat{\mu}_{t-1}^W)^2}$$

$$\widehat{\sigma}^{DD} = \left(\frac{1}{C(DD)} \sum_{t\in N(DD)} (S_{i,t} - \widehat{\mu}_t^D - \widehat{\theta}^{DD}(S_{i,t-1} - \widehat{\mu}_{t-1}^D))^2 \right)^{\frac{1}{2}}$$

$$\widehat{\sigma}^{WW} = \left(\frac{1}{C(WW)} \sum_{t\in N(WW)} (S_{i,t} - \widehat{\mu}_t^W - \widehat{\theta}^{WW}(S_{i,t-1} - \widehat{\mu}_{t-1}^W))^2 \right)^{\frac{1}{2}}$$

$$\widehat{\sigma}^{DW} = \left(\frac{1}{C(DW)} \sum_{t\in N(DW)} (S_{i,t} - \widehat{\mu}_t^W - \widehat{\theta}^{DW}(S_{i,t-1} - \widehat{\mu}_{t-1}^D))^2 \right)^{\frac{1}{2}}$$

$$\widehat{\sigma}^{WD} = \left(\frac{1}{C(WD)} \sum_{t\in N(WD)} (S_{i,t} - \widehat{\mu}_t^D - \widehat{\theta}^{WD}(S_{i,t-1} - \widehat{\mu}_{t-1}))^2 \right)^{\frac{1}{2}}$$

$$\begin{aligned}
\widehat{\alpha}_J^D = \Bigg\{ & \frac{1}{(\widehat{\sigma}^{DD})^2} \sum_{t\in N(DD)} \varphi_j(t)^2 - \frac{2}{(\widehat{\sigma}^{DD})^2} \widehat{\theta}^{DD} \sum_{t\in N(DD)} \varphi_j(t)\varphi_j(t-1) \\
& + \frac{1}{(\widehat{\sigma}^{DD})^2} (\widehat{\theta}^{DD})^2 \sum_{t\in N(DD)} \varphi_j(t-1)^2 \\
& + \frac{1}{(\widehat{\sigma}^{DW})^2} (\widehat{\theta}^{DW})^2 \sum_{t\in N(DW)} \varphi_j(t-1)^2 \\
& + \frac{1}{(\widehat{\sigma}^{WD})^2} \sum_{t\in N(WD)} \varphi_j(t)^2 \Bigg\}^{-1} [-A - M]
\end{aligned}$$

where

$$
\begin{aligned}
A = &-\frac{1}{(\widehat{\sigma}^{DD})^2} \sum_{t\in N(DD)} (S_{i,t} - \widehat{\theta}^{DD} S_{i,t-1})\varphi_j(t) \\
&+ \frac{1}{(\widehat{\sigma}^{DD})^2}\widehat{\theta}^{DD} \sum_{t\in N(DD)} (S_{i,t} - \widehat{\theta}^{DD} S_{i,t-1})\varphi_j(t-1) \\
&+ \frac{1}{(\widehat{\sigma}^{DW})^2}\widehat{\theta}^{DW} \sum_{t\in N(DW)} (S_{i,t} - \widehat{\theta}^{DW} S_{i,t-1})\varphi_j(t-1) \\
&- \frac{1}{(\widehat{\sigma}^{WD})^2} \sum_{t\in N(WD)} (S_{i,t} - \widehat{\theta}^{WD} S_{i,t-1})\varphi_j(t) \\
&- \frac{1}{(\widehat{\sigma}^{DW})^2}\widehat{\theta}^{DW} \sum_{t\in N(DW)} \widehat{\mu}_t^{W}\varphi_j(t-1) \\
&- \frac{1}{(\widehat{\sigma}^{WD})^2}\widehat{\theta}^{WD} \sum_{t\in N(WD)} \widehat{\mu}_{t-1}^{W}\varphi_j(t)
\end{aligned}
$$

and

$$
\begin{aligned}
M = &-\frac{1}{(\widehat{\sigma}^{DD})^2}\widehat{\theta}^{DD} \sum_{t\in N(DD)} \left(\sum_{\substack{k=1\\k\neq j}}^{L} \widehat{\alpha}_k^{D}\varphi_k(t-1)\right)\varphi_j(t) \\
&- \frac{1}{(\widehat{\sigma}^{DD})^2}\widehat{\theta}^{DD} \sum_{t\in N(DD)} \left(\sum_{\substack{k=1\\k\neq j}}^{L} \widehat{\alpha}_k^{D}\varphi_k(t)\right)\varphi_j(t-1) \\
&+ \frac{1}{(\widehat{\sigma}^{DD})^2} \sum_{t\in N(DD)} \left(\sum_{\substack{k=1\\k\neq j}}^{L} \widehat{\alpha}_k^{D}\varphi_k(t)\right)\varphi_j(t) \\
&+ \frac{1}{(\widehat{\sigma}^{DD})^2}(\widehat{\theta}^{DD})^2 \sum_{t\in N(DD)} \left(\sum_{\substack{k=1\\k\neq j}}^{L} \widehat{\alpha}_k^{D}\varphi_k(t-1)\right)\varphi_j(t-1) \\
&+ \frac{1}{(\widehat{\sigma}^{DW})^2}(\widehat{\theta}^{DW})^2 \sum_{t\in N(DW)} \left(\sum_{\substack{k=1\\k\neq j}}^{L} \widehat{\alpha}_k^{D}\varphi_k(t-1)\right)\varphi_j(t-1) \\
&+ \frac{1}{(\widehat{\sigma}^{WD})^2} \sum_{t\in N(WD)} \left(\sum_{\substack{k=1\\k\neq j}}^{L} \widehat{\alpha}_k^{D}\varphi_k(t)\right)\varphi_j(t).
\end{aligned}
$$

The estimate for α_j^W is given by

$$\begin{aligned}\widehat{\alpha}_j^W = \Bigg\{ & \frac{1}{(\widehat{\sigma}^{WW})^2} \sum_{t\in N(WW)} (\varphi_j(t))^2 - \frac{2}{(\widehat{\sigma}^{WW})^2}\widehat{\theta}^{WW} \sum_{t\in N(WW)} \varphi_j(t-1)\varphi_j(t) \\ & + \frac{1}{(\widehat{\sigma}^{WW})^2}(\widehat{\theta}^{WW})^2 \sum_{t\in N(WW)} (\varphi_j(t-1))^2 \\ & + \frac{1}{(\widehat{\sigma}^{DW})^2} \sum_{t\in N(DW)} (\varphi_j(t))^2 \\ & + \frac{1}{(\widehat{\sigma}^{WD})^2}(\widehat{\theta}^{WD})^2 \sum_{t\in N(WD)} (\varphi_j(t-1))^2 \Bigg\}^{-1} [-A_2 - M_2]\end{aligned}$$

where

$$\begin{aligned}A_2 = & -\frac{1}{(\widehat{\sigma}^{WW})^2} \sum_{t\in N(WW)} (S_{i,t} - \widehat{\theta}^{WW} S_{i,t-1})\varphi_j(t) \\ & + \frac{1}{(\widehat{\sigma}^{WW})^2}\widehat{\theta}^{WW} \sum_{t\in N(WW)} (S_{i,t} - \widehat{\theta}^{WW} S_{i,t-1})\varphi_j(t-1) \\ & - \frac{1}{(\widehat{\sigma}^{DW})^2} \sum_{t\in N(DW)} (S_{i,t} - \widehat{\theta}^{DW} S_{i,t-1})\varphi_j(t) \\ & - \frac{1}{(\widehat{\sigma}^{DW})^2}\widehat{\theta}^{DW} \sum_{t\in N(DW)} \widehat{\mu}_{t-1}^D \varphi_j(t) \\ & + \frac{1}{(\widehat{\sigma}^{WD})^2}\widehat{\theta}^{WD} \sum_{t\in N(WD)} (S_{i,t} - \widehat{\theta}^{WD} S_{i,t-1})\varphi_j(t-1) \\ & - \frac{1}{(\widehat{\sigma}^{WD})^2}\widehat{\theta}^{WD} \sum_{t\in N(WD)} (\widehat{\mu}_t^D)\varphi_j(t-1)\end{aligned}$$

and

$$
\begin{aligned}
M_2 = {} & \frac{1}{(\widehat{\sigma}^{WW})^2} \sum_{t \in N(WW)} \left(\sum_{\substack{k=1 \\ k \neq j}}^{L} \widehat{\alpha}_k^W \varphi_k(t) \right) \varphi_j(t) \\
& - \frac{1}{(\widehat{\sigma}^{WW})^2} \widehat{\theta}^{WW} \left[\sum_{t \in N(WW)} \left(\sum_{\substack{k=1 \\ k \neq j}}^{L} \widehat{\alpha}_k^W \varphi_k(t-1) \right) \varphi_j(t) \right. \\
& \left. + \sum_{t \in N(WW)} \left(\sum_{\substack{k=1 \\ k \neq j}}^{L} \widehat{\alpha}_k^W \varphi_k(t) \right) \varphi_j(t-1) \right] \\
& + \frac{1}{(\widehat{\sigma}^{WW})^2} (\widehat{\theta}^{WW})^2 \sum_{t \in N(WW)} \left(\sum_{\substack{k=1 \\ k \neq j}}^{L} \widehat{\alpha}_k^W \varphi_k(t-1) \right) \varphi_j(t-1) \\
& + \frac{1}{(\widehat{\sigma}^{DW})^2} \sum_{t \in N(DW)} \left(\sum_{\substack{k=1 \\ k \neq j}}^{L} \widehat{\alpha}_k^W \varphi_k(t) \right) \varphi_j(t) \\
& + \frac{1}{(\widehat{\sigma}^{WD})^2} (\widehat{\theta}^{WD})^2 \sum_{t \in N(WD)} \left(\sum_{\substack{k=1 \\ k \neq j}}^{L} \widehat{\alpha}_k^W \varphi_k(t-1) \right) \varphi_j(t-1).
\end{aligned}
$$

These equations cannot be solved explicitly and therefore the Newton–Raphson iteration method is used to solve them. The second partial derivatives are required to use this method and these are given by

$$
\begin{aligned}
\frac{\partial^2 \ell(\psi)}{\partial \theta^{DD} \partial \theta^{DD}} &= - \sum_{t \in N(DD)} \left(\frac{S_{i,t-1} - \mu_{t-1}^D}{\sigma^{DD}} \right)^2 \\
\frac{\partial^2 \ell(\psi)}{\partial \theta^{DD} \partial \theta^{WW}} &= \frac{\partial^2 \ell(\psi)}{\partial \theta^{DD} \partial \theta^{DW}} = \frac{\partial^2 \ell(\psi)}{\partial \theta^{DD} \partial \theta^{WD}} = 0 \\
\frac{\partial^2 \ell(\psi)}{\partial \theta^{DD} \partial \alpha_j^D} &= \sum_{t \in N(DD)} \left\{ \left(\frac{S_{i,t} - \mu_t^D}{\sigma^{DD}} - \theta^{DD} \frac{S_{i,t-1} - \mu_{t-1}^D}{\sigma^{DD}} \right) \left(\frac{-\varphi_j(t-1)}{\sigma^{DD}} \right) \right. \\
& \quad \left. + \left(\frac{-\varphi_j(t)}{\sigma^{DD}} + \frac{\theta^{DD} \varphi_j(t-1)}{\sigma^{DD}} \right) \left(\frac{S_{i,t-1} - \mu_{t-1}^D}{\sigma^{DD}} \right) \right\} \\
\frac{\partial^2 \ell(\psi)}{\partial \theta^{DD} \partial \alpha_j^W} &= 0
\end{aligned}
$$

$$\frac{\partial^2 \ell(\psi)}{\partial\theta^{DD}\partial\sigma^{DD}} = \sum_{t\in N(DD)} \left\{ \left(\frac{S_{i,t}-\mu_t^D}{\sigma^{DD}} - \theta^{DD}\frac{S_{i,t-1}-\mu_{t-1}^D}{\sigma^{DD}} \right) \left(-\frac{(S_{i,t-1}-\mu_{t-1}^D)}{(\sigma^{DD})^2} \right) + \left(-\frac{(S_{i,t}-\mu_t^D)}{(\sigma^{DD})^2} + \theta^{DD}\frac{S_{i,t-1}-\mu_{t-1}^D}{(\sigma^{DD})^2} \right) \left(\frac{S_{i,t-1}-\mu_{t-1}^D}{\sigma^{DD}} \right) \right\}$$

$$\frac{\partial^2 \ell(\psi)}{\partial\theta^{DD}\partial\sigma^{WW}} = \frac{\partial^2 \ell(\psi)}{\partial\theta^{DD}\partial\sigma^{DW}} = \frac{\partial^2 \ell(\psi)}{\partial\theta^{DD}\partial\sigma^{WD}} = 0$$

$$\frac{\partial^2 \ell(\psi)}{\partial\theta^{WW}\partial\theta^{WW}} = -\sum_{t\in N(WW)} \left(\frac{S_{i,t-1}-\mu_{t-1}^W}{\sigma^{WW}} \right)^2$$

$$\frac{\partial^2 \ell(\psi)}{\partial\theta^{WW}\partial\theta^{DW}} = \frac{\partial^2 \ell(\psi)}{\partial\theta^{WW}\partial\theta^{WD}} = \frac{\partial^2 \ell(\psi)}{\partial\theta^{WW}\partial\alpha_j^D} = 0$$

$$\frac{\partial^2 \ell(\psi)}{\partial\theta^{WW}\partial\alpha_j^W} = \sum_{t\in N(WW)} \left\{ \left(\frac{S_{i,t}-\mu_t^W}{\sigma^{WW}} - \theta^{WW}\frac{S_{i,t-1}-\mu_{t-1}^W}{\sigma^{WW}} \right) \left(\frac{-\varphi_j(t-1)}{\sigma^{WW}} \right) + \left(\frac{-\varphi_j(t)}{\sigma^{WW}} + \frac{\theta^{WW}\varphi_j(t-1)}{\sigma^{WW}} \right) \left(\frac{S_{i,t-1}-\mu_{t-1}^W}{\sigma^{WW}} \right) \right\}$$

$$\frac{\partial^2 \ell(\psi)}{\partial\theta^{WW}\partial\sigma^{DD}} = \frac{\partial^2 \ell(\psi)}{\partial\theta^{WW}\partial\sigma^{DW}} = \frac{\partial^2 \ell(\psi)}{\partial\theta^{WW}\partial\sigma^{WD}} = 0$$

$$\frac{\partial^2 \ell(\psi)}{\partial\theta^{WW}\partial\sigma^{WW}} = \sum_{t\in N(WW)} \left\{ \left(\frac{S_{i,t}-\mu_t^W}{\sigma^{WW}} - \theta^{WW}\frac{S_{i,t-1}-\mu_{t-1}^W}{\sigma^{WW}} \right) \left(-\frac{(S_{i,t-1}-\mu_{t-1}^W)}{(\sigma^{WW})^2} \right) + \left(-\frac{(S_{i,t}-\mu_t^W)}{(\sigma^{WW})^2} + \theta^{WW}\frac{S_{i,t-1}-\mu_{t-1}^W}{(\sigma^{WW})^2} \right) \left(\frac{S_{i,t-1}-\mu_{t-1}^W}{\sigma^{WW}} \right) \right\}$$

$$\frac{\partial^2 \ell(\psi)}{\partial\theta^{DW}\partial\theta^{DW}} = -\sum_{t\in N(DW)} \left(\frac{S_{i,t-1}-\mu_{t-1}^{D}}{\sigma^{DW}}\right)^2$$

$$\frac{\partial^2 \ell(\psi)}{\partial\theta^{DW}\partial\theta^{WD}} = 0$$

$$\frac{\partial^2 \ell(\psi)}{\partial\theta^{DW}\partial\alpha_j^{D}} = \sum_{t\in N(DW)} \left\{\left(\frac{S_{i,t}-\mu_t^{W}}{\sigma^{DW}} - \theta^{DW}\frac{S_{i,t-1}-\mu_{t-1}^{D}}{\sigma^{DW}}\right)\left(\frac{-\varphi_j(t-1)}{\sigma^{DW}}\right) + \left(\frac{\theta^{DW}\varphi_j(t-1)}{\sigma^{DW}}\right)\left(\frac{S_{i,t-1}-\mu_{t-1}^{D}}{\sigma^{DW}}\right)\right\}$$

$$\frac{\partial^2 \ell(\psi)}{\partial\theta^{DW}\partial\alpha_j^{W}} = \sum_{t\in N(DW)} \left(\frac{-\varphi_j(t)}{\sigma^{DW}}\right)\left(\frac{S_{i,t-1}-\mu_{t-1}^{D}}{\sigma^{DW}}\right)$$

$$\frac{\partial^2 \ell(\psi)}{\partial\theta^{DW}\partial\sigma^{DD}} = \frac{\partial^2 \ell(\psi)}{\partial\theta^{DW}\partial\sigma^{WW}} = \frac{\partial^2 \ell(\psi)}{\partial\theta^{DW}\partial\sigma^{WD}} = 0$$

$$\frac{\partial^2 \ell(\psi)}{\partial\theta^{DW}\partial\sigma^{DW}} = \sum_{t\in N(DW)} \left\{\left(\frac{S_{i,t}-\mu_t^{W}}{\sigma^{DW}} - \theta^{DW}\frac{S_{i,t-1}-\mu_{t-1}^{D}}{\sigma^{DW}}\right)\left(-\frac{(S_{i,t-1}-\mu_{t-1}^{D})}{(\sigma^{DW})^2}\right) + \left(-\frac{(S_{i,t}-\mu_{i,t}^{W})}{(\sigma^{DW})^2} + \theta^{DW}\frac{(S_{i,t-1}-\mu_{t-1}^{D})}{(\sigma^{DW})^2}\right)\left(\frac{S_{i,t-1}-\mu_{t-1}^{D}}{\sigma^{DW}}\right)\right\}$$

$$\frac{\partial^2 \ell(\psi)}{\partial\theta^{WD}\partial\theta^{WD}} = -\sum_{t\in N(WD)} \left(\frac{S_{i,t-1}-\mu_{t-1}^{W}}{\sigma^{WD}}\right)^2$$

$$\frac{\partial^2 \ell(\psi)}{\partial\theta^{WD}\partial\alpha_j^{D}} = \sum_{t\in N(WD)} \left(\frac{-\varphi_j(t)}{\sigma^{WD}}\right)\left(\frac{S_{i,t-1}-\mu_{t-1}^{W}}{\sigma^{WD}}\right)$$

$$\frac{\partial^2 \ell(\psi)}{\partial\theta^{WD}\partial\alpha_j^{W}} = \sum_{t\in N(WD)} \left\{\left(\frac{\varphi_j(t-1)\theta^{WD}}{\sigma^{WD}}\right)\left(\frac{S_{i,t-1}-\mu_{t-1}^{W}}{\sigma^{WD}}\right) + \left(\frac{S_{i,t}-\mu_t^{D}}{\sigma^{WD}} - \theta^{WD}\frac{S_{i,t-1}-\mu_{t-1}^{W}}{\sigma^{WD}}\right)\left(\frac{-\varphi_j(t-1)}{\sigma^{WD}}\right)\right\}$$

$$\frac{\partial^2 \ell(\psi)}{\partial\theta^{WD}\partial\sigma^{DD}} = \frac{\partial^2 \ell(\psi)}{\partial\theta^{WD}\partial\sigma^{WW}} = \frac{\partial^2 \ell(\psi)}{\partial\theta^{WD}\partial\sigma^{DW}} = 0$$

$$\frac{\partial^2 \ell(\psi)}{\partial\theta^{WD}\partial\sigma^{WD}} = \sum_{t\in N(WD)} \left\{\left(\frac{S_{i,t}-\mu_t^{D}}{\sigma^{WD}} - \theta^{WD}\frac{S_{i,t-1}-\mu_{t-1}^{W}}{\sigma^{WD}}\right)\left(-\frac{(S_{i,t-1}-\mu_{t-1}^{W})}{(\sigma^{WD})^2}\right) + \left(-\frac{(S_{i,t}-\mu_t^{D})}{(\sigma^{WD})^2} + \theta^{WD}\frac{S_{i,t-1}-\mu_{t-1}^{W}}{(\sigma^{WD})^2}\right)\left(\frac{S_{i,t-1}-\mu_{t-1}^{W}}{\sigma^{WD}}\right)\right\}$$

$$\frac{\partial^2 \ell(\psi)}{\partial \alpha_j^D \partial \alpha_k^D} = -\left[\sum_{t\in N(DD)} \left(\frac{-\varphi_j(t)}{\sigma^{DD}} + \frac{\theta^{DD}\varphi_j(t-1)}{\sigma^{DD}}\right)\left(\frac{-\varphi_k(t)}{\sigma^{DD}} + \frac{\theta^{DD}\varphi_k(t-1)}{\sigma^{DD}}\right) + \sum_{t\in N(DW)} \left(\frac{\theta^{DW}\varphi_j(t-1)}{\sigma^{DW}}\right)\left(\frac{\theta^{DW}\varphi_k(t-1)}{\sigma^{DW}}\right) + \sum_{t\in N(WD)} \left(\frac{-\varphi_j(t)}{\sigma^{WD}}\right)\left(\frac{-\varphi_k(t)}{\sigma^{WD}}\right)\right]$$

$$\frac{\partial^2 \ell(\psi)}{\partial \alpha_j^D \partial \alpha_k^W} = -\left[\sum_{t\in N(DW)} \left(\frac{\theta^{DW}\varphi_j(t-1)}{\sigma^{DW}}\right)\left(\frac{-\varphi_k(t)}{\sigma^{DW}}\right) + \sum_{t\in N(WD)} \left(\frac{-\varphi_j(t)}{\sigma^{WD}}\right)\left(\frac{\theta^{WD}\varphi_k(t-1)}{\sigma^{WD}}\right)\right]$$

$$\frac{\partial^2 \ell(\psi)}{\partial \alpha_j^D \partial \sigma^{DD}} = -\left[\sum_{t\in N(DD)} \left\{\left(\frac{S_{i,t}-\mu_t^D}{\sigma^{DD}} - \theta^{DD}\frac{S_{i,t-1}-\mu_{t-1}^D}{\sigma^{DD}}\right)\left(\frac{\varphi_j(t)}{(\sigma^{DD})^2} - \frac{\theta^{DD}\varphi_j(t-1)}{(\sigma^{DD})^2}\right) + \left(-\frac{(S_{i,t}-\mu_t^D)}{(\sigma^{DD})^2} + \theta^{DD}\frac{S_{i,t-1}-\mu_{t-1}^D}{(\sigma^{DD})^2}\right)\left(\frac{-\varphi_j(t)}{\sigma^{DD}} + \frac{\theta^{DD}\varphi_j(t-1)}{\sigma^{DD}}\right)\right\}\right]$$

$$\frac{\partial^2 \ell(\psi)}{\partial \alpha_j^D \partial \sigma^{WW}} = 0$$

$$\frac{\partial^2 \ell(\psi)}{\partial \alpha_j^D \partial \sigma^{DW}} = -\left[\sum_{t\in N(DW)} \left\{\left(\frac{S_{i,t}-\mu_t^W}{\sigma^{DW}} - \theta^{DW}\frac{S_{i,t-1}-\mu_{t-1}^D}{\sigma^{DW}}\right)\left(\frac{-\theta^{DW}\varphi_j(t-1)}{(\sigma^{DW})^2}\right) + \left(\frac{\theta^{DW}\varphi_j(t-1)}{\sigma^{DW}}\right)\left(-\frac{(S_{i,t}-\mu_t^W)}{(\sigma^{DW})^2} + \theta^{DW}\frac{S_{i,t-1}-\mu_{t-1}^D}{(\sigma^{DW})^2}\right)\right\}\right]$$

$$\frac{\partial^2 \ell \psi)}{\partial \alpha_j^D \partial \sigma^{WD}} = -\left[\sum_{t\in N(WD)} \left\{\left(\frac{S_{i,t}-\mu_t^D}{\sigma^{WD}} - \theta^{WD}\frac{S_{i,t-1}-\mu_{t-1}^W}{\sigma^{WD}}\right)\left(\frac{\varphi_j(t)}{(\sigma^{WD})^2}\right) + \left(\frac{-\varphi_j(t)}{\sigma^{WD}}\right)\left(-\frac{(S_{i,t}-\mu_t^D)}{(\sigma^{WD})^2} + \theta^{WD}\frac{S_{i,t-1}-\mu_{t-1}^W}{(\sigma^{WD})^2}\right)\right\}\right]$$

$$\frac{\partial^2 \ell(\psi)}{\partial \alpha_j^W \partial \alpha_k^W} = -\Bigg[\sum_{t\in N(WW)} \left(\frac{-\varphi_k(t)}{\sigma^{WW}} + \frac{\varphi_k(t-1)\theta^{WW}}{\sigma^{WW}}\right) \left(\frac{-\varphi_j(t)}{\sigma^{WW}} + \frac{\theta^{WW}\varphi_j(t-1)}{\sigma^{WW}}\right) + \sum_{t\in N(DW)} \left(\frac{-\varphi_k(t)}{\sigma^{WW}}\right)\left(\frac{-\varphi_j(t)}{\sigma^{WW}}\right) + \sum_{t\in N(WD)} \left(\frac{\theta^{WD}\varphi_k(t-1)}{\sigma^{WD}}\right)\left(\frac{\theta^{WD}\varphi_j(t-1)}{\sigma^{WD}}\right)\Bigg]$$

$$\frac{\partial^2 \ell(\psi)}{\partial \alpha_j^W \partial \sigma^{DD}} = 0$$

$$\frac{\partial^2 \ell(\psi)}{\partial \alpha_j^W \partial \sigma^{WW}} = -\Bigg[\sum_{t\in N(WW)} \Bigg\{\left(\frac{S_{i,t}-\mu_t^W}{\sigma^{WW}} - \theta^{WW}\frac{S_{i,t-1}-\mu_{t-1}^W}{\sigma^{WW}}\right) \left(\frac{\varphi_j(t)}{(\sigma^{WW})^2} - \frac{\theta^{WW}\varphi_j(t-1)}{(\sigma^{WW})^2}\right) + \left(\frac{-(S_{i,t}-\mu_t^W)}{(\sigma^{WW})^2} + \theta^{WW}\frac{S_{i,t-1}-\mu_{t-1}^W}{(\sigma^{WW})^2}\right) \left(\frac{-\varphi_j(t)}{\sigma^{WW}} + \frac{\theta^{WW}\varphi_j(t-1)}{\sigma^{WW}}\right)\Bigg\}\Bigg]$$

$$\frac{\partial^2 \ell(\psi)}{\partial \alpha_j^W \partial \sigma^{DW}} = -\Bigg[\sum_{t\in N(DW)} \left(\frac{S_{i,t}-\mu_t^W}{\sigma^{DW}} - \theta^{DW}\frac{S_{i,t-1}-\mu_{t-1}^D}{\sigma^{DW}}\right)\left(\frac{\varphi_j(t)}{(\sigma^{DW})^2}\right) + \left(\frac{-(S_{i,t}-\mu_t^W)}{(\sigma^{DW})^2} + \theta^{DW}\frac{S_{i,t-1}-\mu_{t-1}^D}{(\sigma^{DW})^2}\right)\left(\frac{-\varphi_j(t)}{\sigma^{DW}}\right)\Bigg]$$

$$\frac{\partial^2 \ell(\psi)}{\partial \alpha_j^W \partial \sigma^{WD}} = -\Bigg[\sum_{t\in N(WD)} \left(\frac{S_{i,t}-\mu_t^D}{\sigma^{WD}} - \theta^{WD}\frac{S_{i,t-1}-\mu_{t-1}^W}{\sigma^{WD}}\right)\left(\frac{-\theta^{WD}\varphi_j(t-1)}{(\sigma^{WD})^2}\right) + \left(-\frac{(S_{i,t}-\mu_t^D)}{(\sigma^{WD})^2} + \theta^{WD}\frac{S_{i,t-1}-\mu_{t-1}^W}{(\sigma^{WD})^2}\right)\left(\frac{\theta^{WD}\varphi_j(t-1)}{\sigma^{WD}}\right)\Bigg]$$

$$\frac{\partial^2 \ell(\psi)}{\partial \sigma^{DD} \partial \sigma^{DD}} = \frac{C(DD)}{(\sigma^{DD})^2} - \Bigg[\sum_{t\in N(DD)} \Bigg\{\left(\frac{S_{i,t}-\mu_t^D}{\sigma^{DD}} - \theta^{DD}\frac{S_{i,t-1}-\mu_{t-1}^D}{\sigma^{DD}}\right) \left(2\frac{S_{i,t}-\mu_t^D}{(\sigma^{DD})^3} - 2\theta^{DD}\frac{S_{i,t-1}-\mu_{t-1}^D}{(\sigma^{DD})^3}\right) + \left(-\frac{(S_{i,t}-\mu_t^D)}{(\sigma^{DD})^2} + \theta^{DD}\frac{S_{i,t-1}-\mu_{t-1}^D}{(\sigma^{DD})^2}\right)^2\Bigg\}\Bigg]$$

$$\frac{\partial^2 \ell(\psi)}{\partial \sigma^{DD} \partial \sigma^{WW}} = \frac{\partial^2 \ell(\psi)}{\partial \sigma^{DD} \partial \sigma^{DW}} = \frac{\partial^2 \ell(\psi)}{\partial \sigma^{DD} \partial \sigma^{WD}} = 0$$

$$\frac{\partial^2 \ell(\psi)}{\partial \sigma^{WW} \partial \sigma^{WW}} = \frac{C(WW)}{(\sigma^{WW})^2} - \left[\sum_{t \in N(WW)} \left\{ \left(\frac{S_{i,t} - \mu_t^W}{\sigma^{WW}} - \theta^{WW} \frac{S_{i,t-1} - \mu_{t-1}^W}{\sigma^{WW}} \right) \right. \right.$$
$$\left(2 \frac{S_{i,t} - \mu_t^W}{(\sigma^{WW})^3} - 2\theta^{WW} \frac{S_{i,t-1} - \mu_{t-1}^W}{(\sigma^{WW})^3} \right)$$
$$\left. \left. + \left(-\frac{(S_{i,t} - \mu_t^W)}{(\sigma^{WW})^2} + \theta^{WW} \frac{S_{i,t-1} - \mu_{t-1}^W}{(\sigma^{WW})^2} \right)^2 \right\} \right]$$

$$\frac{\partial^2 \ell(\psi)}{\partial \sigma^{WW} \partial \sigma^{DW}} = \frac{\partial^2 \ell(\psi)}{\partial \sigma^{WW} \partial \sigma^{WD}} = 0$$

$$\frac{\partial^2 \ell(\psi)}{\partial \sigma^{DW} \partial \sigma^{DW}} = \frac{C(DW)}{(\sigma^{DW})^2} - \left[\sum_{t \in N(DW)} \left\{ \left(\frac{S_{i,t} - \mu_t^W}{\sigma^{DW}} - \theta^{DW} \frac{S_{i,t-1} - \mu_{t-1}^D}{\sigma^{DW}} \right) \right. \right.$$
$$\left(2 \frac{S_{i,t} - \mu_t^W}{(\sigma^{DW})^3} - 2\theta^{DW} \frac{S_{i,t-1} - \mu_{t-1}^D}{(\sigma^{DW})^3} \right)$$
$$\left. \left. + \left(-\frac{(S_{i,t} - \mu_t^W)}{(\sigma^{DW})^2} + \theta^{DW} \frac{S_{i,t-1} - \mu_{t-1}^D}{(\sigma^{DW})^2} \right)^2 \right\} \right]$$

$$\frac{\partial^2 \ell(\psi)}{\partial \sigma^{DW} \partial \sigma^{WD}} = 0$$

$$\frac{\partial^2 \ell(\psi)}{\partial \sigma^{WD} \partial \sigma^{WD}} = \frac{C(WD)}{(\sigma^{WD})^2} - \left[\sum_{t \in N(WD)} \left\{ \left(\frac{S_{i,t} - \mu_t^D}{\sigma^{WD}} - \theta^{WD} \frac{S_{i,t-1} - \mu_{t-1}^W}{\sigma^{WD}} \right) \right. \right.$$
$$\left(2 \frac{S_{i,t} - \mu_t^D}{(\sigma^{WD})^3} - 2\theta^{WD} \frac{S_{i,t-1} - \mu_{t-1}^W}{(\sigma^{WD})^3} \right)$$
$$\left. \left. + \left(-\frac{(S_{i,t} - \mu_t^D)}{(\sigma^{WD})^2} + \theta^{WD} \frac{S_{i,t-1} - \mu_{t-1}^W)}{(\sigma^{WD})^2} \right)^2 \right\} \right]$$

The following algorithm is used to estimate the parameters.

Algorithm

Step 1: Estimate initial $\widehat{\mu}_t$ by approximating by its Fourier series representation and estimating the parameters α_i by the method mentioned in the previous models.

Step 2: Estimate initial $\widehat{\theta}^{DD}, \widehat{\theta}^{WW}, \widehat{\theta}^{DW}$ and $\widehat{\theta}^{WD}$ using the following formula:

$$\widehat{\theta}^{DD} = \frac{\sum_{t \in N(DD)} (S_{i,t} - \widehat{\mu}_t^D)(S_{i,t-1} - \widehat{\mu}_{t-1}^D)}{\sum_{t \in N(DD)} (S_{i,t-1} - \widehat{\mu}_{t-1}^D)^2}$$

Similarly for $\widehat{\theta}^{WW}, \widehat{\theta}^{DW}$ and $\widehat{\theta}^{WD}$.

Step 3: Estimate initial $\hat{\sigma}^{DD}, \hat{\sigma}^{WW}, \hat{\sigma}^{DW}$ and $\hat{\sigma}^{WD}$ using the following formula:

$$\hat{\sigma}^{DD} = \left[\frac{1}{C(DD)-1} \sum_{t \in N(DD)} (S_{i,t} - \hat{\mu}_t^D - \hat{\theta}^{DD}(S_{i,t-1} - \hat{\mu}_{t-1}^D))^2 \right]^{\frac{1}{2}}$$

Similarly for $\hat{\sigma}^{WW}, \hat{\sigma}^{DW}$ and $\hat{\sigma}^{WD}$.

Step 4: Compute $f^{(k)}$ and $F^{(k)}$, where $f^{(k)}$ is the vector of first partial derivatives and $F^{(k)}$ is the matrix of second partial derivatives, computed at the kth iteration.

Step 5: Compute the vector $\delta^{(k)}$ which is the solution to the system of NP linear equations

$$F^{(k)}\delta^{(k)} = f^{(k)},$$

where NP represents the number of parameters.

Step 6: Set $\beta^{(k+1)} = \beta^{(k)} - \delta^{(k)}$, where $\beta^{(k)}$ contains the parameter estimates at the kth iteration.

Step 7: Test for convergence, for example, if the elements of $f^{(k)}$ are sufficiently close to zero. If the convergence criterion is met then stop, otherwise increase k by 1 and return to step 4.

The cross-correlation matrix, $\hat{\Sigma}$, has elements given by:

$$R_{jk} = \frac{\frac{1}{T}\sum_{i=1}^{NY}\sum_{t=1}^{NT} e_{i,t}^{(j)} e_{i,t}^{(k)} - \frac{1}{T^2}\sum_{i=1}^{NY}\sum_{t=1}^{NT} e_{i,t}^{(j)} \sum_{i=1}^{NY}\sum_{t=1}^{NT} e_{i,t}^{(k)}}{\left[\frac{1}{T}\sum_{i=1}^{NY}\sum_{t=1}^{NT}(e_{i,t}^{(j)})^2 - \frac{1}{T^2}\left(\sum_{i=1}^{NY}\sum_{t=1}^{NT} e_{i,t}^{(j)}\right)^2\right]^{\frac{1}{2}} \left[\frac{1}{T}\sum_{i=1}^{NY}\sum_{t=1}^{NT}(e_{i,t}^{(k)})^2 - \frac{1}{T^2}\left(\sum_{i=1}^{NY}\sum_{t=1}^{NT} e_{i,t}^{(k)}\right)^2\right]^{\frac{1}{2}}}$$

where $e_{i,t}^{(j)}$ denotes the residual time series of variable j, $j = 1, 2, \ldots, NV$
and $e_{i,t}^{(k)}$ denotes the residual time series of variable k, $k = 1, 2, \ldots, NV$.

(d) Model Selection

The order of the autoregressive process is selected in the same way as in the previous models, as is the order of the Fourier series approximation.

Model 4: Multivariate model for climate data

(a) Notation

Partition the year into $NT(=365)$ equal intervals, denoted by $t = 1, 2, \ldots NT$.

NV is the number of variables.

NY is the number of years observed.

W represents the occurrence of rain.

D represents the non–occurrence of rain.

DD represents the sequence when day $t-1$ was dry and day t was dry.

WW represents the sequence when day $t-1$ was wet and day t was wet.

DW represents the sequence when day $t-1$ was dry and day t was wet.

WD represents the sequence when day $t-1$ was wet and day t was dry.

T represents the total number of observations, i.e. $NT\ NY$.

$N(DD)$ is the set of time periods t such that period t was dry and period $t-1$ was dry, $t = 1, 2, \ldots, T$.

$N(WW)$ is the set of time periods t such that period t was wet and period $t-1$ was wet.

$N(DW)$ is the set of time periods t such that period t was wet and period $t-1$ was dry.

$N(WD)$ is the set of time periods t such that period t was dry and period $t-1$ was wet.

$Y_{i,t}$ is the precipitation amount on period t of year i, $i = 1, 2, \ldots, NY$.

$S_{i,t}$ is the generic name for the observation at time t of the ith year.

μ_t^D is the generic name for the mean for a dry day on perid t.

μ_t^W is the generic name for the mean for a wet day on period t.

σ_t^D is the generic name for the standard deviation for a dry day on period t.

σ_t^W is the generic name for the standard deviation for a wet day on period t.

θ is the coefficient of the AR(1) process.

(b) Model and assumptions

Following the procedure suggested by Richardson (1981), the time series $S_{i,t}$ is reduced to a time series of residual elements, $\chi_{i,t}$, by removing the periodic means and standard deviations, i.e.

$$\chi_{i,t} = \frac{S_{i,t} - \mu_t}{\sigma_t}.$$

This standardization leads to a time series for each variable that is stationary in the mean and standard deviation with mean zero and standard deviation of unity.

The model proposed is assumed to follow an AR(1) process, i.e.

$$\chi_{i,t} = \theta\, \chi_{i,t-1} + e_{i,t}$$

where $e_{i,t} \sim NID(0,1) \quad i = 1,2,\ldots,NY;\ t = 1,2,\ldots,NT$.

The model that incorporates the different rain sequences is then given by:

$$e_{i,t} = \frac{S_{i,t} - \mu_t^D}{\sigma_t^D} - \theta\, \frac{S_{i,t-1} - \mu_{t-1}^D}{\sigma_{t-1}^D} \quad \text{if day } t-1 \text{ was dry and day } t \text{ was dry.}$$

or

$$e_{i,t} = \frac{S_{i,t} - \mu_t^W}{\sigma_t^W} - \theta\, \frac{S_{i,t-1} - \mu_{t-1}^W}{\sigma_{t-1}^W} \quad \text{if day } t-1 \text{ was wet and day } t \text{ was wet.}$$

or

$$e_{i,t} = \frac{S_{i,t} - \mu_t^W}{\sigma_t^W} - \theta\, \frac{S_{i,t-1} - \mu_{t-1}^D}{\sigma_{t-1}^D} \quad \text{if day } t-1 \text{ was dry and day } t \text{ was wet.}$$

or

$$e_{i,t} = \frac{S_{i,t} - \mu_t^D}{\sigma_t^D} - \theta\, \frac{S_{i,t-1} - \mu_{t-1}^W}{\sigma_{t-1}^W} \quad \text{if day } t-1 \text{ was wet and day } t \text{ was dry.}$$

Here again the mean and standard deviation functions, μ_t and σ_t , are approximated by their respective truncated Fourier representation, i.e.

$$\left.\begin{aligned} \mu_t^D &= \sum_{i=1}^{L} \alpha_i^D \varphi_i(t) \\ \sigma_t^D &= \sum_{i=1}^{L} \xi_i^D \varphi_i(t) \end{aligned}\right\} \quad \text{if } t \text{ dry}$$

$$\left.\begin{aligned} \mu_t^W &= \sum_{i=1}^{L} \alpha_i^W \varphi_i(t) \\ \sigma_t^W &= \sum_{i=1}^{L} \xi_i^W \varphi_i(t) \end{aligned}\right\} \quad \text{if } t \text{ wet}$$

where $\varphi_i(t)$ is defined as before and L is the order of the Fourier series approximation.

(c) Estimation

Since $e_{i,t} \sim NID(0,1)$, the density function of $e_{i,t}$ is given by

$$f(e_{i,t}) = \frac{1}{\sqrt{2\pi}} \exp(-\frac{1}{2}e_{i,t}^2).$$

The joint likelihood function, conditioned on the four different sequences is given by:

$$\begin{aligned} L(\psi) &= L(\alpha_j^D, \alpha_j^W, \xi_j^D, \xi_j^W, \theta; e_{i,t}) \\ &= \prod_{t\in N(DD)} f(e_{i,t}|DD) \prod_{t\in N(WW)} f(e_{i,t}|WW) \\ &\quad \prod_{t\in N(DW)} f(e_{i,t}|DW) \prod_{t\in N(WD)} f(e_{i,t}|WD) \end{aligned}$$

where $f(e_{i,t}|DD)$ represents the density function of $e_{i,t}$ given that the sequence DD has been observed, and similarly for the others.

$$= \left(\frac{1}{\sqrt{2\pi}}\right)^T \exp\left\{-\frac{1}{2}\left[\sum_{t\in N(DD)} (e_{i,t}|DD)^2 + \sum_{t\in N(WW)} (e_{i,t}|WW)^2 + \sum_{t\in N(DW)} (e_{i,t}|DW)^2 + \sum_{t\in N(WD)} (e_{i,t}|WD)^2\right]\right\}.$$

One now makes the following transformation:

$$e_{i,t} = \frac{S_{i,t} - \mu_t}{\sigma_t} - \theta\ \frac{S_{i,t-1} - \mu_{t-1}}{\sigma_{t-1}}.$$

The Jacobian of the transformation is given by

$$\left|\frac{\partial e_{i,t}}{\partial S_{i,p}}\right| = \begin{vmatrix} 1/\sigma_1 & 0 & & & 0 \\ -\theta/\sigma_1 & 1/\sigma_2 & 0 & \dots & 0 \\ \vdots & & & & \vdots \\ 0 & \dots & \dots & -\theta/\sigma_{364} & 1/\sigma_{365} \\ \vdots & & & & \vdots \\ 1/\sigma_1 & 0 & \dots & \dots & 0 \\ -\theta/\sigma_1 & 1/\sigma_2 & 0 & \dots & 0 \\ \vdots & & & & \vdots \\ 0 & \dots & \dots & -\theta/\sigma_{364} & 1/\sigma_{365} \end{vmatrix}$$

$$= \prod_{i=1}^{NY} \prod_{t=1}^{NT} \frac{1}{\sigma_t}$$

Taking into account the conditional sequences imposed on $e_{i,t}$, the Jacobian is then given by

$$\left|\frac{\partial e_{i,t}}{\partial S_{i,p}}\right| = \prod_{\substack{t\in N(DD)\\ t\in N(WD)}} \frac{1}{\sigma_t^D} \prod_{\substack{t\in N(WW)\\ t\in N(DW)}} \frac{1}{\sigma_t^W}.$$

The joint probability density function is thus given by:

$$
\begin{aligned}
L(\psi) = \left(\frac{1}{\sqrt{2\pi}}\right)^T & \prod_{\substack{t\in N(DD)\\ t\in N(WD)}} \frac{1}{\sigma_t^D} \prod_{\substack{t\in N(WW)\\ t\in N(DW)}} \frac{1}{\sigma_t^W} \\
& \exp\left\{-\frac{1}{2}\left[\sum_{t\in N(DD)} \left(\frac{S_{i,t}-\mu_t^D}{\sigma_t^D} - \theta\,\frac{S_{i,t-1}-\mu_{t-1}^D}{\sigma_{t-1}^D}\right)^2\right.\right. \\
& + \sum_{t\in N(WW)} \left(\frac{S_{i,t}-\mu_t^W}{\sigma_t^W} - \theta\,\frac{S_{i,t-1}-\mu_{t-1}^W}{\sigma_{t-1}^W}\right)^2 \\
& + \sum_{t\in N(DW)} \left(\frac{S_{i,t}-\mu_t^W}{\sigma_t^W} - \theta\,\frac{S_{i,t-1}-\mu_{t-1}^D}{\sigma_{t-1}^D}\right)^2 \\
& \left.\left. + \sum_{t\in N(WD)} \left(\frac{S_{i,t}-\mu_t^D}{\sigma_t^D} - \theta\,\frac{S_{i,t-1}-\mu_{t-1}^W}{\sigma_{t-1}^W}\right)^2\right]\right\}
\end{aligned}
$$

and the log–likelihood is given by:

$$
\begin{aligned}
\ell(\psi) = -\frac{T}{2}\log(2\pi) & - \sum_{\substack{t\in N(DD)\\ t\in N(WD)}} \log(\sigma_t^D) - \sum_{\substack{t\in N(WW)\\ t\in N(DW)}} \log(\sigma_t^W) \\
& - \frac{1}{2}\left[\sum_{t\in N(DD)} \left(\frac{S_{i,t}-\mu_t^D}{\sigma_t^D} - \theta\,\frac{S_{i,t-1}-\mu_{t-1}^D}{\sigma_{t-1}^D}\right)^2\right. \\
& + \sum_{t\in N(WW)} \left(\frac{S_{i,t}-\mu_t^W}{\sigma_t^W} - \theta\,\frac{S_{i,t-1}-\mu_{t-1}^W}{\sigma_{t-1}^W}\right)^2 \\
& + \sum_{t\in N(DW)} \left(\frac{S_{i,t}-\mu_t^W}{\sigma_t^W} - \theta\,\frac{S_{i,t-1}-\mu_{t-1}^D}{\sigma_{t-1}^D}\right)^2 \\
& \left. + \sum_{t\in N(WD)} \left(\frac{S_{i,t}-\mu_t^D}{\sigma_t^D} - \theta\,\frac{S_{i,t-1}-\mu_{t-1}^W}{\sigma_{t-1}^W}\right)^2\right].
\end{aligned}
$$

Maximum likelihood estimates can be obtained by minimising $\ell(\psi)$ and this is achieved by setting its partial derivatives with respect to the parameters equal to zero.

The first partial derivatives with respect to the parameters are given by:

$$\begin{aligned}\frac{\partial \ell(\psi)}{\partial \theta} = & \sum_{t\in N(DD)} \left(\frac{S_{i,t}-\mu_t^D}{\sigma_t^D} - \theta\ \frac{S_{i,t-1}-\mu_{t-1}^D}{\sigma_{t-1}^D}\right)\left(\frac{S_{i,t-1}-\mu_{t-1}^D}{\sigma_{t-1}^D}\right) \\ & + \sum_{t\in N(WW)} \left(\frac{S_{i,t}-\mu_t^W}{\sigma_t^W} - \theta\ \frac{S_{i,t-1}-\mu_{t-1}^W}{\sigma_{t-1}^W}\right)\left(\frac{S_{i,t-1}-\mu_{t-1}^W}{\sigma_{t-1}^W}\right) \\ & + \sum_{t\in N(DW)} \left(\frac{S_{i,t}-\mu_t^W}{\sigma_t^W} - \theta\ \frac{S_{i,t-1}-\mu_{t-1}^D}{\sigma_{t-1}^D}\right)\left(\frac{S_{i,t-1}-\mu_{t-1}^D}{\sigma_{t-1}^D}\right) \\ & + \sum_{t\in N(WD)} \left(\frac{S_{i,t}-\mu_t^D}{\sigma_t^D} - \theta\ \frac{S_{i,t-1}-\mu_{t-1}^W}{\sigma_{t-1}^W}\right)\left(\frac{S_{i,t-1}-\mu_{t-1}^W}{\sigma_{t-1}^W}\right).\end{aligned}$$

$$\begin{aligned}\frac{\partial \ell(\psi)}{\partial \alpha_j^D} = & -\left[\sum_{t\in N(DD)} \left(\frac{S_{i,t}-\mu_t^D}{\sigma_t^D} - \theta\ \frac{S_{i,t-1}-\mu_{t-1}^D}{\sigma_{t-1}^D}\right)\left(\frac{-\varphi_j(t)}{\sigma_t^D} + \frac{\theta\varphi_j(t-1)}{\sigma_{t-1}^D}\right)\right. \\ & + \sum_{t\in N(DW)} \left(\frac{S_{i,t}-\mu_t^W}{\sigma_t^W} - \theta\ \frac{S_{i,t-1}-\mu_{t-1}^D}{\sigma_{t-1}^D}\right)\left(\frac{\theta\varphi_j(t-1)}{\sigma_{t-1}^D}\right) \\ & \left. + \sum_{t\in N(WD)} \left(\frac{S_{i,t}-\mu_t^D}{\sigma_t^D} - \theta\ \frac{S_{i,t-1}-\mu_{t-1}^W}{\sigma_{t-1}^W}\right)\left(\frac{-\varphi_j(t)}{\sigma_t^D}\right)\right]\end{aligned}$$

$$\begin{aligned}\frac{\partial \ell(\psi)}{\partial \alpha_j^W} = & -\left[\sum_{t\in N(WW)} \left(\frac{S_{i,t}-\mu_t^W}{\sigma_t^W} - \theta\ \frac{S_{i,t-1}-\mu_{t-1}^W}{\sigma_{t-1}^W}\right)\left(\frac{-\varphi_j(t)}{\sigma_t^W} + \frac{\theta\varphi_j(t-1)}{\sigma_{t-1}^W}\right)\right. \\ & + \sum_{t\in N(DW)} \left(\frac{S_{i,t}-\mu_t^W}{\sigma_t^W} - \theta\ \frac{S_{i,t-1}-\mu_{t-1}^D}{\sigma_{t-1}^D}\right)\left(\frac{-\varphi_j(t)}{\sigma_t^W}\right) \\ & \left. + \sum_{t\in N(WD)} \left(\frac{S_{i,t}-\mu_t^D}{\sigma_t^D} - \theta\ \frac{S_{i,t-1}-\mu_{t-1}^W}{\sigma_{t-1}^W}\right)\left(\frac{\theta\varphi_j(t-1)}{\sigma_{t-1}^W}\right)\right]\end{aligned}$$

$$\begin{aligned}\frac{\partial \ell(\psi)}{\partial \xi_j^D} = & -\sum_{\substack{t\in N(DD)\\ t\in N(WD)}} \frac{\varphi_j(t)}{\sigma_t^D} - \left[\sum_{t\in N(DD)} \left(\frac{S_{i,t}-\mu_t^D}{\sigma_t^D} - \theta\ \frac{S_{i,t-1}-\mu_{t-1}^D}{\sigma_{t-1}^D}\right)\right. \\ & \left(-\frac{(S_{i,t}-\mu_t^D)}{(\sigma_t^D)^2}\ \varphi_j(t) + \theta\ \frac{S_{i,t-1}-\mu_{t-1}^D}{(\sigma_{t-1}^D)^2}\ \varphi_j(t-1)\right) \\ & + \sum_{t\in N(DW)} \left(\frac{S_{i,t}-\mu_t^W}{\sigma_t^W} - \theta\ \frac{S_{i,t-1}-\mu_{t-1}^D}{\sigma_{t-1}^D}\right)\left(\theta\ \frac{S_{i,t-1}-\mu_{t-1}^D}{(\sigma_{t-1}^D)^2}\ \varphi_j(t-1)\right) \\ & \left. + \sum_{t\in N(WD)} \left(\frac{S_{i,t}-\mu_t^D}{\sigma_t^D} - \theta\ \frac{S_{i,t-1}-\mu_{t-1}^W}{\sigma_{t-1}^W}\right)\left(-\frac{(S_{i,t}-\mu_t^D)}{(\sigma_t^D)^2}\ \varphi_j(t)\right)\right]\end{aligned}$$

$$\frac{\partial \ell(\psi)}{\partial \xi_j^W} = - \sum_{\substack{t\in N(WW)\\ t\in N(DW)}} \frac{\varphi_j(t)}{\sigma_t^W} - \Bigg[\sum_{t\in N(WW)} \left(\frac{S_{i,t} - \mu_t^W}{\sigma_t^W} - \theta \, \frac{S_{i,t-1} - \mu_{t-1}^W}{\sigma_{t-1}^W} \right)$$
$$\left(- \frac{(S_{i,t} - \mu_t^W)}{(\sigma_t^W)^2} \, \varphi_j(t) + \theta \, \frac{S_{i,t-1} - \mu_{t-1}^W}{(\sigma_{t-1}^W)^2} \, \varphi_j(t-1) \right)$$
$$+ \sum_{t\in N(DW)} \left(\frac{S_{i,t} - \mu_t^W}{\sigma_t^W} - \theta \, \frac{S_{i,t-1} - \mu_{t-1}^D}{\sigma_{t-1}^D} \right) \left(- \frac{(S_{i,t} - \mu_t^W)}{(\sigma_t^W)^2} \, \varphi_j(t) \right)$$
$$+ \sum_{t\in N(WD)} \left(\frac{S_{i,t} - \mu_t^D}{\sigma_t^D} - \theta \, \frac{S_{i,t-1} - \mu_{t-1}^W}{\sigma_{t-1}^W} \right) \left(\theta \, \frac{S_{i,t-1} - \mu_{t-1}^W}{(\sigma_{t-1}^W)^2} \, \varphi_j(t-1) \right) \Bigg]$$

The equations obtained when the partial derivatives are set to zero can be solved using the Newton–Raphson iteration method. For this, the second partial derivatives are required. These are given by:

$$\frac{\partial^2 \ell(\psi)}{\partial \theta \partial \theta} = - \left[\sum_{\substack{t\in N(DD)\\ t\in N(DW)}} \left(\frac{S_{i,t-1} - \mu_{t-1}^D}{\sigma_{t-1}^D} \right)^2 + \sum_{\substack{t\in N(WW)\\ t\in N(WD)}} \left(\frac{S_{i,t-1} - \mu_{t-1}^W}{\sigma_{t-1}^W} \right)^2 \right]$$

$$\frac{\partial^2 \ell(\psi)}{\partial \theta \partial \alpha_j^D} = \sum_{t\in N(DD)} \left\{ \left(\frac{S_{i,t} - \mu_t^D}{\sigma_t^D} - \theta \, \frac{S_{i,t-1} - \mu_{t-1}^D}{\sigma_{t-1}^D} \right) \left(\frac{-\varphi_j(t)}{\sigma_t^D} \right) \right.$$
$$\left. + \left(\frac{-\varphi_j(t)}{\sigma_t^D} + \frac{\theta \varphi_j(t-1)}{\sigma_{t-1}^D} \right) \left(\frac{S_{i,t-1} - \mu_{t-1}^D}{\sigma_{t-1}^D} \right) \right\}$$
$$+ \sum_{t\in N(DW)} \left\{ \left(\frac{S_{i,t} - \mu_t^W}{\sigma_t^W} - \theta \, \frac{S_{i,t-1} - \mu_{t-1}^D}{\sigma_{t-1}^D} \right) \left(\frac{-\varphi_j(t-1)}{\sigma_{t-1}^D} \right) \right.$$
$$\left. + \left(\frac{\theta \varphi_j(t-1)}{\sigma_{t-1}^D} \right) \left(\frac{S_{i,t-1} - \mu_{t-1}^D}{\sigma_{t-1}^D} \right) \right\}$$
$$+ \sum_{t\in N(WD)} \left(\frac{-\varphi_j(t)}{\sigma_t^D} \right) \left(\frac{S_{i,t-1} - \mu_{t-1}^W}{\sigma_{t-1}^W} \right)$$

$$\frac{\partial^2 \ell(\psi)}{\partial\theta\partial\alpha_j^W} = \sum_{t\in N(WW)} \left\{ \left(\frac{S_{i,t}-\mu_t^W}{\sigma_t^W} - \theta\,\frac{S_{i,t-1}-\mu_{t-1}^W}{\sigma_{t-1}^W} \right) \left(\frac{-\varphi_j(t-1)}{\sigma_{t-1}^W} \right) + \left(\frac{-\varphi_j(t)}{\sigma_t^W} + \frac{\theta\varphi_j(t-1)}{\sigma_{t-1}^W} \right) \left(\frac{S_{i,t-1}-\mu_{t-1}^W}{\sigma_{t-1}^W} \right) \right\} + \sum_{t\in N(DW)} \left(\frac{-\varphi_j(t)}{\sigma_t^W} \right) \left(\frac{S_{i,t-1}-\mu_{t-1}^D}{\sigma_{t-1}^D} \right) + \sum_{t\in N(WD)} \left\{ \left(\frac{S_{i,t}-\mu_t^D}{\sigma_t^D} - \theta\,\frac{S_{i,t-1}-\mu_{t-1}^W}{\sigma_{t-1}^W} \right) \left(\frac{-\varphi_j(t-1)}{\sigma_{t-1}^W} \right) + \left(\frac{\theta\varphi_j(t-1)}{\sigma_{t-1}^W} \right) \left(\frac{S_{i,t-1}-\mu_{t-1}^W}{\sigma_{t-1}^W} \right) \right\}$$

$$\frac{\partial^2 \ell(\psi)}{\partial\theta\partial\xi_j^D} = \sum_{t\in N(DD)} \left\{ \left(\frac{S_{i,t}-\mu_t^D}{\sigma_t^D} - \theta\,\frac{S_{i,t-1}-\mu_{t-1}^D}{\sigma_{t-1}^D} \right) \left(-\varphi_j(t-1)\frac{S_{i,t-1}-\mu_{t-1}^D}{(\sigma_{t-1}^D)^2} \right) + \left(-\varphi_j(t)\frac{S_{i,t}-\mu_t^D}{(\sigma_t^D)^2} + \theta\varphi_j(t-1)\frac{S_{i,t-1}-\mu_{t-1}^D}{(\sigma_{t-1}^D)^2} \right) \left(\frac{S_{i,t-1}-\mu_{t-1}^D}{\sigma_{t-1}^D} \right) \right\} + \sum_{t\in N(DW)} \left\{ \left(\frac{S_{i,t}-\mu_t^W}{\sigma_t^W} - \theta\,\frac{S_{i,t-1}-\mu_{t-1}^D}{\sigma_{t-1}^D} \right) \left(-\varphi_j(t-1)\frac{S_{i,t-1}-\mu_{t-1}^D}{(\sigma_{t-1}^D)^2} \right) + \left(\theta\varphi_j(t-1)\frac{S_{i,t-1}-\mu_{t-1}^D}{(\sigma_{t-1}^D)^2} \right) \left(\frac{S_{i,t-1}-\mu_{t-1}^D}{\sigma_{t-1}^D} \right) \right\} + \sum_{t\in N(WD)} \left(-\varphi_j(t)\frac{S_{i,t}-\mu_t^D}{(\sigma_t^D)^2} \right) \left(\frac{S_{i,t-1}-\mu_{t-1}^W}{\sigma_{t-1}^W} \right)$$

$$\frac{\partial^2 \ell(\psi)}{\partial\theta\partial\xi_j^W} = \sum_{t\in N(WW)} \left\{ \left(\frac{S_{i,t}-\mu_t^W}{\sigma_t^W} - \theta\,\frac{S_{i,t-1}-\mu_{t-1}^W}{\sigma_{t-1}^W} \right) \left(-\varphi_j(t-1)\frac{S_{i,t-1}-\mu_{t-1}^W}{(\sigma_{t-1}^W)^2} \right) + \left(-\varphi_j(t)\frac{S_{i,t}-\mu_t^W}{(\sigma_t^W)^2} + \theta\varphi_j(t-1)\frac{S_{i,t-1}-\mu_{t-1}^W}{(\sigma_{t-1}^W)^2} \right) \left(\frac{S_{i,t-1}-\mu_{t-1}^W}{\sigma_{t-1}^W} \right) \right\} + \sum_{t\in N(DW)} \left(-\varphi_j(t)\frac{S_{i,t}-\mu_t^W}{(\sigma_t^W)^2} \right) \left(\frac{S_{i,t-1}-\mu_{t-1}^D}{\sigma_{t-1}^D} \right) + \sum_{t\in N(WD)} \left\{ \left(\frac{S_{i,t}-\mu_t^D}{\sigma_t^D} - \theta\,\frac{S_{i,t-1}-\mu_{t-1}^W}{\sigma_{t-1}^W} \right) \left(-\varphi_j(t-1)\frac{S_{i,t-1}-\mu_{t-1}^W}{(\sigma_{t-1}^W)^2} \right) + \left(\theta\varphi_j(t-1)\frac{S_{i,t-1}-\mu_{t-1}^W}{(\sigma_{t-1}^W)^2} \right) \left(\frac{S_{i,t-1}-\mu_{t-1}^W}{\sigma_{t-1}^W} \right) \right\}$$

$$\frac{\partial^2\ell(\psi)}{\partial\alpha_j^D\partial\alpha_k^D} = -\left[\sum_{t\in N(DD)}\left(\frac{-\varphi_j(t)}{\sigma_t^D}+\frac{\theta\varphi_j(t-1)}{\sigma_{t-1}^D}\right)\left(\frac{-\varphi_k(t)}{\sigma_t^D}+\frac{\theta\varphi_k(t-1)}{\sigma_{t-1}^D}\right)\right.$$
$$+\sum_{t\in N(DW)}\left(\frac{\theta\varphi_j(t-1)}{\sigma_{t-1}^D}\right)\left(\frac{\theta\varphi_k(t-1)}{\sigma_{t-1}^D}\right)$$
$$\left.+\sum_{t\in N(WD)}\left(\frac{-\varphi_j(t)}{\sigma_t^D}\right)\left(\frac{-\varphi_k(t)}{\sigma_t^D}\right)\right]$$

$$\frac{\partial^2\ell(\psi)}{\partial\alpha_j^D\partial\alpha_k^W} = -\left[\sum_{t\in N(DW)}\left(\frac{-\varphi_k(t)}{\sigma_t^W}\right)\left(\frac{\theta\varphi_j(t-1)}{\sigma_{t-1}^D}\right)+\sum_{t\in N(WD)}\left(\frac{\theta\varphi_k(t-1)}{\sigma_{t-1}^W}\right)\left(\frac{-\varphi_j(t)}{\sigma_t^D}\right)\right]$$

$$\frac{\partial^2\ell(\psi)}{\partial\alpha_j^D\partial\xi_k^D} = -\left[\sum_{t\in N(DD)}\left\{\left(\frac{S_{i,t}-\mu_t^D}{\sigma_t^D}-\theta\,\frac{S_{i,t-1}-\mu_{t-1}^D}{\sigma_{t-1}^D}\right)\right.\right.$$
$$\left(\frac{\varphi_k(t)\varphi_j(t)}{(\sigma_t^D)^2}-\frac{\theta\varphi_k(t-1)\varphi_j(t-1)}{(\sigma_{t-1}^D)^2}\right)$$
$$\left.+\left(-\varphi_k(t)\frac{S_{i,t}-\mu_t^D}{(\sigma_t^D)^2}+\theta\varphi_k(t-1)\frac{S_{i,t-1}-\mu_{t-1}^D}{(\sigma_{t-1}^D)^2}\right)\left(\frac{-\varphi_j(t)}{\sigma_t^D}+\frac{\theta\varphi_j(t-1)}{\sigma_{t-1}^D}\right)\right\}$$
$$+\sum_{t\in N(DW)}\left\{\left(\frac{S_{i,t}-\mu_t^W}{\sigma_t^W}-\theta\,\frac{S_{i,t-1}-\mu_{t-1}^D}{\sigma_{t-1}^D}\right)\left(\frac{-\theta\varphi_k(t-1)\varphi_j(t-1)}{(\sigma_{t-1}^D)^2}\right)\right.$$
$$\left.+\left(\theta\varphi_k(t-1)\frac{S_{i,t-1}-\mu_{t-1}^D}{(\sigma_{t-1}^D)^2}\right)\left(\frac{\theta\varphi_j(t-1)}{\sigma_{t-1}^D}\right)\right\}$$
$$+\sum_{t\in N(WD)}\left\{\left(-\varphi_k(t)\frac{S_{i,t}-\mu_t^D}{(\sigma_t^D)^2}\right)\left(\frac{-\varphi_j(t)}{\sigma_t^D}\right)\right.$$
$$\left.\left.+\left(\frac{S_{i,t}-\mu_t^D}{\sigma_t^D}-\theta\,\frac{S_{i,t-1}-\mu_{t-1}^W}{\sigma_{t-1}^W}\right)\left(\frac{\varphi_j(t)\varphi_k(t)}{(\sigma_t^D)^2}\right)\right\}\right]$$

$$\frac{\partial^2\ell(\psi)}{\partial\alpha_j^D\partial\xi_k^W} = -\left[\sum_{t\in N(DW)}\left(-\varphi_k(t)\frac{S_t-\mu_t^W}{(\sigma_t^W)^2}\right)\left(\frac{\theta\varphi_j(t-1)}{\sigma_{t-1}^D}\right)\right.$$
$$\left.+\sum_{t\in N(WD)}\left(\theta\varphi_k(t-1)\frac{S_{t-1}-\mu_{t-1}^W}{(\sigma_{t-1}^W)^2}\right)\left(\frac{-\varphi_j(t)}{\sigma_t^D}\right)\right]$$

$$\frac{\partial^2 \ell(\psi)}{\partial \alpha_j^W \partial \alpha_k^W} = -\left[\sum_{t\in N(WW)} \left(\frac{-\varphi_k(t)}{\sigma_{t'}^W} + \frac{\theta\varphi_k(t-1)}{\sigma_{t-1}^W}\right)\left(\frac{-\varphi_j(t)}{\sigma_t^W} + \frac{\theta\varphi_j(t-1)}{\sigma_{t-1}^W}\right) + \sum_{t\in N(DW)} \left(\frac{-\varphi_k(t)}{\sigma_t^W}\right)\left(\frac{-\varphi_j(t)}{\sigma_t^W}\right) + \sum_{t\in N(WD)} \left(\frac{\theta\varphi_k(t-1)}{\sigma_{t-1}^W}\right)\left(\frac{\theta\varphi_j(t-1)}{\sigma_{t-1}^W}\right)\right]$$

$$\frac{\partial^2 \ell(\psi)}{\partial \alpha_j^W \partial \xi_k^D} = -\left[\sum_{t\in N(DW)} \left(\theta\varphi_k(t-1)\frac{S_{i,t-1}-\mu_{t-1}^D}{(\sigma_{t-1}^D)^2}\right)\left(\frac{-\varphi_j(t)}{\sigma_t^W}\right) + \sum_{t\in N(WD)} \left(-\varphi_k(t)\frac{S_{i,t-1}-\mu_t^D}{(\sigma_t^D)^2}\right)\left(\frac{\theta\varphi_j(t-1)}{\sigma_{t-1}^W}\right)\right]$$

$$\begin{aligned}\frac{\partial^2 \ell(\psi)}{\partial \alpha_j^W \partial \xi_k^W} = -\Bigg[&\sum_{t\in N(WW)} \Bigg\{\left(\frac{S_{i,t}-\mu_t^W}{\sigma_t^W} - \theta\,\frac{S_{i,t-1}-\mu_{t-1}^W}{\sigma_{t-1}^W}\right) \\ &\left(\frac{\varphi_j(t)\varphi_k(t)}{(\sigma_t^W)^2} - \frac{\theta\varphi_j(t-1)\varphi_k(t-1)}{(\sigma_{t-1}^W)^2}\right) \\ &+ \left(\frac{-\varphi_j(t)}{\sigma_t^W} + \frac{\theta\varphi_j(t-1)}{\sigma_{t-1}^W}\right)\left(-\varphi_k(t)\frac{S_{i,t}-\mu_t^W}{(\sigma_t^W)^2} + \theta\varphi_k(t-1)\frac{S_{i,t-1}-\mu_{t-1}^W}{(\sigma_{t-1}^W)^2}\right)\Bigg\} \\ &+ \sum_{t\in N(DW)} \Bigg\{\left(\frac{S_{i,t}-\mu_t^W}{\sigma_t^W} - \theta\,\frac{S_{i,t-1}-\mu_{t-1}^D}{\sigma_{t-1}^D}\right)\left(\frac{\varphi_j(t)\varphi_k(t)}{(\sigma_t^W)^2}\right) \\ &+ \left(-\varphi_k(t)\frac{S_{i,t}-\mu_t^W}{(\sigma_t^W)^2}\right)\left(\frac{-\varphi_j(t)}{\sigma_t^W}\right)\Bigg\} \\ &+ \sum_{t\in N(WD)} \Bigg\{\left(\frac{S_{i,t}-\mu_t^D}{\sigma_t^D} - \theta\,\frac{S_{i,t-1}-\mu_{t-1}^W}{\sigma_{t-1}^W}\right)\left(\frac{-\theta\varphi_j(t-1)\varphi_k(t-1)}{(\sigma_{t-1}^W)^2}\right) \\ &+ \left(\frac{\theta\varphi_j(t-1)}{\sigma_{t-1}^W}\right)\left(\theta\varphi_k(t-1)\frac{S_{i,t-1}-\mu_{t-1}^W}{(\sigma_{t-1}^W)^2}\right)\Bigg\}\Bigg]\end{aligned}$$

$$\frac{\partial^2 \ell(\psi)}{\partial \xi_j^D \partial \xi_k^D} = \sum_{\substack{t\in N(DD)\\ t\in N(WD)}} \frac{\varphi_j(t)\varphi_k(t)}{(\sigma_t^D)^2} - \Bigg[\sum_{t\in N(DD)} \Bigg\{ \left(\frac{S_{i,t}-\mu_t^D}{\sigma_t^D} - \theta\, \frac{S_{i,t-1}-\mu_{t-1}^D}{\sigma_{t-1}^D} \right)$$

$$\left(2\varphi_j(t)\varphi_k(t)\frac{S_{i,t}-\mu_t^D}{(\sigma_t^D)^3} - 2\theta\varphi_j(t-1)\varphi_k(t-1)\frac{S_{i,t-1}-\mu_{t-1}^D}{(\sigma_{t-1}^D)^3} \right)$$

$$+ \left(-\varphi_j(t)\frac{S_{i,t}-\mu_t^D}{(\sigma_t^D)^2} + \theta\varphi_j(t-1)\frac{S_{i,t-1}-\mu_{t-1}^D}{(\sigma_{t-1}^D)^2} \right)$$

$$\left(-\varphi_k(t)\frac{S_{i,t}-\mu_t^D}{(\sigma_t^D)^2} + \theta\varphi_k(t-1)\frac{S_{i,t-1}-\mu_{t-1}^D}{(\sigma_{t-1}^D)^2} \right) \Bigg\}$$

$$+ \sum_{t\in N(DW)} \Bigg\{ \left(\frac{S_{i,t}-\mu_t^W}{\sigma_t^W} - \theta\, \frac{S_{i,t-1}-\mu_{t-1}^D}{\sigma_{t-1}^D} \right)$$

$$\left(-2\theta\varphi_j(t-1)\varphi_k(t-1)\frac{S_{i,t-1}-\mu_{t-1}^D}{(\sigma_{t-1}^D)^3} \right)$$

$$+ \left(\theta\frac{S_{i,t-1}-\mu_{t-1}^D}{(\sigma_{t-1}^D)^2} \right)^2 \varphi_j(t-1)\varphi_k(t-1) \Bigg\}$$

$$+ \sum_{t\in N(WD)} \Bigg\{ \left(\frac{S_{i,t}-\mu_t^D}{\sigma_t^D} - \theta\, \frac{S_{i,t-1}-\mu_{t-1}^W}{\sigma_{t-1}^W} \right) \left(2\varphi_j(t)\varphi_k(t)\frac{S_{i,t}-\mu_t^D}{(\sigma_t^D)^3} \right)$$

$$+ \left(\frac{S_{i,t}-\mu_t^D}{(\sigma_t^D)^2} \right)^2 \varphi_j(t)\varphi_k(t) \Bigg\} \Bigg]$$

$$\frac{\partial^2 \ell(\psi)}{\partial \xi_j^D \partial \xi_k^W} = - \Bigg[\sum_{t\in N(DW)} \left(-\varphi_k(t)\frac{S_{i,t}-\mu_t^W}{(\sigma_t^W)^2} \right) \left(\theta\varphi_j(t-1)\frac{S_{i,t-1}-\mu_{t-1}^D}{(\sigma_{t-1}^D)^2} \right)$$

$$+ \sum_{t\in N(WD)} \left(\theta\varphi_k(t-1)\frac{S_{i,t-1}-\mu_{t-1}^W}{(\sigma_{t-1}^W)^2} \right) \left(-\varphi_j(t)\frac{S_{i,t}-\mu_t^D}{(\sigma_t^D)^2} \right) \Bigg]$$

$$\frac{\partial^2 \ell(\psi)}{\partial \xi_j^W \partial \xi_k^W} = \sum_{\substack{t\in N(WW)\\ t\in N(DW)}} \frac{\varphi_j(t)\varphi_k(t)}{(\sigma_t^W)^2} - \left[\sum_{t\in N(WW)} \left\{\left(\frac{S_{i,t}-\mu_t^W}{\sigma_t^W} - \theta\,\frac{S_{i,t-1}-\mu_{t-1}^W}{\sigma_{t-1}^W}\right)\right.\right.$$
$$\left(2\varphi_j(t)\varphi_k(t)\frac{S_{i,t}-\mu_t^W}{(\sigma_t^W)^3} - 2\theta\varphi_j(t-1)\varphi_k(t-1)\frac{S_{i,t-1}-\mu_{t-1}^W}{(\sigma_{t-1}^W)^3}\right)$$
$$+\left(-\varphi_j(t)\frac{S_{i,t}-\mu_t^W}{(\sigma_t^W)^2} + \theta\varphi_j(t-1)\frac{S_{i,t-1}-\mu_{t-1}^W}{(\sigma_{t-1}^W)^2}\right)$$
$$\left.\left(-\varphi_k(t)\frac{S_{i,t}-\mu_t^W}{(\sigma_t^W)^2} + \theta\varphi_k(t-1)\frac{S_{i,t-1}-\mu_{t-1}^W}{(\sigma_{t-1}^W)^2}\right)\right\}$$
$$+\sum_{t\in N(DW)} \left\{\left(\frac{S_{i,t}-\mu_t^W}{\sigma_t^W} - \theta\,\frac{S_{i,t-1}-\mu_{t-1}^D}{\sigma_{t-1}^D}\right)\left(2\varphi_j(t)\varphi_k(t)\frac{S_{i,t}-\mu_t^W}{(\sigma_t^W)^3}\right)\right.$$
$$\left.+\left(-\varphi_j(t)\frac{S_{i,t}-\mu_t^W}{(\sigma_t^W)^2}\right)\left(-\varphi_k(t)\frac{S_{i,t}-\mu_t^W}{(\sigma_t^W)^2}\right)\right\}$$
$$+\sum_{t\in N(WD)} \left\{\left(\frac{S_{i,t}-\mu_t^D}{\sigma_t^D} - \theta\,\frac{S_{i,t-1}-\mu_{t-1}^W}{\sigma_{t-1}^W}\right)\right.$$
$$\left(-2\theta\varphi_j(t-1)\varphi_k(t-1)\frac{S_{i,t-1}-\mu_{t-1}^W}{(\sigma_{t-1}^W)^3}\right)$$
$$\left.\left.+\left(\theta\varphi_j(t-1)\frac{S_{i,t-1}-\mu_{t-1}^W}{(\sigma_{t-1}^W)^2}\right)\left(\theta\varphi_k(t-1)\frac{S_{i,t-1}-\mu_{t-1}^W}{(\sigma_{t-1}^W)^2}\right)\right\}\right]$$

The following algorithm is used to estimate the parameters.

Algorithm

Step 1: Estimate initial $\widehat{\mu}_t$ and $\widehat{\sigma}_t$ by aproximating by its Fourier series representation and estimating the parameters α_i and ξ_i by the method mentioned in the previous models.

Step 2: Estimate initial $\widehat{\theta}$ using the following formula:

$$\widehat{\theta} = \frac{T\sum_{t=2}^{T}(S_{i,t}-\widehat{\mu}_t)(S_{i,t-1}-\widehat{\mu}_{t-1})}{(T-1)\sum_{t=2}^{T+1}(S_{i,t-1}-\widehat{\mu}_{t-1})^2}$$

where $\widehat{\mu}_t$ depends on the status of day t and $\widehat{\mu}_{t-1}$ depends on the status of day $t-1$.

Step 3: Compute $f^{(k)}$ and $F^{(k)}$ where $f^{(k)}$ is the vector of first partial derivatives and $F^{(k)}$ is the matrix of second partial derivatives, computed at the kth iteration.

Step 4: Compute the vector $\delta^{(k)}$ which is the solution to the system of NP linear equations

$$F^{(k)}\delta^{(k)} = f^{(k)}$$

where NP is the number of parameters in the model.

Step 5: Set $\beta^{(k+1)} = \beta^{(k)} - \delta^{(k)}$, where $\beta^{(k)}$ contains the parameter estimates computed at the kth iteration.

Step 6 Test for convergence, for example, if the elements of $f^{(k)}$ are sufficiently close to zero. If the convergence criterion is met then stop, otherwise increase k by 1 and return to step 3.

The cross-correlation matrix, $\widehat{\Sigma}$, has elements given by

$$R_{jk} = \frac{\frac{1}{T}\sum_{i=1}^{NY}\sum_{t=1}^{NT} e_{i,t}^{(j)} e_{i,t}^{(k)} - \frac{1}{T^2}\sum_{i=1}^{NY}\sum_{t=1}^{NT} e_{i,t}^{(j)} \sum_{i=1}^{NY}\sum_{t=1}^{NT} e_{i,t}^{(k)}}{\left[\frac{1}{T}\sum_{i=1}^{NY}\sum_{t=1}^{NT}(e_{i,t}^{(j)})^2 - \frac{1}{T^2}\left(\sum_{i=1}^{NY}\sum_{t=1}^{NT} e_{i,t}^{(j)}\right)^2\right]^{\frac{1}{2}} \left[\frac{1}{T}\sum_{i=1}^{NY}\sum_{t=1}^{NT}(e_{i,t}^{(k)})^2 - \frac{1}{T^2}\left(\sum_{i=1}^{NY}\sum_{t=1}^{NT} e_{i,t}^{(k)}\right)^2\right]^{\frac{1}{2}}}$$

where

$e_{i,t}^{(j)}$ denotes the residual time series of variable j ; $j = 1, 2, \ldots, NV$

and

$e_{i,t}^{(k)}$ denotes the residual time series of variable k; $k = 1, 2, \ldots, NV$.

(d) Model Selection

The order of the autoregressive process is chosen in the same way as in the previous model as is the order of the Fourier series approximation.

Model 5: Multivariate model for climate data

(a) Notation

Partition the year into $NT(=365)$ equal intervals, denoted by $t = 1, 2, \ldots NT$.

NV is the number of variables.

NY is the number of years observed.

W represents the occurrence of rain.

D represents the non–occurrence of rain.

DD represents the sequence when day $t-1$ was dry and day t was dry.

WW represents the sequence when day $t-1$ was wet and day t was wet.

DW represents the sequence when day $t-1$ was dry and day t was wet.

WD represents the sequence when day $t-1$ was wet and day t was dry.

T represents the total number of observations, i.e. $NT\ NY$.

$N(DD)$ is the set of time periods t such that period t was dry and period $t-1$ was dry, $t = 1, 2, \ldots, T$.

$N(WW)$ is the set of time periods t such that period t was wet and period $t-1$ was wet.

$N(DW)$ is the set of time periods t such that period t was wet and period $t-1$ was dry.

$N(WD)$ is the set of time periods t such that period t was dry and period $t-1$ was wet.

$Y_{i,t}$ is the precipitation amount on period t of year i, $i = 1, 2, \ldots, NY$.

$S_{i,t}$ is the generic name for the observation at time t of the ith year.

μ_t^D is the generic name for the mean for a dry day on period t.

μ_t^W is the generic name for the mean for a wet day on period t.

σ_t^D is the generic name for the standard deviation for a dry day on period t.

σ_t^W is the generic name for the standard deviation for a wet day on period t.

θ^{DD} is the coefficient of the AR(1) process, given sequence DD.

θ^{WW} is the coefficient of the AR(1) process given sequence WW.

θ^{DW} is the coefficient of the AR(1) process given sequence DW.

θ^{WD} is the coefficient of the AR(1) process given sequence WD.

(b) Model and assumptions

Model 5 is formulated in the same manner as that of Model 4, except that here it is assumed that the coefficient of the AR(1) process, θ , varies according to the wet/dry status of the present and previous day.

Therefore, the time series $S_{i,t}$ is once again reduced to a residual time series $\chi_{i,t}$ by removing the periodic mean and standard deviation, i.e.

$$\chi_{i,t} = \frac{S_{i,t} - \mu_t}{\sigma_t}.$$

Assume that this residual time series follows an AR(1) process, i.e.

$$\chi_{i,t} = \theta\ \chi_{i,t-1} + e_{i,t}$$

where $e_{i,t} \sim NID(0,1) \quad i = 1,2,\ldots,NY;\ t = 1,2,\ldots,NT$.

The model incorporating the different wet/dry sequences is given by:

$$e_{i,t} = \frac{S_{i,t} - \mu_t^D}{\sigma_t^D} - \theta^{DD}\ \frac{S_{i,t-1} - \mu_{t-1}^D}{\sigma_{t-1}^D} \quad \text{if day } t-1 \text{ was dry and day } t \text{ was dry.}$$

or

$$e_{i,t} = \frac{S_{i,t} - \mu_t^W}{\sigma_t^W} - \theta^{WW}\ \frac{S_{i,t-1} - \mu_{t-1}^W}{\sigma_{t-1}^W} \quad \text{if day } t-1 \text{ was wet and day } t \text{ was wet.}$$

or

$$e_{i,t} = \frac{S_{i,t} - \mu_t^W}{\sigma_t^W} - \theta^{DW}\ \frac{S_{i,t-1} - \mu_{t-1}^D}{\sigma_{t-1}^D} \quad \text{if day } t-1 \text{ was dry and day } t \text{ was wet.}$$

or

$$e_{i,t} = \frac{S_{i,t} - \mu_t^D}{\sigma_t^D} - \theta^{WD}\ \frac{S_{i,t-1} - \mu_{t-1}^W}{\sigma_{t-1}^W} \quad \text{if day } t-1 \text{ was wet and day } t \text{ was dry.}$$

The mean and standard deviation functions are approximated by their respective truncated Fourier representation, i.e.

$$\left.\begin{aligned} \mu_t^D &= \sum_{i=1}^{L} \alpha_i^D \varphi_i(t) \\ \sigma_t^D &= \sum_{i=1}^{L} \xi_i^D \varphi_i(t) \end{aligned}\right\} \quad \text{if } t \text{ dry}$$

$$\left.\begin{aligned} \mu_t^W &= \sum_{i=1}^{L} \alpha_i^W \varphi_i(t) \\ \sigma_t^W &= \sum_{i=1}^{L} \xi_i^W \varphi_i(t) \end{aligned}\right\} \quad \text{if } t \text{ wet}$$

where $\varphi_i(t)$ is defined as before and L is the order of the Fourier series approximation.

(c) Estimation

The density function of $e_{i,t}$ is given by

$$f(e_{i,t}) = \frac{1}{\sqrt{2\pi}} \exp(-\frac{1}{2}e_{i,t}^2)$$

since $e_{i,t} \sim NID(0,1)$.

The joint likelihood function, conditioned on the four different sequences is given by:

$$\begin{aligned}
L(\psi) &= L(\alpha_j^D, \alpha_j^W, \xi_j^D, \xi_j^W, \theta^{DD}, \theta^{WW}, \theta^{DW}, \theta^{WD}; e_{i,t}) \\
&= \prod_{t\in N(DD)} f(e_{i,t}|DD) \prod_{t\in N(WW)} f(e_{i,t}|WW) \\
&\quad \prod_{t\in N(DW)} f(e_{i,t}|DW) \prod_{t\in N(WD)} f(e_{i,t}|WD) \\
&= \left(\frac{1}{\sqrt{2\pi}}\right)^T \exp\left\{-\frac{1}{2}\left[\sum_{t\in N(DD)} (e_{i,t}|DD)^2\right.\right. \\
&\quad + \sum_{t\in N(WW)} (e_{i,t}|WW)^2 + \sum_{t\in N(DW)} (e_{i,t}|DW)^2 \\
&\quad \left.\left. + \sum_{t\in N(WD)} (e_{i,t}|WD)^2\right]\right\}.
\end{aligned}$$

Make the following transformation

$$e_{i,t} = \frac{S_{i,t} - \mu_t}{\sigma_t} - \theta \frac{S_{i,t-1} - \mu_{t-1}}{\sigma_{t-1}}.$$

The Jacobian of the transformation is

$$\left|\frac{\partial e_{i,t}}{\partial S_{i,p}}\right| = \begin{vmatrix}
1/\sigma_1 & 0 & & & 0 \\
-\theta/\sigma_1 & 1/\sigma_2 & 0 & \dots & 0 \\
\vdots & & & & \vdots \\
0 & \dots & 0 & -\theta/\sigma_{364} & 1/\sigma_{365} \\
\vdots & & & & \vdots \\
1/\sigma_1 & 0 & \dots & \dots & 0 \\
-\theta/\sigma_1 & 1/\sigma_2 & 0 & \dots & 0 \\
\vdots & & & & \vdots \\
0 & \dots & 0 & -\theta/\sigma_{364} & 1/\sigma_{365}
\end{vmatrix}$$

$$= \prod_{i=1}^{NY} \prod_{t=1}^{NT} \frac{1}{\sigma_t}$$

Taking into account the conditional sequences imposed on $e_{i,t}$, the Jacobian is then given by

$$\left|\frac{\partial e_{i,t}}{\partial S_{i,p}}\right| = \prod_{\substack{t\in N(DD)\\ t\in N(WD)}} \frac{1}{\sigma_t^D} \prod_{\substack{t\in N(WW)\\ t\in N(DW)}} \frac{1}{\sigma_t^W}.$$

The joint probability density function is thus given by:

$$L(\psi) = \left(\frac{1}{\sqrt{2\pi}}\right)^T \prod_{\substack{t\in N(DD)\\ t\in N(WD)}} \frac{1}{\sigma_t^D} \prod_{\substack{t\in N(WW)\\ t\in N(DW)}} \frac{1}{\sigma_t^W}$$

$$\exp\left\{-\frac{1}{2}\left[\sum_{t\in N(DD)} \left(\frac{S_{i,t}-\mu_t^D}{\sigma_t^D} - \theta^{DD}\frac{S_{i,t-1}-\mu_{t-1}^D}{\sigma_{t-1}^D}\right)^2\right.\right.$$

$$+ \sum_{t\in N(WW)} \left(\frac{S_{i,t}-\mu_t^W}{\sigma_t^W} - \theta^{WW}\frac{S_{i,t-1}-\mu_{t-1}^W}{\sigma_{t-1}^W}\right)^2$$

$$+ \sum_{t\in N(DW)} \left(\frac{S_{i,t}-\mu_t^W}{\sigma_t^W} - \theta^{DW}\frac{S_{i,t-1}-\mu_{t-1}^D}{\sigma_{t-1}^D}\right)^2$$

$$\left.\left.+ \sum_{t\in N(WD)} \left(\frac{S_{i,t}-\mu_t^D}{\sigma_t^D} - \theta^{WD}\frac{S_{i,t-1}-\mu_{t-1}^W}{\sigma_{t-1}^W}\right)^2\right]\right\}$$

and the log–likelihood is given by:

$$\ell(\psi) = -\frac{T}{2}\log(2\pi) - \sum_{\substack{t\in N(DD)\\ t\in N(WD)}} \log(\sigma_t^D) - \sum_{\substack{t\in N(WW)\\ t\in N(DW)}} \log(\sigma_t^W)$$

$$-\frac{1}{2}\left[\sum_{t\in N(DD)} \left(\frac{S_{i,t}-\mu_t^D}{\sigma_t^D} - \theta^{DD}\frac{S_{i,t-1}-\mu_{t-1}^D}{\sigma_{t-1}^D}\right)^2\right.$$

$$+ \sum_{t\in N(WW)} \left(\frac{S_{i,t}-\mu_t^W}{\sigma_t^W} - \theta^{WW}\frac{S_{i,t-1}-\mu_{t-1}^W}{\sigma_{t-1}^W}\right)^2$$

$$+ \sum_{t\in N(DW)} \left(\frac{S_{i,t}-\mu_t^W}{\sigma_t^W} - \theta^{DW}\frac{S_{i,t-1}-\mu_{t-1}^D}{\sigma_{t-1}^D}\right)^2$$

$$\left.+ \sum_{t\in N(WD)} \left(\frac{S_{i,t}-\mu_t^D}{\sigma_t^D} - \theta^{WD}\frac{S_{i,t-1}-\mu_{t-1}^W}{\sigma_{t-1}^W}\right)^2\right].$$

To obtain maximum likelihood estimates for the parameters, $\ell(\psi)$ is minimized. To minimize $\ell(\psi)$, its first partial derivatives with respect to the parameters are set to zero.

The first partial derivatives with respect to the parameters are given by

$$\frac{\partial \ell(\psi)}{\partial \theta^{DD}} = \sum_{t\in N(DD)} \left(\frac{S_{i,t}-\mu_t^D}{\sigma_t^D} - \theta^{DD}\frac{S_{i,t-1}-\mu_{t-1}^D}{\sigma_{t-1}^D}\right)\left(\frac{S_{i,t-1}-\mu_{t-1}^D}{\sigma_{t-1}^D}\right)$$

$$\frac{\partial \ell(\psi)}{\partial \theta^{WW}} = \sum_{t\in N(WW)} \left(\frac{S_{i,t}-\mu_t^W}{\sigma_t^W} - \theta^{WW}\frac{S_{i,t-1}-\mu_{t-1}^W}{\sigma_{t-1}^W}\right)\left(\frac{S_{i,t-1}-\mu_{t-1}^W}{\sigma_{t-1}^W}\right)$$

$$\frac{\partial \ell(\psi)}{\partial \theta^{DW}} = \sum_{t\in N(DW)} \left(\frac{S_{i,t}-\mu_t^W}{\sigma_t^W} - \theta^{DW}\frac{S_{i,t-1}-\mu_{t-1}^D}{\sigma_{t-1}^D}\right)\left(\frac{S_{i,t-1}-\mu_{t-1}^D}{\sigma_{t-1}^D}\right)$$

$$\frac{\partial \ell(\psi)}{\partial \theta^{WD}} = \sum_{t\in N(WD)} \left(\frac{S_{i,t}-\mu_t^D}{\sigma_t^D} - \theta^{WD}\frac{S_{i,t-1}-\mu_{t-1}^W}{\sigma_{t-1}^W}\right)\left(\frac{S_{i,t-1}-\mu_{t-1}^W}{\sigma_{t-1}^W}\right)$$

$$\begin{aligned}\frac{\partial \ell(\psi)}{\partial \alpha_j^D} = -\Bigg[&\sum_{t\in N(DD)} \left(\frac{S_{i,t}-\mu_t^D}{\sigma_t^D} - \theta^{DD}\frac{S_{i,t-1}-\mu_{t-1}^D}{\sigma_{t-1}^D}\right)\left(\frac{-\varphi_j(t)}{\sigma_t^D} + \frac{\theta^{DD}\varphi_j(t-1)}{\sigma_{t-1}^D}\right)\\
&+ \sum_{t\in N(DW)} \left(\frac{S_{i,t}-\mu_t^W}{\sigma_t^W} - \theta^{DW}\frac{S_{i,t-1}-\mu_{t-1}^D}{\sigma_{t-1}^D}\right)\left(\frac{\theta^{DW}\varphi_j(t-1)}{\sigma_{t-1}^D}\right)\\
&+ \sum_{t\in N(WD)} \left(\frac{S_{i,t}-\mu_t^D}{\sigma_t^D} - \theta^{WD}\frac{S_{i,t-1}-\mu_{t-1}^W}{\sigma_{t-1}^W}\right)\left(\frac{-\varphi_j(t)}{\sigma_t^D}\right)\Bigg]\end{aligned}$$

$$\begin{aligned}\frac{\partial \ell(\psi)}{\partial \alpha_j^W} = -\Bigg[&\sum_{t\in N(WW)} \left(\frac{S_{i,t}-\mu_t^W}{\sigma_t^W} - \theta^{WW}\frac{S_{i,t-1}-\mu_{t-1}^W}{\sigma_{t-1}^W}\right)\\
&\left(\frac{-\varphi_j(t)}{\sigma_t^W} + \frac{\theta^{WW}\varphi_j(t-1)}{\sigma_{t-1}^W}\right)\\
&+ \sum_{t\in N(DW)} \left(\frac{S_{i,t}-\mu_t^W}{\sigma_t^W} - \theta^{DW}\frac{S_{i,t-1}-\mu_{t-1}^D}{\sigma_{t-1}^D}\right)\left(\frac{-\varphi_j(t)}{\sigma_t^W}\right)\\
&+ \sum_{t\in N(WD)} \left(\frac{S_{i,t}-\mu_t^D}{\sigma_t^D} - \theta^{WD}\frac{S_{i,t-1}-\mu_{t-1}^W}{\sigma_{t-1}^W}\right)\left(\frac{\theta^{WD}\varphi_j(t-1)}{\sigma_{t-1}^W}\right)\Bigg]\end{aligned}$$

$$\begin{aligned}\frac{\partial \ell(\psi)}{\partial \xi_j^D} = -\sum_{\substack{t\in N(DD)\\ t\in N(WD)}} \frac{\varphi_j(t)}{\sigma_t^D} - \Bigg[&\sum_{t\in N(DD)} \left(\frac{S_{i,t}-\mu_t^D}{\sigma_t^D} - \theta^{DD}\frac{S_{i,t-1}-\mu_{t-1}^D}{\sigma_{t-1}^D}\right)\\
&\left(-\varphi_j(t)\frac{S_{i,t}-\mu_t^D}{(\sigma_t^D)^2} + \theta^{DD}\varphi_j(t-1)\frac{S_{i,t-1}-\mu_{t-1}^D}{(\sigma_{t-1}^D)^2}\right)\\
&+ \sum_{t\in N(DW)} \left(\frac{S_{i,t}-\mu_t^W}{\sigma_t^W} - \theta^{DW}\frac{S_{i,t-1}-\mu_{t-1}^D}{\sigma_{t-1}^D}\right)\left(\theta^{DW}\varphi_j(t-1)\frac{S_{i,t-1}-\mu_{t-1}^D}{(\sigma_{t-1}^D)^2}\right)\\
&+ \sum_{t\in N(WD)} \left(\frac{S_{i,t}-\mu_t^D}{\sigma_t^D} - \theta^{WD}\frac{S_{i,t-1}-\mu_{t-1}^W}{\sigma_{t-1}^W}\right)\left(-\varphi_j(t)\frac{S_{i,t}-\mu_t^D}{(\sigma_t^D)^2}\right)\Bigg]\end{aligned}$$

$$\frac{\partial \ell(\psi)}{\partial \xi_j^W} = -\sum_{\substack{t\in N(WW)\\ t\in N(DW)}} \frac{\varphi_j(t)}{\sigma_t^W} - \left[\sum_{t\in N(WW)} \left(\frac{S_{i,t}-\mu_t^W}{\sigma_t^W} - \theta^{WW}\frac{S_{i,t-1}-\mu_{t-1}^W}{\sigma_{t-1}^W}\right)\right.$$
$$\left(-\varphi_j(t)\frac{S_{i,t}-\mu_t^W}{(\sigma_t^W)^2} + \theta^{WW}\varphi_j(t-1)\frac{S_{i,t-1}-\mu_{t-1}^W}{(\sigma_{t-1}^W)^2}\right)$$
$$+ \sum_{t\in N(DW)} \left(\frac{S_{i,t}-\mu_t^W}{\sigma_t^W} - \theta^{DW}\frac{S_{i,t-1}-\mu_{t-1}^D}{\sigma_{t-1}^D}\right)\left(-\varphi_j(t)\frac{S_{i,t}-\mu_t^W}{(\sigma_t^W)^2}\right)$$
$$\left.+ \sum_{t\in N(WD)} \left(\frac{S_{i,t}-\mu_t^D}{\sigma_t^D} - \theta^{WD}\frac{S_{i,t-1}-\mu_{t-1}^W}{\sigma_{t-1}^W}\right)\left(\theta^{WD}\varphi_j(t-1)\frac{S_{i,t-1}-\mu_{t-1}^W}{(\sigma_{t-1}^W)^2}\right)\right]$$

The Newton–Raphson method is used to solve the system of equation. For this the second partial derivatives are required and these are given by

$$\frac{\partial^2 \ell(\psi)}{\partial \theta^{DD}\partial \theta^{DD}} = -\sum_{t\in N(DD)} \left(\frac{S_{i,t-1}-\mu_{t-1}^D}{\sigma_{t-1}^D}\right)^2$$

$$\frac{\partial^2 \ell(\psi)}{\partial \theta^{DD}\partial \theta^{WW}} = \frac{\partial^2 \ell(\psi)}{\partial \theta^{DD}\partial \theta^{DW}} = \frac{\partial^2 \ell(\psi)}{\partial \theta^{DD}\partial \theta^{WD}} = 0$$

$$\frac{\partial^2 \ell(\psi)}{\partial \theta^{DD}\partial \alpha_j^D} = \sum_{t\in N(DD)} \left\{\left(\frac{S_{i,t}-\mu_t^D}{\sigma_t^D} - \theta^{DD}\frac{S_{i,t-1}-\mu_{t-1}^D}{\sigma_{t-1}^D}\right)\left(\frac{-\varphi_j(t-1)}{\sigma_{t-1}^D}\right)\right.$$
$$\left.+ \left(\frac{-\varphi_j(t)}{\sigma_t^D} + \frac{\theta^{DD}\varphi_j(t-1)}{\sigma_{t-1}^D}\right)\left(\frac{S_{i,t-1}-\mu_{t-1}^D}{\sigma_{t-1}^D}\right)\right\}$$

$$\frac{\partial^2 \ell(\psi)}{\partial \theta^{DD}\partial \alpha_j^W} = \frac{\partial^2 \ell(\psi)}{\partial \theta^{DD}\partial \xi_j^W} = 0$$

$$\frac{\partial^2 \ell(\psi)}{\partial \theta^{DD}\partial \xi_j^D} = \sum_{t\in N(DD)} \left\{\left(\frac{S_{i,t}-\mu_t^D}{\sigma_t^D} - \theta^{DD}\frac{S_{i,t-1}-\mu_{t-1}^D}{\sigma_{t-1}^D}\right)\left(-\varphi_j(t-1)\frac{S_{i,t-1}-\mu_{t-1}^D}{(\sigma_{t-1}^D)^2}\right)\right.$$
$$\left.+ \left(-\varphi_j(t)\frac{S_{i,t}-\mu_t^D}{(\sigma_t^D)^2} + \theta^{DD}\varphi_j(t-1)\frac{S_{i,t-1}-\mu_{t-1}^D}{(\sigma_{t-1}^D)^2}\right)\left(\frac{S_{i,t-1}-\mu_{t-1}^D}{\sigma_{t-1}^D}\right)\right\}$$

$$\frac{\partial^2 \ell(\psi)}{\partial \theta^{WW}\partial \theta^{WW}} = -\sum_{t\in N(WW)} \left(\frac{S_{i,t-1}-\mu_{t-1}^W}{\sigma_{t-1}^W}\right)^2$$

$$\frac{\partial^2 \ell(\psi)}{\partial \theta^{WW}\partial \theta^{DW}} = \frac{\partial^2 \ell(\psi)}{\partial \theta^{WW}\partial \theta^{WD}} = \frac{\partial^2 \ell(\psi)}{\partial \theta^{WW}\partial \alpha_j^D} = 0$$

$$\frac{\partial^2 \ell(\psi)}{\partial \theta^{WW}\partial \alpha_j^W} = \sum_{t\in N(WW)} \left\{\left(\frac{S_{i,t}-\mu_t^W}{\sigma_t^W} - \theta^{WW}\frac{S_{i,t-1}-\mu_{t-1}^W}{\sigma_{t-1}^W}\right)\left(\frac{-\varphi_j(t-1)}{\sigma_{t-1}^W}\right)\right.$$
$$\left.+ \left(\frac{-\varphi_j(t)}{\sigma_t^W} + \frac{\theta^{WW}\varphi_j(t-1)}{\sigma_{t-1}^W}\right)\left(\frac{S_{i,t-1}-\mu_{t-1}^W}{\sigma_{t-1}^W}\right)\right\}$$

$$\frac{\partial^2 \ell(\psi)}{\partial \theta^{WW}\partial \xi_j^D} = 0$$

$$\frac{\partial^2 \ell(\psi)}{\partial \theta^{WW} \partial \xi_j^W} = \sum_{t \in N(WW)} \left\{ \left(\frac{S_{i,t} - \mu_t^W}{\sigma_t^W} - \theta^{WW} \frac{S_{i,t-1} - \mu_{t-1}^W}{\sigma_{t-1}^W} \right) \left(-\varphi_j(t-1) \frac{S_{i,t-1} - \mu_{t-1}^W}{(\sigma_{t-1}^W)^2} \right) + \left(-\varphi_j(t) \frac{S_{i,t} - \mu_t^W}{(\sigma_t^W)^2} + \theta^{WW} \varphi_j(t-1) \frac{S_{i,t-1} - \mu_{t-1}^W}{(\sigma_{t-1}^W)^2} \right) \left(\frac{S_{i,t-1} - \mu_{t-1}^W}{\sigma_{t-1}^W} \right) \right\}$$

$$\frac{\partial^2 \ell(\psi)}{\partial \theta^{DW} \partial \theta^{DW}} = - \sum_{t \in N(DW)} \left(\frac{S_{i,t-1} - \mu_{t-1}^D}{\sigma_{t-1}^D} \right)^2$$

$$\frac{\partial^2 \ell(\psi)}{\partial \theta^{DW} \partial \theta^{WD}} = 0$$

$$\frac{\partial^2 \ell(\psi)}{\partial \theta^{DW} \partial \alpha_j^D} = \sum_{t \in N(DW)} \left\{ \left(\frac{S_{i,t} - \mu_t^W}{\sigma_t^W} - \theta^{DW} \frac{S_{i,t-1} - \mu_{t-1}^D}{\sigma_{t-1}^D} \right) \left(\frac{-\varphi_j(t-1)}{\sigma_{t-1}^D} \right) + \left(\frac{\theta^{DW} \varphi_j(t-1)}{\sigma_{t-1}^D} \right) \left(\frac{S_{i,t-1} - \mu_{t-1}^D}{\sigma_{t-1}^D} \right) \right\}$$

$$\frac{\partial^2 \ell(\psi)}{\partial \theta^{DW} \partial \alpha_j^W} = \sum_{t \in N(DW)} \left(\frac{-\varphi_j(t)}{\sigma_t^W} \right) \left(\frac{S_{i,t-1} - \mu_{t-1}^D}{\sigma_{t-1}^D} \right)$$

$$\frac{\partial^2 \ell(\psi)}{\partial \theta^{DW} \partial \xi_j^D} = \sum_{t \in N(DW)} \left\{ \left(\frac{S_{i,t} - \mu_t^W}{\sigma_t^W} - \theta^{DW} \frac{S_{i,t-1} - \mu_{t-1}^D}{\sigma_{t-1}^D} \right) \left(-\varphi_j(t-1) \frac{S_{i,t-1} - \mu_{t-1}^D}{(\sigma_{t-1}^D)^2} \right) + \left(\theta^{DW} \varphi_j(t-1) \frac{S_{i,t-1} - \mu_{t-1}^D}{(\sigma_{t-1}^D)^2} \right) \left(\frac{S_{i,t-1} - \mu_{t-1}^D}{\sigma_{t-1}^D} \right) \right\}$$

$$\frac{\partial^2 \ell(\psi)}{\partial \theta^{DW} \partial \xi_j^W} = \sum_{t \in N(DW)} \left(-\varphi_j(t) \frac{S_{i,t} - \mu_t^W}{\sigma_t^W} \right) \left(\frac{S_{i,t-1} - \mu_{t-1}^D}{\sigma_{t-1}^D} \right)$$

$$\frac{\partial^2 \ell(\psi)}{\partial \theta^{WD} \partial \theta^{WD}} = - \sum_{t \in N(WD)} \left(\frac{S_{i,t-1} - \mu_{t-1}^W}{\sigma_{t-1}^W} \right)^2$$

$$\frac{\partial^2 \ell(\psi)}{\partial \theta^{WD} \partial \alpha_j^D} = \sum_{t \in N(WD)} \left(\frac{-\varphi_j(t)}{\sigma_t^D} \right) \left(\frac{S_{i,t-1} - \mu_{t-1}^W}{\sigma_{t-1}^W} \right)$$

$$\frac{\partial^2 \ell(\psi)}{\partial \theta^{WD} \partial \alpha_j^W} = \sum_{t \in N(WD)} \left\{ \left(\frac{S_{i,t} - \mu_t^D}{\sigma_t^D} - \theta^{WD} \frac{S_{i,t-1} - \mu_{t-1}^W}{\sigma_{t-1}^W} \right) \left(\frac{-\varphi_j(t-1)}{\sigma_{t-1}^W} \right) + \left(\frac{\theta^{WD} \varphi_j(t-1)}{\sigma_{t-1}^W} \right) \left(\frac{S_{i,t-1} - \mu_{t-1}^W}{\sigma_{t-1}^W} \right) \right\}$$

$$\frac{\partial^2 \ell(\psi)}{\partial \theta^{WD} \partial \xi_j^D} = \sum_{t \in N(WD)} \left(-\varphi_j(t) \frac{S_{i,t} - \mu_t^D}{(\sigma_t^D)^2} \right) \left(\frac{S_{i,t-1} - \mu_{t-1}^W}{\sigma_{t-1}^W} \right)$$

$$\frac{\partial^2 \ell(\psi)}{\partial \theta^{WD} \partial \xi_j^W} = \sum_{t \in N(WD)} \left\{ \left(\frac{S_{i,t} - \mu_t^D}{\sigma_t^D} - \theta^{WD} \frac{S_{i,t-1} - \mu_{t-1}^W}{\sigma_{t-1}^W} \right) \left(-\varphi_j(t-1) \frac{S_{i,t-1} - \mu_{t-1}^W}{(\sigma_{t-1}^W)^2} \right) + \left(\theta^{WD} \varphi_j(t-1) \frac{S_{i,t-1} - \mu_{t-1}^W}{(\sigma_{t-1}^W)^2} \right) \left(\frac{S_{i,t-1} - \mu_{t-1}^W}{\sigma_{t-1}^W} \right) \right\}$$

$$\frac{\partial^2 \ell(\psi)}{\partial \alpha_j^D \partial \alpha_k^D} = -\left[\sum_{t\in N(DD)} \left(\frac{-\varphi_j(t)}{\sigma_t^D} + \frac{\theta^{DD}\varphi_j(t-1)}{\sigma_{t-1}^D}\right)\left(\frac{-\varphi_k(t)}{\sigma_t^D} + \frac{\theta^{DD}\varphi_k(t-1)}{\sigma_{t-1}^D}\right) + \sum_{t\in N(DW)} \left(\frac{\theta^{DW}\varphi_j(t-1)}{\sigma_{t-1}^D}\right)\left(\frac{\theta^{DW}\varphi_k(t-1)}{\sigma_{t-1}^D}\right) + \sum_{t\in N(WD)} \left(\frac{-\varphi_k(t)}{\sigma_t^D}\right)\left(\frac{-\varphi_j(t)}{\sigma_t^D}\right)\right]$$

$$\frac{\partial^2 \ell(\psi)}{\partial \alpha_j^D \partial \alpha_k^W} = -\left[\sum_{t\in N(DW)} \left(\frac{-\varphi_k(t)}{\sigma_t^W}\right)\left(\frac{\theta^{DW}\varphi_j(t-1)}{\sigma_{t-1}^D}\right) + \sum_{t\in N(WD)} \left(\frac{\theta^{WD}\varphi_k(t-1)}{\sigma_{t-1}^W}\right)\left(\frac{-\varphi_j(t)}{\sigma_t^D}\right)\right]$$

$$\begin{aligned}
\frac{\partial^2 \ell(\psi)}{\partial \alpha_j^D \partial \xi_k^D} = -\Bigg[&\sum_{t\in N(DD)} \Bigg\{\left(\frac{S_{i,t}-\mu_t^D}{\sigma_t^D} - \theta^{DD}\frac{S_{i,t-1}-\mu_{t-1}^D}{\sigma_{t-1}^D}\right) \\
&\left(\frac{\varphi_j(t)\varphi_k(t)}{(\sigma_t^D)^2} - \frac{\theta^{DD}\varphi_j(t-1)\varphi_k(t-1)}{(\sigma_{t-1}^D)^2}\right) \\
&+ \left(\frac{-\varphi_j(t)}{\sigma_t^D} + \frac{\theta^{DD}\varphi_j(t-1)}{\sigma_{t-1}^D}\right)\left(-\varphi_k(t)\frac{S_{i,t}-\mu_t^D}{(\sigma_t^D)^2}\right. \\
&\left.+\theta^{DD}\varphi_k(t-1)\frac{S_{i,t-1}-\mu_{t-1}^D}{(\sigma_{t-1}^D)^2}\right)\Bigg\} \\
&+ \sum_{t\in N(DW)} \Bigg\{\left(\frac{S_{i,t}-\mu_t^W}{\sigma_t^W} - \theta^{DW}\frac{S_{i,t-1}-\mu_{t-1}^D}{\sigma_{t-1}^D}\right)\left(\frac{-\theta^{DW}\varphi_k(t-1)\varphi_j(t-1)}{(\sigma_{t-1}^D)^2}\right) \\
&+ \left(\frac{\theta^{DW}\varphi_j(t-1)}{\sigma_{t-1}^D}\right)\left(\theta^{DW}\varphi_k(t-1)\frac{S_{i,t-1}-\mu_{t-1}^D}{(\sigma_{t-1}^D)^2}\right)\Bigg\} \\
&+ \sum_{t\in N(WD)} \Bigg\{\left(\frac{S_{i,t}-\mu_t^D}{\sigma_t^D} - \theta^{WD}\frac{S_{i,t-1}-\mu_{t-1}^W}{\sigma_{t-1}^W}\right)\left(\frac{\varphi_j(t)\varphi_k(t)}{(\sigma_t^D)^2}\right) \\
&+ \left(\frac{-\varphi_j(t)}{\sigma_t^D}\right)\left(-\varphi_k(t)\frac{S_{i,t}-\mu_t^D}{(\sigma_t^D)^2}\right)\Bigg\}\Bigg]
\end{aligned}$$

$$\frac{\partial^2 \ell(\psi)}{\partial \alpha_j^D \partial \xi_k^W} = -\left[\sum_{t\in N(DW)} \left(\frac{\theta^{DW}\varphi_j(t-1)}{\sigma_{t-1}^D}\right)\left(-\varphi_k(t)\frac{S_{i,t}-\mu_t^W}{(\sigma_t^W)^2}\right) + \sum_{t\in N(WD)} \left(\frac{-\varphi_j(t)}{\sigma_t^D}\right)\left(\theta^{WD}\varphi_k(t-1)\frac{S_{i,t-1}-\mu_{t-1}^W}{(\sigma_{t-1}^W)^2}\right)\right]$$

$$\frac{\partial^2 \ell(\psi)}{\partial \alpha_j^W \partial \alpha_k^W} = -\left[\sum_{t\in N(WW)} \left(\frac{-\varphi_k(t)}{\sigma_t^W} + \frac{\theta^{WW}\varphi_k(t-1)}{\sigma_{t-1}^W}\right)\right.$$
$$\left(\frac{-\varphi_j(t)}{\sigma_t^W} + \frac{\theta^{WW}\varphi_j(t-1)}{\sigma_{t-1}^W}\right)$$
$$+ \sum_{t\in N(DW)} \left(\frac{-\varphi_k(t)}{\sigma_t^W}\right)\left(\frac{-\varphi_j(t)}{\sigma_t^W}\right)$$
$$\left. + \sum_{t\in N(WD)} \left(\frac{\theta^{WD}\varphi_k(t-1)}{\sigma_{t-1}^W}\right)\left(\frac{\theta^{WD}\varphi_j(t-1)}{\sigma_{t-1}^W}\right)\right]$$

$$\frac{\partial^2 \ell(\psi)}{\partial \alpha_j^W \partial \xi_k^D} = -\left[\sum_{t\in N(DW)} \left(\frac{-\varphi_j(t)}{\sigma_t^W}\right)\left(\theta^{DW}\varphi_k(t-1)\frac{S_{i,t-1}-\mu_{t-1}^D}{(\sigma_{t-1}^D)^2}\right)\right.$$
$$\left. + \sum_{t\in N(WD)} \left(\frac{\theta^{WD}\varphi_j(t-1)}{\sigma_{t-1}^W}\right)\left(-\varphi_k(t)\frac{S_{i,t}-\mu_t^D}{(\sigma_t^D)^2}\right)\right]$$

$$\frac{\partial^2 \ell(\psi)}{\partial \alpha_j^W \partial \xi_k^W} = -\left[\sum_{t\in N(WW)} \left\{\left(\frac{S_{i,t}-\mu_t^W}{\sigma_t^W} - \theta^{WW}\frac{S_{i,t-1}-\mu_{t-1}^W}{\sigma_{t-1}^W}\right)\right.\right.$$
$$\left(\frac{\varphi_j(t)\varphi_k(t)}{(\sigma_t^W)^2} - \frac{\theta^{WW}\varphi_j(t-1)\varphi_k(t-1)}{(\sigma_{t-1}^W)^2}\right)$$
$$+ \left(\frac{-\varphi_j(t)}{\sigma_t^W} + \frac{\theta^{WW}\varphi_j(t-1)}{\sigma_{t-1}^W}\right)$$
$$\left.\left(-\varphi_k(t)\frac{S_{i,t}-\mu_t^W}{(\sigma_t^W)^2} + \theta^{WW}\varphi_k(t-1)\frac{S_{i,t-1}-\mu_{t-1}^W}{(\sigma_{t-1}^W)^2}\right)\right\}$$
$$+ \sum_{t\in N(DW)} \left\{\left(\frac{S_{i,t}-\mu_t^W}{\sigma_t^W} - \theta^{DW}\frac{S_{i,t-1}-\mu_{t-1}^D}{\sigma_{t-1}^D}\right)\left(\frac{\varphi_j(t)\varphi_k(t)}{(\sigma_t^W)^2}\right)\right.$$
$$\left. + \left(\frac{-\varphi_j(t)}{\sigma_t^W}\right)\left(-\varphi_k(t)\frac{S_{i,t}-\mu_t^W}{(\sigma_t^W)^2}\right)\right\}$$
$$+ \sum_{t\in N(WD)} \left\{\left(\frac{S_{i,t}-\mu_t^D}{\sigma_t^D} - \theta^{WD}\frac{S_{i,t-1}-\mu_{t-1}^W}{\sigma_{t-1}^W}\right)\left(\frac{-\theta^{WD}\varphi_j(t-1)\varphi_k(t-1)}{(\sigma_{t-1}^W)^2}\right)\right.$$
$$\left.\left. + \left(\frac{\theta^{WD}\varphi_j(t-1)}{\sigma_{t-1}^W}\right)\left(\theta^{WD}\varphi_k(t-1)\frac{S_{i,t-1}-\mu_{t-1}^W}{(\sigma_{t-1}^W)^2}\right)\right\}\right]$$

$$
\begin{aligned}
\frac{\partial^2 \ell(\psi)}{\partial \xi_j^D \partial \xi_k^D} = & \sum_{\substack{t \in N(DD) \\ t \in N(WD)}} \frac{\varphi_k(t)\varphi_j(t)}{(\sigma_t^D)^2} - \Bigg[\sum_{t \in N(DD)} \Bigg\{ \left(\frac{S_{i,t} - \mu_t^D}{\sigma_t^D} - \theta^{DD} \frac{S_{i,t-1} - \mu_{t-1}^D}{\sigma_{t-1}^D} \right) \\
& \left(2\varphi_j(t)\varphi_k(t) \frac{S_{i,t} - \mu_t^D}{(\sigma_t^D)^3} - 2\theta^{DD} \varphi_j(t-1)\varphi_k(t-1) \frac{S_{i,t-1} - \mu_{t-1}^D}{(\sigma_{t-1}^D)^3} \right) \\
& + \left(-\varphi_j(t) \frac{S_{i,t} - \mu_t^D}{(\sigma_t^D)^2} + \theta^{DD} \varphi_j(t-1) \frac{S_{i,t-1} - \mu_{t-1}^D}{(\sigma_{t-1}^D)^2} \right) \\
& \left(-\varphi_k(t) \frac{S_{i,t} - \mu_t^D}{(\sigma_t^D)^2} + \theta^{DD} \varphi_k(t-1) \frac{S_{i,t-1} - \mu_{t-1}^D}{(\sigma_{t-1}^D)^2} \right) \Bigg\} \\
& + \sum_{t \in N(DW)} \Bigg\{ \left(\frac{S_{i,t} - \mu_t^W}{\sigma_t^W} - \theta^{DW} \frac{S_{i,t-1} - \mu_{t-1}^D}{\sigma_{t-1}^D} \right) \\
& \left(-2\theta^{DW} \varphi_j(t-1)\varphi_k(t-1) \frac{S_{i,t-1} - \mu_{t-1}^D}{(\sigma_{t-1}^D)^3} \right) \\
& + \left(\theta^{DW} \varphi_j(t-1) \frac{S_{i,t-1} - \mu_{t-1}^D}{(\sigma_{t-1}^D)^2} \right) \left(\theta^{DW} \varphi_k(t-1) \frac{S_{i,t-1} - \mu_{t-1}^D}{(\sigma_{t-1}^D)^2} \right) \Bigg\} \\
& + \sum_{t \in N(WD)} \Bigg\{ \left(\frac{S_{i,t} - \mu_t^D}{\sigma_t^D} - \theta^{WD} \frac{S_{i,t-1} - \mu_{t-1}^W}{\sigma_{t-1}^W} \right) \left(2\varphi_j(t)\varphi_k(t) \frac{S_{i,t} - \mu_t^D}{(\sigma_t^D)^3} \right) \\
& + \left(-\varphi_j(t) \frac{S_{i,t} - \mu_t^D}{(\sigma_t^D)^2} \right) \left(-\varphi_k(t) \frac{S_{i,t} - \mu_t^D}{(\sigma_t^D)^2} \right) \Bigg\} \Bigg]
\end{aligned}
$$

$$\frac{\partial^2 \ell(\psi)}{\partial \xi_j^D \partial \xi_k^W} = -\left[\sum_{t\in N(DW)} \left(\theta^{DW}\varphi_j(t-1)\frac{S_{i,t-1}-\mu_{t-1}^D}{(\sigma_{t-1}^D)^2}\right)\left(-\varphi_k(t)\frac{S_{i,t}-\mu_t^W}{(\sigma_t^W)^2}\right)\right.$$
$$\left. + \sum_{t\in N(WD)} \left(-\varphi_j(t)\frac{S_{i,t}-\mu_t^D}{(\sigma_t^D)^2}\right)\left(\theta^{WD}\varphi_k(t-1)\frac{S_{i,t-1}-\mu_{t-1}^W}{(\sigma_{t-1}^W)^2}\right)\right]$$

$$\frac{\partial^2 \ell(\psi)}{\partial \xi_j^W \partial \xi_k^W} = \sum_{\substack{t\in N(WW)\\ t\in N(DW)}} \frac{\varphi_j(t)\varphi_k(t)}{(\sigma_t^W)^2}$$
$$-\left[\sum_{t\in N(WW)}\left\{\left(\frac{S_{i,t}-\mu_t^W}{\sigma_t^W} - \theta^{WW}\frac{S_{i,t-1}-\mu_{t-1}^W}{\sigma_{t-1}^W}\right)\right.\right.$$
$$\left(2\varphi_j(t)\varphi_k(t)\frac{S_{i,t}-\mu_t^W}{(\sigma_t^W)^3} - 2\theta^{WW}\varphi_j(t-1)\varphi_k(t-1)\frac{S_{i,t-1}-\mu_{t-1}^W}{(\sigma_{t-1}^W)^3}\right)$$
$$+\left(-\varphi_j(t)\frac{S_{i,t}-\mu_t^W}{(\sigma_t^W)^2} + \theta^{WW}\varphi_j(t-1)\frac{S_{i,t-1}-\mu_{t-1}^W}{(\sigma_{t-1}^W)^2}\right)$$
$$\left.\left(-\varphi_k(t)\frac{S_{i,t}-\mu_t^W}{(\sigma_t^W)^2} + \theta^{WW}\varphi_k(t-1)\frac{S_{i,t-1}-\mu_{t-1}^W}{(\sigma_{t-1}^W)^2}\right)\right\}$$
$$+\sum_{t\in N(DW)}\left\{\left(\frac{S_{i,t}-\mu_t^W}{\sigma_t^W} - \theta^{WW}\frac{S_{i,t-1}-\mu_{t-1}^D}{\sigma_{t-1}^D}\right)\left(2\varphi_j(t)\varphi_k(t)\frac{S_{i,t}-\mu_t^W}{(\sigma_t^W)^3}\right)\right.$$
$$\left.+\left(-\varphi_j(t)\frac{S_{i,t}-\mu_t^W}{(\sigma_t^W)^2}\right)\left(-\varphi_k(t)\frac{S_{i,t}-\mu_t^W}{(\sigma_t^W)^2}\right)\right\}$$
$$+\sum_{t\in N(WD)}\left\{\left(\frac{S_{i,t}-\mu_t^D}{\sigma_t^D} - \theta^{WD}\frac{S_{i,t-1}-\mu_{t-1}^W}{\sigma_{t-1}^W}\right)\right.$$
$$\left(-2\theta^{WD}\varphi_j(t-1)\varphi_k(t-1)\frac{S_{i,t-1}-\mu_{t-1}^W}{(\sigma_{t-1}^W)^3}\right)$$
$$\left.\left.+\left(\theta^{WD}\varphi_j(t-1)\frac{S_{i,t-1}-\mu_{t-1}^W}{(\sigma_{t-1}^W)^2}\right)\left(\theta^{WD}\varphi_k(t-1)\frac{S_{i,t-1}-\mu_{t-1}^W}{(\sigma_{t-1}^W)^2}\right)\right\}\right]$$

The following algorithm is used to estimate the parameters

Algorithm

Step 1: Estimate initial $\widehat{\mu}_t$ and $\widehat{\sigma}_t$ by approximating by its Fourier series representation and estimating the parameters α_i and ξ_i by the method mentioned in the previous models.

Step 2: Estimate initial $\widehat{\theta}^{DD}, \widehat{\theta}^{WW}, \widehat{\theta}^{DW}$ and $\widehat{\theta}^{WD}$ using the following formula:

$$\widehat{\theta}^{DD} = \frac{\sum_{t\in N(DD)}(S_{i,t}-\widehat{\mu}_t^D)(S_{i,t-1}-\widehat{\mu}_{t-1}^D)}{\sum_{t\in N(DD)}(S_{i,t-1}-\widehat{\mu}_{t-1}^D)^2}$$

Similarly for $\widehat{\theta}^{WW}, \widehat{\theta}^{DW}$ and $\widehat{\theta}^{WD}$.

Step 3: Compute $f^{(k)}$ and $F^{(k)}$, where $f^{(k)}$ is the vector of first partial derivatives and $F^{(k)}$ is the matrix of second partial derivatives, computed at the kth iteration.

Step 4: Compute the vector $\delta^{(k)}$ which is the solution to the system of NP linear equations

$$F^{(k)}\delta^{(k)} = f^{(k)}$$

where NP represents the number of parameters.

Step 5: Set $\beta^{(k+1)} = \beta^{(k)} - \delta^{(k)}$, where $\beta^{(k)}$ contains the parameter estimates at the kth iteration.

Step 6: Test for convergence, for example, if the elements of $f^{(k)}$ are sufficiently close to zero. If the convergence criterion is met then stop, otherwise increase k by 1 and return to step 3.

The cross–correlation matrix, $\hat{\Sigma}$, has elements given by

$$R_{jk} = \frac{\frac{1}{T}\sum_{i=1}^{NY}\sum_{t=1}^{NT} e_{i,t}^{(j)} e_{i,t}^{(k)} - \frac{1}{T^2}\sum_{i=1}^{NY}\sum_{t=1}^{NT} e_{i,t}^{(j)} \sum_{i=1}^{NY}\sum_{t=1}^{NT} e_{i,t}^{(k)}}{\left[\frac{1}{T}\sum_{i=1}^{NY}\sum_{t=1}^{NT}(e_{i,t}^{(j)})^2 - \frac{1}{T^2}\left(\sum_{i=1}^{NY}\sum_{t=1}^{NT} e_{i,t}^{(j)}\right)^2\right]^{\frac{1}{2}} \left[\frac{1}{T}\sum_{i=1}^{NY}\sum_{t=1}^{NT}(e_{i,t}^{(k)})^2 - \frac{1}{T^2}\left(\sum_{i=1}^{NY}\sum_{t=1}^{NT} e_{i,t}^{(k)}\right)^2\right]^{\frac{1}{2}}}$$

where $e_{i,t}^{(j)}$ denotes the residual time series of variable j, $j = 1, 2, \ldots, NV$.
and $e_{i,t}^{(k)}$ denotes the residual time series of variable k, $k = 1, 2, \ldots, NV$.

(d) Model Selection

The order of the autoregressive process is selected in the same way as in the previous models as is the order of the Fourier series approximation.

CHAPTER 4

MODEL IMPLEMENTATION

This chapter gives details of the implementation of the proposed time series models to describe historical climate series. In particular, the model selection process is described step by step, the parameter estimates are given and the results of tests to check the model assumptions are discussed.

Six stations were chosen for study which broadly represent the various climate regions of South Africa. Table 4.1 lists them together with the years for which simultaneous observations of all climate variables were recorded. The stations marked with an asterisk indicate those stations for which the variable evaporation was not available.

Considerable difficulties were experienced in obtaining suitable data sets for model implementation. This refers to problems in obtaining stations for which all the climate variables of interest are recorded as well as to the quantity and the quality of the available data. Thus, one is restricted by the stations one can fit the climate models to, and the quantity and quality of the historical records determines the performance of the models. As already mentioned, the models are sensitive to "unlcean" data records and relatively short historical records lead to three problems. Firstly, the precision of the estimates decreases as a large number of parameters are estimated using very few data values. Secondly, the effective record length for the conditioned estimates is further reduced as the models separate the sequences into wet and dry sequences. Thirdly, the fact that the record length of the stations are quite small, combined with the fact that there are missing observations in the records means that the historical data might not wholly be representative of the long term climate for that particular location.

Since rainfall was considered to be the primary variable and all other variables are conditioned on whether a given day was wet or dry, it was modelled independently of all other variables.

Simple Markov chain to describe the occurrence of wet and dry sequences of days.

The logit transformation of the probabilities $\pi(t)$, $t = 1, 2, \ldots, NT$ is given by

$$\lambda(t) = \log\left(\frac{\pi(t)}{1-\pi(t)}\right)$$

Table 4.1 Climate Stations

Station	Province	Years available
Elsenburg	Cape	1979–1984
Kakamas	Cape	1975–1986
Middelburg	Cape	1977–1986
Nelspruit*	Transvaal	1981–1987
Cedara*	Natal	1980–1989
Hoopstad*	Orange Free State	1981–1989

where $\lambda(t)$ is represented by a Fourier series approximation, i.e.

$$\lambda(t) = \sum_{i=1}^{L} \gamma_i \phi_i(t), \qquad t = 1, 2, \ldots, NT$$

and $\phi_i(t)$ is defined as in Chapter 3.

The parameters γ_i have to be estimated for the probability that a wet day is preceded by a wet day $(P(R|R))$ and for the probability that a wet day is preceded by a dry day $(P(R|\overline{R}))$. For each of the probabilities the order of the Fourier series approximation, L, has to be selected.

The selection of the appropriate L was based on Akaike's Information Criterion, where

$$AIC = -\ell(\gamma; M(t)) + L$$

where $\ell(\gamma; M(t))$ is the log likelihood function of a particular model. The criterion is computed for $L = 1, 3, 5, \ldots$ and the model which leads to the smallest value of the criterion is selected.

Table 4.2 gives the optimal number of parameters for $P(R|R)$ and $P(R|\overline{R})$. The values of L for $P(R|R)$ and $P(R|\overline{R})$ ranged between 1 and 3 and between 1 and 5 respectively, with modes 3 and 5. A choice of 3 parameters for both models was decided upon for the following reasons. Firstly, the method of model selection employed here is less stringent than conventional tests of hypotheses, and therefore generally leads to a selection of more parameters. Thus a choice of 3 parameters would be preferable to 5. Secondly, the length of the historical record plays a role in determining α and it must be kept in mind that the results here have been obtained with a relatively small data set, thus making the selection of fewer parameters inevitable. Zucchini and Adamson (1984a) chose 5 parameters for both models, but their data sets (typically 40 years) were large enough to warrant that number of parameters.

TABLE 4.2 Optimal number of parameters to estimate $P(R|R)$ and $P(R|\overline{R})$

Station	Model	
	$P(R\|R)$	$P(R\|\overline{R})$
Elsenburg	3	5
Kakamas	–	1
Middelburg	1	3
Nelspruit	3	5
Cedara	3	5
Hoopstad	1	3

The station Kakamas presented a problem in obtaining convergence when estimating the parameters for $P(R|R)$. This can be explained by the rare occurrence of rainfall, and in particular that of consecutive days of rainfall in Kakamas. Moreover, the few years of records available for estimation intensify this problem. That is, when preparing the array $NRR(t)$ required for parameter estimation, where $NRR(t)$ represents the number of times it was wet in period $t-1$ and wet in period t, most of the entries are zero and therefore there are very few values on which to compute parameter estimates leading to difficulties in achieving convergence.

Zucchini and Adamson (1984b) have computed parameter estimates for Kakamas, and as rainfall is modelled independently of the other climate variables, these estimates were used.

It is important to note that the readings were recorded by multiplying each value by ten, i.e. a record of 10.2 is given as 102. This convention was used throughout the study and applied to all results given in this report with the exceptions indicated below. This does not affect the generation of climate sequences which can be easily converted to the original units by dividing by ten. The only exceptions to this were the variables wind run, maximum humidity and minimum humidity for the stations Nelspruit, Cedara and Hoopstad.

The parameter estimates for the probability that a wet day follows a wet day and that a wet day follows a dry day are given in Table 4.3.

The distribution for rainfall on days when rain occurs.

The mean rainfall per rainy day in period $t, \mu(t)$, can be approximated by its truncated Fourier series representation

$$\mu(t) = \sum_{i=1}^{L} \mu_i \phi_i(t), \qquad t = 1, 2, \ldots, NT$$

where $\phi_i(t)$ is defined as in Chapter 3.

The parameters μ_i need to be estimated and the order of the Fourier series approximation selected. A 3-term Fourier series approximation was chosen following arguments similar to those in the previous section.

Table 4.4 shows the parameter estimates for mean rainfall and the estimate for the coefficient of variation. It is sometimes easier to work with the Fourier series coefficients in their polar form, therefore the amplitude and phase representation of the mean rainfall is also given. From these parameter estimates, parameters of the corresponding Weibull distribution can then be estimated by the method of moments (see Zucchini and Adamson 1984a)).

MODEL FOR CLIMATE SEQUENCES

Transforming the data set

Preliminary work carried out to asssess the feasibility of modelling climate on a daily basis highlighted some weaknesses in the models. Firstly, although the models satisfactorily preserved the mean and standard deviation, they failed to preserve the extreme values. This problem arises because some climate variables lie within permissible boundaries with some variables having a high frequency of values near or on an upper or lower limit so that it is expected that simulated sequences will occasionally have values that exceed these boundaries. Secondly, some minimum temperature values were slightly higher than the corresponding maximum temperature value. The same occurred with the humidity variable.

The problem that generated values fall outside their respective admissible range can of course be easily overcome by simply setting the generated values to the appropriate boundary

TABLE 4.3 Parameter estimates for $P(R|R)$ and $P(R|\overline{R})$

Station	Variable	Parameters			Polar Form				
		$\hat{\gamma}_1$	$\hat{\gamma}_2$	$\hat{\gamma}_3$	amplitudes			phases	
					(0)	(1)	(2)	(1)	(2)
Elsenburg	$P(R\|R)$	-0.143	-0.398	-0.148	-0.143	0.425		203.19	
	$P(R\|\overline{R})$	-1.593	-0.487	-0.249	-1.593	0.547		210.00	
Kakamas	$P(R\|R)$				-1.194	0.241	0.221	106.59	84.75
	$P(R\|\overline{R})$				-3.367	0.810	0.321	51.69	92.34
Middelburg	$P(R\|R)$	-0.281	0.175	0.032	-0.281	0.178		10.57	
	$P(R\|\overline{R})$	-2.054	0.558	0.195	-2.054	0.591		19.54	
Nelspruit	$P(R\|R)$	-0.204	0.391	-0.133	-0.204	0.413		345.94	
	$P(R\|\overline{R})$	-1.567	1.294	-0.037	-1.567	1.295		363.34	
Cedara	$P(R\|R)$	0.293	0.918	-0.180	0.293	0.935		353.76	
	$P(R\|\overline{R})$	-0.888	1.488	-0.139	-0.888	1.494		359.61	
Hoopstad	$P(R\|R)$	-0.192	0.251	-0.017	-0.192	0.252		361.03	
	$P(R\|\overline{R})$	-1.927	1.201	0.190	-1.927	1.216		9.11	

value whenever they fall outside the range. Such a procedure is easy to implement but it does change the parameter functions of the generated process (for example the mean), unless the percentage of such points is quite small in which case the resultant bias will be small. Alternatively, one can transform the data. The transformation used ensures that the generated climate sequences lie within the admissible regions and that maximum temperature/humidity values will be greater than minimum temperature/humidity, while the characteristics displayed by the climate series remain unchanged. No transformation was performed on variables which the models described adequately.

TABLE 4.4 Parameter estimates for the distribution of rainfall on days when rain occurs

Station	Variable	Parameters $\widehat{\mu}_1$	$\widehat{\mu}_2$	$\widehat{\mu}_3$	Polar Form amplitudes (0)	(1)	(2)	phases (1)	(2)
Elsenburg	mean	64.73	-12.65	14.64	64.73	19.35		132.66	
	coeff.var	1.2216							
Kakamas	mean				62.56	20.86	2.60	30.99	128.25
	coeff.var	1.0637							
Middelburg	mean	52.94	17.85	-1.01	52.94	17.88		361.72	
	coeff.var	1.3688							
Nelspruit	mean	64.63	20.35	6.34	64.63	21.31		17.54	
	coeff.var	1.5305							
Cedara	mean	53.98	5.01	-0.21	53.98	5.01		362.60	
	coeff.var	2.1594							
Hoopstad	mean	63.91	0.45	-4.94	63.91	4.96		279.01	
	coeff.var	1.4534							

The general transformation used is of the form

$$V_{TF} = \log\left(\frac{a - V_{NTF}}{V_{NTF} - b}\right)$$

where a is the upper bound of the variable and b is the lower bound. V_{TF} represents the variable in its transformed state and V_{NTF} represents the variable in its original form.

The models are then implemented on the transformed time series. The simulated sequences are easily changed back to the original units by reversing the transformation, that is

$$S_{NTF} = \frac{a + b\, e^{S_{TF}}}{e^{S_{TF}} + 1}$$

where S_{NTF} represents the simulated series in the original form and S_{TF} represents the simulated series in a transformed state.

The above transformation has the property that

$$a > S_{NTF} > b.$$

By a suitable choice of a and b, one can prevent maximum temperature being less than minimum temperature. Similarly for humidity. For example, specifying $a = \text{max temp}_{NTF}$ when transforming minimum temperature (i.e. condition minimum temperature on maximum temperature), one obtains that

$$\text{max temp} > \text{min temp} > b.$$

Alternatively, one can condition maximum temperature on minimum temperature by specifying $b = \text{min temp}_{NTF}$ when transforming maximum temperature, obtaining

$$a > \text{max temp} > \text{min temp}.$$

Unfortunately the choice of which variable should be conditioned is not obvious. An option can only be verified by implementing the model and then examining the simulated sequences to check whether the properties of the climate sequences are being preserved. Usually one can get an indication of which variable to condition when one fits the model to the untransformed time series. If for one variable it is noted that the properties are not being retained as well as for its corresponding variable, then it would be advisable to first try the transformation where the "worse behaving" variable is conditioned.

In the case of sunshine duration, the upper bound was allowed to vary seasonally with time instead of being a constant. Define the upper limit by $B(t)$, where

$$B(t) = \text{ave} + 5 + \left(\frac{\text{amp}}{2}\right) \cos\left(\left(\frac{2\pi}{NT}\right)(t + 11)\right), \qquad t = 1, 2, \ldots, NT$$

where

$$\text{ave} = \frac{\text{smax} + \text{smin}}{2},$$

$$\text{amp} = \text{smax} - \text{smin} \quad \text{and}$$

smax and smin are chosen so that $B(t) \geq$ sunshine duration observed at time t .

Care must be taken that one does not divide by zero, which can happen at times when the lower limit is zero and a zero observation occurs. This problem can be overcome by adding a small value (e.g. 0.01) to all observations.

Model 1: Multivariate model for climate data proposed by Richardson (1981).

Table 4.5 shows the transformations used for each station. Only the lower and upper bounds are given in the table as the form of the transformation is given above.

The historical data for each of the climate variables was conditioned on the wet or dry status of the day, thus obtaining a mean function and a standard deviation function for each of the conditioned data sets. The mean and standard deviation were both approximated by a truncated Fourier series representation. That is

$$\mu_t = \sum_{i=1}^{L} \alpha_i \phi_i(t) \qquad \text{and}$$

$$\sigma_t = \sum_{i=1}^{L} \xi_i \phi_i(t), \qquad t = 1, 2, \ldots, NT$$

where $\phi_i(t)$ is defined as in Chapter 3 and where L does not have to be of the same order for both of the mean and the standard deviation function.

For the purposes of model selection the truncation level L , which determines the family of approximating models being fitted, was varied and the fit in each case was examined. The decision on which order of approximation to use was based on Akaike's Information Criterion (AIC). Tables 4.6 – 4.9 show the value of AIC and the choice of the order of approximation is given in square brackets. The percentage decrease of the criterion is given in parentheses whenever the value of AIC continued to decrease after five parameters had already been fitted. Here the number of parameters selected is based on the model which leads to a decrease in the criterion of more than 5 percent. This decision was taken for reasons mentioned in the previous section on the undesirability of fitting a large number of parameters to the models.

The values of L ranged between 1 and 5, with a mode of 3 for both the mean function given a dry day and for the mean function given a wet day. Therefore, a 3–term Fourier series approximation is estimated to be appropriate. Different L values for each variable for a particular station were not chosen in order to simplify the implementation and interpretation of the complete (multivariate time series) model.

TABLE 4.5 Transformations for Model 1

Variable	Station					
	Elsenburg	Kakamas	Middelburg	Nelspruit	Cedara	Hoopstad
Max Temp	unchanged	a=490	a=400	a=420	a=400	a=410
		b=min temp	b=min temp	b=min temp	b=min temp	b=min temp
Min Temp	a=max temp	a=320	a=250	a=250	unchanged	a=230
	b=0	b=-50	b=-90	b=0		b=-100
Evapo	square	a=300	square	N/A	N/A	N/A
	root	b=0	root	N/A	N/A	N/A
Sun	a=B(t)	a=B(t)	a=B(t)	a=B(t)	a=B(t)	a=B(t)
	smax=134	smax=136	smax=139	smax=130	smax=132	smax=135
	smin=94	smin=100	smin=100	smin=110	smin=102	smin=110
	b=0	b=0	b=0	b=0	b=0	b=0
Wind	a=10000	a=10000	a=10000	a=1000	a=1000	a=1000
	b=0	b=0	b=0	b=0	b=0	b=0
Max Hum	a=1001	a=1001	a=1001	a=101	a=101	a=101
	b=min hum	b=min hum	b=min hum	b=0	b=0	b=min hum
Min Hum	a=1000	a=1000	a=1000	a=max hum	a=max hum	a=100
	b=0	b=0	b=0	b=0	b=0	b=0

TABLE 4.6 Model selection criteria (AIC) for the mean function for non–rainy days

Variable	L	Elsenburg	Kakamas	Middelburg	Nelspruit	Cedara	Hoopstad
	1	327261	345	357	341	339	351
	3	78867	331	335	343	341	338
Max Temp	5	72626	333	337			339
	7	72133(0.4%)					
Selected		[5]	[3]	[3]	[1]	[1]	[3]
	1	355	407	397	465	350496	464
Min Temp	3	346	333	336	344	49924	339
	5	348	333	337	340(1%)	34256	338(0.3%)
	7					33719(2%)	
Selected		[3]	[3]	[3]	[3]	[5]	[3]
	1	1098	414	737			
Evapo	3	391	332	370			
	5	390(0.3%)	334	379			
Selected		[3]	[3]	[3]			
	1	403	343	370	528	494	359
Sun	3	399	345	372	479	475	359
	5	400			473	476	
Selected		[3]	[1]	[1]	[3]	[3]	[1]
	1	340	339	334	331	330	328
Wind	3	337	332	334	332	329	330
	5	338	334			330	
Selected		[3]	[3]	[1]	[1]	[3]	[1]
	1	376	383	385	347	389	389
Max Hum	3	375	356	364	343	371	345
	5	377	358	365	344	373	345
Selected		[3]	[3]	[3]	[3]	[3]	[3]
	1	345	338	336	361	370	343
Min Hum	3	341	332	333	341	346	340
	5	342	333	335	342	348	342
Selected		[3]	[3]	[3]	[3]	[3]	[3]

TABLE 4.7 Model selection criteria (AIC) for the mean function for rain days

Variable	L	Elsenburg	Kakamas	Middelburg	Nelspruit	Cedara	Hoopstad
	1	285827	341	372	357	347	354
Max Temp	3	107685	341	347	351	346	340
	5	104150(3%)		349	354	346	343
Selected		[3]	[1]	[3]	[3]	[3]	[3]
	1	340	340	348	349	215468	348
Min Temp	3	342	333	332	330	42050	330
	5		335	333	332	36230	332
	7					35482(2%)	
Selected		[1]	[3]	[3]	[3]	[5]	[3]
	1	1196	355	1154			
Evapo	3	756	353	841			
	5	748(1%)	355	832(1%)			
Selected		[3]	[3]	[3]			
	1	951	457	970	1541	1251	661
Sun	3	865	458	947	1354	1177	657
	5	855(1%)		940(0.7%)	1331(2%)	1130(4%)	654(0.5%)
Selected		[3]	[1]	[3]	[3]	[3]	[3]
	1	338	327	333	328	328	337
Wind	3	340	330	334	330	330	337
Selected		[1]	[1]	[1]	[1]	[1]	[1]
	1	371	504	446	346	352	366
Max Hum	3	375	504	438	347	352	359
	5			439			356(0.8%)
Selected		[1]	[1]	[3]	[1]	[1]	[3]
	1	339	337	349	372	392	360
Min Hum	3	337	339	350	375	388	360
	5	338				389	
Selected		[3]	[1]	[1]	[1]	[3]	[1]

TABLE 4.8 Model selection criteria (AIC) for the standard deviation function for non–rainy days

Variable	L	Station: Elsenburg	Kakamas	Middelburg	Nelspruit	Cedara	Hoopstad
	1	30600	327	327	327	321	327
Max Temp	3	29683	329	329	329	323	329
	5	29003(2%)					
Selected		[3]	[1]	[1]	[1]	[1]	[1]
	1	327	327	327	332	20944	327
Min Temp	3	329	329	329	333	19404	329
	5					18588(4%)	
Selected		[1]	[1]	[1]	[1]	[3]	[1]
	1	339	327	328			
Evapo	3	340	329	329			
Selected		[1]	[1]	[1]			
	1	367	327	354	504	442	333
Sun	3	365	329	355	489	439	331
	5	367			484(1%)	441	333
Selected		[3]	[1]	[1]	[3]	[3]	[3]
	1	327	327	327	327	321	327
Wind	3	329	329	329	329	323	329
Selected		[1]	[1]	[1]	[1]	[1]	[1]
	1	348	333	332	327	321	327
Max Hum	3	342	334	331	329	324	329
	5	344		332			
Selected		[3]	[1]	[3]	[1]	[1]	[1]
	1	327	327	327	327	321	327
Min Hum	3	329	329	329	329	323	329
Selected		[1]	[1]	[1]	[1]	[1]	[1]

TABLE 4.9 Model selection criteria (AIC) for the standard deviation function for rain days

Variable	no of parameters	Station Elsenburg	Kakamas	Middelburg	Nelspruit	Cedara	Hoopstad
	1	51315	332	329	329	327	329
Max Temp	3	48654	334	331	331	329	331
	5	47971(1%)					
Selected		[3]	[1]	[1]	[1]	[1]	[1]
	1	327	327	328	327	15941	327
Min Temp	3	329	329	329	329	15330	329
	5					15221(1%)	
Selected		[1]	[1]	[1]	[1]	[3]	[1]
	1	457	332	497			
Evapo	3	457	333	497			
Selected		[1]	[1]	[1]			
	1	619	382	702	744	665	572
Sun	3	547	381	691	686	625	572
	5	545(0.4%)	383	689(0.3%)	683(0.4%)	602(4%)	
Selected		[3]	[3]	[3]	[3]	[3]	[1]
	1	329	327	327	327	327	330
Wind	3	330	329	329	329	329	332
Selected		[1]	[1]	[1]	[1]	[1]	[1]
	1	360	392	357	328	328	333
Max Hum	3	355	392	360	330	332	335
	5	355					
Selected		[3]	[1]	[1]	[1]	[1]	[1]
	1	327	329	329	342	338	333
Min Hum	3	330	331	332	342	336	335
	5					338	
Selected		[1]	[1]	[1]	[1]	[3]	[1]

The values of L ranged between 1 and 3, with a mode of 1, for both the standard deviation function given a dry day and for the standard deviation function given a wet day.

Again a 3–term Fourier series approximation was chosen to simplify programming by having a common approximation order.

Tables 4.10–4.15 show the parameter estimates for the mean function and for the standard deviation function, both conditioned on the wet or dry status of period t .

The resulting time series obtained by subtracting the fitted mean function and by dividing through by the fitted standard deviation function should be a time series with a mean of zero and a standard deviation of unity. Since the mean value functions and the standard deviation functions which were fitted are based on truncated Fourier series, that is, on approximating models, the means of the residual series would not be exactly zero and the standard deviations would not be exactly one. However, deviations in this respect were found to be quite small. (Table 4.16.)

TABLE 4.10 Parameter estimates for the mean and standard deviation function for Nelspruit

Variable	Day Status	Mean Function $\hat{\alpha}_1$	$\hat{\alpha}_2$	$\hat{\alpha}_3$	Standard deviation function $\hat{\xi}_1$	$\hat{\xi}_2$	$\hat{\xi}_3$
Max Temp	Dry	1.1406	0.0491	0.0021	0.5492	0.0722	-0.1045
	Wet	0.7439	-0.3820	-0.2312	0.7346	-0.0129	-0.1027
Min Temp	Dry	-0.0768	-1.1420	-0.2058	0.4849	-0.1264	0.0172
	Wet	-0.3966	-0.8404	-0.2213	0.3666	-0.0469	0.0026
Sun	Dry	-0.8305	0.6966	-0.2708	1.5700	0.4211	-0.2742
	Wet	2.2995	-1.2331	-1.1324	3.1566	-0.8673	-0.5615
Wind	Dry	1.9637	0.0005	0.1367	0.2404	0.0175	-0.0155
	Wet	2.0005	-0.0579	0.1554	0.2789	-0.0069	-0.0338
Max Hum	Dry	-1.4305	-0.0110	-0.2614	0.5754	-0.0945	-0.0809
	Wet	-2.1826	0.1480	-0.0467	0.7114	0.0075	-0.0685
Min Hum	Dry	-0.2314	-0.4867	-0.0223	0.5471	-0.0377	-0.0490
	Wet	-0.9915	0.0100	0.1507	0.7902	-0.1771	-0.0605

TABLE 4.11 Parameter estimates for the mean and standard deviation function for Kakamas

Variable	Day Status	Mean Function			Standard deviation function		
		$\widehat{\alpha}_1$	$\widehat{\alpha}_2$	$\widehat{\alpha}_3$	$\widehat{\xi}_1$	$\widehat{\xi}_2$	$\widehat{\xi}_3$
Max Temp	Dry	0.1820	-0.4165	-0.0206	0.3912	0.0299	-0.0215
	Wet	0.7169	-0.1589	-0.1176	0.5849	0.0534	-0.1207
Min Temp	Dry	0.0561	-0.8786	-0.2398	0.4488	0.0249	-0.0311
	Wet	-0.2287	-0.6914	-0.2603	0.3976	0.0583	-0.1232
Evapo	Dry	0.9503	-0.9578	0.0770	0.4293	0.0004	-0.0106
	Wet	1.4806	-0.3786	-0.1060	0.7570	-0.0814	-0.1750
Sun	Dry	-1.6980	-0.0062	0.0217	0.8455	0.0029	0.1077
	Wet	0.5187	-0.0863	-0.1922	1.3270	-0.2649	0.1156
Wind	Dry	1.4858	-0.3014	0.0995	0.4098	-0.0406	-0.0231
	Wet	1.2941	0.0066	0.0779	0.3349	-0.0073	-0.0103
Max Hum	Dry	-0.0944	0.5496	-0.1002	1.3666	-0.2549	0.0597
	Wet	-0.7814	0.0296	-0.2719	1.4821	-0.2785	-0.1268
Min Hum	Dry	1.1816	0.2893	-0.0566	0.4990	0.0249	-0.0066
	Wet	0.4772	-0.0064	-0.0173	0.5523	-0.0303	-0.0673

TABLE 4.12 Parameter estimates for the mean and standard deviation function for Middelburg

Variable	Day Status	Mean Function $\widehat{\alpha}_1$	$\widehat{\alpha}_2$	$\widehat{\alpha}_3$	Standard deviation function $\widehat{\xi}_1$	$\widehat{\xi}_2$	$\widehat{\xi}_3$
Max Temp	Dry	-0.1424	-0.5116	-0.0217	0.5336	0.0718	-0.0156
	Wet	0.4120	-0.7261	-0.1572	0.6608	0.0509	-0.0500
Min Temp	Dry	0.2575	-0.8132	-0.1849	0.5279	-0.1576	-0.0616
	Wet	-0.1811	-0.7344	-0.2009	0.4172	-0.0803	-0.0499
Evapo	Dry	7.9960	1.9682	-0.4032	1.3287	-0.1210	-0.1113
	Wet	6.1770	2.1295	0.0773	2.0379	-0.2645	-0.2099
Sun	Dry	-2.0452	0.0167	0.0293	1.2232	0.0837	0.0880
	Wet	0.6383	-0.5181	-0.3849	1.9803	-0.4339	-0.2604
Wind	Dry	1.4977	-0.0610	0.1297	0.4592	-0.0855	-0.0228
	Wet	1.4042	0.0481	0.1328	0.4821	-0.1459	-0.0122
Max Hum	Dry	-1.3432	-0.4132	-0.2987	1.2128	-0.2591	-0.1682
	Wet	-1.2860	-0.3606	-0.1579	1.1107	-0.1790	-0.0987
Min Hum	Dry	1.1904	0.1649	-0.1501	0.5000	-0.0057	-0.0314
	Wet	0.5364	0.2806	0.0152	0.6164	-0.0767	-0.0799

TABLE 4.13 Parameter estimates for the mean and standard deviation function for Elsenburg

Variable	Day Status	Mean Function $\hat{\alpha}_1$	$\hat{\alpha}_2$	$\hat{\alpha}_3$	Standard deviation function $\hat{\xi}_1$	$\hat{\xi}_2$	$\hat{\xi}_3$
Max Temp	Dry	238.22	48.57	19.04	38.38	3.179	-0.0134
	Wet	193.53	46.05	21.63	25.71	5.296	3.317
Min Temp	Dry	0.3177	-0.2871	-0.1851	0.5278	-0.1475	-0.0490
	Wet	-0.4954	-0.0572	-0.1013	0.5552	-0.1558	-0.0066
Evapo	Dry	7.680	2.787	0.0901	1.0026	-0.0536	0.0270
	Wet	4.961	2.506	0.1371	1.8950	0.0165	-0.0899
Sun	Dry	-1.6113	-0.2528	-0.0514	0.9575	-0.2123	0.0104
	Wet	1.1820	-1.0978	0.2018	1.9234	-0.9789	0.3166
Wind	Dry	1.6157	-0.2502	0.0099	0.3923	-0.0539	-0.0276
	Wet	1.2347	0.0298	0.1007	0.4737	-0.1934	-0.0040
Max Hum	Dry	-2.2258	0.1167	-0.1301	0.8527	-0.1969	0.1334
	Wet	-2.0879	0.0697	-0.0156	0.4737	-0.1934	-0.0040
Min Hum	Dry	0.5553	0.2438	0.0887	0.4858	-0.0842	-0.0051
	Wet	-0.1166	0.4444	0.0339	0.6925	-0.3514	0.0798

TABLE 4.14 Parameter estimates for the mean and standard deviation function for Cedara

Variable	Day Status	Mean Function $\widehat{\alpha}_1$	$\widehat{\alpha}_2$	$\widehat{\alpha}_3$	Standard deviation function $\widehat{\xi}_1$	$\widehat{\xi}_2$	$\widehat{\xi}_3$
Max Temp	Dry	0.0677	-0.1084	0.0099	0.5117	0.0419	-0.0775
	Wet	0.8100	-0.1902	-0.1313	0.8119	0.0835	-0.1253
Min Temp	Dry	96.367	56.733	11.619	26.661	-3.558	-2.103
	Wet	111.379	46.644	13.587	22.168	-0.926	-2.553
Sun	Dry	-1.383	0.4607	-0.1772	1.3118	0.2792	-0.1978
	Wet	2.0787	0.2088	-0.9158	3.3718	0.3074	-0.6914
Wind	Dry	1.7602	-0.0816	0.1776	0.2919	-0.0758	-0.0153
	Wet	1.6711	-0.0878	0.1973	0.3219	-0.0127	-0.0353
Max Hum	Dry	-2.0807	-0.3614	-0.2952	0.9552	-0.1621	-0.0217
	Wet	-3.0247	-0.2189	-0.1216	0.8009	0.1149	0.0001
Min Hum	Dry	-0.0543	-0.5256	-0.0813	0.6235	-0.0317	-0.0688
	Wet	-1.1142	-0.2502	0.0721	1.0571	-0.0468	-0.1675

TABLE 4.15 Parameter estimates for the mean and standard deviation function for Hoopstad

Variable	Day	Mean Function			Standard deviation function		
	Status	$\hat{\alpha}_1$	$\hat{\alpha}_2$	$\hat{\alpha}_3$	$\hat{\xi}_1$	$\hat{\xi}_2$	$\hat{\xi}_3$
Max Temp	Dry	-0.3482	-0.4120	0.0216	0.4415	0.1293	-0.0081
	Wet	0.2494	-0.5653	-0.0547	0.6324	0.0222	0.0372
Min Temp	Dry	-1.4750	-1.1625	-0.2069	0.4701	-0.0122	-0.0174
	Wet	-0.6601	-0.8644	-0.1761	0.3764	-0.0442	-0.0151
Sun	Dry	-1.7544	0.1099	0.1206	0.7901	0.2790	0.0360
	Wet	0.6600	-0.4369	-0.0254	1.8220	-0.1887	0.1484
Wind	Dry	2.1065	-0.2612	0.2238	0.5323	-0.0128	-0.0153
	Wet	1.8836	0.0616	0.3142	0.4281	0.1357	0.0514
Max Hum	Dry	-0.4700	0.5763	-0.4229	0.7188	-0.1011	0.0364
	Wet	-0.7645	0.2807	-0.3057	0.7501	-0.0202	0.0191
Min Hum	Dry	0.9651	0.0142	-0.2402	0.5565	0.0808	-0.0154
	Wet	0.0503	0.1797	-0.0793	0.7935	-0.0143	0.0135

TABLE 4.16 Mean and standard deviation of residual time series obtained by standardizing the data

Variable		Station					
		Elsenburg	Kakamas	Middelburg	Nelspruit	Cedara	Hoopstad
Max Temp	M	-0.010	0.002	-0.007	0.0007	-0.02	0.006
	SD	1.09	1.04	1.05	1.08	1.08	1.06
Min Temp	M	0.004	-0.004	0.002	-0.0005	-0.01	-0.002
	SD	1.08	1.03	1.07	1.10	1.08	1.06
Evapo	M	-0.003	-0.0004	0.006			
	SD	1.12	1.05	1.05			
Sun	M	-0.007	0.006	-0.002	0.011	0.02	0.0009
	SD	1.16	1.06	1.12	1.22	1.17	1.13
Wind	M	-0.015	-0.003	0.0002	0.192	-0.007	0.013
	SD	1.09	1.03	1.05	1.12	1.11	1.15
Max Hum	M	0.012	-0.002	0.002	0.004	-0.009	-0.004
	SD	1.19	1.05	1.06	1.07	1.06	1.09
Min Hum	M	-0.010	-0.0003	0.004	0.003	-0.01	-0.006
	SD	1.10	1.05	1.06	1.10	1.08	1.07

Another assumption made by the model is that the residual time series follows an autoregressive process of order 1. If this is true then $\rho_k = \rho_1^k$ where ρ_k is the autocorrelation with lag k . This assumption (or more precisely, this approximation) was checked by comparing $\hat{\rho}_k$, $k = 1, 2, 3, 4$ with $\hat{\rho}_1^k$, and was found to be reasonable except for a few cases (Table 4.17). It is possible to increase the order of the autoregressive process to these cases, but this has be be done at cost of increasing the complexity and number of parameters in the model, and therefore not advisable.

The results of the above checks would suggest that the residual series do seem to satisfy the required assumptions of the model. It is therefore reasonable to approximate each of the seven series by the sum of a seasonal component and a residual component, to approximate the seasonal component by a 3–term Fourier series and finally to approximate the standard deviation of the residual series by a 3–term Fourier approximation.

TABLE 4.17 Comparison of $\hat{\rho}_k$ and $\hat{\rho}_1^*$ of the residual series

Station	Max Temp		Min Temp		Evapo		Sun		Wind		Max Hum		Min Hum	
	$\hat{\rho}_k$	$\hat{\rho}_1^*$	$\hat{\rho}_k$	$\hat{\rho}_1^*$	$\hat{\rho}_k$	$\hat{\rho}_1^*$	$\hat{\rho}_k$	$\hat{\rho}_1^*$	$\hat{\rho}_k$	$\hat{\rho}_1^*$	$\hat{\rho}_k$	$\hat{\rho}_1^*$	$\hat{\rho}_k$	$\hat{\rho}_1^*$
	0.42	0.42	0.14	0.14	0.19	0.19	0.21	0.21	0.26	0.26	0.28	0.28	0.38	0.38
Elsenburg	0.15	0.18	0.01	0.02	0.11	0.04	0.06	0.04	0.04	0.07	0.19	0.08	0.23	0.14
	0.04	0.07	-0.01	0.00	0.05	0.01	0.01	0.01	0.02	0.13	0.02	0.20	0.05	
	0.01	0.03	-0.03	0.00	0.02	0.00	0.00	0.00	0.01	0.01	0.14	0.01	0.18	0.02
	0.43	0.43	0.66	0.66	0.29	0.29	0.30	0.30	0.26	0.26	0.46	0.46	0.62	0.62
Kakamas	0.20	0.18	0.42	0.44	0.23	0.08	0.12	0.09	0.03	0.07	0.29	0.21	0.49	0.38
	0.12	0.08	0.29	0.29	0.19	0.02	0.08	0.03	0.03	0.02	0.23	0.10	0.45	0.24
	0.08	0.03	0.21	0.19	0.16	0.01	0.05	0.01	0.02	0.00	0.21	0.04	0.42	0.15
	0.34	0.34	0.34	0.34	0.36	0.36	0.22	0.22	0.31	0.31	0.24	0.24	0.46	0.46
Middelburg	0.12	0.12	0.14	0.12	0.11	0.13	0.10	0.05	0.03	0.10	0.15	0.06	0.33	0.21
	0.09	0.04	0.07	0.04	0.10	0.05	0.07	0.01	0.03	0.03	0.14	0.01	0.30	0.10
	0.03	0.01	0.04	0.01	0.07	0.02	0.02	0.00	0.03	0.01	0.09	0.00	0.26	0.04
	0.24	0.24	0.45	0.45			0.15	0.15	0.78	0.78	0.24	0.24	0.17	0.17
Nelspruit	0.04	0.06	0.31	0.20			0.02	0.02	0.66	0.61	0.09	0.06	0.06	0.03
	0.01	0.01	0.25	0.09			-0.01	0.00	0.56	0.47	0.06	0.01	0.05	0.01
	0.02	0.00	0.18	0.04			0.02	0.00	0.48	0.37	0.02	0.00	0.02	0.00
	0.13	0.13	0.48	0.48			0.10	0.10	0.76	0.76	0.12	0.12	0.14	0.14
Cedara	0.00	0.02	0.29	0.23			-0.01	0.01	0.62	0.58	0.02	0.01	0.04	0.02
	-0.01	0.00	0.20	0.11			-0.02	0.00	0.52	0.44	0.03	0.00	0.01	0.00
	-0.01	0.00	0.15	0.05			0.00	0.00	0.43	0.33	0.02	0.00	0.03	0.00
	0.50	0.50	0.50	0.50			0.23	0.23	0.75	0.75	0.40	0.40	0.51	0.51
Hoopstad	0.31	0.25	0.35	0.25			0.14	0.05	0.63	0.56	0.31	0.16	0.35	0.26
	0.22	0.13	0.28	0.13			0.09	0.01	0.54	0.42	0.23	0.06	0.26	0.13
	0.18	0.06	0.22	0.06			0.07	0.00	0.46	0.32	0.21	0.03	0.18	0.07

Model 1, proposed by Richardson (1981) is given by:

$$\chi_{i,t} = A\ \chi_{i,t-1} + B\ \epsilon_{i,t}$$

where $\chi_{i,t}$ is the residual series at time period t of year i. Display 4.1 gives the estimated A matrix for the various stations and Display 4.2 gives the estimated B matrix. The order of the climate variables in these displays is as follows: maximum temperature, minimum temperature, evaporation, sunshine duration, windrun, maximum humidity and finally minimum humidity.

Model T: Multivariate model for climate data

Models 3, 4 and 5 were developed as an alternative to Model 1 in an attempt to deal with a deficiency in Model 1, namely the assumption that the autocorrelation function of each variable is assumed invariant with respect to wet/dry and dry/wet day boundaries. Each model varies in complexity and emphasizes a slightly different aspect of the joint distribution of the variables. Table 4.18 shows the fundamental assumptions of each model.

The models depicted here are complex, describing several distinguishing properties of the climate series. No one model will be "best" in all respects or for all sites. In general, simpler models can be expected to outperform the more complex ones when the historical record at the site is small, whereas the opposite will be true when the record is long. Statistically one can select the appropriate model for each variable using Akaike's Information Criterion (AIC) where

$$\text{AIC} = -\ell(\psi; e_{i,t}) + L$$

where $\ell(\psi; e_{i,t})$ is the loglikelihood function of the model and L is the number of parameters. The model producing the lowest AIC value is chosen as the model that best described that particular climate variable. It must be noted that Models 3, 4 and 5 are not hierarchical and therefore a model with a larger number of parameters does not imply that the value of its loglikelihood function will be smaller.

$$\text{Elsenburg}\begin{bmatrix} 0.69 & -0.49 & -0.02 & 0.05 & -0.03 & -0.08 & -0.04 \\ 0.17 & 0.09 & -0.03 & 0.01 & -0.03 & -0.05 & 0.02 \\ 0.26 & -0.25 & -0.02 & -0.03 & -0.08 & 0.14 & 0.19 \\ -0.15 & -0.07 & 0.02 & 0.13 & 0.03 & -0.01 & -0.05 \\ 0.22 & 0.19 & -0.01 & 0.07 & 0.16 & -0.19 & -0.22 \\ 0.04 & -0.02 & 0.01 & 0.07 & 0.00 & 0.26 & 0.02 \\ 0.12 & -0.21 & -0.07 & 0.08 & -0.08 & 0.08 & 0.51 \end{bmatrix}$$

$$\text{Kakamas}\begin{bmatrix} 0.40 & 0.25 & 0.05 & -0.07 & 0.06 & -0.10 & 0.04 \\ -0.09 & 0.74 & 0.02 & 0.04 & 0.00 & 0.17 & -0.07 \\ -0.07 & 0.13 & 0.26 & -0.01 & 0.02 & -0.14 & 0.07 \\ 0.16 & -0.20 & 0.06 & 0.18 & -0.01 & -0.15 & 0.03 \\ -0.45 & -0.06 & 0.09 & 0.06 & 0.17 & -0.02 & -0.02 \\ 0.06 & -0.26 & 0.01 & 0.00 & -0.03 & 0.32 & 0.13 \\ 0.11 & -0.05 & 0.10 & 0.09 & -0.06 & 0.24 & 0.64 \end{bmatrix}$$

$$\text{Middelburg}\begin{bmatrix} 0.40 & 0.35 & -0.01 & -0.06 & 0.03 & -0.18 & -0.03 \\ -0.04 & 0.39 & -0.07 & 0.01 & -0.04 & 0.19 & 0.07 \\ 0.23 & -0.18 & 0.25 & -0.05 & 0.03 & 0.27 & 0.11 \\ 0.11 & -0.08 & -0.02 & 0.15 & 0.03 & -0.16 & 0.01 \\ -0.58 & 0.05 & -0.09 & 0.14 & 0.12 & -0.23 & -0.12 \\ 0.11 & -0.01 & 0.04 & 0.00 & 0.05 & 0.24 & 0.06 \\ 0.10 & -0.02 & -0.08 & 0.10 & -0.04 & 0.28 & 0.59 \end{bmatrix}$$

$$\text{Nelspruit}\begin{bmatrix} 0.43 & 0.23 & -0.15 & 0.38 & -0.08 & 0.07 \\ 0.13 & 0.46 & 0.00 & -0.03 & 0.06 & 0.04 \\ 0.22 & 0.07 & 0.03 & 0.38 & 0.03 & 0.01 \\ -0.04 & -0.07 & -0.02 & 0.15 & -0.06 & -0.02 \\ -0.03 & 0.02 & 0.06 & -0.30 & 0.16 & 0.02 \\ -0.10 & -0.04 & 0.04 & -0.27 & 0.17 & 0.08 \end{bmatrix}$$

$$\text{Cedara}\begin{bmatrix} 0.16 & -0.28 & -0.10 & 0.16 & -0.05 & -0.07 \\ -0.10 & 0.45 & 0.05 & -0.14 & -0.07 & -0.04 \\ 0.07 & -0.03 & 0.03 & 0.14 & -0.02 & -0.07 \\ -0.03 & 0.13 & 0.02 & 0.22 & 0.05 & 0.01 \\ 0.01 & -0.05 & 0.08 & -0.18 & 0.09 & 0.09 \\ -0.03 & 0.08 & 0.12 & -0.15 & 0.06 & 0.19 \end{bmatrix}$$

$$\text{Hoopstad}\begin{bmatrix} 0.40 & 0.24 & -0.07 & 0.12 & -0.19 & -0.21 \\ 0.07 & 0.52 & 0.06 & 0.11 & 0.04 & 0.04 \\ 0.09 & -0.12 & 0.09 & -0.03 & -0.10 & -0.10 \\ -0.01 & -0.11 & 0.02 & 0.43 & 0.05 & 0.07 \\ -0.05 & -0.07 & 0.06 & -0.05 & 0.36 & 0.13 \\ -0.09 & -0.01 & 0.11 & -0.13 & 0.23 & 0.47 \end{bmatrix}$$

DISPLAY 4.1 The estimated matrix A

$$\text{Elsenburg}\begin{bmatrix} 0.76 & & & & & & \\ 0.46 & 0.86 & & & & & \\ 0.36 & 0.00 & 0.84 & & & & \\ -0.34 & -0.33 & -0.34 & 0.75 & & & \\ 0.20 & 0.18 & -0.35 & -0.03 & 0.80 & & \\ 0.05 & -0.19 & 0.07 & 0.05 & -0.05 & 0.93 & \\ 0.57 & 0.27 & 0.14 & -0.05 & 0.04 & -0.08 & 0.60 \end{bmatrix}$$

$$\text{Kakamas}\begin{bmatrix} 0.80 & & & & & & \\ -0.15 & 0.73 & & & & & \\ 0.40 & 0.31 & 0.80 & & & & \\ 0.31 & -0.19 & 0.15 & 0.83 & & & \\ 0.01 & 0.30 & 0.41 & -0.09 & 0.74 & & \\ -0.25 & -0.27 & -0.05 & 0.05 & -0.06 & 0.75 & \\ -0.43 & -0.05 & -0.08 & -0.08 & 0.02 & 0.10 & 0.59 \end{bmatrix}$$

$$\text{Middelburg}\begin{bmatrix} 0.77 & & & & & & \\ -0.34 & 0.87 & & & & & \\ -0.36 & -0.34 & 0.68 & & & & \\ 0.44 & -0.07 & -0.32 & 0.77 & & & \\ 0.00 & 0.30 & -0.49 & -0.11 & 0.58 & & \\ 0.00 & -0.41 & 0.20 & 0.01 & -0.11 & 0.84 & \\ -0.56 & -0.10 & 0.20 & -0.09 & 0.08 & -0.02 & 0.55 \end{bmatrix}$$

$$\text{Nelspruit}\begin{bmatrix} 0.84 & & & & & \\ -0.40 & 0.80 & & & & \\ 0.61 & 0.00 & 0.66 & & & \\ -0.03 & -0.02 & -0.08 & 0.98 & & \\ -0.31 & 0.00 & -0.08 & -0.27 & 0.83 & \\ -0.66 & -0.02 & -0.07 & 0.02 & -0.08 & 0.63 \end{bmatrix}$$

$$\text{Cedara}\begin{bmatrix} 0.92 & & & & & \\ 0.20 & 0.83 & & & & \\ 0.66 & 0.08 & 0.72 & & & \\ -0.22 & -0.17 & -0.02 & 0.95 & & \\ -0.45 & 0.06 & 0.04 & -0.10 & 0.87 & \\ -0.71 & 0.01 & -0.16 & -0.03 & 0.01 & 0.63 \end{bmatrix}$$

$$\text{Hoopstad}\begin{bmatrix} 0.77 & & & & & \\ -0.34 & 0.80 & & & & \\ 0.40 & -0.20 & 0.89 & & & \\ -0.21 & 0.07 & 0.01 & 0.84 & & \\ -0.05 & -0.05 & -0.04 & -0.06 & 0.93 & \\ -0.56 & 0.04 & -0.12 & -0.01 & 0.04 & 0.57 \end{bmatrix}$$

DISPLAY 4.2 The estimated matrix B

Thus, one does not restrict the generation of the climate variables to any particular

TABLE 4.18 Assumption of the different models

Model	Assumptions
Model 1	– seasonal mean function
	– seasonal standard deviation function
	– constant autocorrelation coefficient across different rain – no rain sequences
	– conditioned on wet and dry sequences only
Model 3	– seasonal mean function
	– constant standard deviation function
	– different autocorrelation coefficients across different rain – no rain sequences
	– conditioned on wet/wet, dry/dry, wet/dry and dry/wet sequences
Model 4	– seasonal mean function
	– seasonal standard deviation function
	– constant autocorrelation coefficients across different rain – no rain sequences
	– conditioned on wet/wet, dry/dry, wet/dry and dry/wet sequences
Model 5	– seasonal mean function
	– seasonal standard deviation function
	– different autocorrelation coefficients across different rain – no rain sequences
	– conditioned on wet/wet, dry/dry, wet/dry and dry/wet sequences.

model, but each variable is generated according to the model that "best" describes it. The multivariate model to generate simultaneous daily climate sequences will be referred to as Model T, where, for each climate variable, Model T is constructed by selecting between Models 3, 4 and 5 for the model that produces the lowest AIC.

The following algorithm is used to implement Model T.

Algorithm

Step 1: Implement Model 3 to obtain parameter estimates and AIC for each variable.

Step 2: Implement Model 4 to obtain parameter estimates and AIC for each variable.

Step 3: Implement Model 5 to obtain parameter estimates and AIC for each variable.

Step 4: Construct Model T by choosing for each variable the model producing the lowest AIC.

Step 5: Obtain the estimated cross–correlation matrix, $\widehat{\Sigma}$, whose elements are given by:

$$R_{jk} = \frac{\frac{1}{T}\sum_{i=1}^{NY}\sum_{t=1}^{NT} e_{i,t}^{(j)} e_{i,t}^{(k)} - \frac{1}{T^2}\sum_{i=1}^{NY}\sum_{t=1}^{NT} e_{i,t}^{(j)} \sum_{i=1}^{NY}\sum_{t=1}^{NT} e_{i,t}^{(k)}}{\left[\frac{1}{T}\sum_{i=1}^{NY}\sum_{t=1}^{NT}(e_{i,t}^{(j)})^2 - \frac{1}{T^2}\left(\sum_{i=1}^{NY}\sum_{t=1}^{NT} e_{i,t}^{(j)}\right)^2\right]^{\frac{1}{2}} \left[\frac{1}{T}\sum_{i=1}^{NY}\sum_{t=1}^{NT}(e_{i,t}^{(k)})^2 - \frac{1}{T^2}\left(\sum_{i=1}^{NY}\sum_{t=1}^{NT} e_{i,t}^{(k)}\right)^2\right]^{\frac{1}{2}}}$$

where $e_{i,t}^{(j)}$ denotes the residual time series of variable j, $j = 1,2,\ldots,NV$ and where for each j the residual series is the series obtained after the model producing the lowest AIC for variable j has been fitted and $e_{i,t}^{(k)}$ denotes the residual time series of variable k, $k = 1,2,\ldots,NV$, the residual series obtained in the same way as above.

Implementing Models 3, 4 and 5

The transformations applied to these models are the same as those in the previous model except for the variables maximum temperature and minimum temperature of the station Nelspruit. Here the bounds are given by

$$a = 420, \quad \text{and}$$

$$b = 0$$

for maximum temperature, and

$$a = \text{max temp}$$

$$b = 0$$

for minimum temperature.

Models 3, 4 and 5 are implemented by following the respective algorithms given in Chapter 3. The initial estimates for the mean function and for the seasonal standard deviation function are the same as the estimates of Model 1 and therefore need not be recomputed. Only when a different transformation to that applied in Model 1 is used, is it necessary to compute initial estimates for the mean function and the seasonal standard deviation function.

The selection of the order of the Fourier series approximation for the mean function and the standard deviation function was based on the initial estimates of these functions and so a choice of a 3–term Fourier series was made for both functions.

The estimation of the parameters is accomplished by iteration. The procedure described in the algorithms is that of Newton–Raphson. This method, although converging within a few iterations, was found to be sensitive to the initial values given. On occasions where convergence was not reached, a conjugate gradient method was used for parameter estimation. The computer programs for this procedure are given in Appendix D. The advantages of this method is that one gets convergence and only the vector of first derivatives needs to be computed. The disadvantage is that it is time consuming and takes a large number of iterations to converge.

In our implementation a particular parameter estimate was deemed to have converged when its value changed by less than 0.01% in successive iterations. The estimation procedure was deemed to have converged when all estimates had converged.

Akaike's Information Criterion for the selection of a model for each variable is given in Table 4.19. The lowest AIC value is shown in bold and the corresponding model is selected to generate climate sequences for that variable. This model selection is performed for each of the stations.

Values with an asterisk indicate the model that was finally selected, although it did not produce the lowest AIC. This choice was necessary in some instances because of the relatively few occurrences of rainfall in some stations. More importantly, at some sites consecutive rainy days seldom occur. Thus, in Models 3 and 5 the estimation of the autocorrelation coefficient given that the sequence WW was observed, is based on very few observations, and consequently, it is possible to obtain inadmissible estimates. Similarly, when the sequence WD or DW was observed. In such cases, generally Model 4 was chosen as it gave acceptable estimates since here the autocorrelation function is not conditioned and all observations are used in the estimation.

Parameter estimates for the mean function, the standard deviation function and the coefficient of the autoregressive process of order 1 are given in Tables 4.20–4.25 for each station.

Table 4.19 Akaike's Information Criterion for Models 3, 4 and 5

Variable	Model	Station					
		Elsenburg	Kakamas	Middelburg	Nelspruit	Cedara	Hoopstad
	Mod 3	21190	**3393**	5751	2151	5781	**2610**
Max Temp	Mod 4	21174	3537	5796	2030	5745	2988
	Mod 5	**21160**	3502	**5749**	**2006**	**5726**	2773
	Mod 3	3579	2994	5474	4684	26594	3191
Min Temp	Mod 4	3484	2992	5178	4502	26594	3193
	Mod 5	**3467**	**2962**	**5162**	**4472**	**26562**	**3182**
	Mod 3	7102	**5018**	12721			
Evapo	Mod 4	7124	5124	12710			
	Mod 5	**7091**	5080	**12697**			
	Mod 3	7458	**10528**	**12716**	**10742**	**12645**	**8458**
Sun	Mod 4	7263	10669	12841	10951	12786	8511
	Mod 5	**7241**	10665	12822	10953	12758	8505
	Mod 3	2557	4528	4664	**493**	2105	3973
Wind	Mod 4	**2477**	**4499**	**4536**	607	2162	4463
	Mod 5	2483	4501	4539	565	**2057**	**3760**
	Mod 3	5799	13644	11890	4900	8101	6579
Max Hum	Mod 4	5544	13799*	**11712**	4879	8090	6614
	Mod 5	**5542**	**13587**	11717	**4859**	**8078**	**6573**
	Mod 3	3012	3951*	5653	4961	**7047**	**4266**
Min Hum	Mod 4	2952	3972	**4876**	**4903**	7067	4541
	Mod 5	**2926**	**3929**	5668	4904	7054	4312

The estimate of the cross–correlation matrix, $\hat{\Sigma}$, is obtained by following Step 5 of the algorithm given in this chapter. Problems arise when computing this formula when missing observations occur in the residual series. A simple approach to estimate the cross–correlation matrix in the presence of missing values, is to restrict the analysis to time periods t with all variables observed. However, this method discards a considerable amount of data and the estimate obtained is biased. A more efficient approach is to estimate the missing values and to replace them by their estimate.

Makhuvha (1988) investigated several methods of estimating the missing values in rainfall records. She concluded that of the methods compared, the EM algorithm is the most

TABLE 4.20 Parameter estimates of Model T for Elsenburg

Variable	Status	Mean $\widehat{\alpha}_1$	$\widehat{\alpha}_2$	$\widehat{\alpha}_3$	Standard Deviation $\widehat{\xi}_1$ (DD)	$\widehat{\xi}_2$ (WW)	$\widehat{\xi}_3$ (DW)	(WD)	Autocorrelation coefficient (DD)	(WW)	(DW)	(WD)
Max Temp	D	231.18	48.72	19.01	33.55	1.8024	0.2573		0.5794	0.3459	0.3872	0.5942
	W	204.50	53.68	22.74	29.03	6.6864	3.1137					
Min Temp	D	0.3022	-0.2920	-0.1657	0.5212	-0.1354	-0.0546		0.2342	-0.0720	0.2671	0.0520
	W	-0.4868	-0.0160	-0.0984	0.6016	-0.1326	-0.0079					
Evapo	D	7.6192	2.7414	0.0900	1.0359	-0.0949	0.0635		0.3227	-0.0491	0.1168	0.0920
	W	5.0406	2.5638	-0.1071	2.1324	0.2682	-0.0206					
Sun	D	-1.6281	-0.2424	-0.0765	1.0764	-0.2902	0.0525		0.0009	0.1173	0.2898	0.1036
	W	0.9495	-1.1692	0.0193	2.2477	-1.0441	0.2785					
Wind	D	1.6106	-0.2549	0.0180	0.3955	-0.0433	-0.0349		0.2780			
	W	1.2669	-0.0216	0.0709	0.5073	-0.1575	-0.0129					
Max Hum	D	-2.2759	0.0996	-0.1112	0.8724	-0.2045	0.1318		0.3436	0.2144	0.2482	0.2042
	W	-2.0439	0.0681	0.0147	0.8779	-0.3449	0.1664					
Min Hum	D	0.5042	0.2483	0.0683	0.4446	-0.0893	0.0071		0.5027	0.1687	0.3148	0.3223
	W	-0.0733	0.4691	0.0451	0.5791	-0.0833	0.0936					

TABLE 4.21 Parameter estimates of Model T for Kakamas

Variable	Status	Mean $\widehat{\alpha}_1$	$\widehat{\alpha}_2$	$\widehat{\alpha}_3$	Standard Deviation $\widehat{\xi}_1$ (DD)	$\widehat{\xi}_2$ (WW)	$\widehat{\xi}_3$ (DW)	(WD)	Autocorrelation coefficient (DD)	(WW)	(DW)	(WD)
Max Temp	D	0.1627	-0.4202	-0.0276	0.3367	1.0537	0.6734	0.6167	0.4781	0.4354	0.4171	0.4551
	W	0.3463	-0.1978	-0.0721								
Min Temp	D	0.0488	-0.8637	-0.2536	0.3342	0.0154	-0.0237		0.6947	0.3330	0.7680	0.4741
	W	-0.0969	-0.8725	-0.2746	0.3321	0.0307	-0.0707					
Evapo	D	0.9368	-0.9685	0.0712	0.4108	1.2381	0.7372	0.6606	0.2930	0.6665	0.5014	0.3466
	W	1.1645	-0.7191	0.0049								
Sun	D	-1.7212	0.0064	-0.0003	0.7934	1.4352	1.8488	1.6322	0.3372	0.0280	0.0585	-0.0044
	W	0.5558	-0.0440	-0.1465								
Wind	D	1.4873	-0.3020	0.1002	0.3989	-0.0255	-0.0251		0.2629			
	W	1.3139	-0.0322	0.0648	0.3927	-0.0213	-0.0013					
Max Hum	D	-0.1501	0.4878	-0.1316	1.1771	-0.2292	0.0032		0.5277			
	W	0.6801	0.1684	-0.1280	1.6941	-0.4360	0.0031					
Min Hum	D	1.1912	0.2865	-0.0486	0.3697	1.0398	0.5222	0.5177	0.6781	0.7747	0.7354	0.6003
	W	0.9306	0.2392	-0.0246								

TABLE 4.22 Parameter estimates of Model T for Middelburg

Variable	Status	Mean $\widehat{\alpha}_1$	Mean $\widehat{\alpha}_2$	Mean $\widehat{\alpha}_3$	Standard Deviation $\widehat{\xi}_1$ (DD)	Standard Deviation $\widehat{\xi}_2$ (WW)	Standard Deviation $\widehat{\xi}_3$ (DW)	Standard Deviation (WD)	Autocorrelation coefficient (DD)	Autocorrelation coefficient (WW)	Autocorrelation coefficient (DW)	Autocorrelation coefficient (WD)
Max Temp	D	-0.1750	-0.5214	-0.0207	0.4972	0.0700	-0.0271		0.3351	0.6246	0.2688	0.7525
	W	-0.0466	-0.5426	-0.0733	0.7664	-0.0019	-0.0946					
Min Temp	D	0.2309	-0.7890	-0.2229	0.5073	-0.1471	-0.0584		0.3855	0.1991	0.4919	0.2023
	W	-0.1527	-0.7246	-0.1868	0.4616	-0.1031	-0.0569					
Evapo	D	7.9724	1.9607	-0.3904	1.2470	-0.0886	-0.1185		0.3733	0.4681	0.2609	0.5932
	W	6.9672	1.8639	-0.2865	2.2131	-0.1846	-0.2240					
Sun	D	-2.0811	0.0179	0.0070	1.1915	3.0527	1.8217	1.7494	0.2798	0.2948	0.1615	0.1473
	W	0.0118	-0.3549	-0.2383								
Wind	D	1.5048	-0.0641	0.1327	0.4486	-0.0718	-0.0270		0.3153			
	W	1.3822	0.0314	0.1695	0.4822	-0.1150	-0.0313					
Max Hum	D	-1.3835	-0.4246	-0.3093	1.2037	-0.2473	-0.1653		0.2501			
	W	-1.0464	-0.4250	-0.2633	1.2606	-0.1815	-0.1403					
Min Hum	D	1.1584	0.1583	-0.1487	0.4467	0.0138	-0.0290		0.4710			
	W	0.7990	0.1551	-0.0416	0.6503	-0.1175	-0.0614					

TABLE 4.23 Parameter estimates of Model T for Nelspruit

Variable	Status	Mean $\widehat{\alpha}_1$	Mean $\widehat{\alpha}_2$	Mean $\widehat{\alpha}_3$	Standard Deviation $\widehat{\xi}_1$ (DD)	Standard Deviation $\widehat{\xi}_2$ (WW)	Standard Deviation $\widehat{\xi}_3$ (DW)	Standard Deviation (WD)	Autocorrelation coefficient (DD)	Autocorrelation coefficient (WW)	Autocorrelation coefficient (DW)	Autocorrelation coefficient (WD)
Max Temp	D	-0.5530	-0.2991	-0.0618	0.3496	0.0415	-0.0825		0.5130	0.1863	0.3670	0.2989
	W	-0.3810	-0.4587	-0.1973	0.4045	0.0255	-0.0425					
Min Temp	D	0.0386	-0.7716	-0.1403	0.5612	-0.1514	-0.0713		0.3813	0.0797	0.1556	0.0717
	W	-0.5269	-0.2497	-0.0071	0.7003	-0.2251	-0.0918					
Sun	D	-1.0839	0.5190	-0.1531	1.4358	3.9054	2.7535	3.2377	0.1700	0.2423	0.1597	0.1180
	W	1.4841	-0.9117	-1.0326								
Wind	D	2.0032	-0.0014	0.1453	0.2386	0.3743	0.2590	0.3336	0.0001	0.3037	0.1856	0.1613
	W	2.0175	-0.0235	0.1640								
Max Hum	D	-1.4927	-0.0301	-0.2512	0.5749	-0.0730	-0.0896		0.3661	0.1056	0.1539	0.2178
	W	-2.0923	0.1254	-0.0859	0.7741	-0.0250	-0.0894					
Min Hum	D	-0.2500	-0.4921	-0.0472	0.5716	-0.0267	-0.0580		0.1953			
	W	-0.8259	-0.1266	0.1330	0.9221	-0.2392	-0.0932					

TABLE 4.24 Parameter estimates of Model T for Cedara

Variable	Status	Mean			Standard Deviation				Autocorrelation coefficient			
		$\hat{\alpha}_1$	$\hat{\alpha}_2$	$\hat{\alpha}_3$	$\hat{\xi}_1$ (DD)	$\hat{\xi}_2$ (WW)	$\hat{\xi}_3$ (DW)	(WD)	(DD)	(WW)	(DW)	(WD)
Max Temp	D	0.0817	-0.1296	0.0067	0.5511	0.1035	-0.0839		0.2398	0.1117	0.0036	0.0444
	W	0.7722	-0.1986	-0.0917	0.8740	0.0132	-0.0792					
Min Temp	D	97.70	51.43	14.11	23.82	-2.624	-0.781		0.6092	0.3651	0.5537	0.3714
	W	110.28	46.56	14.04	21.60	-2.705	-1.667					
Sun	D	-1.5666	0.2819	-0.2128	1.2486	4.2278	2.8030	2.1725	0.1643	0.1914	-0.1618	-0.0123
	W	1.4590	0.1536	-0.6015								
Wind	D	1.7531	-0.0856	0.1766	0.3146	-0.0541	-0.0150		0.2607	0.0794	0.0000	0.1909
	W	1.6761	-0.0900	0.1703	0.3782	-0.0339	-0.0351					
Max Hum	D	-2.1008	-0.3668	-0.3021	1.0044	-0.1101	-0.0460		0.1672	0.1115	-0.0599	0.1598
	W	-3.0014	-0.2290	-0.0672	0.8444	0.0607	-0.0111					
Min Hum	D	-0.0160	-0.4920	-0.0928	0.6204	1.2461	0.9200	0.7907	0.2142	0.2060	-0.0295	0.0743
	W	-0.8994	-0.2833	0.0296								

TABLE 4.25 Parameter estimates of Model T for Hoopstad

Variable	Status	Mean $\widehat{\alpha}_1$	$\widehat{\alpha}_2$	$\widehat{\alpha}_3$	Standard Deviation $\widehat{\xi}_1$ (DD)	$\widehat{\xi}_2$ (WW)	$\widehat{\xi}_3$ (DW)	(WD)	Autocorrelation coefficient (DD)	(WW)	(DW)	(WD)
Max Temp	D	-0.5078	-0.5004	0.0254	0.2893	0.8638	0.4752	0.5262	0.5808	0.8066	0.7517	0.6951
	W	-0.3431	-0.5375	-0.0210								
Min Temp	D	-0.2257	-1.1626	-0.2329	0.3919	-0.0033	-0.0244		0.5982	0.4282	0.5456	0.3609
	W	-0.5434	-1.0062	-0.2076	0.3728	-0.0538	0.0099					
Sun	D	-1.8242	0.0955	0.0975	0.6753	2.9114	1.6379	1.2875	0.2845	0.2908	0.0003	0.1339
	W	0.0924	-0.3002	-0.0487								
Wind	D	2.0979	-0.2117	0.2139	0.4114	0.0001	-0.0514		0.4733	0.5789	0.0003	0.5387
	W	2.0284	-0.0710	0.2737	0.4468	0.0452	-0.0037					
Max Hum	D	-0.5581	0.4988	-0.4000	0.6267	-0.0181	-0.0186		0.5338	0.2678	0.2790	0.2456
	W	-0.7064	0.2786	-0.2623	0.8009	-0.0600	-0.0189					
Min Hum	D	1.1107	0.0944	-0.2305	0.3903	1.0292	0.6553	0.6209	0.5566	0.8081	0.7663	0.6498
	W	0.7927	0.1593	-0.1760								

efficient method that can be applied for the estimation of missing records, and in terms of accuracy, it performs at least as well as the other methods. The EM algorithm is a very general iterative method for maximum likelihood estimation in incomplete data sets. It comprises of the following steps:

1. Missing values are replaced by estimated values.
2. Parameters are estimated.
3. Missing values are re–estimated assuming that the new parameter estimates are correct.
4. Parameters are re–estimated and so forth, iterating until convergence.

A detailed explanation and the theory of the EM algorithm is given in Appendix E.

The estimates of the cross–correlation matrix for each station are given in Display 4.3. The matrices are symmetrical, therefore only the upper triangle is given. The order of the climate variables in the display is as follows:

maximum temperature, minimum temperature, evaporation, sunshine duration, windrun, maximum humidity and finally minimum humidity.

The results described in this chapter would suggest that the models are not inconsistent with the historical record.

The selected models have the following number of parameters:

The model for rainfall occurrences: has 6 parameters.

The model for rainfall depth: has 4 parameters.

Model 1: has 161 parameters.

Model 3: has 126 parameters.

Model 4: has 119 parameters.

Model 5: has 140 parameters.

Of course the tests described in this chapter cover only some limited aspects of the fit. The issue of model validation is considered more exhaustively in Chapter 6.

$$\text{Elsenburg}\begin{bmatrix} 1.000 & 0.606 & 0.440 & -0.430 & 0.235 & 0.073 & 0.711 \\ & 1.000 & 0.219 & -0.507 & 0.281 & -0.154 & 0.614 \\ & & 1.000 & -0.550 & -0.291 & 0.128 & 0.451 \\ & & & 1.000 & -0.066 & 0.025 & -0.466 \\ & & & & 1.000 & -0.127 & 0.162 \\ & & & & & 1.000 & -0.072 \\ & & & & & & 1.000 \end{bmatrix}$$

$$\text{Kakamas}\begin{bmatrix} 1.000 & -0.038 & 0.385 & 0.193 & 0.000 & 0.006 & -0.027 \\ & 1.000 & 0.294 & -0.234 & 0.012 & -0.010 & 0.028 \\ & & 1.000 & 0.102 & -0.012 & -0.021 & -0.002 \\ & & & 1.000 & -0.007 & 0.112 & 0.062 \\ & & & & 1.000 & 0.040 & 0.188 \\ & & & & & 1.000 & 0.307 \\ & & & & & & 1.000 \end{bmatrix}$$

$$\text{Middelburg}\begin{bmatrix} 1.000 & -0.354 & -0.488 & 0.413 & 0.036 & 0.014 & 0.002 \\ & 1.000 & -0.170 & -0.270 & 0.141 & -0.015 & 0.013 \\ & & 1.000 & -0.386 & -0.349 & -0.027 & 0.002 \\ & & & 1.000 & 0.056 & 0.007 & -0.002 \\ & & & & 1.000 & -0.187 & -0.251 \\ & & & & & 1.000 & -0.231 \\ & & & & & & 1.000 \end{bmatrix}$$

$$\text{Nelspruit}\begin{bmatrix} 1.000 & -0.650 & 0.620 & -0.155 & -0.328 & -0.718 \\ & 1.000 & -0.668 & 0.082 & 0.307 & 0.637 \\ & & 1.000 & -0.123 & -0.349 & -0.594 \\ & & & 1.000 & -0.225 & 0.059 \\ & & & & 1.000 & 0.231 \\ & & & & & 1.000 \end{bmatrix}$$

$$\text{Cedara}\begin{bmatrix} 1.000 & 0.060 & 0.689 & -0.236 & -0.461 & -0.480 \\ & 1.000 & 0.130 & -0.112 & 0.029 & -0.028 \\ & & 1.000 & -0.164 & -0.305 & -0.403 \\ & & & 1.000 & -0.039 & 0.095 \\ & & & & 1.000 & 0.212 \\ & & & & & 1.000 \end{bmatrix}$$

$$\text{Hoopstad}\begin{bmatrix} -1.000 & -0.343 & 0.351 & -0.398 & 0.086 & 0.026 \\ & 1.000 & -0.375 & 0.165 & -0.082 & -0.013 \\ & & 1.000 & -0.139 & -0.043 & 0.023 \\ & & & 1.000 & -0.063 & -0.007 \\ & & & & 1.000 & -0.002 \\ & & & & & 1.000 \end{bmatrix}$$

DISPLAY 4.3 Estimated cross–correlation matrices for each station

CHAPTER 5

ALGORITHMS

This chapter describes the various procedures to be followed during model implementation and later during generation of climate sequences.

"Custom built" computer programs to carry out the preliminary analysis, to fit the models, to validate the models and finally to generate climate sequences have been written in ANSI 77 FORTRAN. The programs written conform to the full ANSI standard except for programs 6 and 8 where the array CLIMA is dimensioned using the HUGE attribute, which is an extension to the full ANSI standard. This was necessary when the climate data sets consisted of more than 9 years of daily data. Standard Fortran programs without this attribute should be no problem on a mainframe. Appendix D gives information where a listing of these programs (referred to in the algorithms below) can be obtained. The algorithms described here were all implemented on an IBM compatible PC micro-computer.

The following algorithms are discussed in this chapter:

- Algorithm for fitting the rainfall model.
- Algorithm for generating artificial rainfall sequences.
- Algorithm for fitting Model 1 to climate sequences.
- Algorithm for generating climate sequences using Model 1.
- Algorithm for fitting Model 3 to climate sequences.
- Algorithm for fitting Model 4 to climate sequences.
- Algorithm for fitting Model 5 to climate sequences.
- Algorithmn for implementing Model T.
- Algorithm for generating climate sequences using Model T.

Algorithm for implementing the rainfall model

The following information is required for the parameter estimation programs and must be computed from the historical record:

NT the number of periods in the year (e.g. 365 for daily data).

NY the number of years of data (including the missing values).

For each $t = 1, 2, \ldots, NT$.

$NW(t)$ the number of times it was wet in period $t-1$ and there was an observation in period t .

$NRR(t)$ the number of times it was wet in period $t-1$ and wet in period t .

$ND(t)$ the number of times it was dry in period $t-1$ and there was an observation in period t .

$N\overline{R}R(t)$ the number of times it was dry in period $t-1$ and wet in period t .

$R(i,t)$ the ith non–zero rainfall depth in period t, $i = 1, 2, \ldots, NR(t)$.

$NR(t)$ the number of times it was wet in period t .

Algorithm for estimating the probabilities of wet and dry sequences

Step 1: Prepare data sets $NW(t)$ and $NRR(t)$.

- Program 1
- if any of the $NW(t)$ are equal to zero, then delete time period t from data set.

Step 2: Estimate the parameters for the probability that a wet period follows a wet period.

- Program 2.

Step 3: Prepare data sets $ND(t)$ and $N\overline{R}R(t)$.

- Program 1
- of any of the $ND(t)$ are equal to zero, delete time period t from the data set.

Step 4: Estimate the parameters for the probability that a wet day follows a dry period.

- Program 2.

Algorithm to estimate the mean rainfall in wet periods

Step 1: Prepare the data sets $NR(t)$ and $R(i,t)$.

- Program 1.

Step 2: Estimate the parameters of the mean.

Step3: Estimate the coefficient of variation.

- Program 3 (does Steps 2 and 3).

Algorithm for generating artificial rainfall sequences

Step 1: Set initial state of day to be dry.

Step 2: Generate uniform random number between 0 and 1, inclusive $(U(0,1))$.

Step 3: If $U(0,1)$ random number is less than the probability of a wet day following a day with the status of the previous time period then

- the status of the present time period is wet.

Otherwise

- the status of the present time period is dry.

Step 4: If present state is wet than determine the rainfall depth.

Step 5: Repeat steps from Step 2 until enough rainfall sequences have geen generated.

- Program 4.

Algorithm for implementing Model 1 to climate sequences

The following information is required for the parameter estimation programs and must be computed from the historical records:

NT the number of periods in the year.

NY the number of years of data.

NV the number of variables in the model.

For each $t = 1, 2, \ldots, NT$ and for each variable

$m(t)$ the mean of the climate variable at time t.

$s(t)$ the standard deviation of the climate variable at time t.

For each variable do:

Step 1: Condition data set according to the wet or dry status of the day. That is, a record is kept of the time periods that had rain and the time periods that had no rain.

- Program 5.

For each conditioned data set do Step 2 – Step 6:

Step 2: Compute the daily mean vector, $m(t)$.

- Program 6.

Step 3: Estimate the parameters of the mean.

- Program 7.

Step 4: Compute the daily standard deviation vector, $s(t)$.

- Program 8.

Step 5: Estimate the parameters of the standard deviation.

- Program 7.

Step 6: Obtain the standardized residual series by subtracting the estimated daily mean function and dividing by the estimated daily standard deviation function.

- Program 9.

Step 7: Once the residual time series has been calculated for each variable, estimate the lag 0 and lag 1 cross–correlation coefficients.

- Program 10.

Step 8: Using estimates obtained in Step 7 compute the matrices A and B .

- Program 11.

Algorithm for generating artificial climate sequences using Model 1

Step 1: Generate rainfall sequence (algorithm given above)

For each variable do:

Step 2: Generate a normal random number from a distribution with a mean of zero and a standard deviation of unity $(N(0,1))$.

Step 3: Generate residual time series by:

$$\chi_{i,t} = \widehat{A}\ \chi_{i,t-1} + \widehat{B}\ \epsilon_{i,t}.$$

- the initial condition of the residual time series is taken to be equal to zero, i.e. $\chi_{1,0} = 0$.

Step 4: Generate climate sequences by:

$$S_{i,t} = \begin{cases} \chi_{i,t}\ \widehat{\sigma}_t^W + \widehat{\mu}_t^W & \text{if wet} \\ \chi_{i,t}\ \widehat{\sigma}_t^D + \widehat{\mu}_t^D & \text{if dry.} \end{cases}$$

Step 5: Repeat all of the above steps until the desired amount of climate sequences have been generated. So that the generating process has a chance of stabilizing itself, the first year of data generated is ignored.

- Program 12.

Algorithm for implementing Model 3 to climate sequences

The following information is required for the parameter estimation programs and must be computed from the historical records:

NT the number of periods in the year.

NY the number of years of data.

NV the number of variables in the model.

T the total number of observations.

$N(DD)$ the set of time periods t such that period t was dry and period $t-1$ was dry, $t = 1, 2, \ldots, T$.

$N(WW)$ the set of time periods t such that period t was wet and period $t-1$ was wet.

$N(DW)$ the set of time periods t such that period t was wet and period $t-1$ was dry.

$N(WD)$ the set of time periods t such that period t was dry and period $t-1$ was wet.

$C(DD)$ number of elements in the set $N(DD)$.

$C(WW)$ number of elements in the set $N(WW)$.

$C(DW)$ number of elements in the set $N(DW)$.

$C(WD)$ number of elements in the set $N(WD)$.

For each variable do:

Step 1: Estimate initial parameters of the mean function by performing Step 1 through to Step 3 of the algorithm for parameter estimation of Model 1.

Step 2: Prepare the data sets of possible sequences, i.e. $N(DD), N(WW), N(DW)$ and $N(WD)$. Compute $C(DD), C(WW), C(DW)$ and $C(WD)$.

- Program 13.

Step: 3: Estimate initial autocorrelation coefficients for each of the possible sequences.

- Program 14.

Step 4: Estimate initial standard deviation function for each of the possible sequences.

– Program 15.

Step 5: Estimate parameters of the mean function, standard deviation function and the autocorrelation coefficients, iterating until convergence is met by all parameters.

– Program 16 (or Program 17).

Step 6: Obtain residual time series by

$$e_{i,t} = \frac{S_{i,t} - \widehat{\mu}_t}{\widehat{\sigma}} - \widehat{\theta}\,\frac{S_{i,t-1} - \widehat{\mu}_{t-1}}{\widehat{\sigma}}$$

where $\widehat{\mu}_t, \widehat{\mu}_{t-1}, \widehat{\theta}$ and $\widehat{\sigma}$ are chosen depending on which sequence the time periods t and $t-1$ satisfy.

Algorithm for implementing Model 4 to climate sequences

The information necessary for parameter estimation programs is the same as for Model 3.

For each variable do:

Step1: Estimate initial parameters of the mean function by performing Step 1 through to Step 3 of the algorithm for parameter estimation of Model 1.

Step 2: Estimate initial parameters of the standard deviation function by performing Step 4 and Step 5 of the algorithm for implementing Model 1.

Step 3: Estimate initial autocorrelation coefficient.

– Program 18.

Step 4: Prepare the data sets of possible sequences, $N(DD), N(WW), N(DW)$ and $N(WD)$. Compute $C(DD), C(WW), C(DW)$ and $C(WD)$.

– Program 13.

Step 5: Estimate parameters of the mean function, standard deviation function and the autocorrelation coefficient, iterating until covergence is met by all parameters.

– Program 19 (or Program 20).

Step 6: Obtain residual time series by:

$$e_{i,t} = \frac{S_{i,t} - \widehat{\mu}_t}{\widehat{\sigma}_t} - \widehat{\theta}\,\frac{S_{i,t-1} - \widehat{\mu}_{t-1}}{\widehat{\sigma}_{t-1}}$$

where $\widehat{\mu}_t, \widehat{\mu}_{t-1}, \widehat{\sigma}_t$ and $\widehat{\sigma}_{t-1}$ are chosen depending on which sequence the time periods t and $t-1$ satisfy.

Algorithm for implementing Model 5 to climate sequences

The information necessary for parameter estimation programs is the same as for Model 3.

For each variable do:

Step 1: Estimate initial parameters of the mean function and of the standard deviation function by performing Step 1 through to Step 5 of the algorithm for implementing Model 1.

Step 2: Estimate initial autocorrelation coefficients for each of the possible sequences by performing Step 3 of the algorithm for implementing Model 3.

Step 3: Prepare the data sets of possible sequences, $N(DD), N(WW), N(DW)$ and $N(WD)$. Compute $C(DD), C(WW), C(DW)$ and $C(WD)$.

– Program 13.

Step 4: Estimate parameters of the mean function, standard deviation function and autocorrelation coefficients, iterating until convergence is met by all parameters.

– Program 21 (or Program 22).

Step 5: Obtain residual time series by:

$$e_{i,t} = \frac{S_{i,t} - \widehat{\mu}_t}{\widehat{\sigma}_t} - \widehat{\theta}\,\frac{S_{i,t-1} - \widehat{\mu}_{t-1}}{\widehat{\sigma}_{t-1}}$$

where $\widehat{\mu}_t, \widehat{\mu}_{t-1}, \widehat{\sigma}_t, \widehat{\sigma}_{t-1}$ and $\widehat{\theta}$ are chosen depending on which sequence the time periods t and $t-1$ satisfy.

Algorithm for implementing Model T

Step 1: From each residual time series obtained after fitting Models 3, 4 and 5, select for each variable the residual series from the model which produced the lowest Akaike's Information Criterion.

Step 2: Record, for each variable the time perioids t for which a missing observation occurs.

– Program 23.

Step 3: Use the EM algorithm to estimate and replace missing values by this estimate.

– Program 24.

Step 4: Estimate the cross–correlation matrix, Σ .

– Program 25.

Algorithm for generating artificial climate sequences using Model T

Step 1: Generate rainfall sequence.

For each variable do:

Step 2: Generate $N(0, \widehat{\Sigma})$ random number.

Step 3: Generate climate values according to the model chosen for that variable. For example, if Model 3 is chosen, then

$$S_{i,t} = \widehat{\mu}_t + \widehat{\sigma}\left[e_{i,t} + \widehat{\theta}\ \frac{S_{i,t-1} - \widehat{\mu}_{t-1}}{\widehat{\sigma}}\right]$$

where $\widehat{\mu}_t, \widehat{\mu}_{t-1}, \widehat{\sigma}$ and $\widehat{\theta}$ are chosen depending on the sequence t and $t-1$ satisfy.

If Model 4 is chosen, then

$$S_{i,t} = \widehat{\mu}_t + \widehat{\sigma}_t\left[e_{i,t} + \widehat{\theta}\ \frac{S_{i,t-1} - \widehat{\mu}_{t-1}}{\widehat{\sigma}_{t-1}}\right]$$

where $\widehat{\mu}_t, \widehat{\mu}_{t-1}, \widehat{\sigma}_t$ and $\widehat{\sigma}_{t-1}$ are chosen depending on the sequence t and $t-1$ satisfy.

If Model 5 is chosen then

$$S_{i,t} = \widehat{\mu}_t + \widehat{\sigma}_t\left[e_{i,t} + \widehat{\theta}\ \frac{S_{i,t-1} - \widehat{\mu}_{t-1}}{\widehat{\sigma}_{t-1}}\right]$$

where $\widehat{\mu}_t, \widehat{\mu}_{t-1}, \widehat{\sigma}_t, \widehat{\sigma}_{t-1}$ and $\widehat{\theta}$ are chosen depending on the sequence t and $t-1$ satisfy.

Step 4: Repeat above steps until the desired amount of climate sequences have been generated.

– Program 26.

CHAPTER 6

GOODNESS OF FIT

Once a model has been identified and the parameters estimated, it remains to decide whether the model is adequate. Model validation is applied with the object of assessing the performance of the model and to uncover any possible lack of fit. In particular one wants to assess whether the model proposed and parameters estimated preserve the properties of the process being examined. This chapter summarizes the results of the checks carried out on Model 1 and Model T described in Chapter 3.

Validation of rainfall model

The rainfall model has been shown to be satisfactory in the various regions of South Africa (Zucchini and Adamson, 1984). They performed extensive checks on the properties of the model such as:

(a) the annual mean and standard deviation and the distribution of annual totals and sum of k running totals, $k = 1, 2, \ldots, 5$,

(b) the monthly means and standard deviations,

(c) the expected number of wet days at different times of the year,

(d) the distribution of runs of wet and dry days,

(e) the distribution of n–day extreme rainfall.

The Markov chain/Weibull model adopted was found to preserve these properties. A number of these checks were repeated in this study. For a more complete model validation procedure see Zucchini and Adamson (1984).

Historical data (daily observations) were obtained for the weather stations Elsenburg, Kakamas, Middelburg, Nelspruit, Cedara and Hoopstad. More information on these records was given in Chapter 4.

Fifty years of simulated daily data were compared with the historical data on an annual, monthly and daily basis.

Table 6.1 gives both the historical and simulated annual mean number of wet days. This property has been adequately preserved by the model.

TABLE 6.1 Mean number of wet days per year

Station	Historical	Simulated
Elsenburg	91	92
Kakamas	16	19
Middelburg	63	63
Nelspruit	96	95
Cedara	150	149
Hoopstad	72	80

The mean number of wet days for each month has also been adequately preserved by the model (Figure 6.1)

It is especially important that the occurrence of wet days by season be adequately modelled as the generation of the other climate variables is conditioned on the occurrence of wet or dry days. The above results indicate that the Markov chain/Weibull model preserves the properties of the rainfall sequence at those locations.

The fits of the truncated Fourier series for the probability of having a wet day given a preceding wet day, and for the probability of having a wet day given the preceding one was dry, for each station, are shown in Figures 6.2 – 6.7. The fits are generally good.

The interpretation of these figures requires some explanation. These are not ordinary regression equation fits with normally distributed residuals. The smooth line indicates the fitted probability for a binomial random variable where the number of trials is also random. The outcomes are *discrete* values representing the number of successes in a series of Bernoulli trials. This is analogous to a situation in which a coin, which has a probability p of landing heads, is tossed n times and the fraction of times the coin landed heads is recorded. The smooth line would then represent the (smoothly varying probability) and the points on the graph the proportion of heads. The visual impression that one gets from such a diagram might suggest that the fit is poor (because one is used to interpreting regressions with normally distributed residuals, that is continuous random residuals) when in fact the fit is very good. The latter is the case in Figures 6.2 – 6.7.

FIGURE 6.1 Historical and simulated mean number of wet days

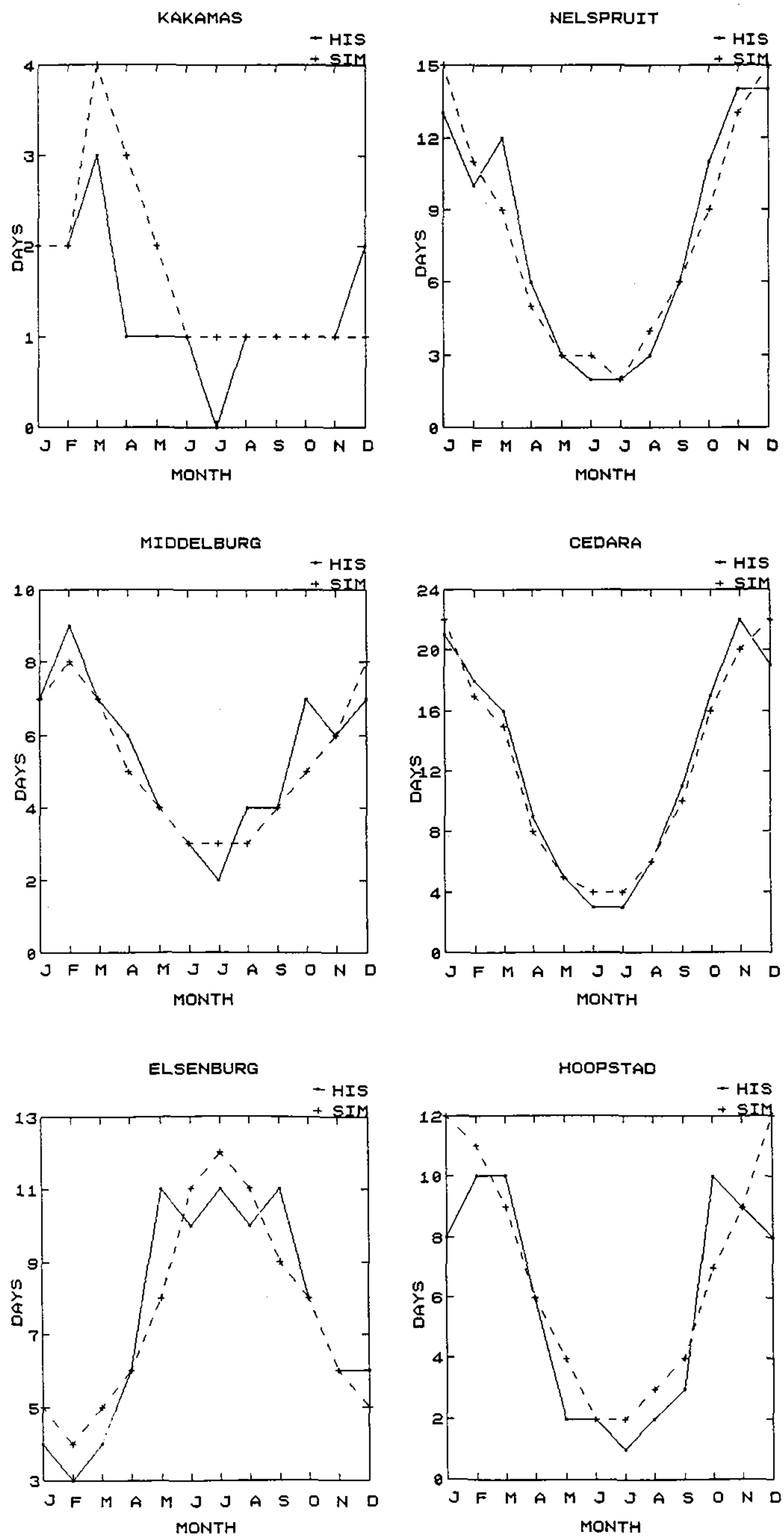
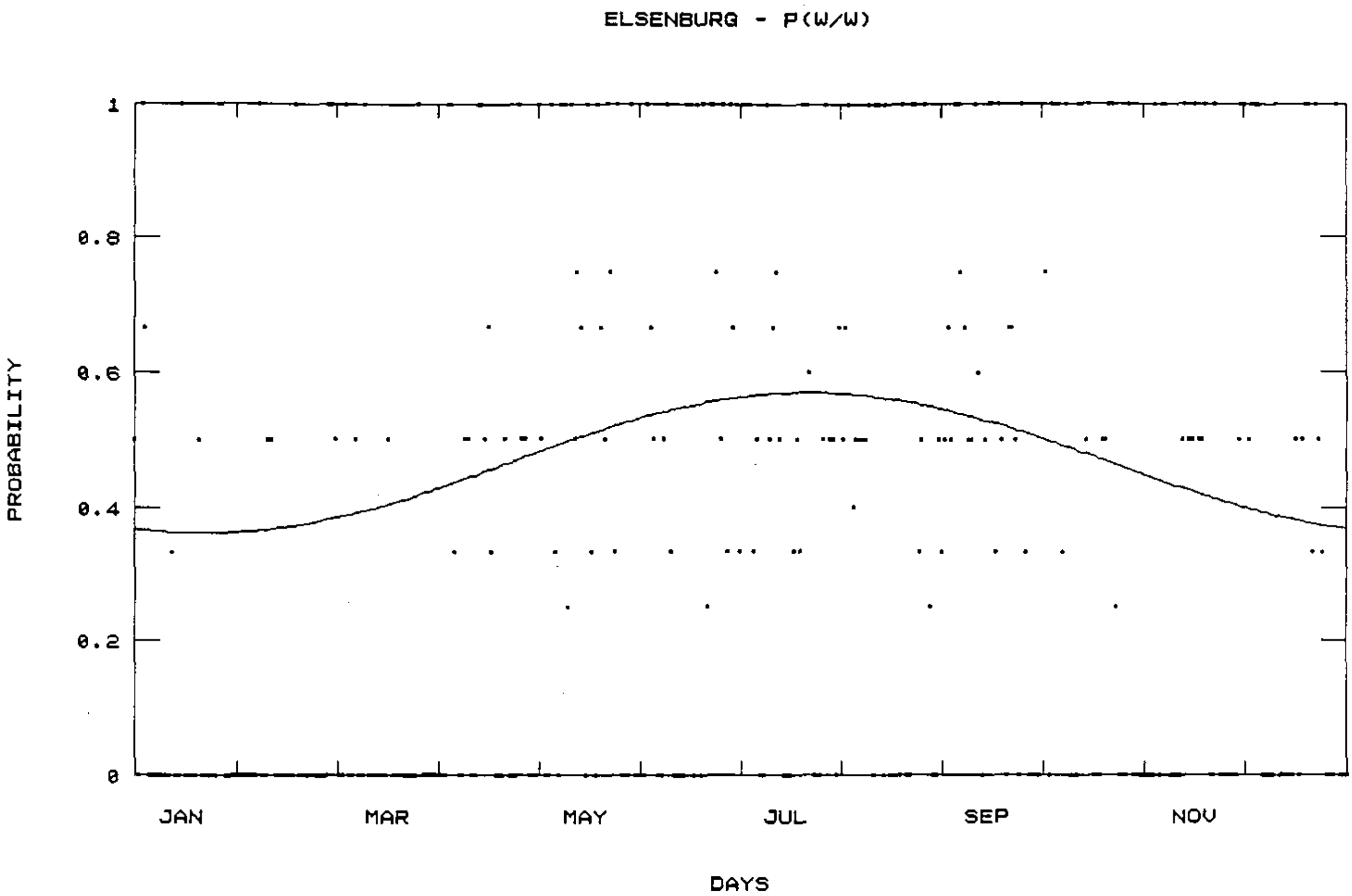


FIGURE 6.2 Empirical probabilities and estimates based on a 3 parameter model for $P(W|W)$ and $P(W|D)$ for Elsenburg



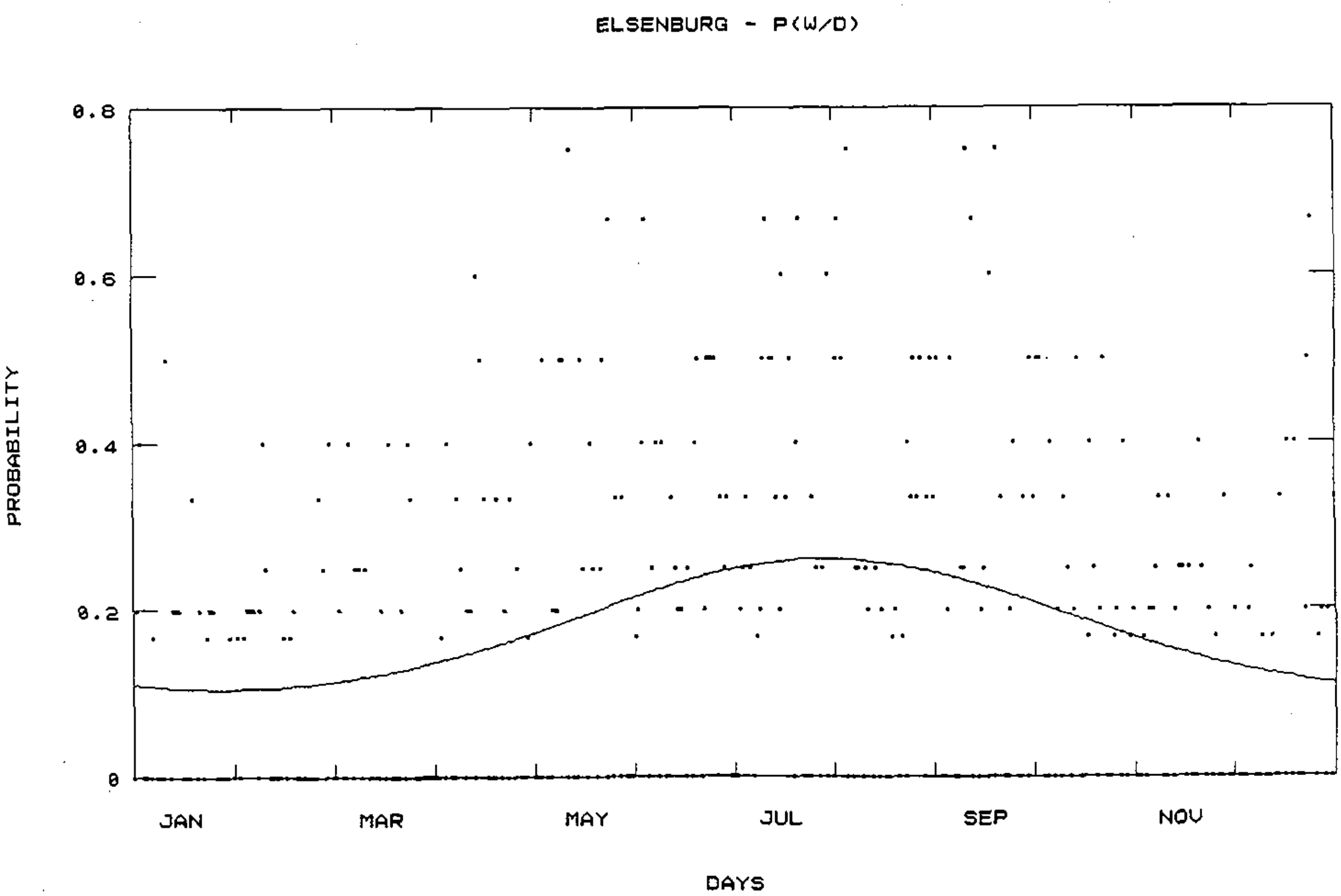
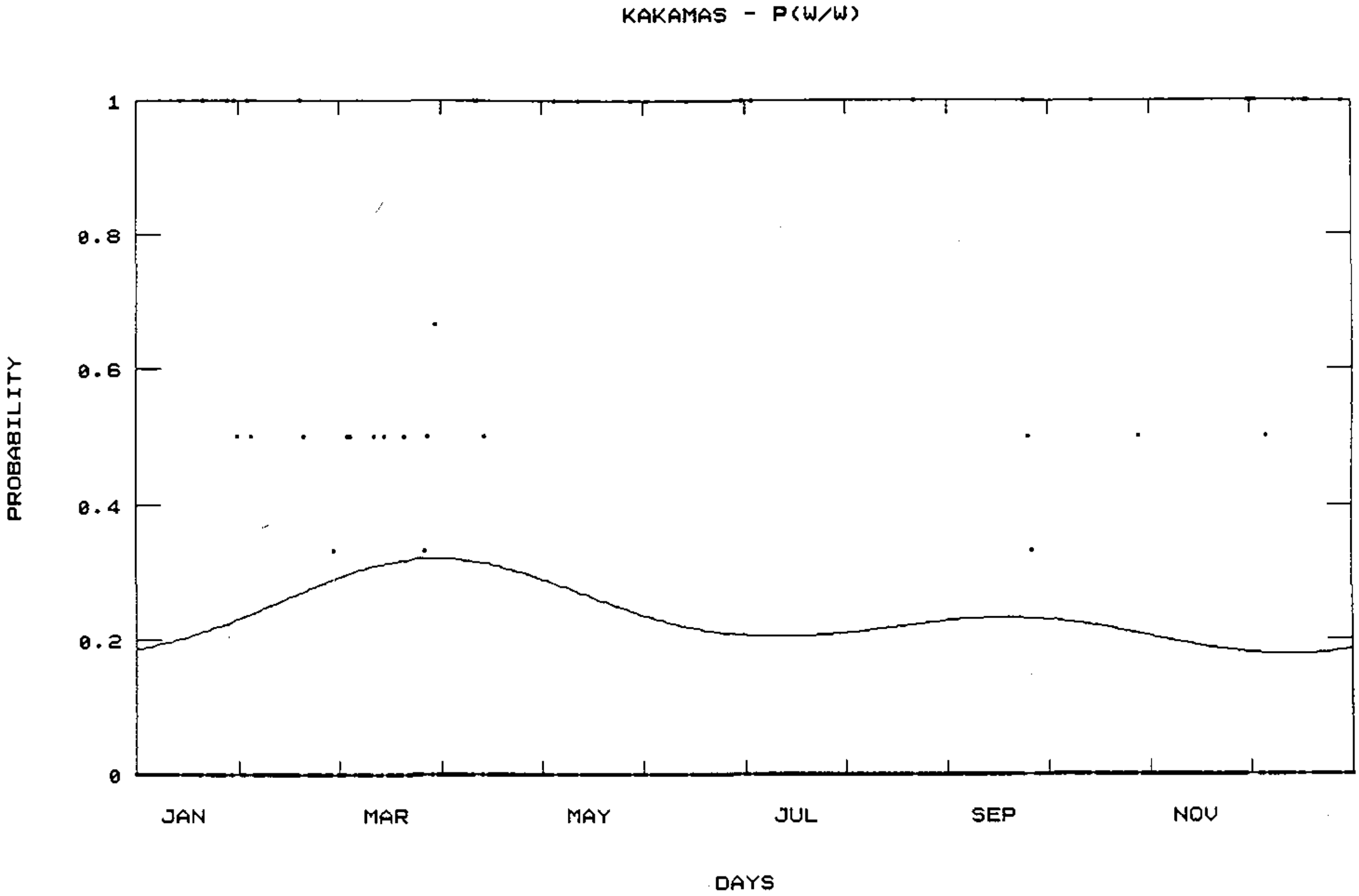


FIGURE 6.3 Empirical probabilities and estimates based on a 3 parameter model for $P(W|W)$ and $P(W|D)$ for Kakamas



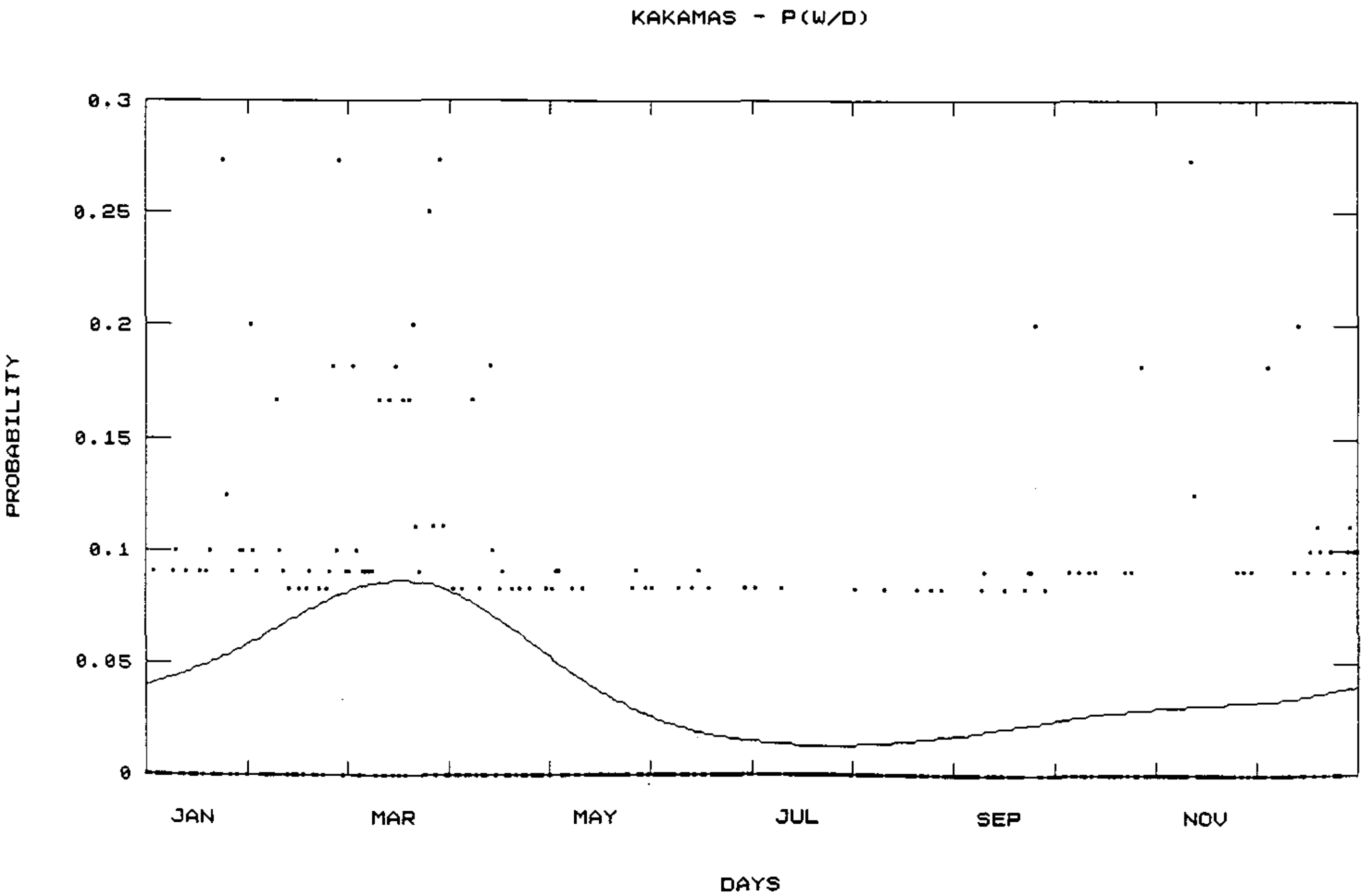
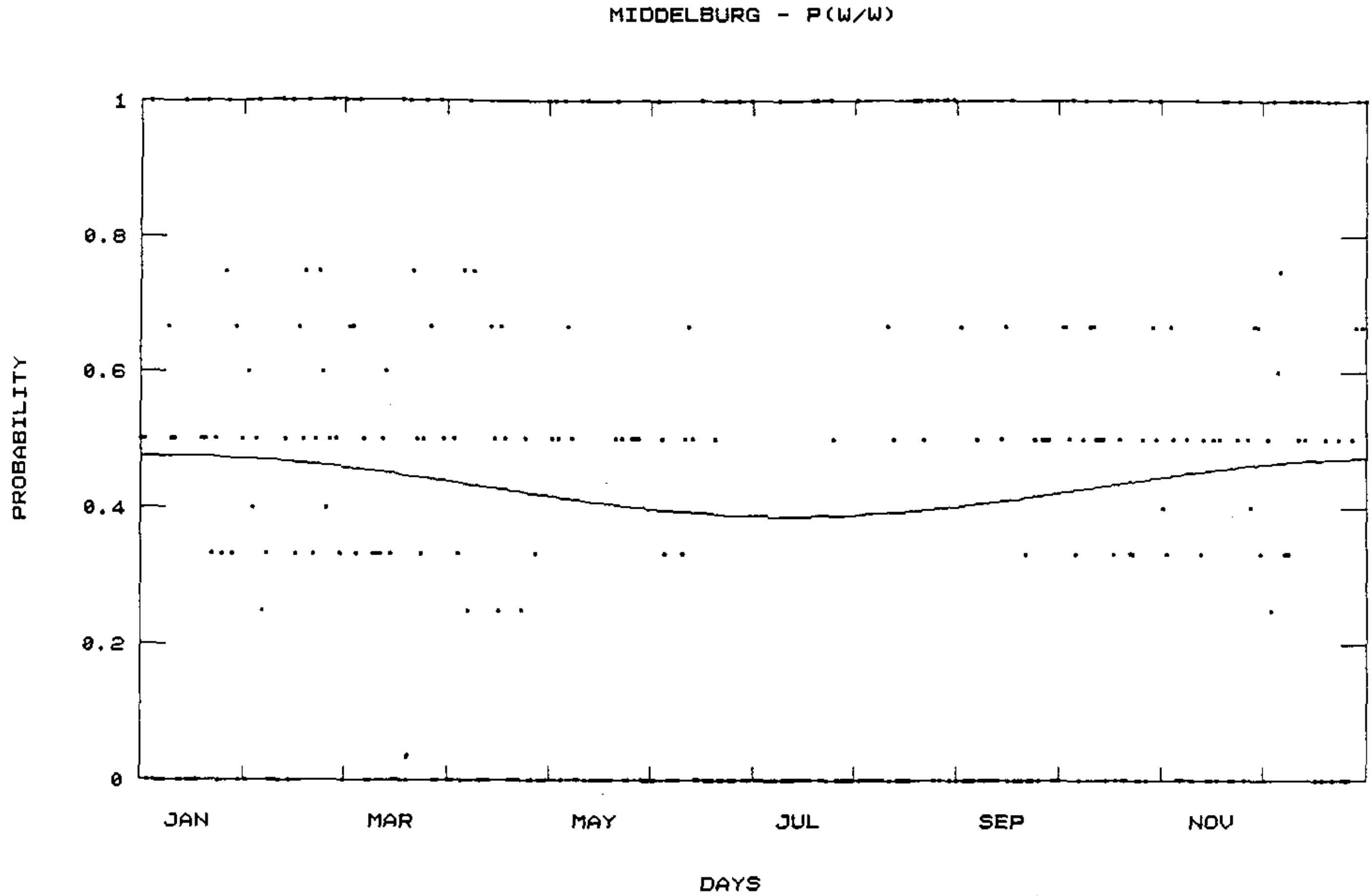


FIGURE 6.4 Empirical probabilities and estimates based on a 3 parameter model for $P(W|W)$ and $P(W|D)$ for Middelburg



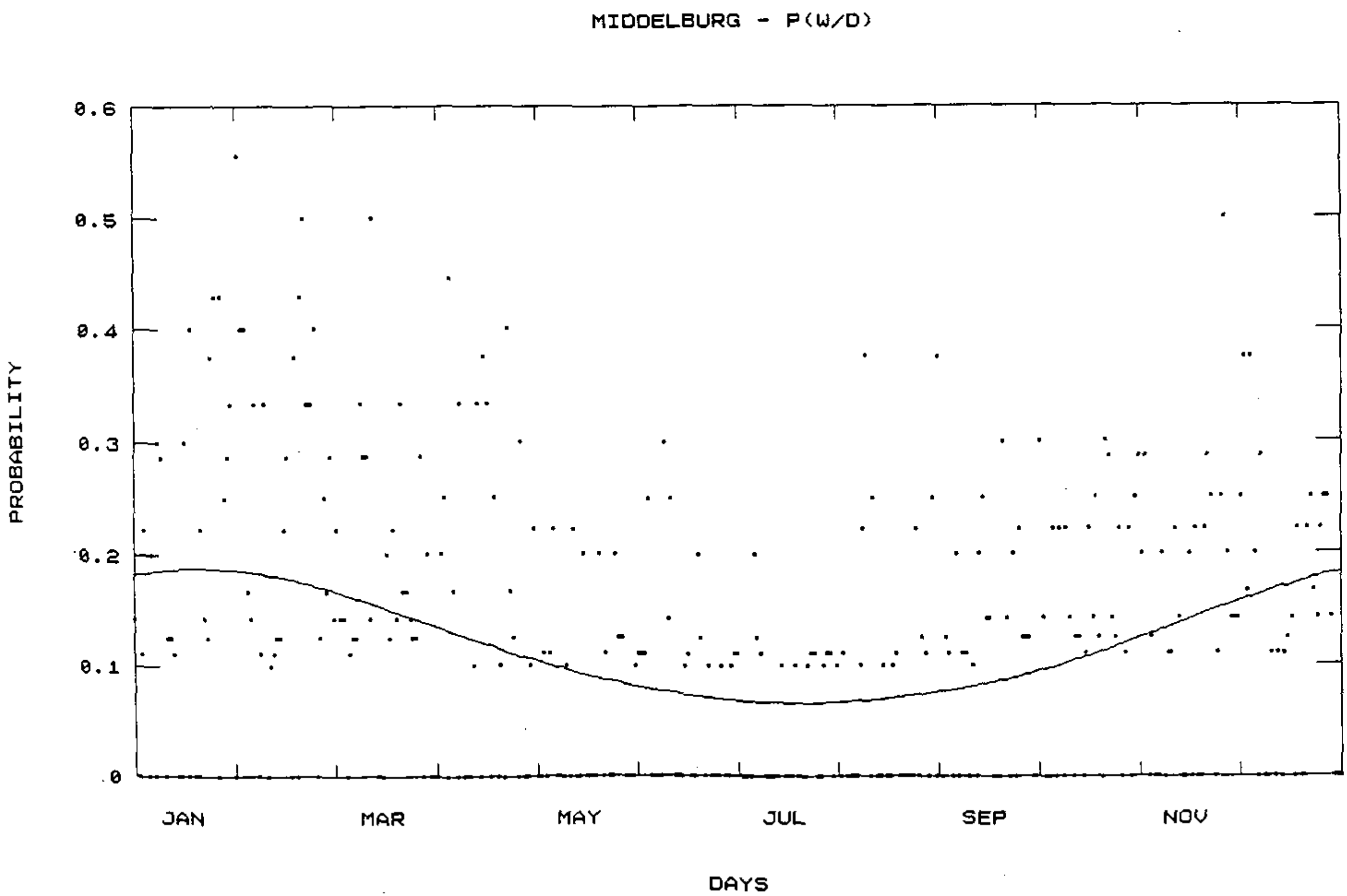
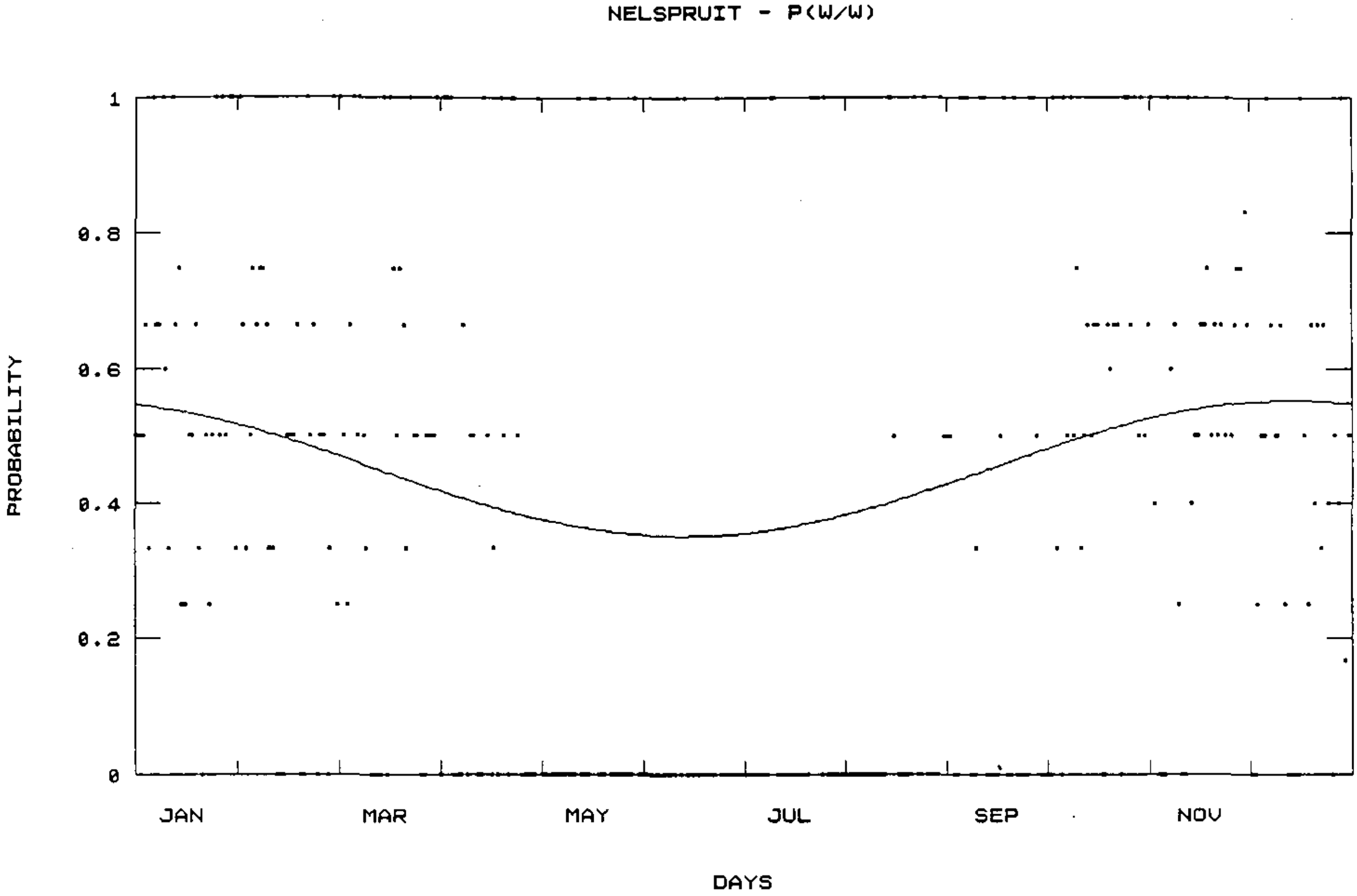


FIGURE 6.5 Empirical probabilities and estimates based on a 3 parameter model for $P(W|W)$ and $PW|D)$ for Nelspruit



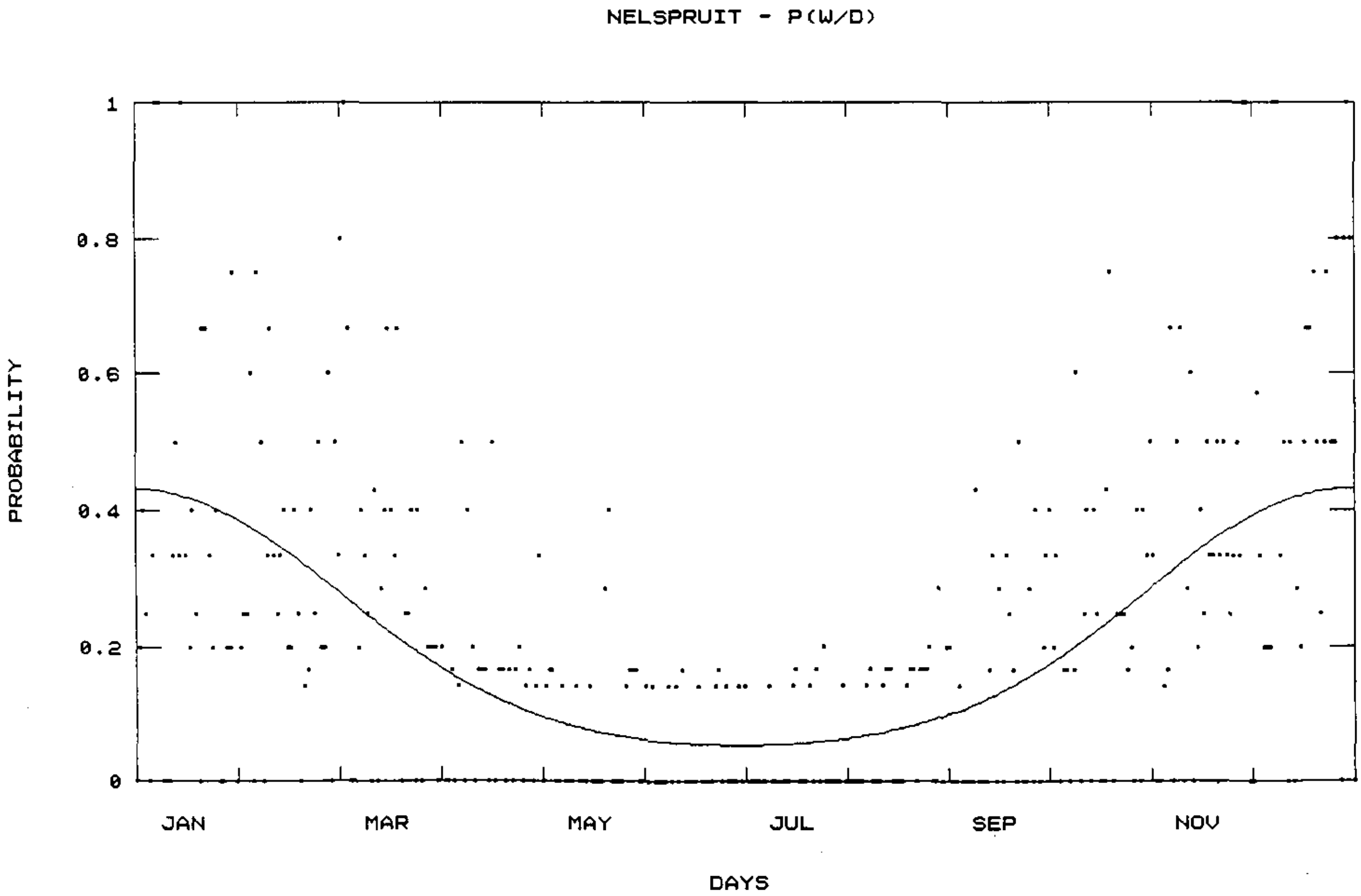
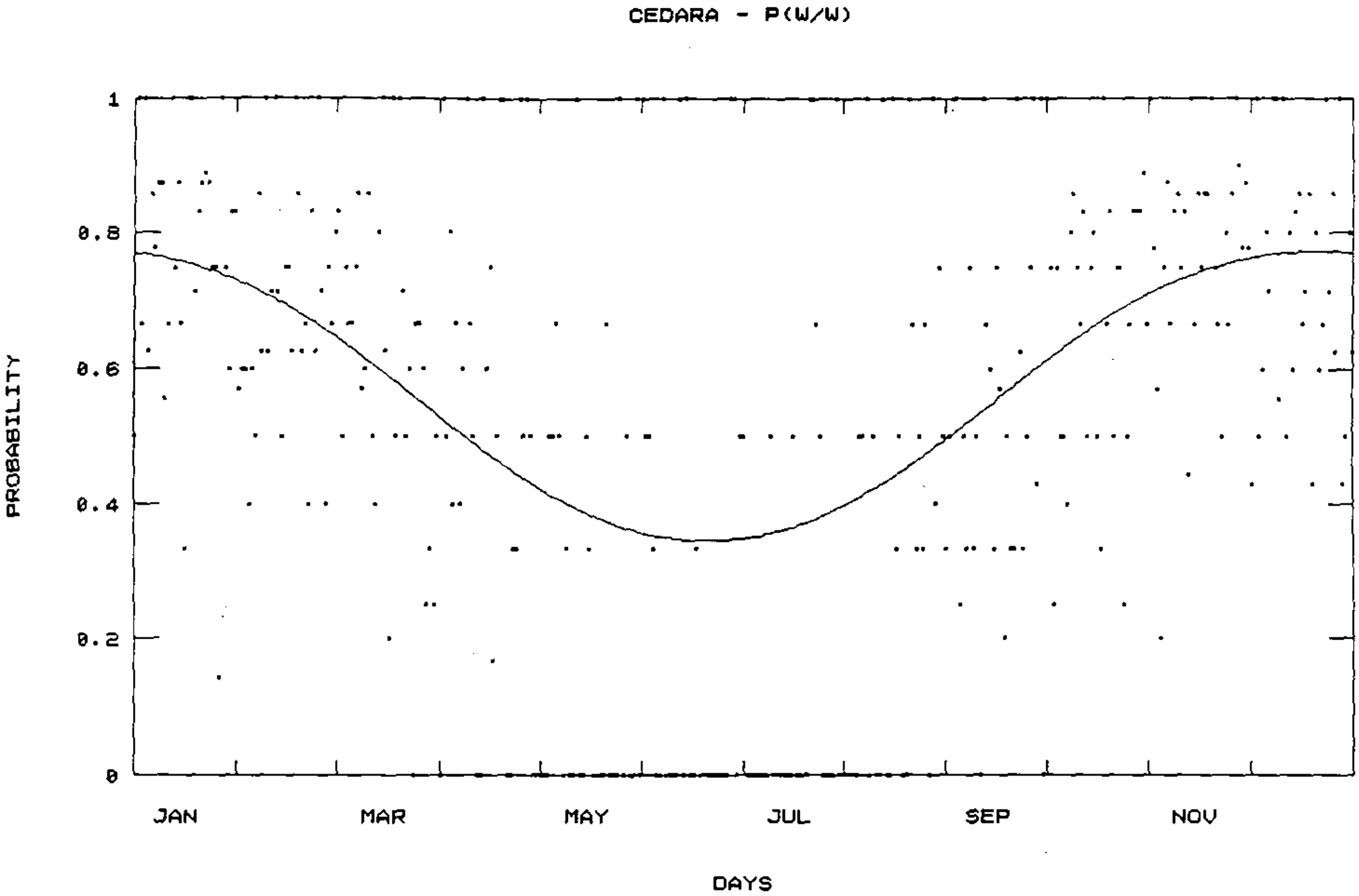


FIGURE 6.6 Empirical probabilities and estimates based on a 3 parameter model for $P(W|W)$ and $P(W|D)$ for Cedara



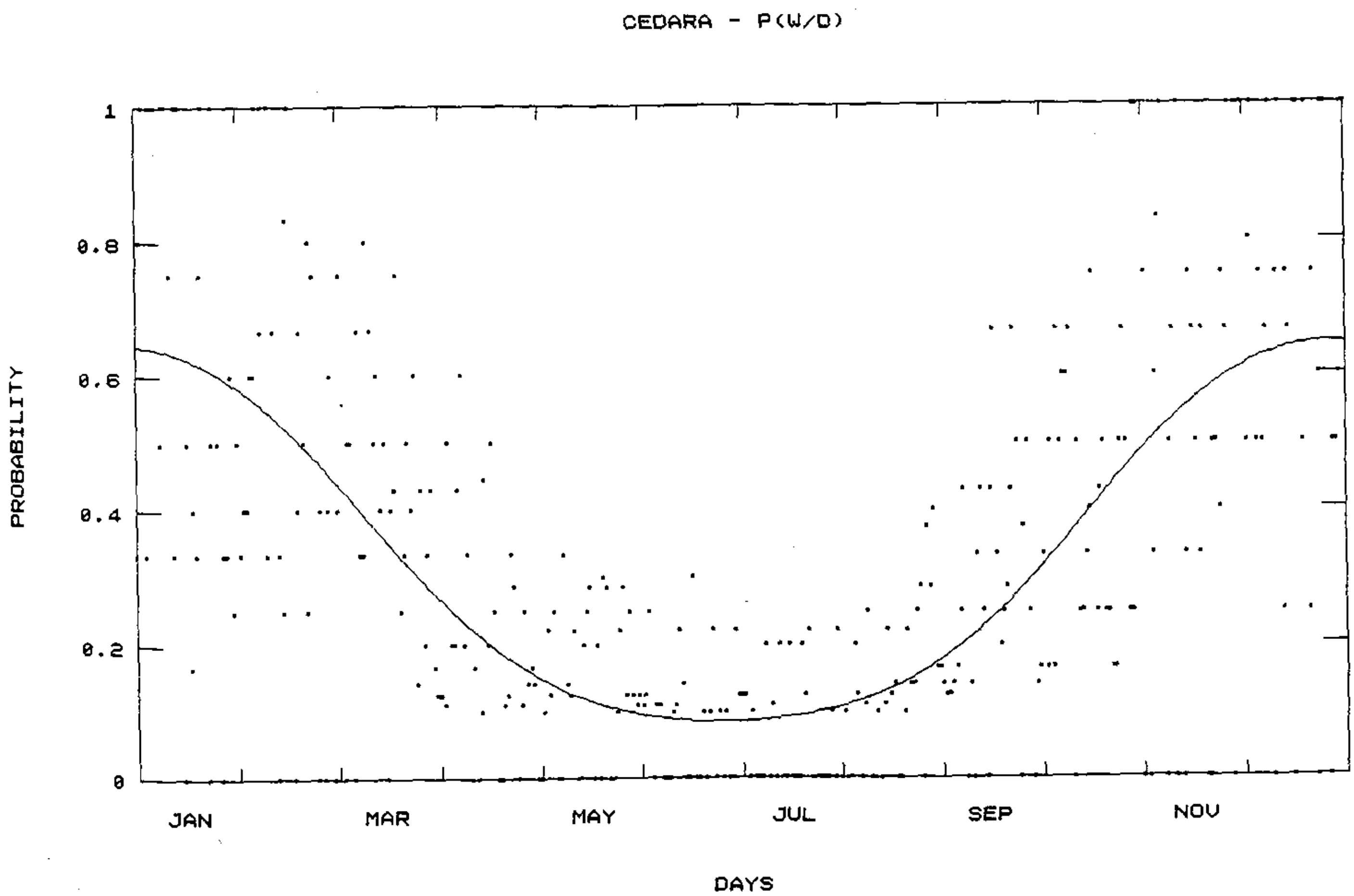
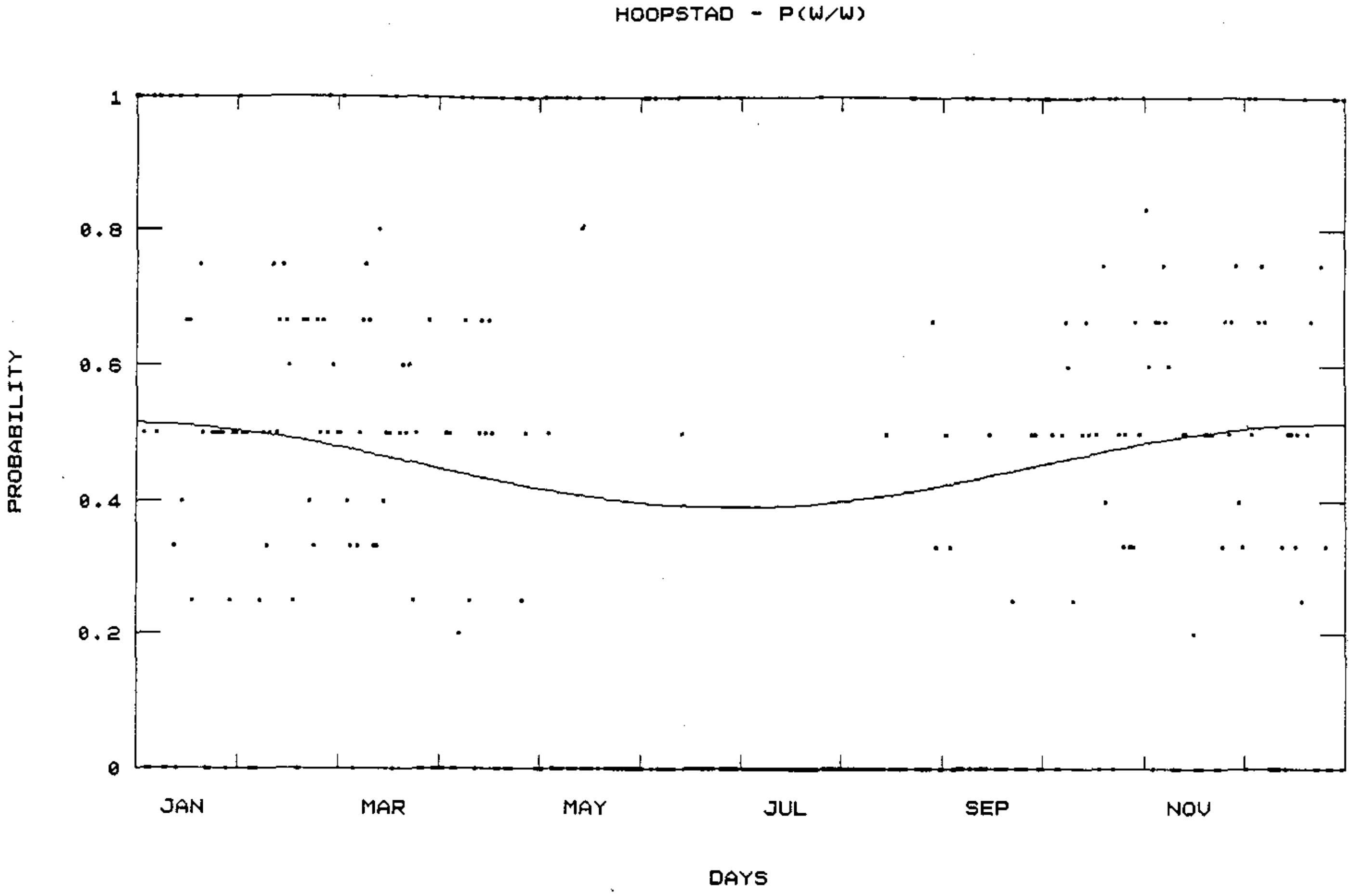
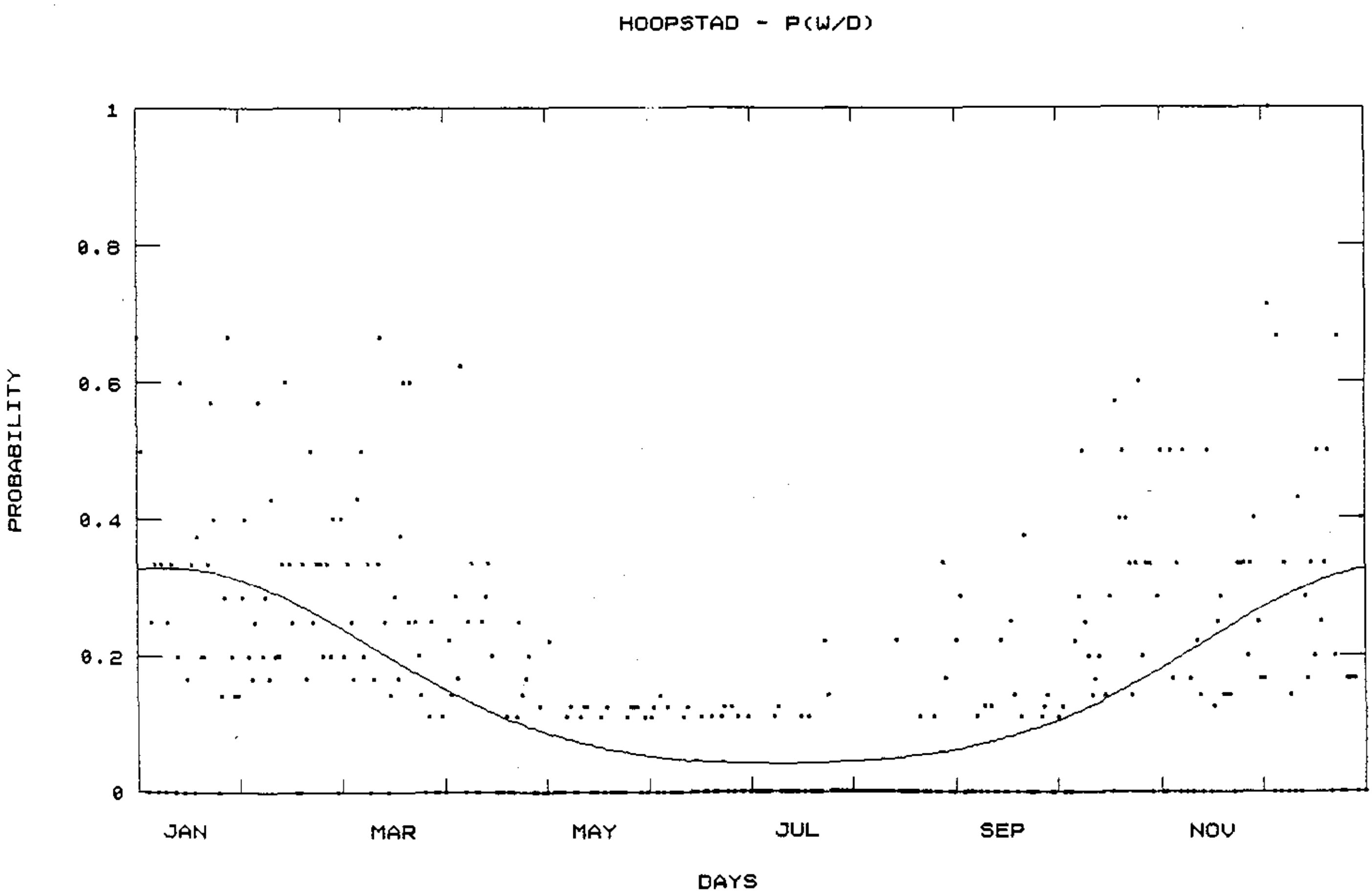


FIGURE 6.7 Empirical probabilities and estimates based on a 3 parameter model for $P(W|W)$ and $P(W|D)$ for Hoopstad





Validation of Climate Model

To consider the climate model as adequate in preserving the characteristics of the climate, the multivariate properties of the weather variables must be investigated as well as the univariate characteristics of each individual variable.

The following parameters and parameter functions must be preserved if one is to consider the climate model as satisfactory:

(a) the annual mean and standard deviation for each climate variable for the unconditioned data and the data conditioned on the wet or dry status of the day,

(b) the monthly means and standard deviations for each variable for the unconditioned data and the data conditioned on the wet or dry status of the day,

(c) the extreme values of each climate variable, i.e. maximum and minimum daily values,

(d) the autocorrelation within each variable for the unconditioned data and the data conditioned on the wet or dry status of the day,

(e) the cross–correlation over all climate variables.

The checks above test either the multivariate part of the climate model, e.g. the cross–correlation over all variables, or the individual characteristics of each variable, e.g. the monthly means and standard deviations for each variable.

Again fifty years of simulated daily climate sequences were compared with the historical data on an annual, monthly and daily basis.

The following abbreviations are used in tables and figures:

Max Temp — Maximum Temperature

Min Temp — Minimum Temperature

Evapo — Evaporation

Sun — Sunshine Duration

Wind — Wind run

Max Hum — Maximum Humidity

Min Hum — Minimum Humidity

His — Historical Data

Mod 1 — Simulated data using Model 1

Mod T — Simulated data using Model T.

Validation of annual properties

Table 6.2 shows the comparison of historical and simulated annual means for each variable and each station. This statistic has been adequately preserved by both models when the variables are conditioned on a wet day and when they are conditioned on the dry status of the day (Tables 6.3 – 6.4). There is however a slight underestimation of the annual mean for wet sequences by Model T for the variables wind run, maximum and minimum humidity at some of the stations. For Middelburg, the annual mean of wind run is slightly overestimated by Model T.

TABLE 6.2 Comparison of historical and simulated annual mean

Variable	Data	Station: Elsenburg	Kakamas	Middelburg	Nelspruit	Cedara	Hoopstad
	His	22.7	28.9	23.7	26.4	22.8	25.9
Max Temp	Mod 1	22.7	28.9	23.5	26.3	22.8	26.1
	Mod T	22.4	29.2	24.4	26.1	22.9	27.9
	His	10.5	13,1	6.8	13.3	10.1	7.9
Min Temp	Mod 1	10.7	13.1	6.7	13.3	10.1	8.1
	Mod T	10.5	13.2	6.8	13.4	10.2	8.4
	His	5.7	8.9	6.4			
Evapo	Mod 1	5.7	8.9	6.3			
	Mod T	5.6	9.1	6.5			
	His	8.3	9.8	9.2	7.3	6.9	9.2
Sun	Mod 1	8.2	9.8	9.1	6.7	6.5	9.0
	Mod T	8.3	9.8	9.3	7.2	6.9	9.4
	His	194.7	194.9	195.8	121.1	158.4	123.8
Wind	Mod 1	193.0	196.0	195.4	125.6	157.5	123.1
	Mod T	192.5	195.0	195.4	122.6	158.3	119.8
	His	92.5	62.2	80.3	81.8	88.8	74.0
Max Hum	Mod 1	92.8	62.9	80.4	82.0	89.0	74.0
	Mod T	92.8	63.2	80.9	82.2	89.0	73.8
	His	41.4	25.6	26.7	47.9	52.6	32.9
Min Hum	Mod 1	41.2	24.9	26.6	48.4	53.2	32.9
	Mod T	41.9	24.9	26.4	48.0	51.1	28.1

TABLE 6.3 Comparison of historical and simulated annual mean given a wet day

		Station					
Variable	Data	Elsenburg	Kakamas	Middelburg	Nelspruit	Cedara	Hoopstad
	His	18.1	28.9	24.0	26.2	22.3	26.8
Max Temp	Mod 1	18.1	27.8	23.5	25.9	22.3	27.1
	Mod T	19.0	30.5	26.5	26.4	22.5	30.2
	His	10.9	17.8	10.9	16.6	13.0	13.6
Min Temp	Mod 1	11.0	16.6	10.8	16.6	12.9	13.7
	Mod T	11.4	16.7	10.6	16.6	12.8	13.2
	His	2.8	7.2	5.2			
Evapo	Mod 1	2.6	6.4	5.0			
	Mod T	2.8	8.8	6.1			
	His	4.2	5.7	6.0	4.6	4.4	6.0
Sun	Mod 1	3.8	5.5	5.3	3.8	3.7	5.4
	Mod T	4.3	5.3	6.3	4.7	4.3	6.5
	His	245.8	219.2	205.1	121.0	171.1	142.8
Wind	Mod 1	242.9	215.0	203.6	126.2	169.9	138.7
	Mod T	235.4	213.5	211.3	123.8	170.3	129.7
	His	94.3	75.8	83.3	87.8	95.0	81.7
Max Hum	Mod 1	94.2	77.8	84.4	88.3	95.2	80.8
	Mod T	93.4	56.2	80.3	87.0	94.9	76.1
	His	55.1	39.2	36.4	61.6	68.6	47.3
Min Hum	Mod 1	55.0	38.4	36.6	62.9	69.5	47.3
	Mod T	54.0	29.7	31.5	59.5	65.8	34.2

TABLE 6.4 Comparison of historical and simulated annual mean given a dry day

Variable	Data	Station					
		Elsenburg	Kakamas	Middelburg	Nelspruit	Cedara	Hoopstad
Max Temp	His	24.1	28.9	23.6	26.5	23.3	25.6
	Mod 1	24.2	28.9	23.5	26.5	23.2	25.8
	Mod T	23.5	29.2	23.9	26.0	23.3	27.2
Min Temp	His	10.4	12.9	5.0	12.2	8.1	6.3
	Mod 1	10.5	12.9	5.8	12.1	8.1	6.6
	Mod T	10.2	13.0	6.0	12.2	8.4	7.1
Evapo	His	6.7	9.0	6.6			
	Mod 1	6.7	9.1	6.6			
	Mod T	6.6	9.1	6.6			
Sun	His	9.7	10.0	9.9	8.2	8.7	10.0
	Mod 1	9.7	10.0	9.9	7.8	8.5	10.0
	Mod T	9.6	10.0	9.9	8.1	8.6	10.1
Wind	His	117.7	193.7	193.9	121.2	149.4	118.6
	Mod 1	176.3	195.0	193.7	125.4	148.7	118.8
	Mod T	178.1	194.1	192.2	122.2	149.9	117.1
Max Hum	His	91.9	61.6	79.6	79.6	84.5	72.1
	Mod 1	92.3	62.1	79.5	79.8	84.7	72.2
	Mod T	92.6	63.5	81.1	80.5	84.9	73.2
Min Hum	His	36.8	25.0	24.7	43.1	41.3	29.1
	Mod 1	36.6	24.2	24.5	43.2	41.6	29.0
	Mod T	37.9	24.6	25.3	43.9	40.9	26.5

We note that there is a much smaller number of wet days in the year than there are dry days at the stations in this study. In particular Kakamas where observations of rainfall constitute only 4% of the data record. It would be therefore (statistically) surprising if all the parameter functions associated with wet days fitted the historical record very closely.

The annual standard deviation has been well described by both models for the cases when the variables are conditioned on the wet and dry status of the day as well as for the case when they are not (Tables 6.5 – 6.7). Again it is seen that for some stations, the standard deviation statistic for the simulated sequences of wind run and minimum humidity differ slightly from that of the historical record. In these instances, Model T generally performs better than Model 1.

TABLE 6.5 Comparison of historical and simulated annual standard deviation

		Station					
Variable	Data	Elsenburg	Kakamas	Middelburg	Nelspruit	Cedara	Hoopstad
	His	5.9	6.8	6..5	4.4	5.2	5.8
Max Temp	Mod 1	5.7	6.8	6.5	4.5	5.1	5.8
	Mod T	5.7	6.7	6.3	4.4	5.2	6.0
	His	3.8	6.6	6.1	5.2	5.1	7.1
Min Temp	Mod 1	3.7	6.6	6.1	5.2	4.9	7.1
	Mod T	3.6	6.5	6.2	5.2	4.8	7.0
	His	3.7	4.6	3.2			
Evapo	Mod 1	3.7	4.7	3.2			
	Mod T	3.7	4.8	3.3			
	His	3.6	2.4	3.0	3.6	3.7	2.7
Sun	Mod 1	3.6	2.2	3.0	3.8	4.1	2.9
	Mod T	3.6	2.2	3.0	3.9	4.1	2.8
	His	86.7	75.2	82.6	32.2	54.5	63.3
Wind	Mod 1	76.7	71.1	74.6	30.1	48.1	61.3
	Mod T	78.6	71.7	76.6	32.4	51.1	57.0
	His	6.9	22.0	17.5	10.7	12.5	15.9
Max Hum	Mod 1	5.7	21.8	16.3	10.4	12.2	15.4
	Mod T	6.1	22.1	15.3	10.8	12.4	14.3
	His	15.2	10.4	11.9	16.9	22.6	15.6
Min Hum	Mod 1	14.4	7.6	11.2	16.7	22.5	15.0
	Mod T	14.7	10.4	10.7	16.6	22.3	14.0

TABLE 6.6 Comparison of historical and simulated annual standard deviation given a wet day

		Station					
Variable	Data	Elsenburg	Kakamas	Middelburg	Nelspruit	Cedara	Hoopstad
	His	4.5	6.2	7.1	5.1	5.6	5.6
Max Temp	Mod 1	4.2	5.8	7.1	5.1	5.6	5.8
	Mod T	5.0	6.9	6.6	4.9	5.7	6.0
	His	3.3	5.4	5.1	3.3	3.5	4.2
Min Temp	Mod 1	3.4	5.1	5.1	3.5	3.6	4.3
	Mod T	3.5	5.5	5.4	4.0	3.8	5.0
	His	2.6	3.9	3.2			
Evapo	Mod 1	2.5	3.4	3.2			
	Mod T	2.8	5.4	4.0			
	His	3.4	3.3	3.6	3.6	3.6	3.5
Sun	Mod 1	3.4	3.2	3.7	4.0	4.3	3.8
	Mod T	3.8	3.6	4.3	4.6	4.6	4.2
	His	113.7	68.2	85.3	37.7	56.5	65.3
Wind	Mod 1	96.6	54.4	75.0	33.7	50.4	60.6
	Mod T	100.8	64.3	82.6	37.4	54.8	62.1
	His	4.1	20.0	16.1	8.0	5.5	13.1
Max Hum	Mod 1	4.5	16.6	12.3	8.0	5.0	12.9
	Mod T	6.1	25.7	14.7	9.5	5.5	14.5
	His	15.4	14.6	15.4	15.1	18.8	19.1
Min Hum	Mod 1	13.4	7.6	13.6	14.6	18.5	17.6
	Mod T	15.4	15.6	14.2	16.9	20.2	20.4

One of the difficulties that arose when modelling climate variables was that the variables are bounded with values lying outside these boundaries being inadmissible, for example, having negative sunshine. Also, some variables have a high frequency of values near or on an upper or lower limit so that it is expected that simulated sequences will occasionally have values that exceed these boundaries. Transformations were applied to the climate variables to overcome this problem. To verify that the simulated sequences of climate variables were adequately restrained within their boundaries and at the same time that extreme values simulated closely resemble those of the historical record, the maximum and minimum values simulated for each variable were compared with those observed in the

TABLE 6.7 Comparison of historical and simulated annual standard deviation given a dry day

		Station					
Variable	Data	Elsenburg	Kakamas	Middelburg	Nelspruit	Cedara	Hoopstad
	His	5.5	6.8	6.3	4.1	4.8	5.8
Max Temp	Mod 1	5.2	6.8	6.4	4.2	4.7	5.8
	Mod T	5.4	6.6	6.2	4.2	4.8	5.9
	His	3.9	6.6	5.9	5.2	5.0	6.9
Min Temp	Mod 1	3.9	6.7	6.0	5.1	4.7	6.9
	Mod T	3.6	6.5	6.0	5.1	4.6	6.9
	His	3.5	4.6	3.2			
Evapo	Mod 1	3.4	4.7	3.1			
	Mod T	3.4	4.8	3.1			
	His	2.5	2.1	2.3	3.0	2.4	1.8
Sun	Mod 1	2.2	1.8	2.1	3.1	2.3	1.5
	Mod T	2.4	1.9	2.3	3.2	2.5	1.5
	His	67.5	75.3	81.9	29.9	51.2	61.6
Wind	Mod 1	60.1	71.7	74.3	28.6	44.3	60.9
	Mod T	63.4	72.0	74.9	30.3	46.6	55.2
	His	7.5	21.9	17.7	10.7	14.0	15.9
Max Hum	Mod 1	6.0	21.8	16.9	10.2	13.9	15.5
	Mod T	6.1	21.8	15.4	10.7	14.1	14.2
	His	12.0	9.7	9.9	14.6	17.8	11.9
Min Hum	Mod 1	11.5	7.0	9.4	14.2	17.3	11.4
	Mod T	11.9	9.9	9.5	14.5	17.4	11.2

historical record. Tables 6.8 and 6.9 show these comparisons.

As can be seen from the tables, the extreme values simulated compare favourably with those observed in the historical record. Even for those variables that show a slight difference in the extreme values, when a count was taken of those values of the simulated sequence that lay either above the maximum or below the minimum values observed in the historical data, the percentage of such values was found to be negligible, that is, the highest percentage observed was 0.4%.

TABLE 6.8 Comparison of historical and simulated minimum values

		Station					
Variable	Data	Elsenburg	Kakamas	Middelburg	Nelspruit	Cedara	Hoopstad
Max Temp	His	10.0	9.5	5.0	13.1	7.8	5.6
	Mod 1	6.9	7.4	2.6	8.5	4.5	5.4
	Mod T	4.8	7.2	1.1	8.6	3.7	3.9
Min Temp	His	1.7	-2.5	-8.0	0.0	-4.0	-8.1
	Mod 1	1.1	-2.3	-8.2	0.9	-10.4	-7.8
	Mod T	1.3	-3.0	-8.3	0.7	-5.5	-8.3
Evapo	His	0.0	0.5	0.0			
	Mod 1	0.0	0.5	0.0			
	Mod T	0.0	0.1	0.0			
Sun	His	0.0	0.0	0.0	0.0	0.0	0.0
	Mod 1	0.0	0.1	0.0	0.0	0.0	0.0
	Mod T	0.0	0.0	0.0	0.0	0.0	0.0
Wind	His	40.0	35.1	62.2	26.0	22.0	14.0
	Mod 1	17.0	34.0	17.3	45.3	41.9	14.0
	Mod T	19.3	32.8	23.0	34.4	37.2	15.7
Max Hum	His	50.0	12.0	18.0	30.0	22.0	20.0
	Mod 1	38.4	11.3	8.9	19.8	10.4	20.3
	Mod T	34.5	1.9	17.5	18.7	6.7	18.9
Min Hum	His	12.0	1.0	5.0	9.0	2.0	7.0
	Mod 1	8.7	7.1	3.7	3.1	1.2	3.6
	Mod T	8.6	0.6	3.0	3.9	0.6	0.3

TABLE 6.9 Comparison of historical and simulated maximum values

		Station					
Variable	Data	Elsenburg	Kakamas	Middelburg	Nelspruit	Cedara	Hoopstad
Max Temp	His	40.8	43.8	38.0	39.8	37.3	39.0
	Mod 1	42.1	44.6	37.8	39.5	38.0	39.5
	Mod T	42.3	47.4	39.3	40.1	38.2	40.6
Min Temp	His	20.9	29.8	22.5	23.3	21.1	21.4
	Mod 1	25.3	28.6	21.1	22.9	23.1	21.1
	Mod T	24.1	28.8	21.6	26.9	22.9	21.1
Evapo	His	18.5	24.0	18.0			
	Mod 1	18.8	23.7	20.2			
	Mod T	22.8	28.2	28.3			
Sun	His	13.3	13.5	13.6	12.9	13.0	13.4
	Mod 1	13.7	14.0	13.9	13.0	13.2	13.4
	Mod T	13.7	14.0	13.9	13.0	13.2	13.5
Wind	His	733.3	531.0	583.1	420.0	453.0	396.0
	Mod 1	705.1	583.7	614.1	301.7	468.3	584.3
	Mod T	681.3	564.6	640.7	341.5	449.7	592.3
Max Hum	His	100.0	100.0	100.0	100.0	100.0	100.0
	Mod 1	100.0	100.0	100.0	100.0	100.0	100.0
	Mod T	100.0	100.0	100.0	100.0	100.0	100.0
Min Hum	His	95.0	80.0	85.0	97.0	100.0	97.0
	Mod 1	93.9	62.5	86.2	94.5	99.7	93.5
	Mod T	95.9	90.2	86.7	98.2	99.3	99.3

Validation of monthly properties

It is important that the monthly characteristics of each climate variable, mainly the mean and standard deviation, be adequately described by the models. The monthly means and standard deviations of the simulated sequences for each station were compared to those of the respective historical record. Figures 6.8 – 6.14 show the monthly means of each station for the various climate variables.

From the figures it can be seen that the monthly means have been successfully preserved by both models. Model T slightly overestimates the monthly means of maximum temperature for the station Hoopstad, but the highest difference between the means of the simulated sequence and that of the historical data still lies within 3°C of the observed monthly mean. Model T fails to preserve the monthly means for the variable minimum humidity of the station Hoopstad. Here the monthly means are underestimated. Model 1 fits the data reasonably well for this variable.

Figures 6.15 – 6.21 show the monthly standard deviations of each station for the various climate variables. The monthly standard deviations have been preserved by both models. The variables wind run, maximum humidity and minimum humidity show the greatest differences between the standard deviations of the observed sequence and those of the generated sequence. For these variables the models tend to slightly underestimate the monthly standard deviations. Looking at the original sequence of these three variables we see that they do not follow an approximate sinusoidal shape, one of the assumptions made when fitting the mean by a truncated Fourier series, so it seems that these observable differences may be accountable for this.

The mean and standard deviation functions of each variable differ significantly depending on the wet or dry status of the day. It is therefore necessary that monthly means and standard deviations should also be preserved by the models when the climate variables are conditioned on wet and dry days. Figures 6.22 – 6.28 show the monthly means for each station when the climate variables are conditioned on wet days. Figures 6.29 – 6.35 gives the monthly means when the climate variables are conditioned on the dry days.

FIGURE 6.8 Monthly means for maximum temperature

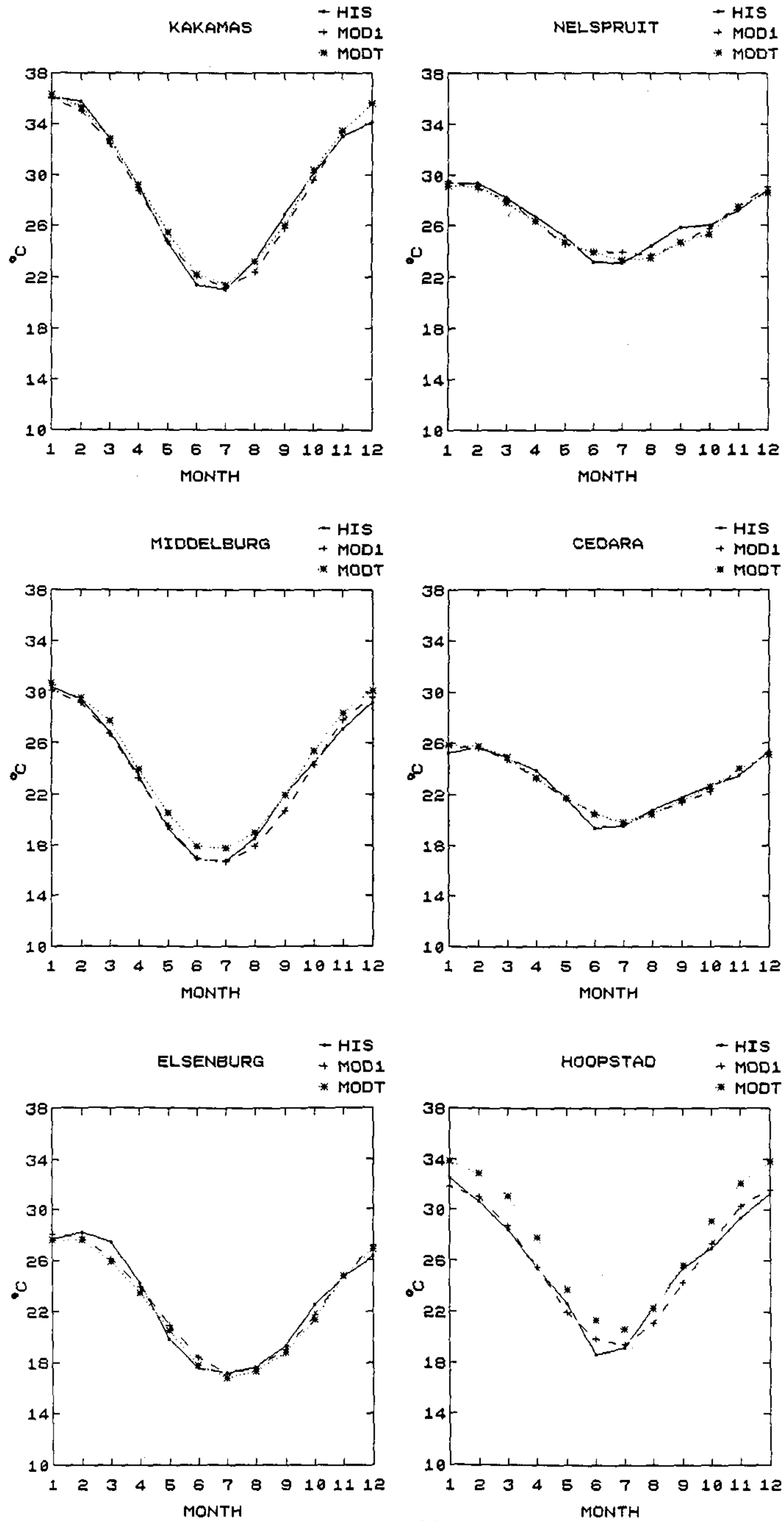


FIGURE 6.9 Monthly means for minimum temperature

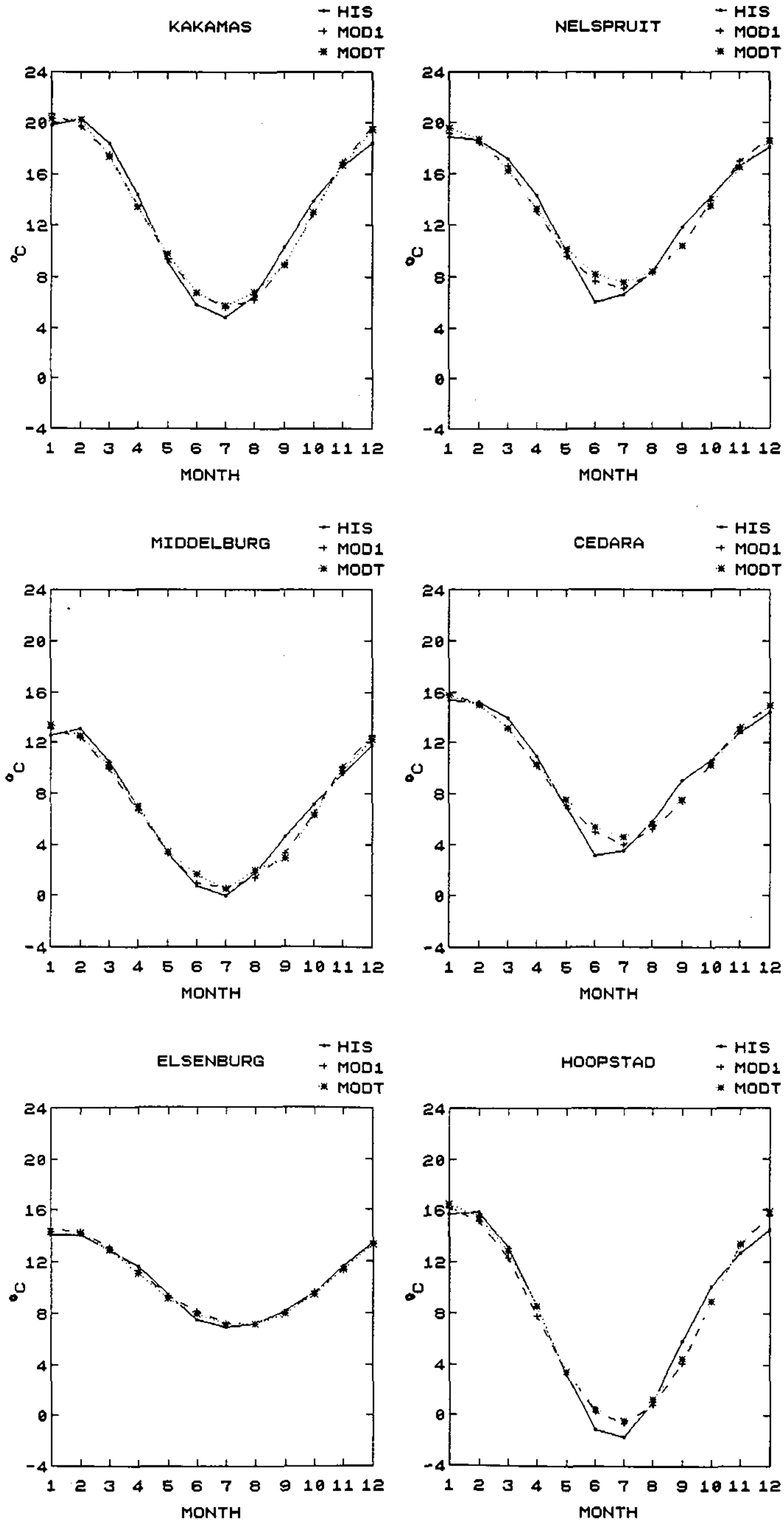


FIGURE 6.10 Monthly means for evaporation

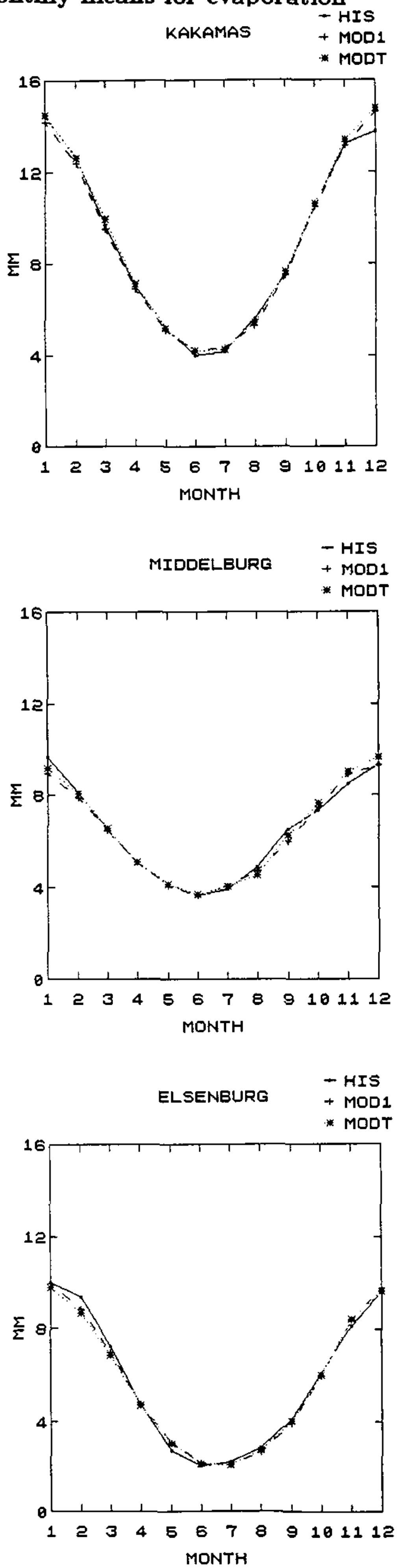


FIGURE 6.11 Monthly means for sunshine duration

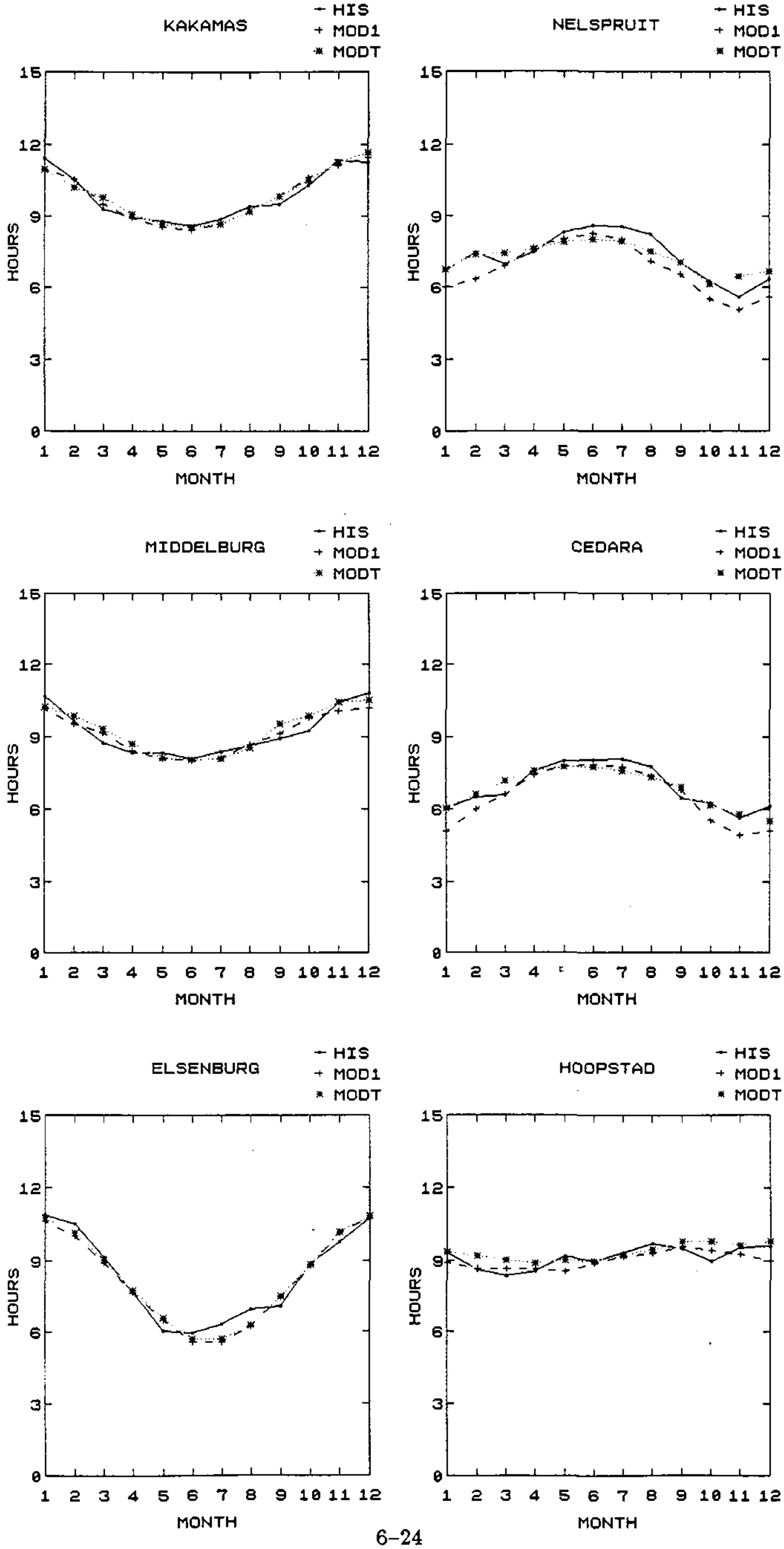


FIGURE 6.12 Monthly means for wind run

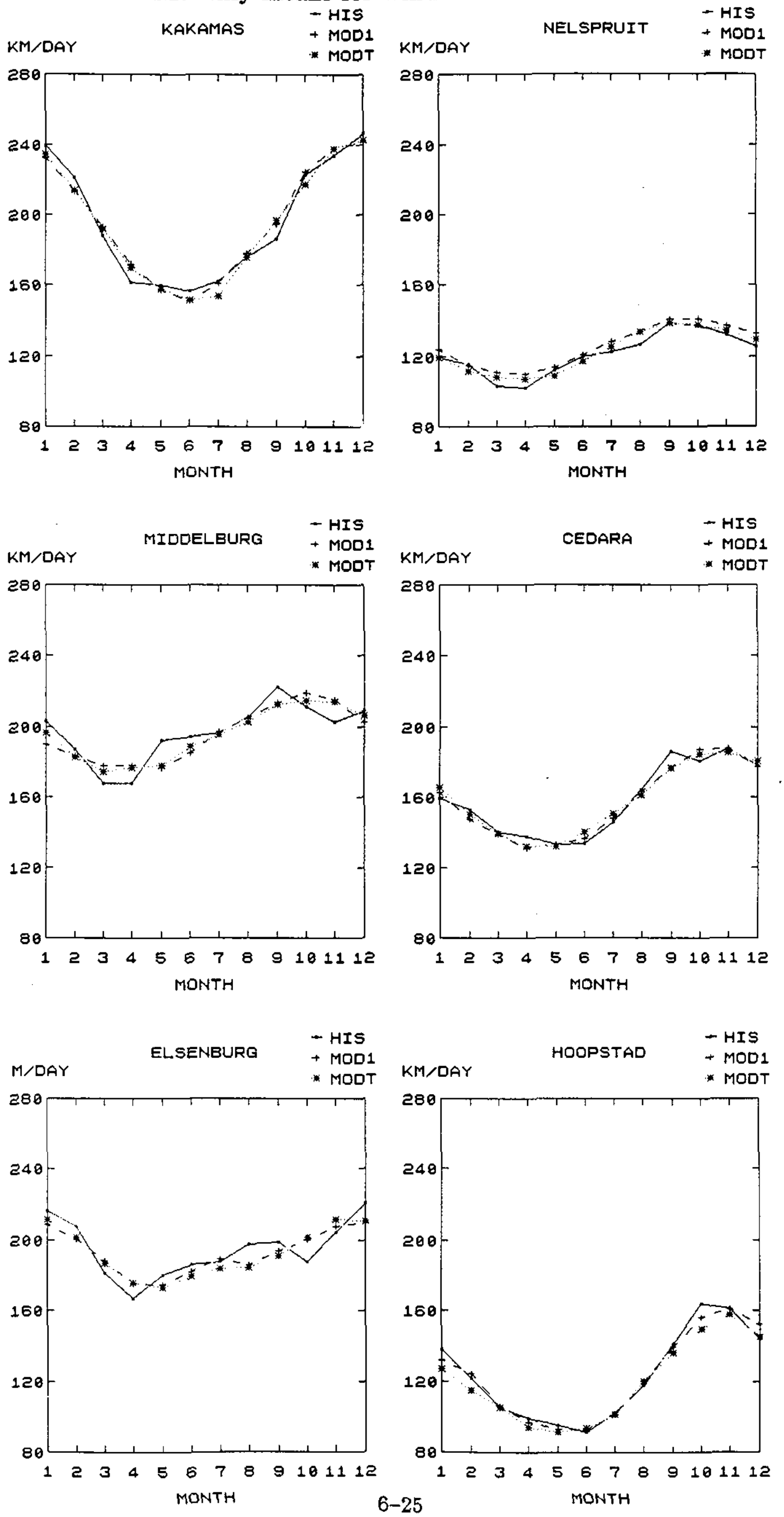


FIGURE 6.13 Monthly means for maximum humidity

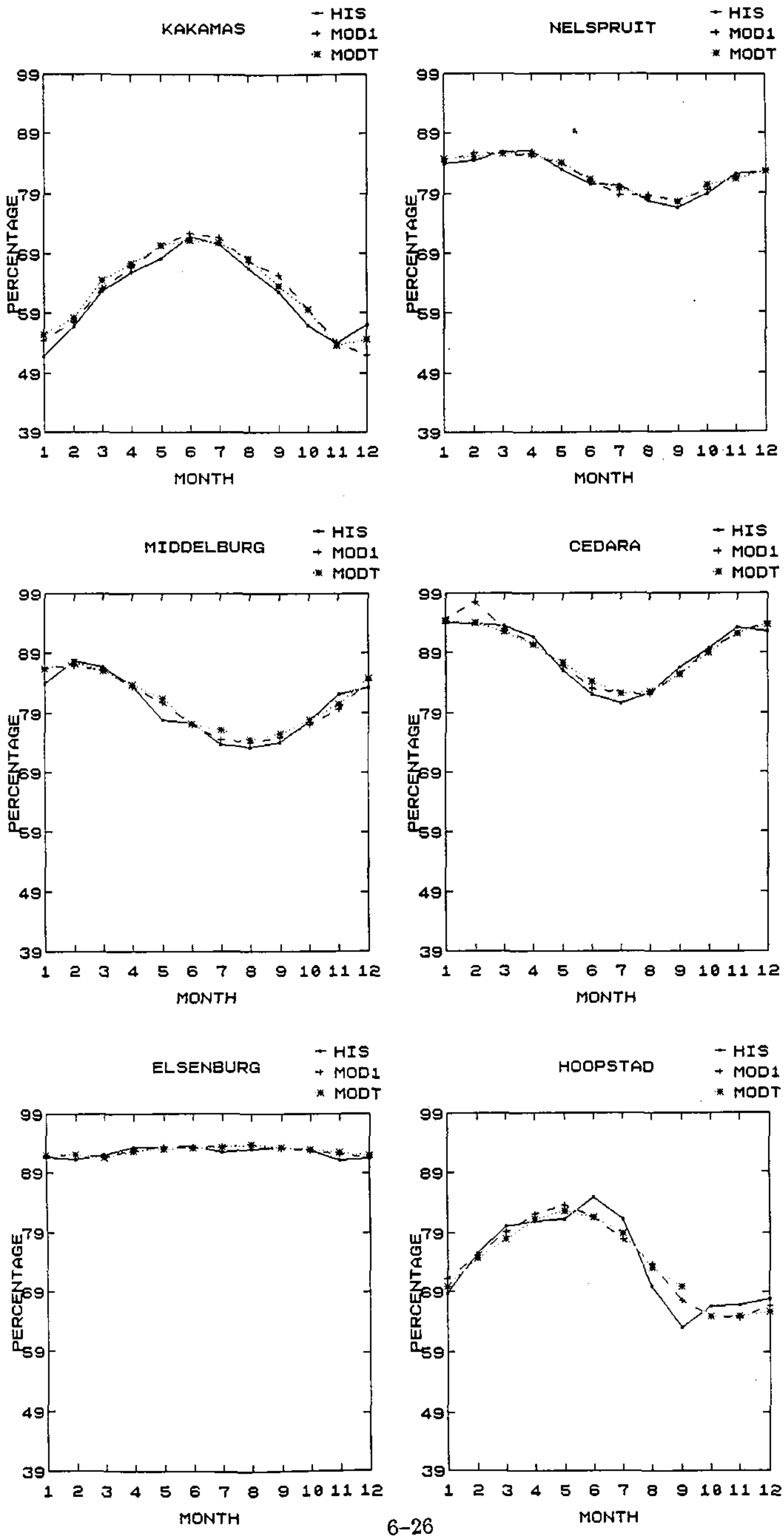


FIGURE 6.14 Monthly means for minimum humidity

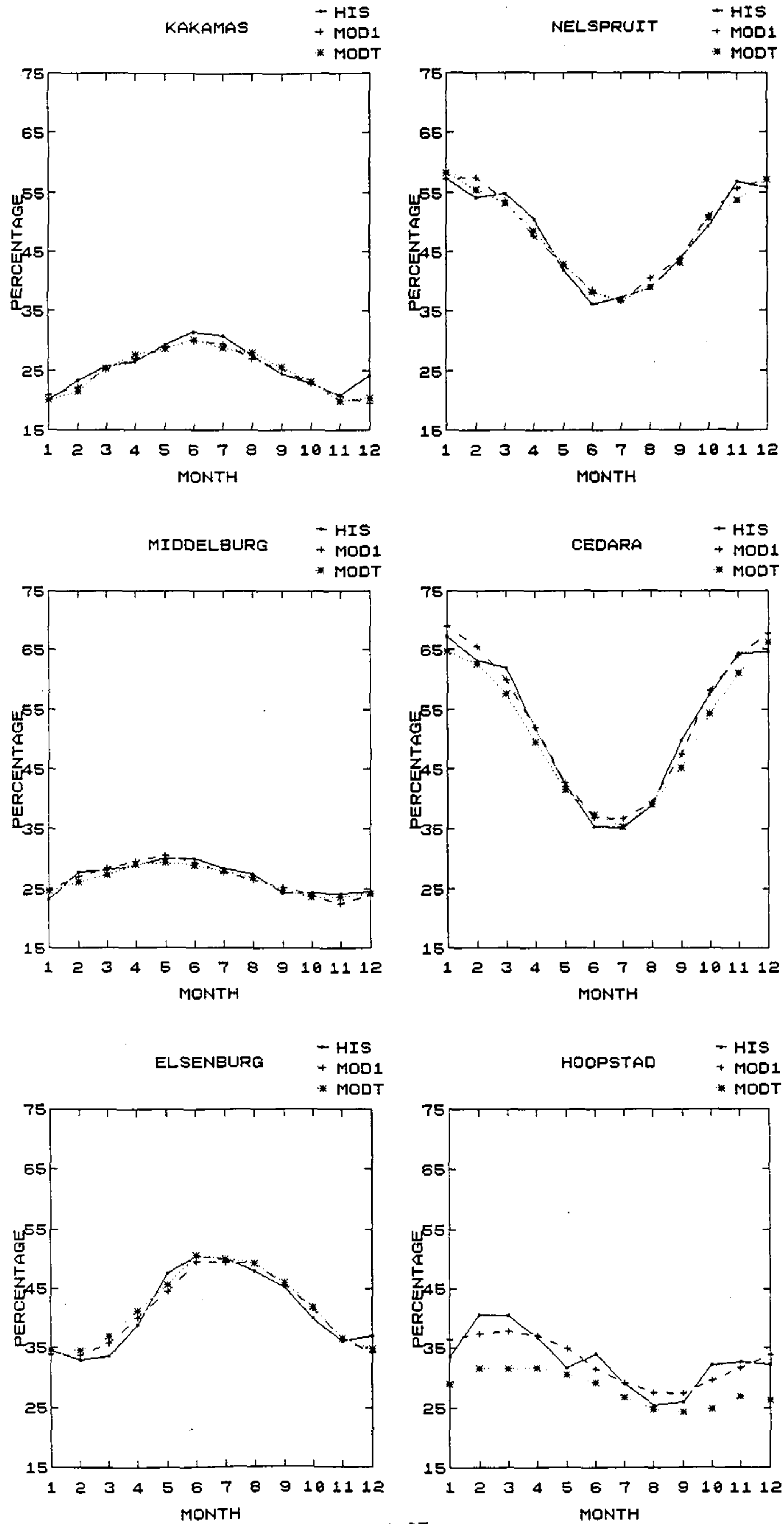


FIGURE 6.15 Monthly standard deviations for maximum temperature

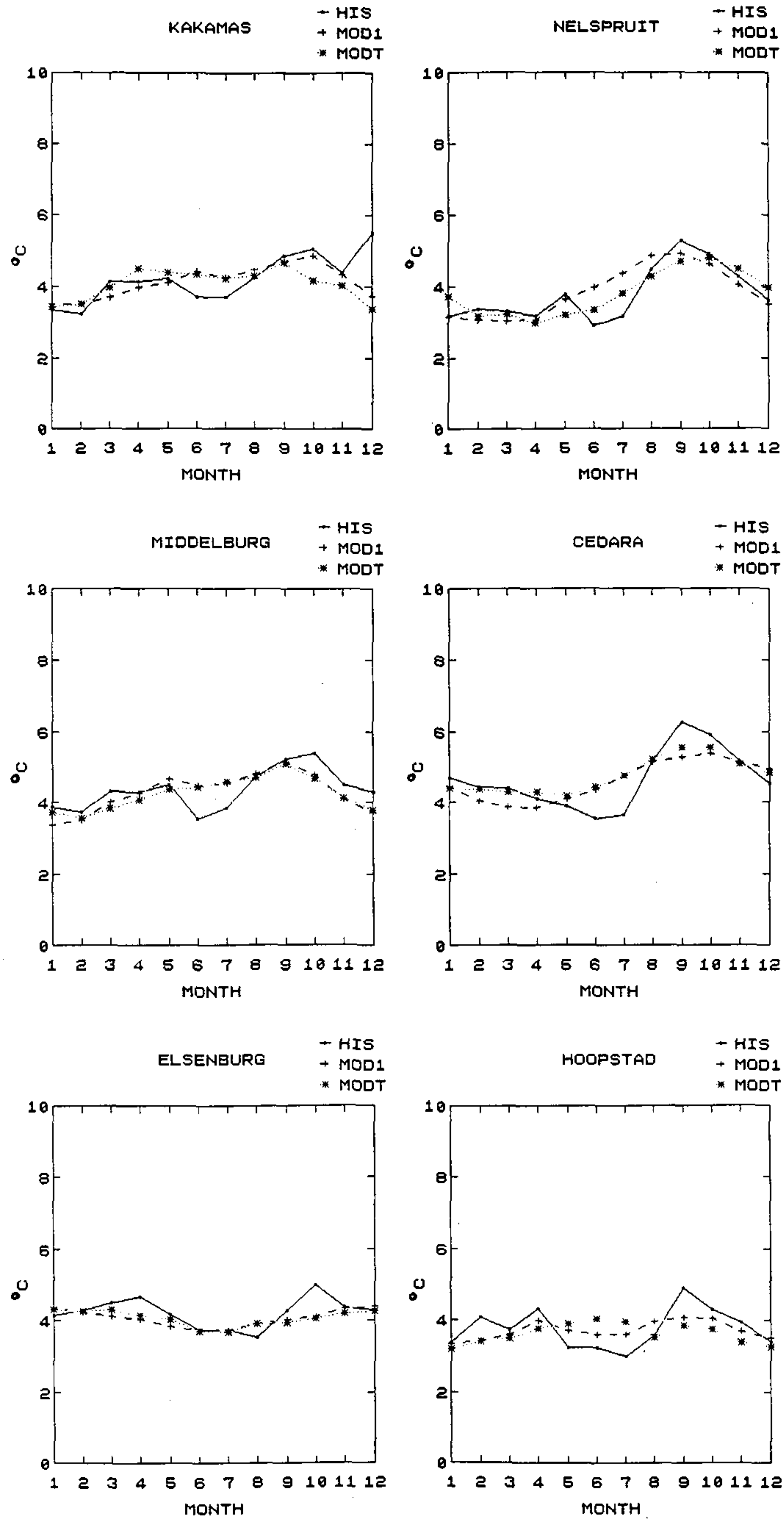


FIGURE 6.16 Monthly standard deviations for minimum temperature

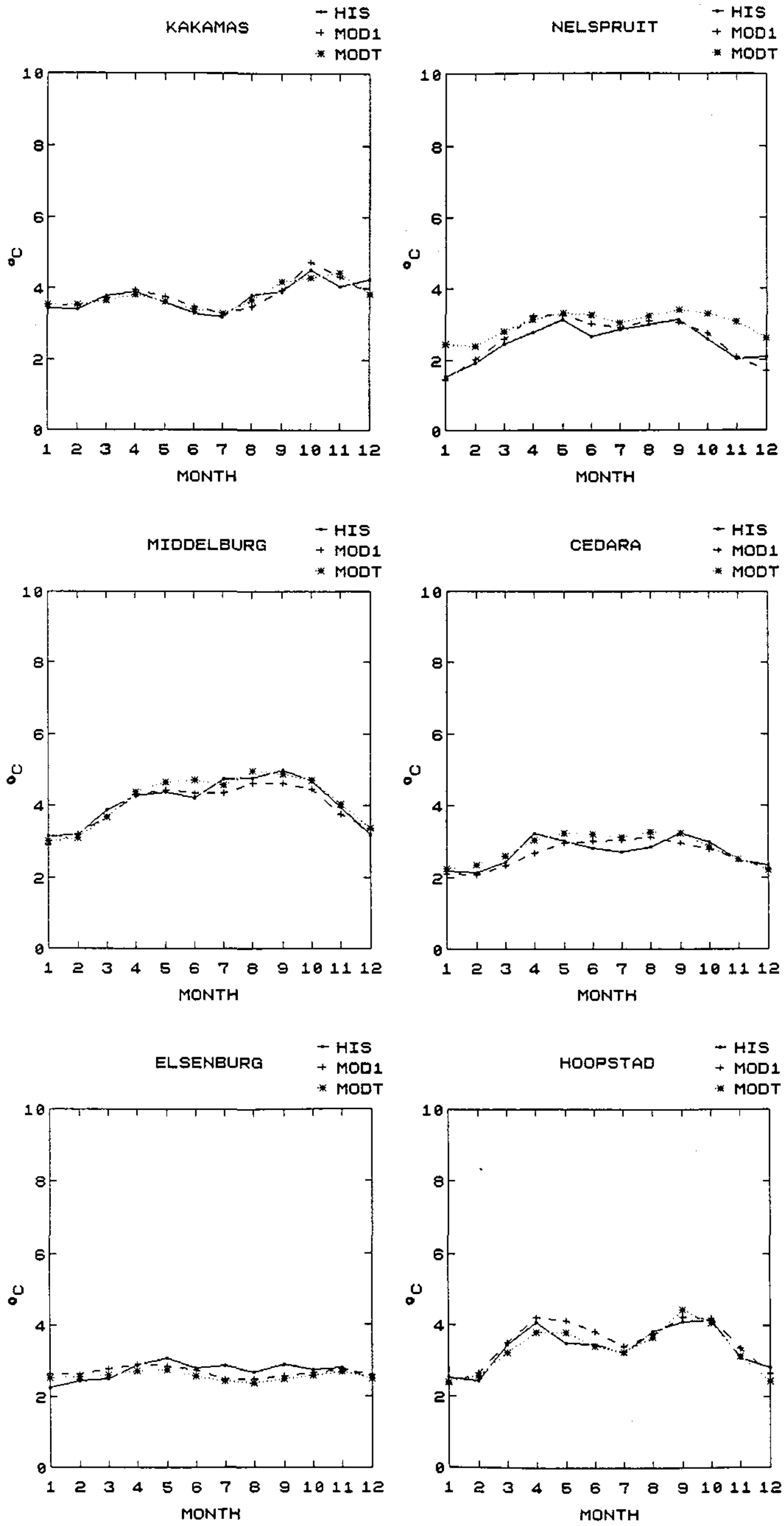


FIGURE 6.17 Monthly standard deviations for evaporation

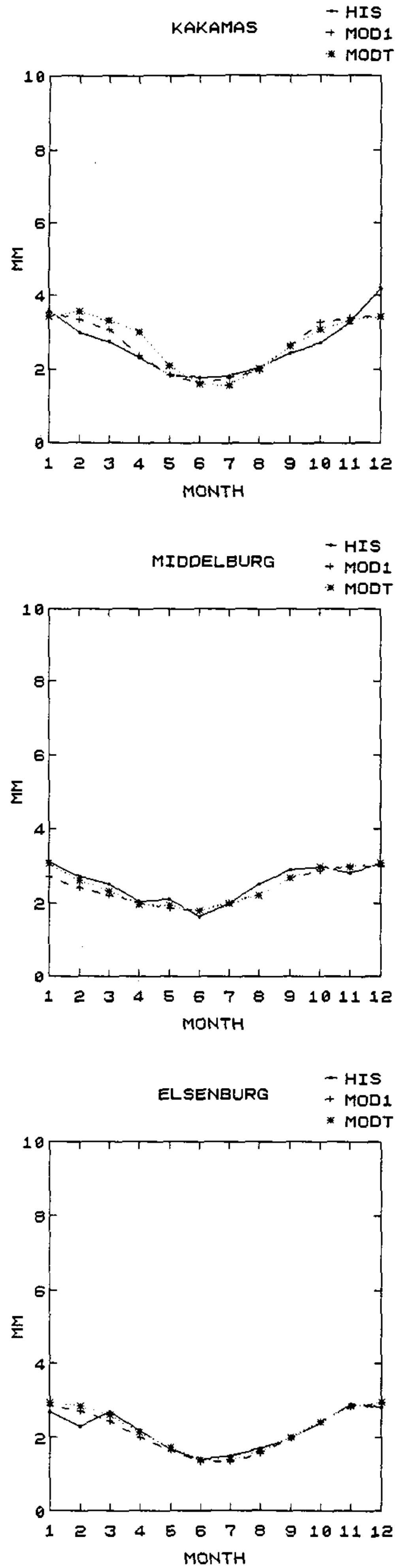


FIGURE 6.18 Monthly standard deviations for sunshine duration

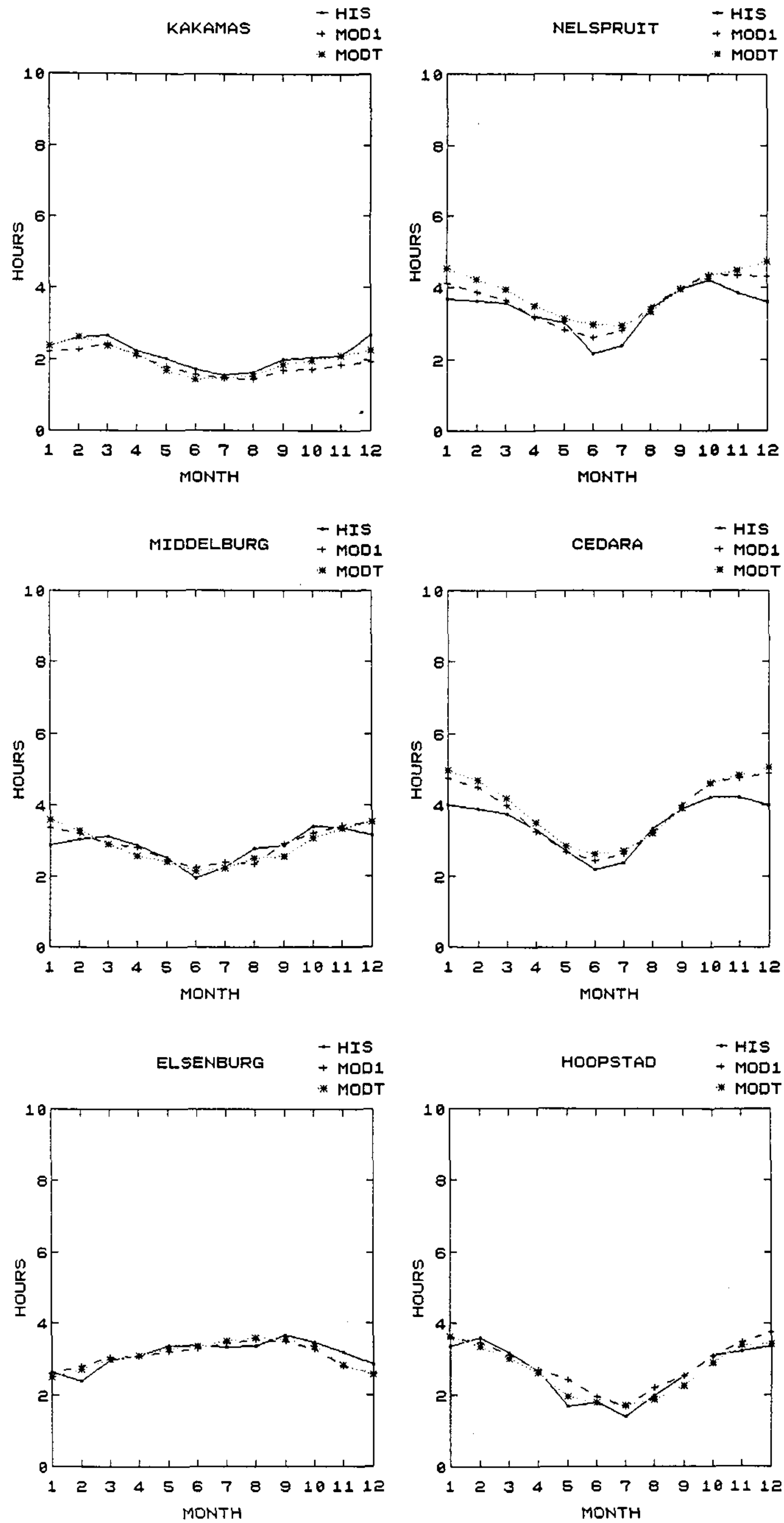


FIGURE 6.19 Monthly standard deviations for wind run

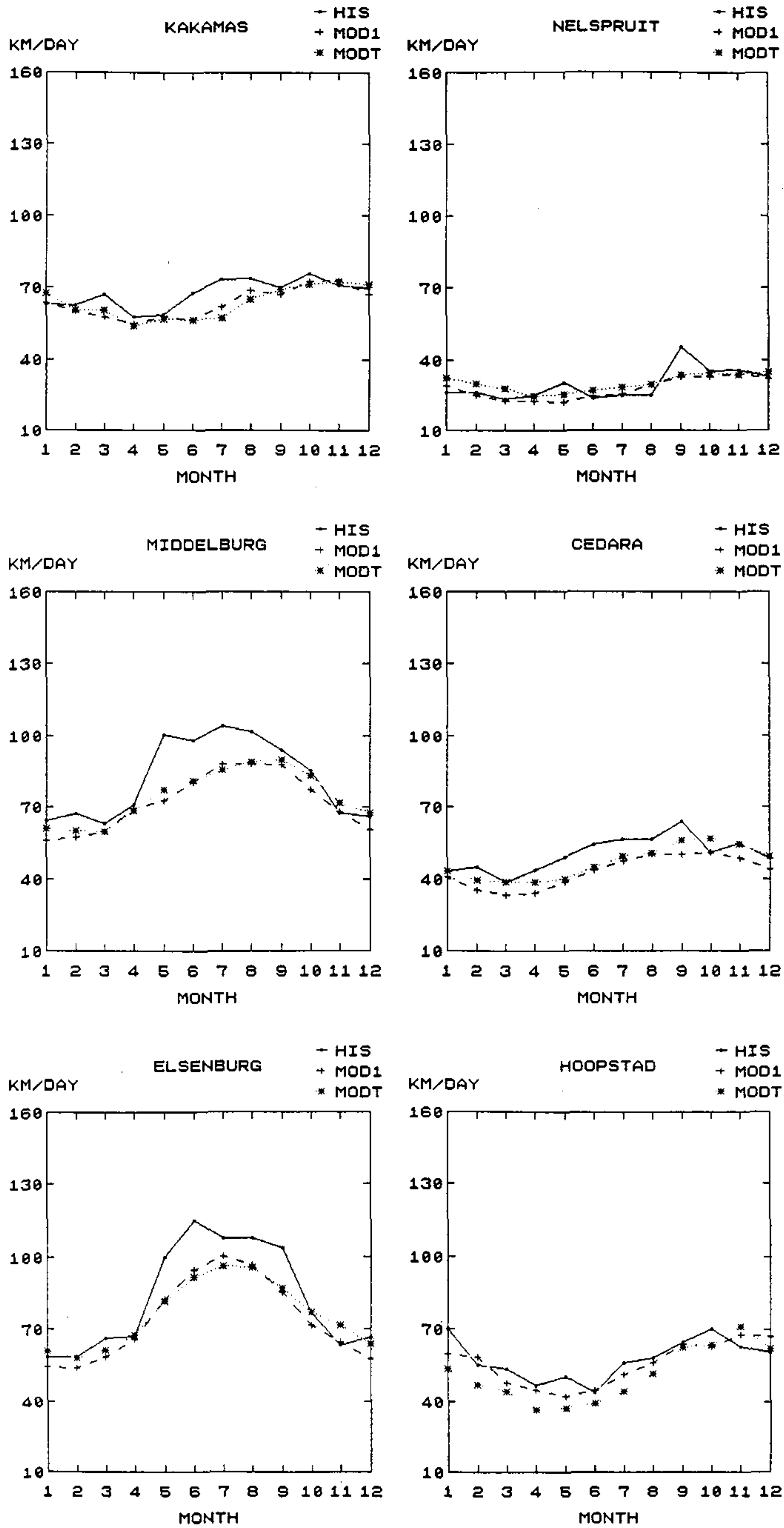


FIGURE 6.20 Monthly standard deviations for maximum humidity

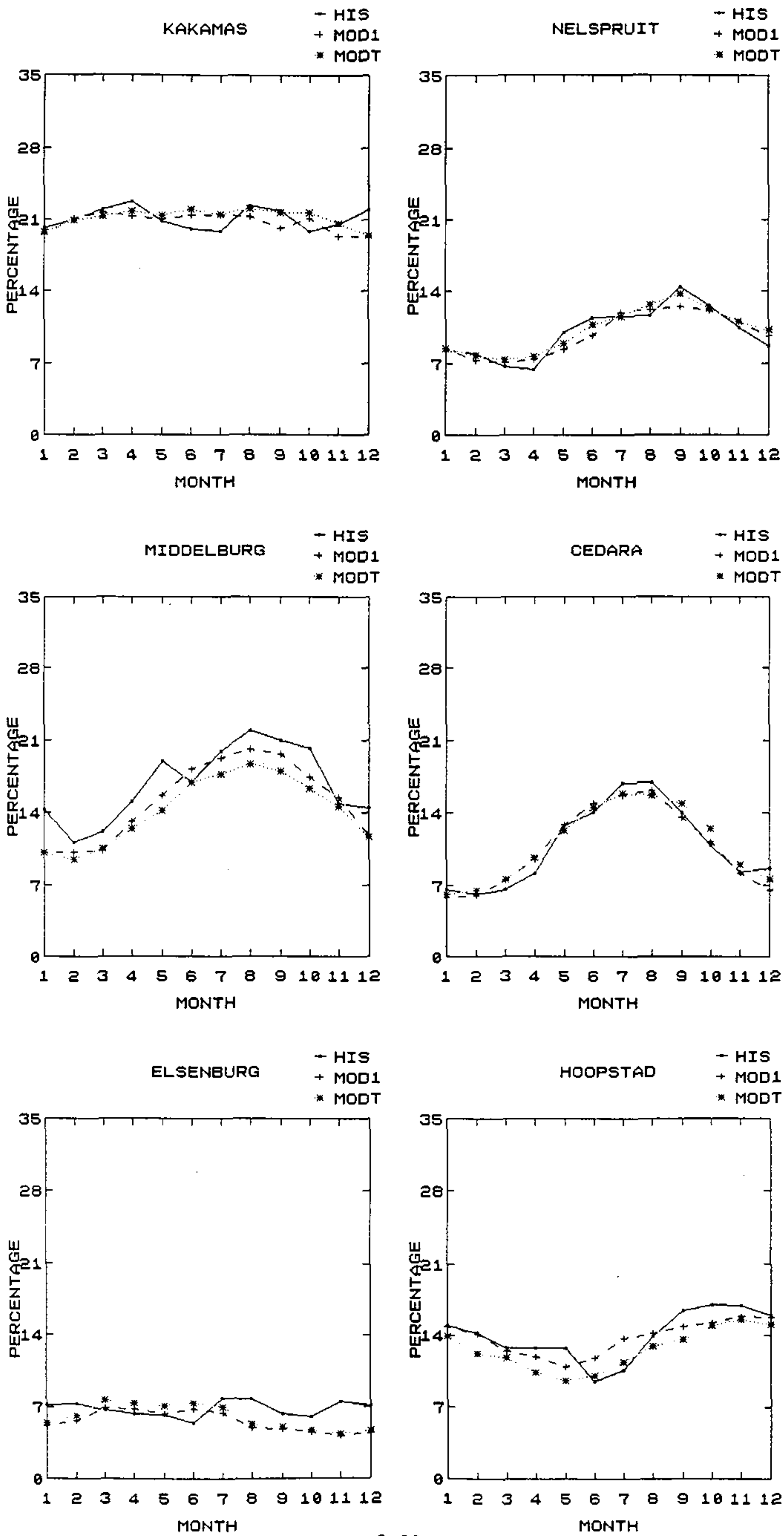


FIGURE 6.21 Monthly standard deviations for minimum humidity

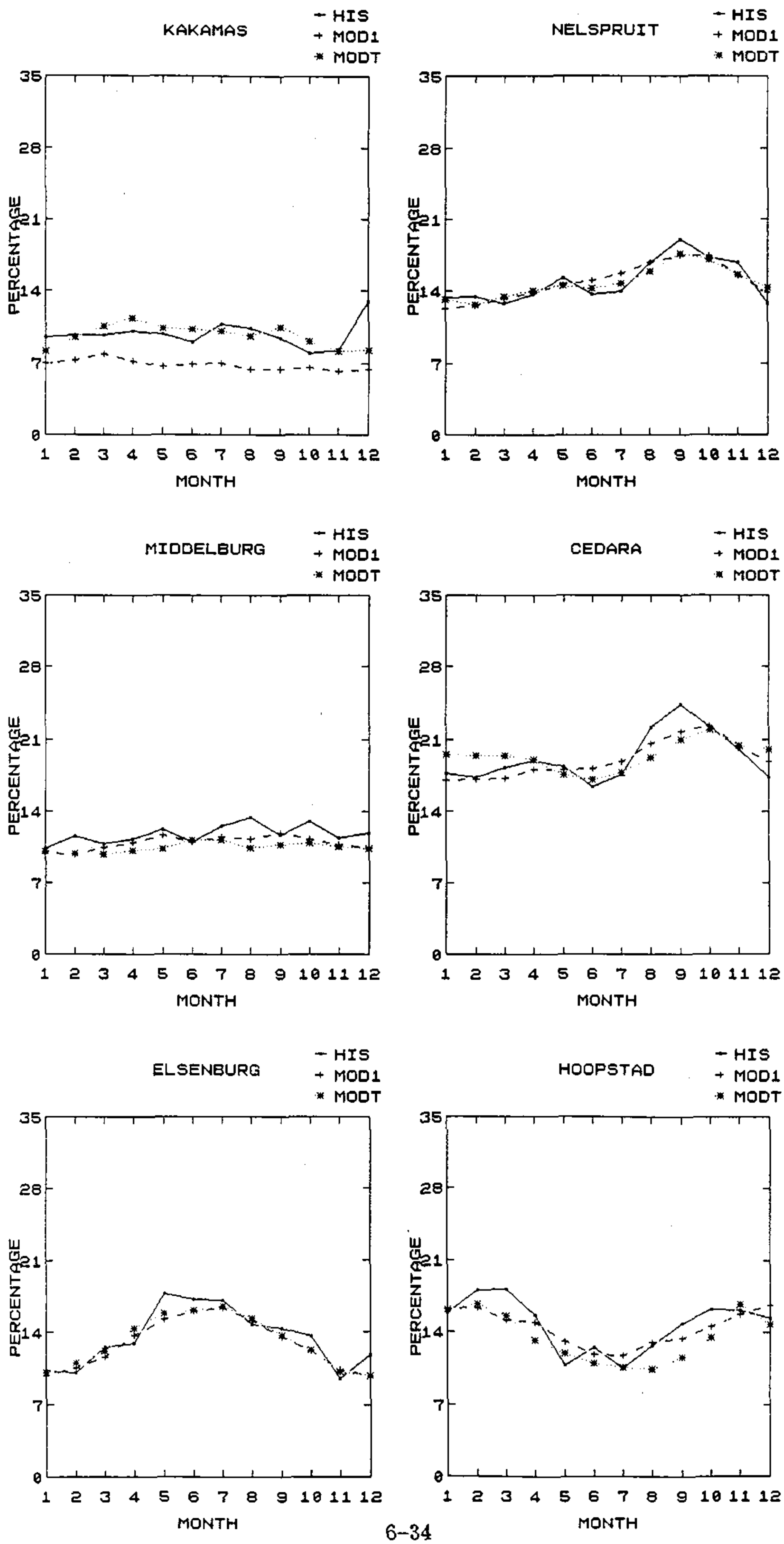


FIGURE 6.22 Monthly means for maximum temperature for wet days

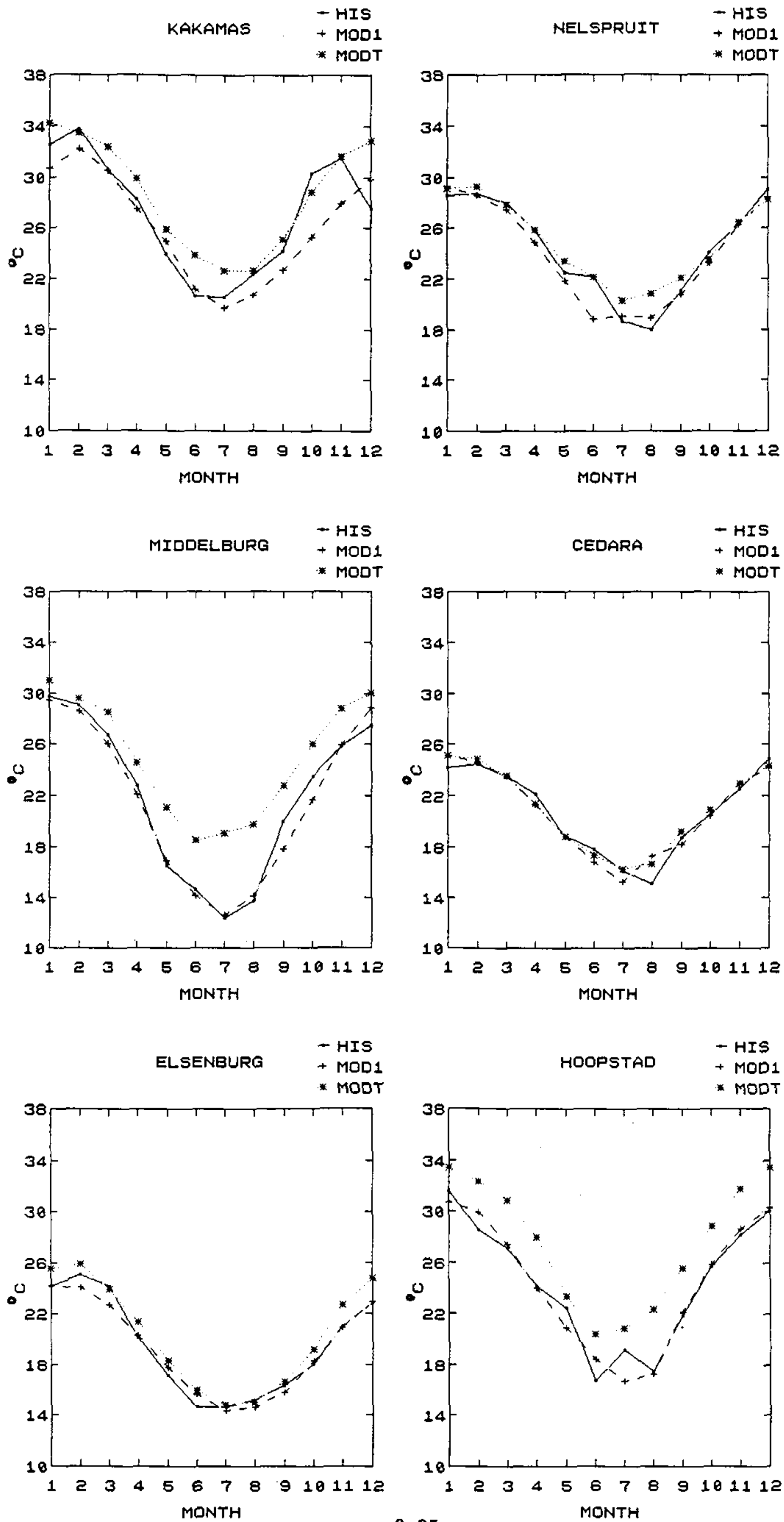


FIGURE 6.23 Monthly means for minimum temperature for wet days

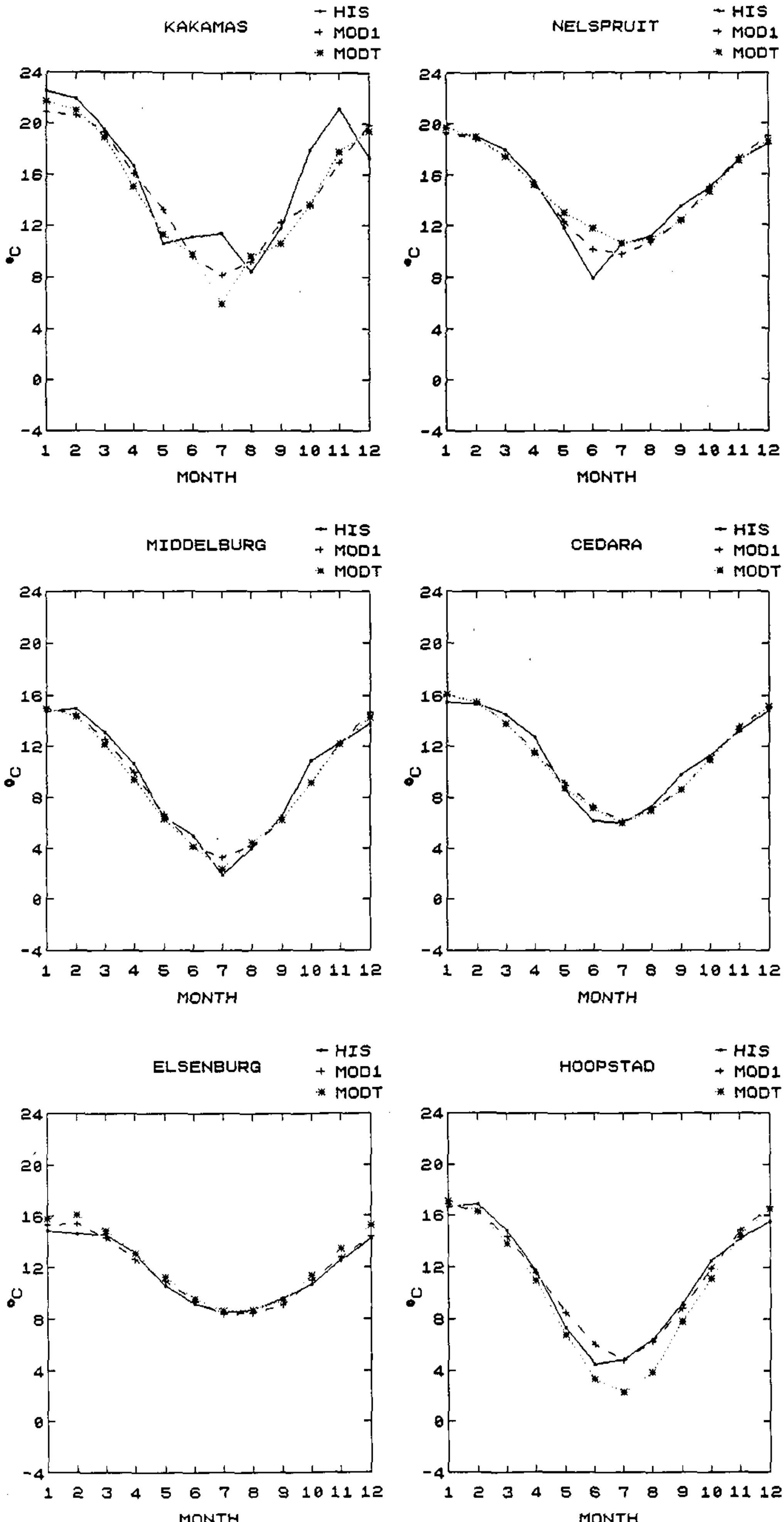


FIGURE 6.24 Monthly means for evaporation for wet days

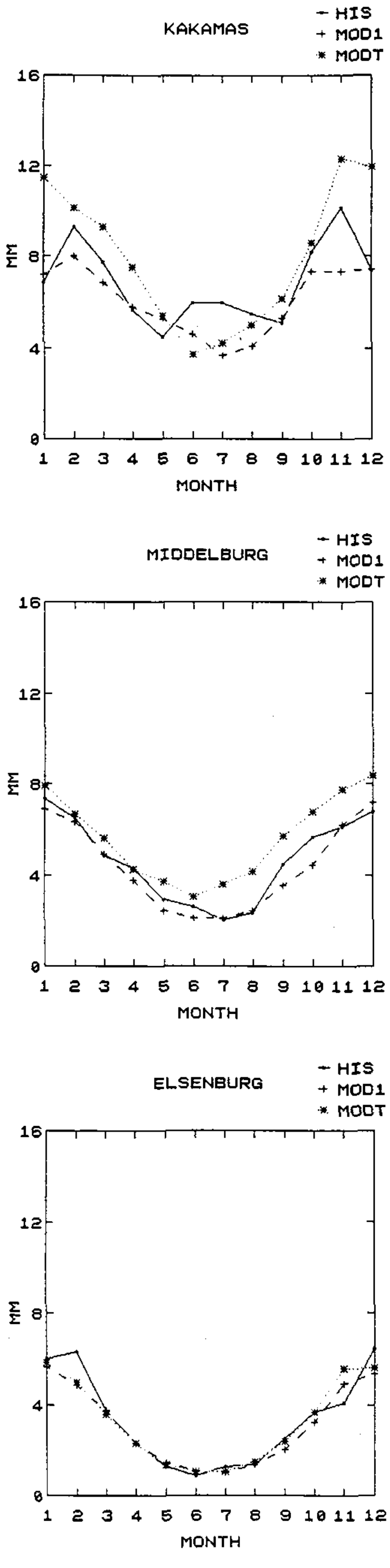


FIGURE 6.25 Monthly means for sunshine duration for wet days

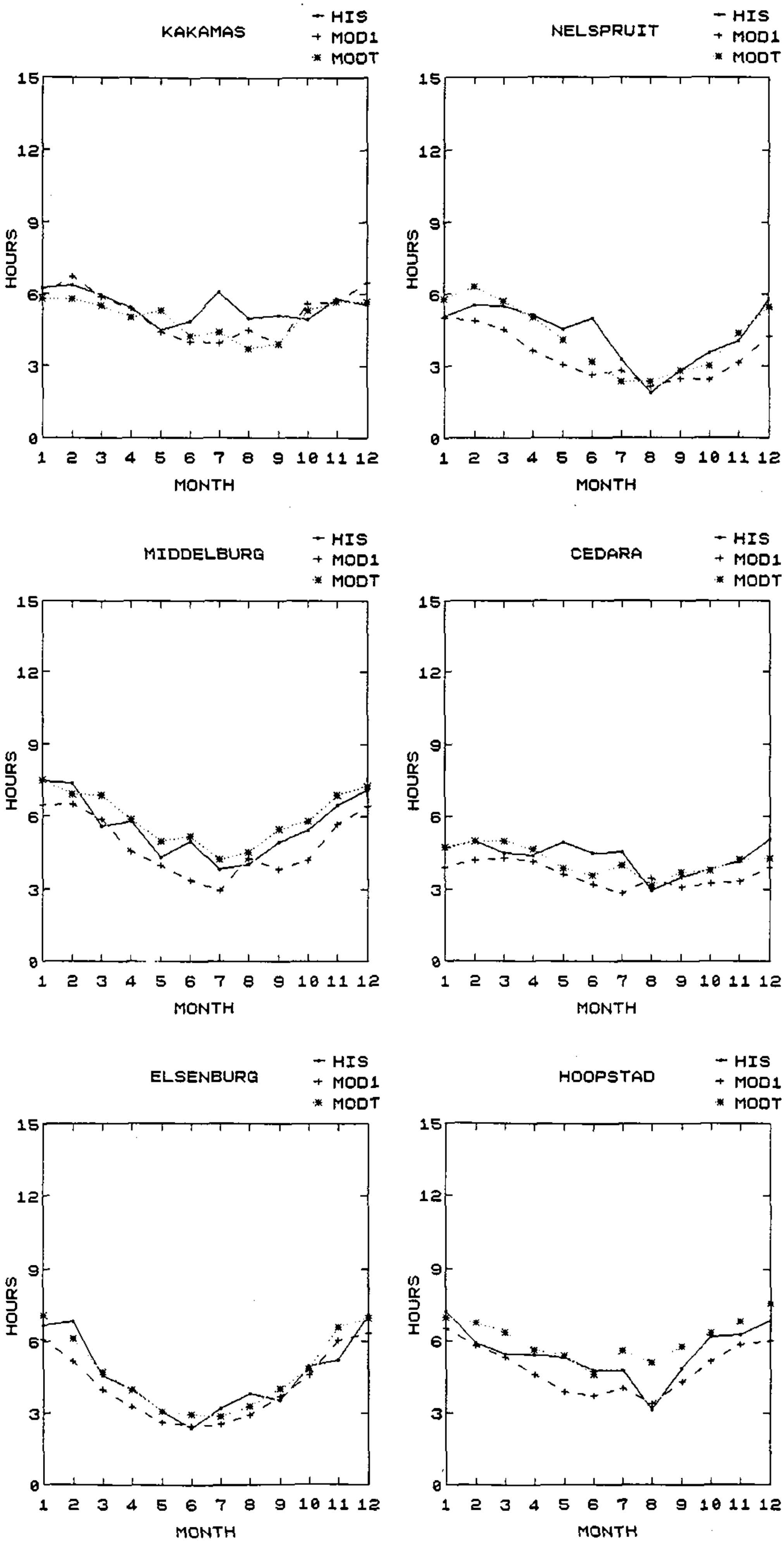


FIGURE 6.26 Monthly means for wind run for wet days

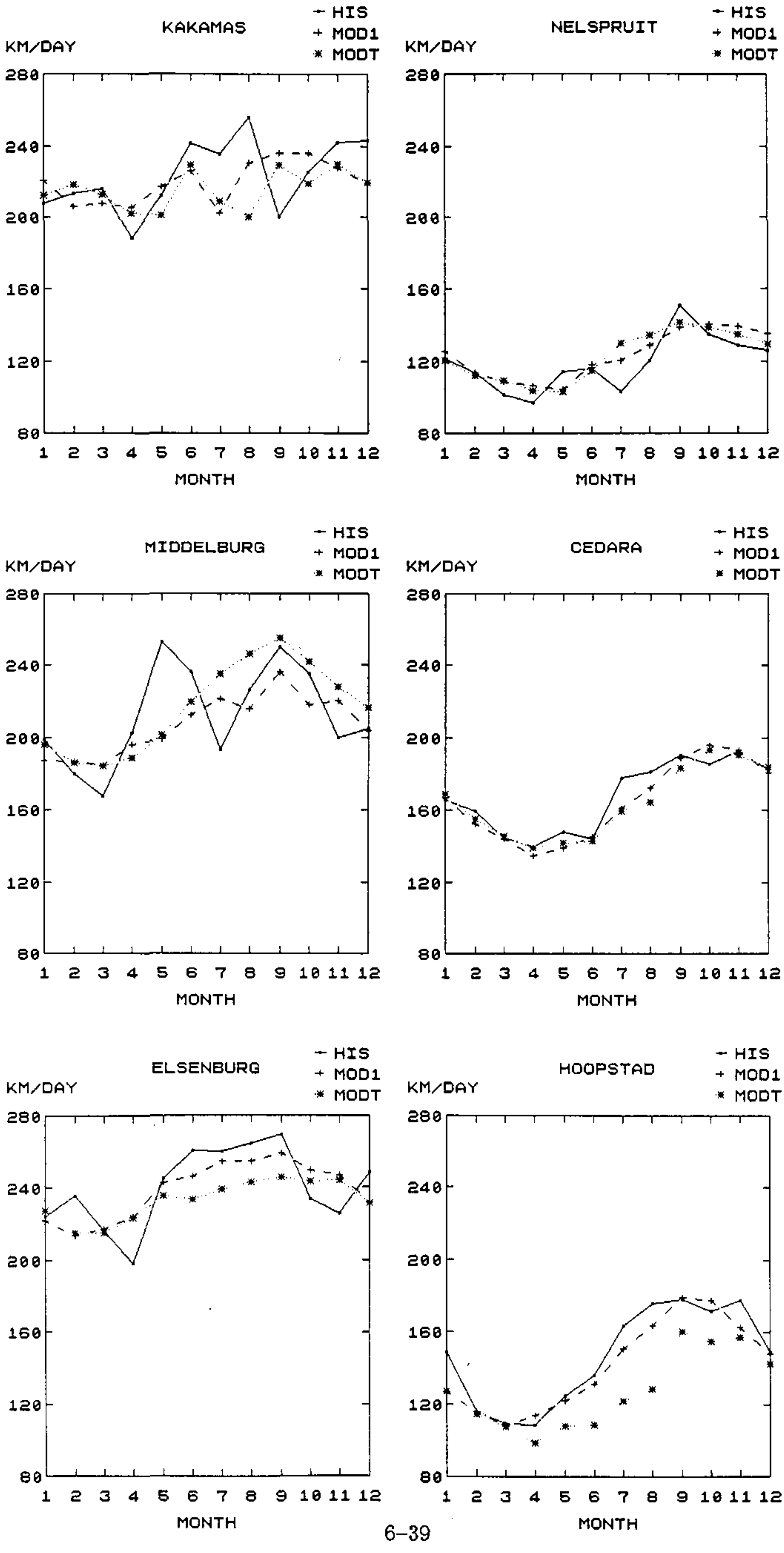


FIGURE 6.27 Monthly means for maximum humidity for wet days

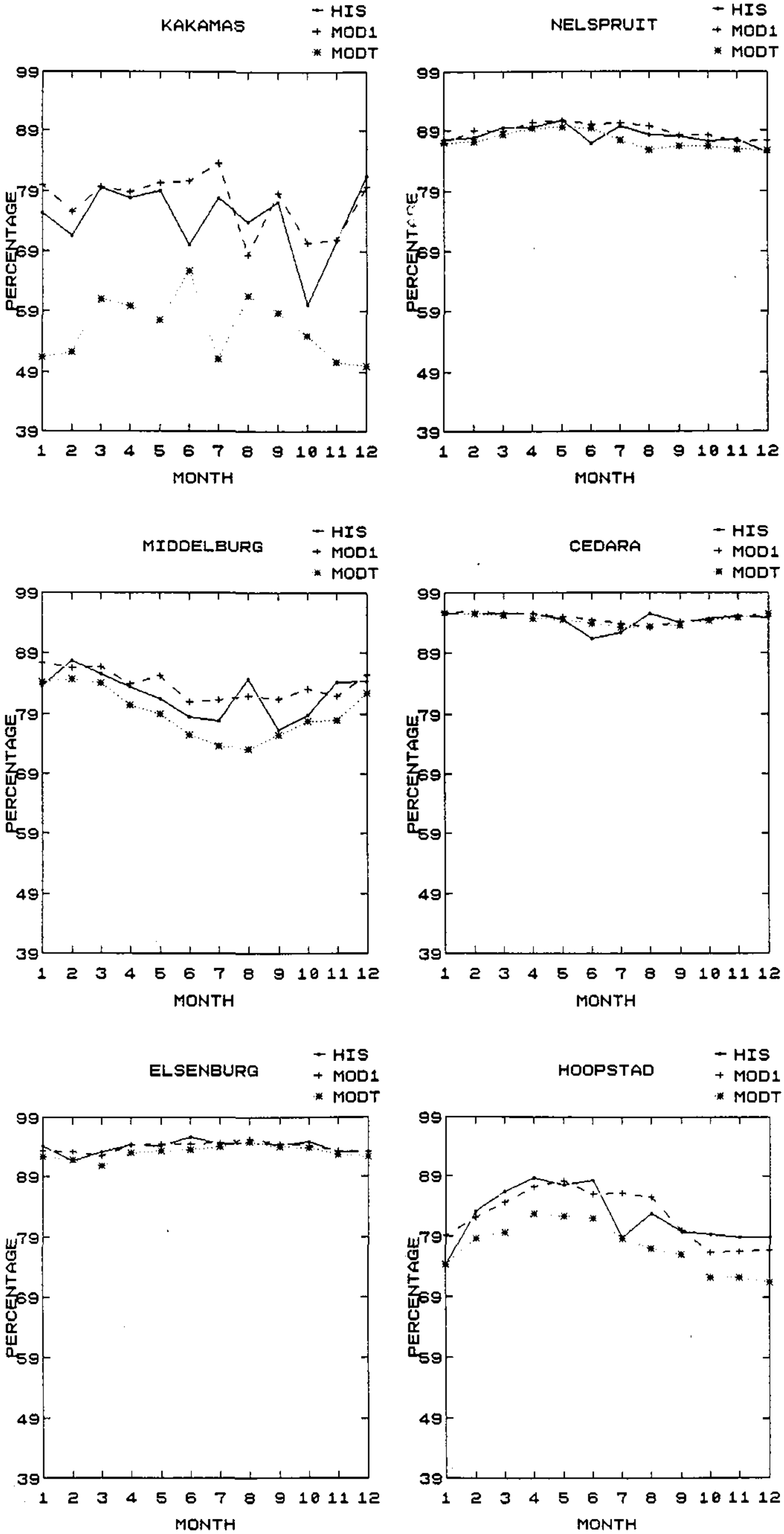


FIGURE 6.28 Monthly means for minimum humidity for wet days

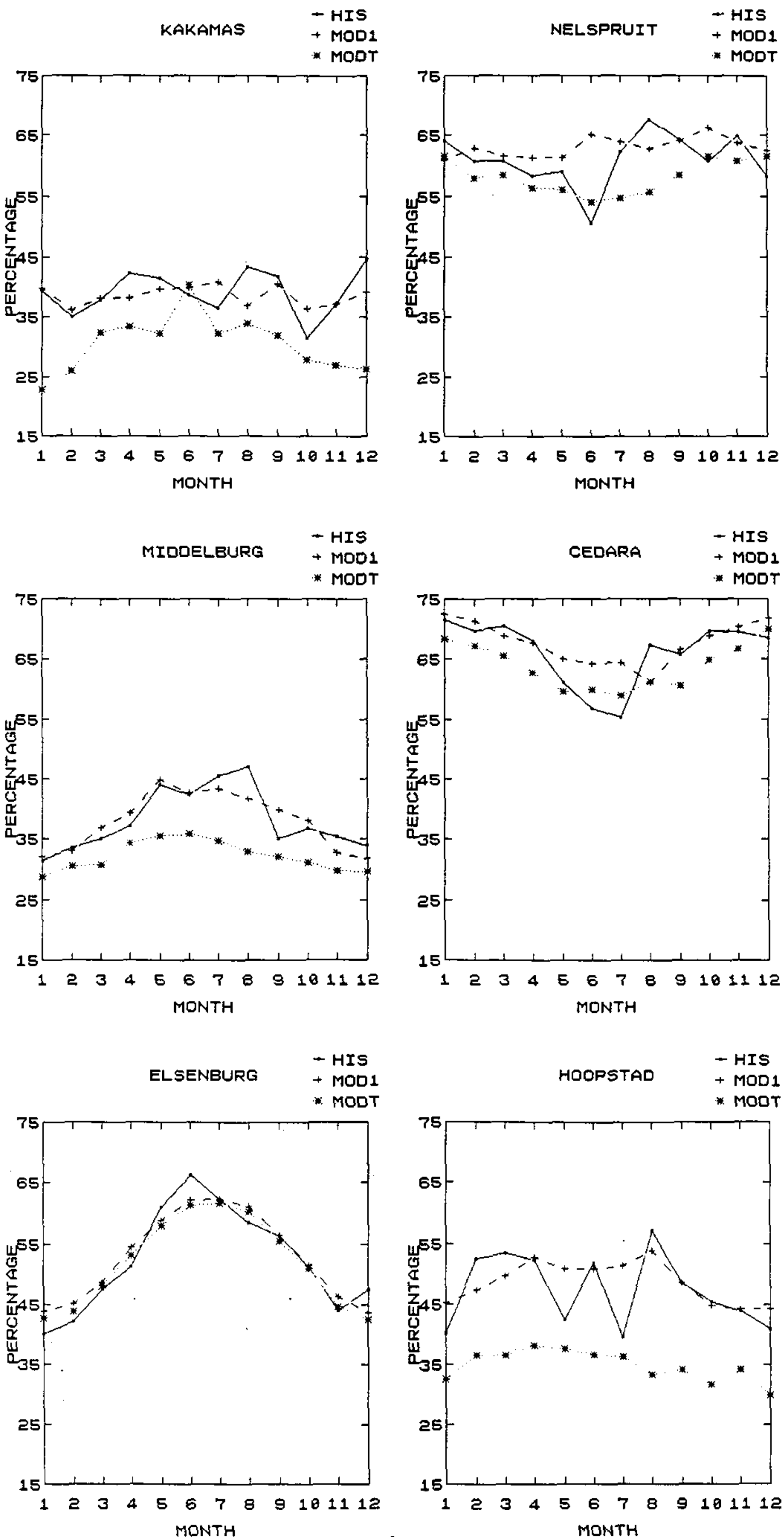


FIGURE 6.29 Monthly means for maximum temperature for dry days

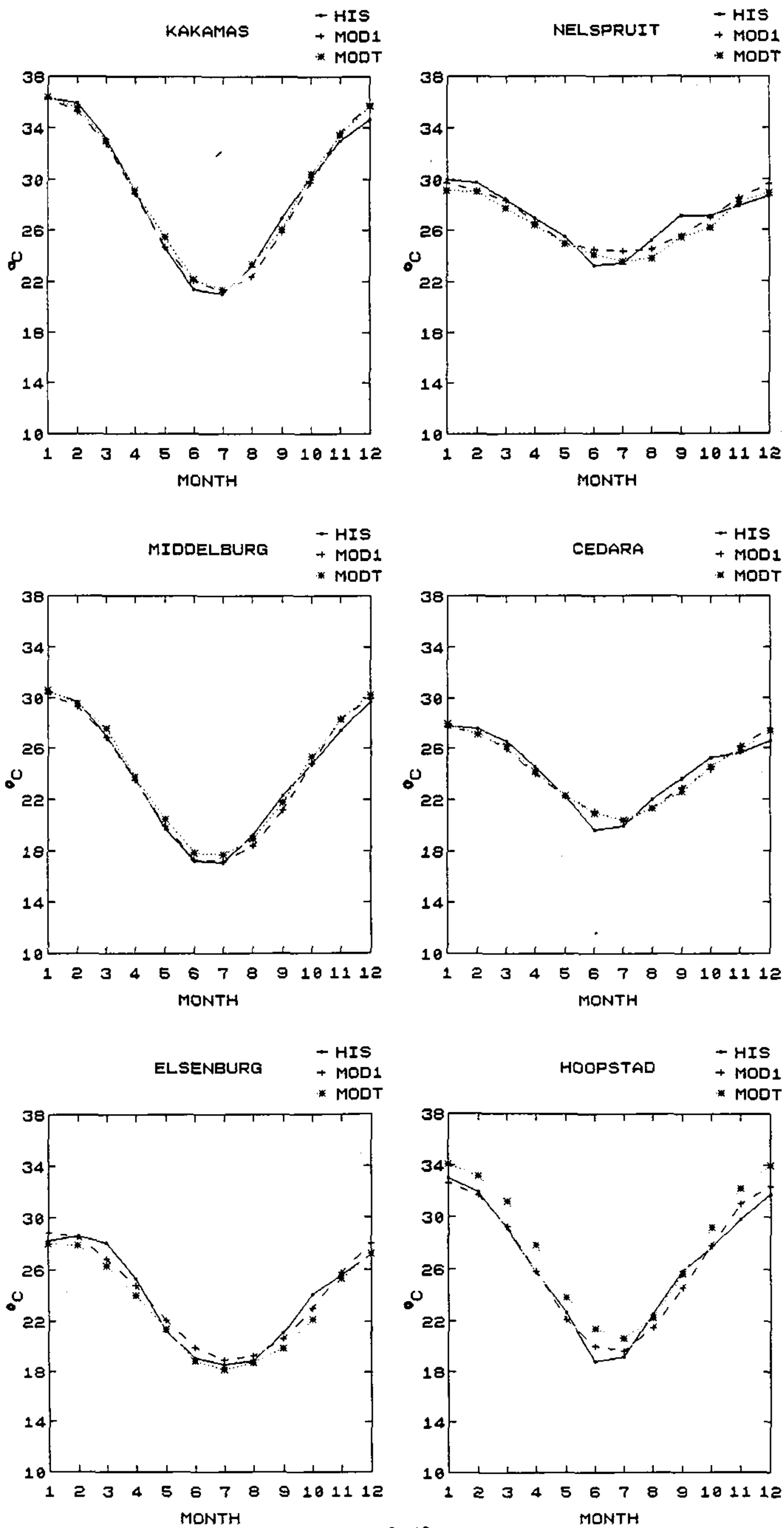
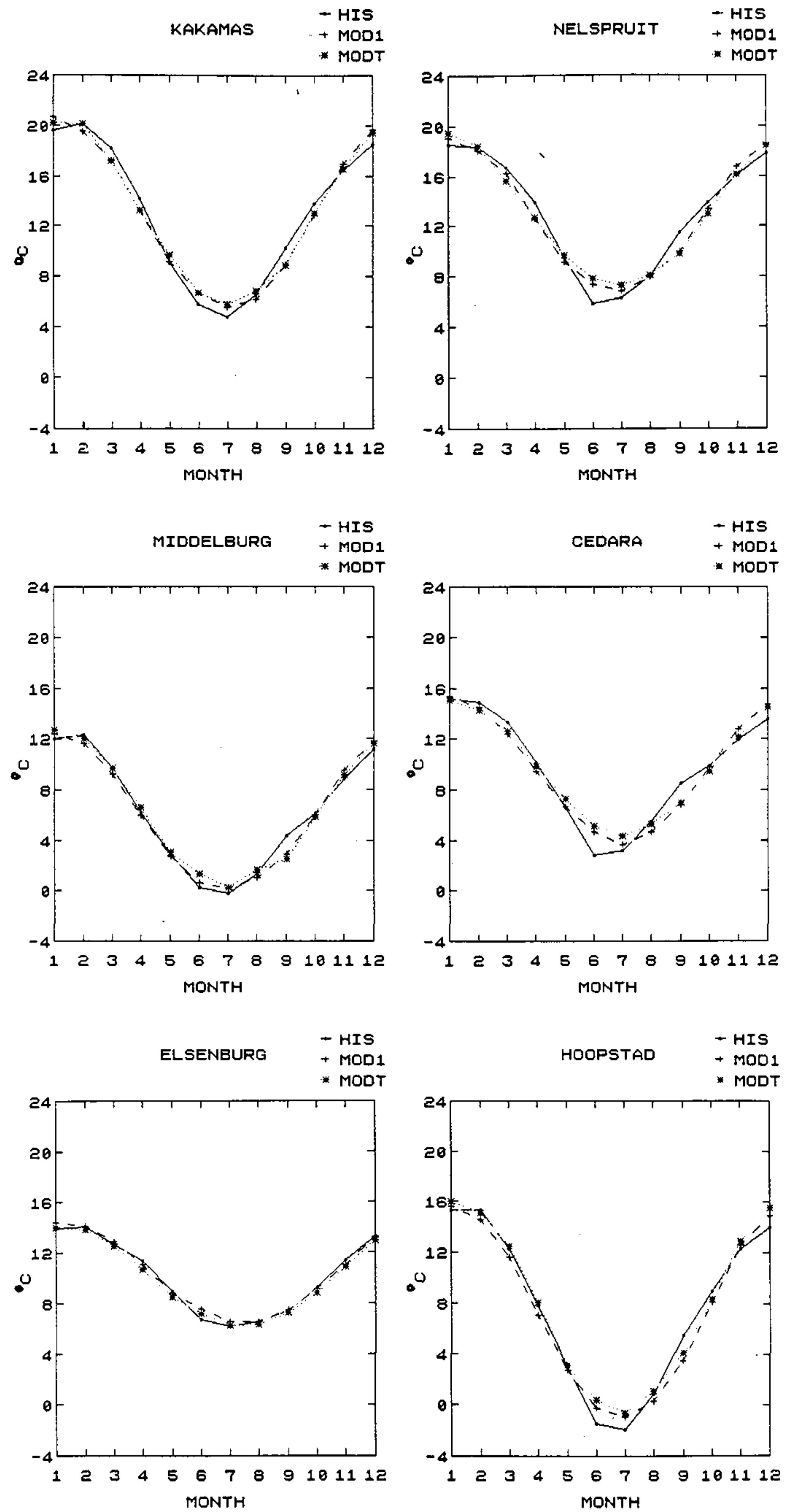
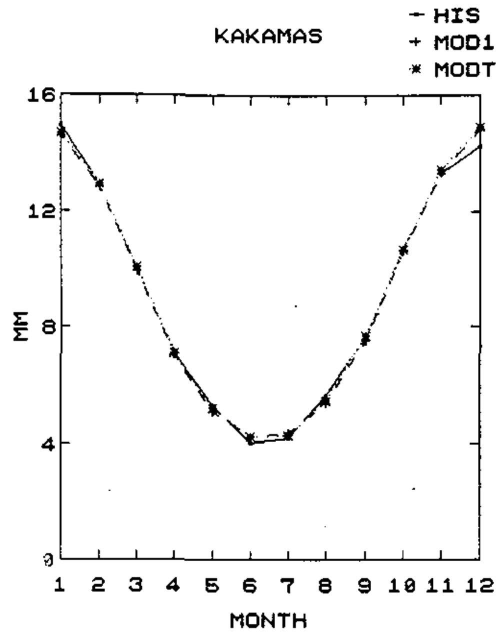
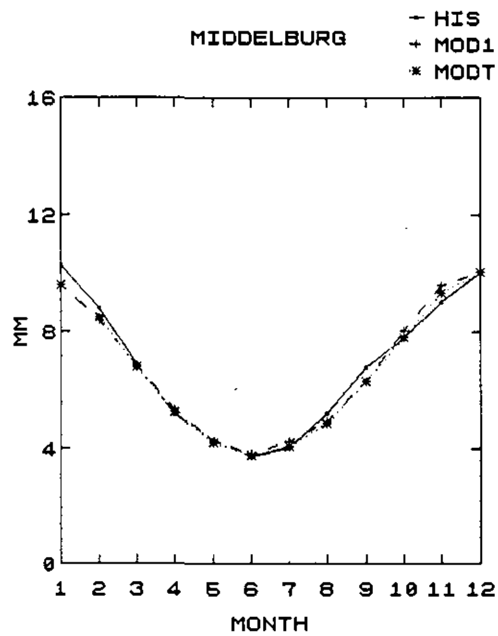


FIGURE 6.30 Monthly means for minimum temperature for dry days



FIGUE 6.31 Monthly means for evaporation for dry days





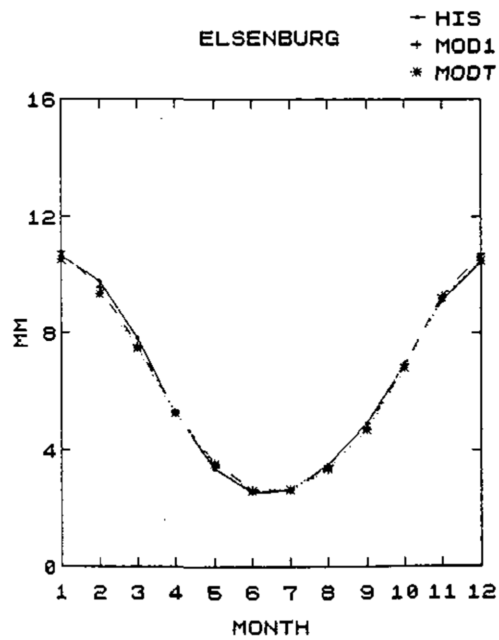


FIGURE 6.32 Monthly means for sunshine duration for dry days

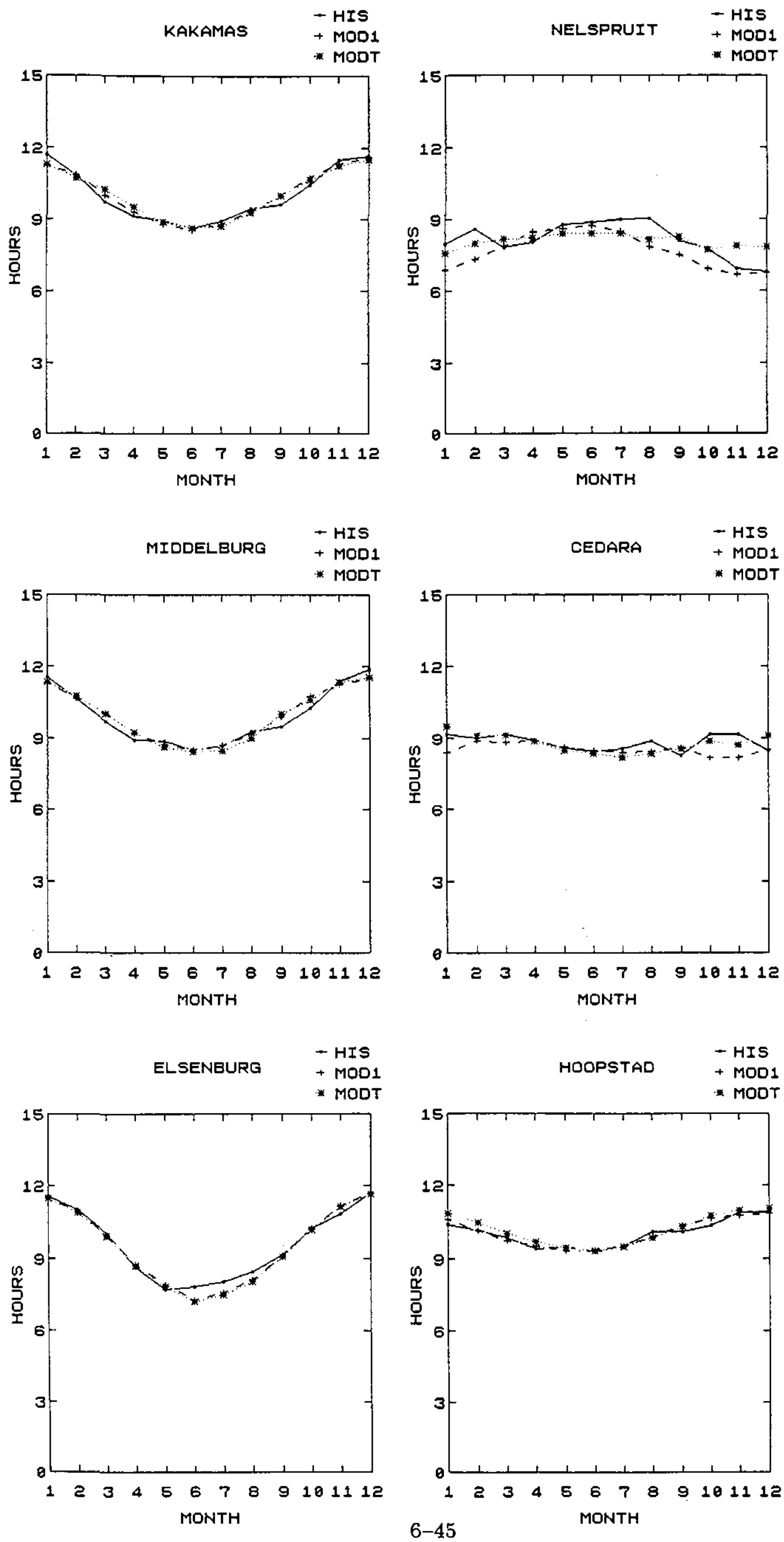


FIGURE 6.33 Monthly means for wind run for dry days

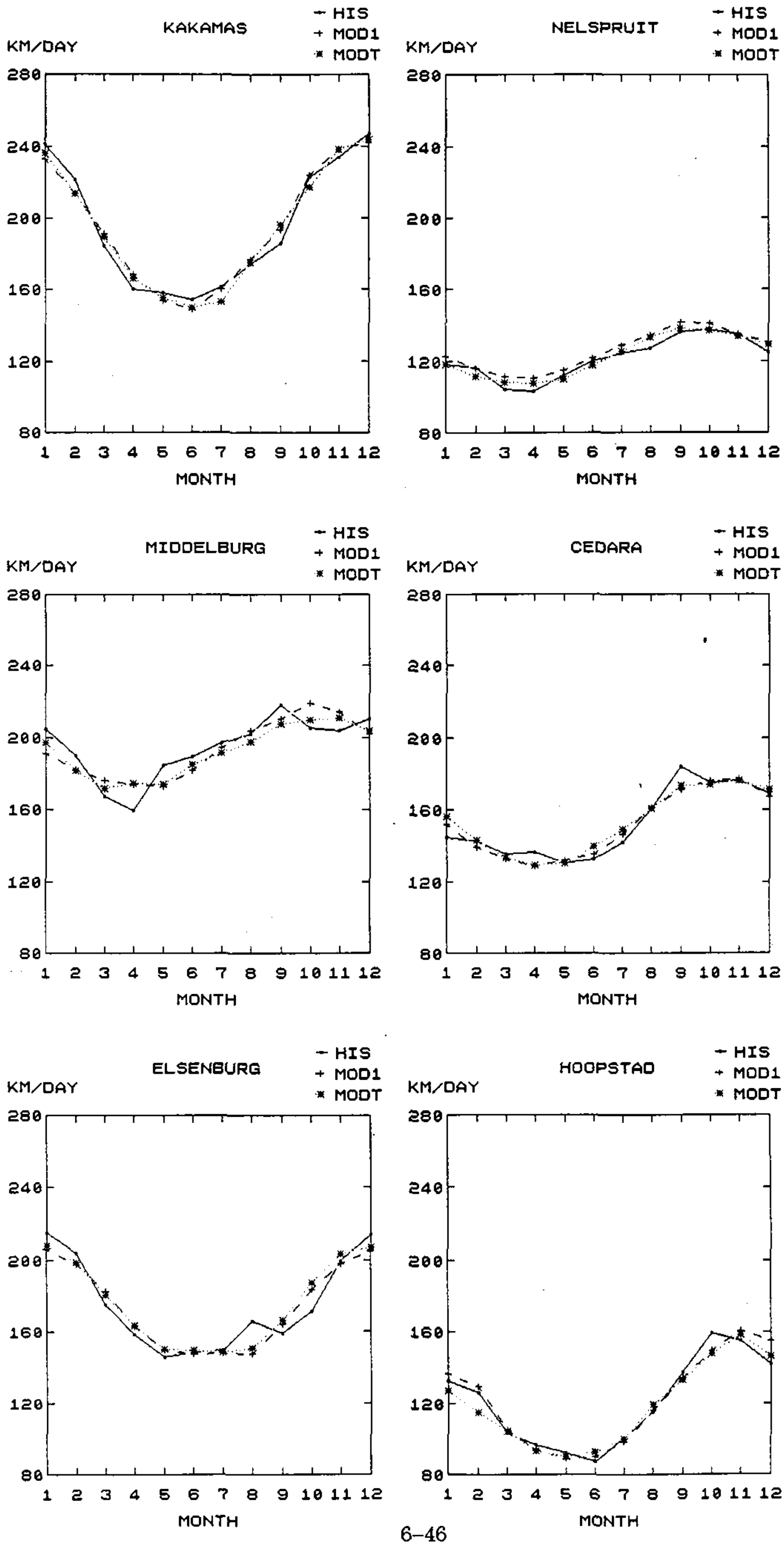


FIGURE 6.34 Monthly means for maximum humidity for dry days

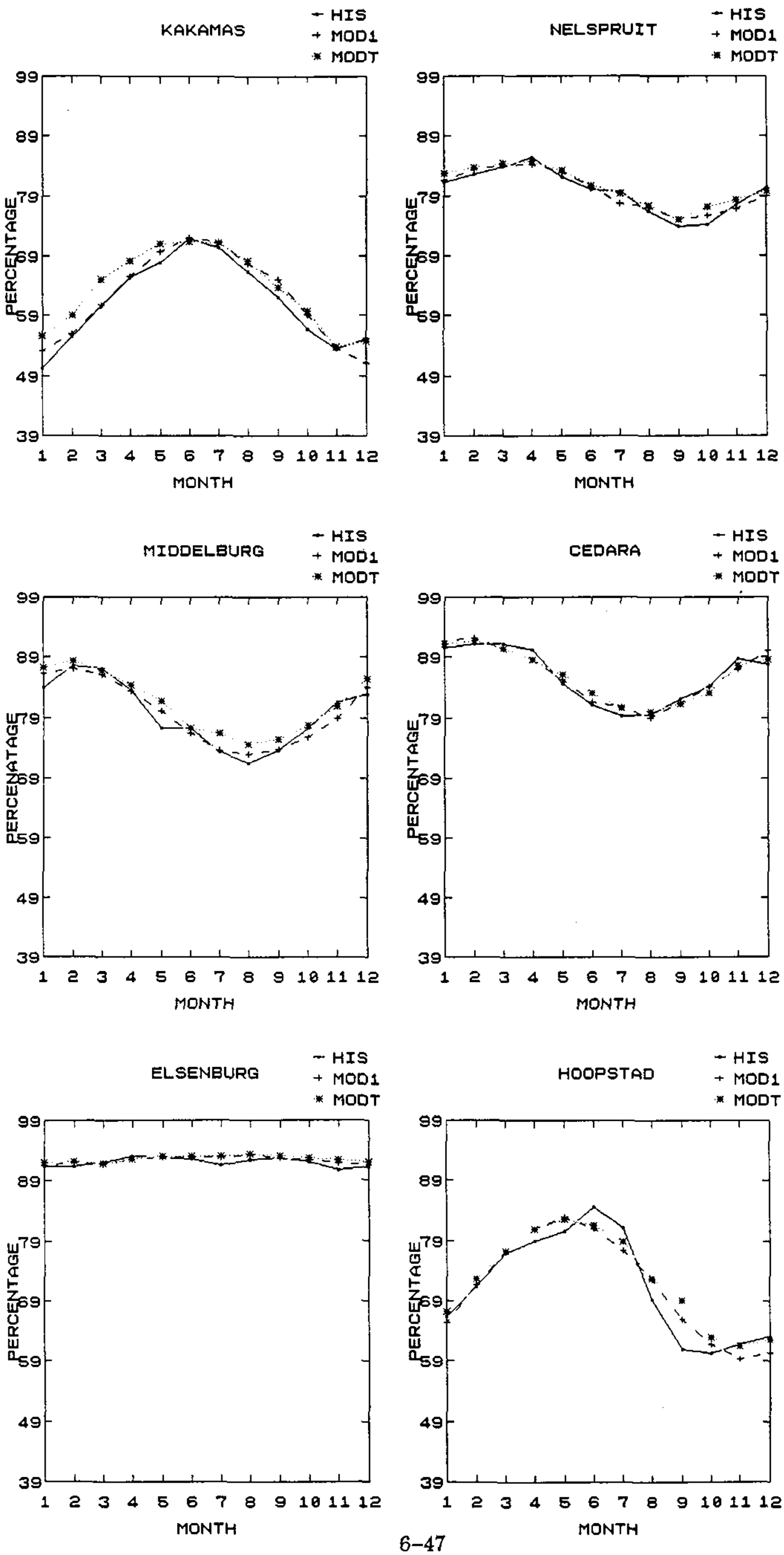
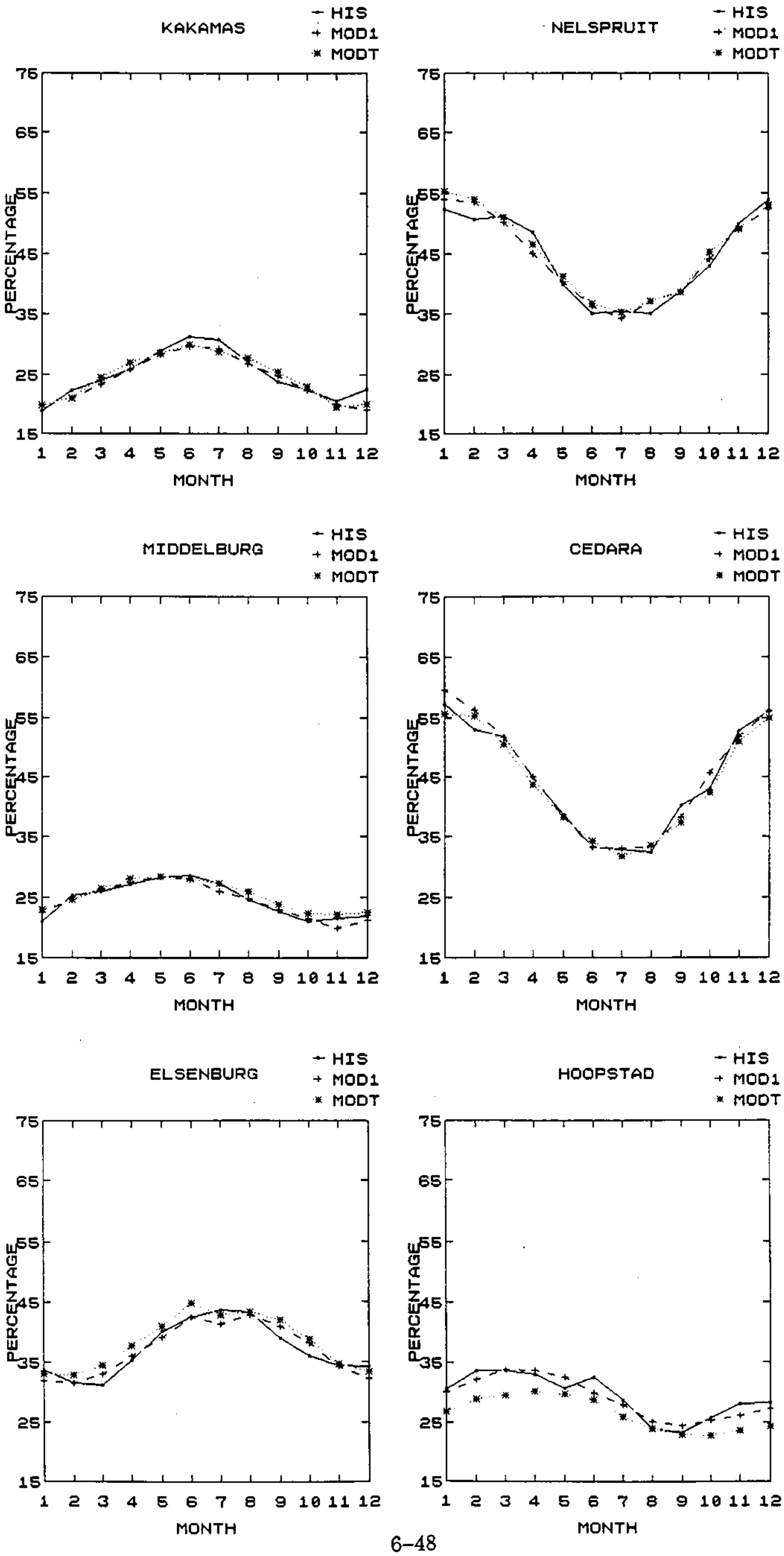


FIGURE 6.35 Monthly means for minimum humidity for dry days



These figures show that monthly means are adequately preserved by both models, in particular, the models fit the monthly means very well on dry days. The same results are observed for the station Hoopstad for the variables maximum temperature and minimum humidity as when the sequences were treated as a whole. For the sequences of wet days, differences are observed between monthly means of the simulated sequences and the monthly means of the observed sequences, in particular for the variables wind run, maximum humidity and minimum humidity. As already mentioned there are relatively few observations of rainfall at these stations and therefore one does not expect the models to fit the historical records on wet days very accurately. This is supported by comparing the results obtained for the stations Kakamas and Cedara when the variables are conditioned on wet days. Kakamas is the station for which fewer rainfall days are observed, and Cedara is the station at which most rainfall days are observed, of the stations in this study. It is clear that both models preserve the monthly means for Cedara but do not perform as well for Kakamas.

The plots also show that generally, Model 1 fits the data better than Model T. This can be explained by observing that for Model 1 one is only separating the sequences into dry and wet days, while for Model T one separates the sequences into four parts, that is, into dry–dry, wet–wet, dry–wet and wet–dry sequences. Therefore the model parameters for Model T are estimated using very few observations especially when dealing with a wet sequence.

Figures 6.36 – 6.42 show the monthly standard deviations when the climate variables are conditioned on wet days. Figures 6.43 – 6.49 show the monthly standard deviations when the climate variables are conditioned on dry days. These plots show that both models have preserved monthly standard deviations when the climate variables are conditioned on dry days. Here again, very similar results are obtained to those when the sequences are taken as a whole. Generally, the models preserve the monthly standard deviations when the variables are conditioned on wet days. Some differences are observed between the simulated sequences and the historical records. These differences can be explained again for the reasons mentioned above. Where differences between simulated and historical sequences occur, it can be noted that generally Model T tends to overestimate the standard deviations, while Model 1 tends to underestimate them.

FIGURE 6.36 Monthly standard deviations for maximum temperature for wet days

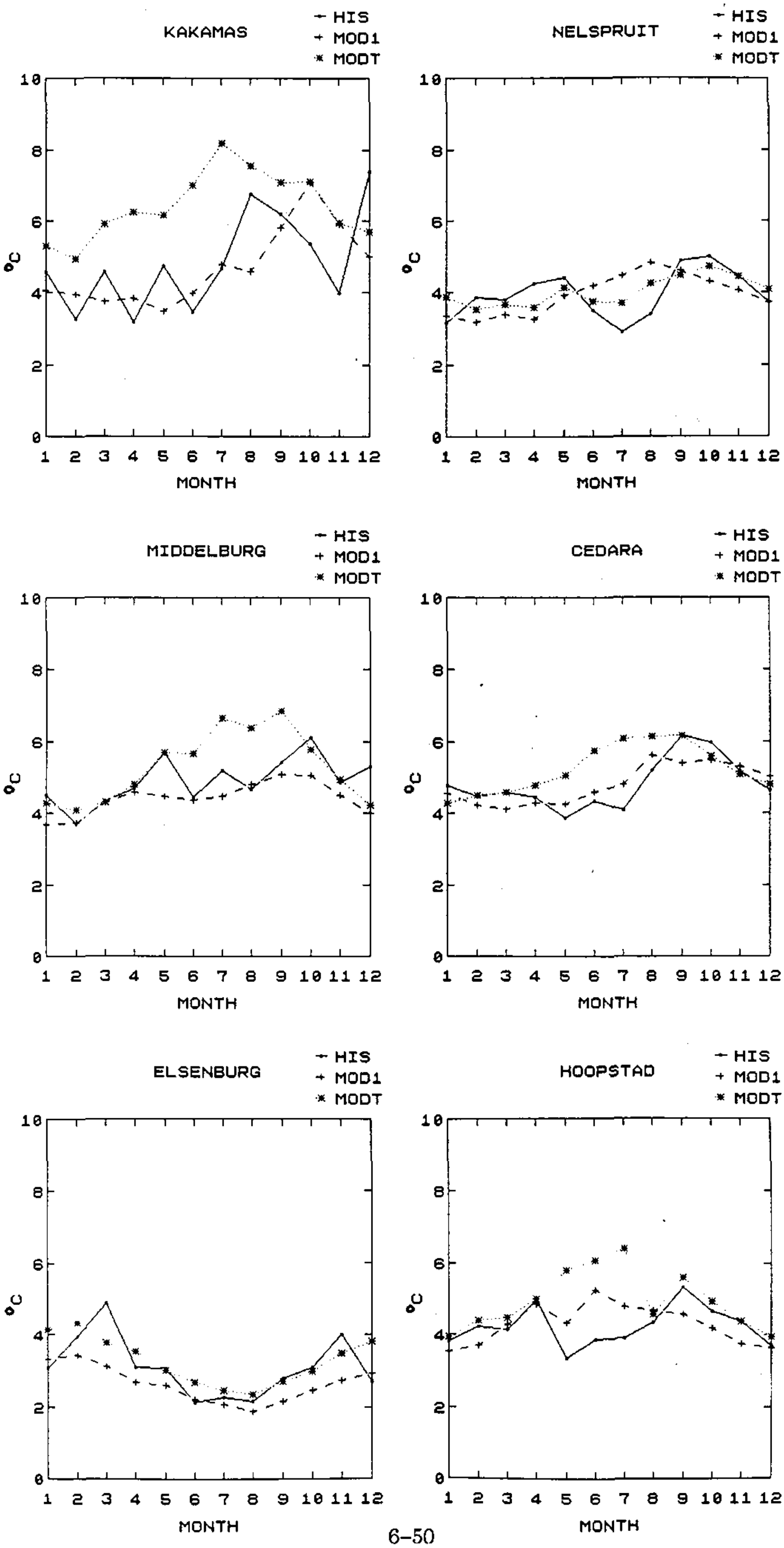


FIGURE 6.37 Monthly standard deviations for minimum temperature for wet days

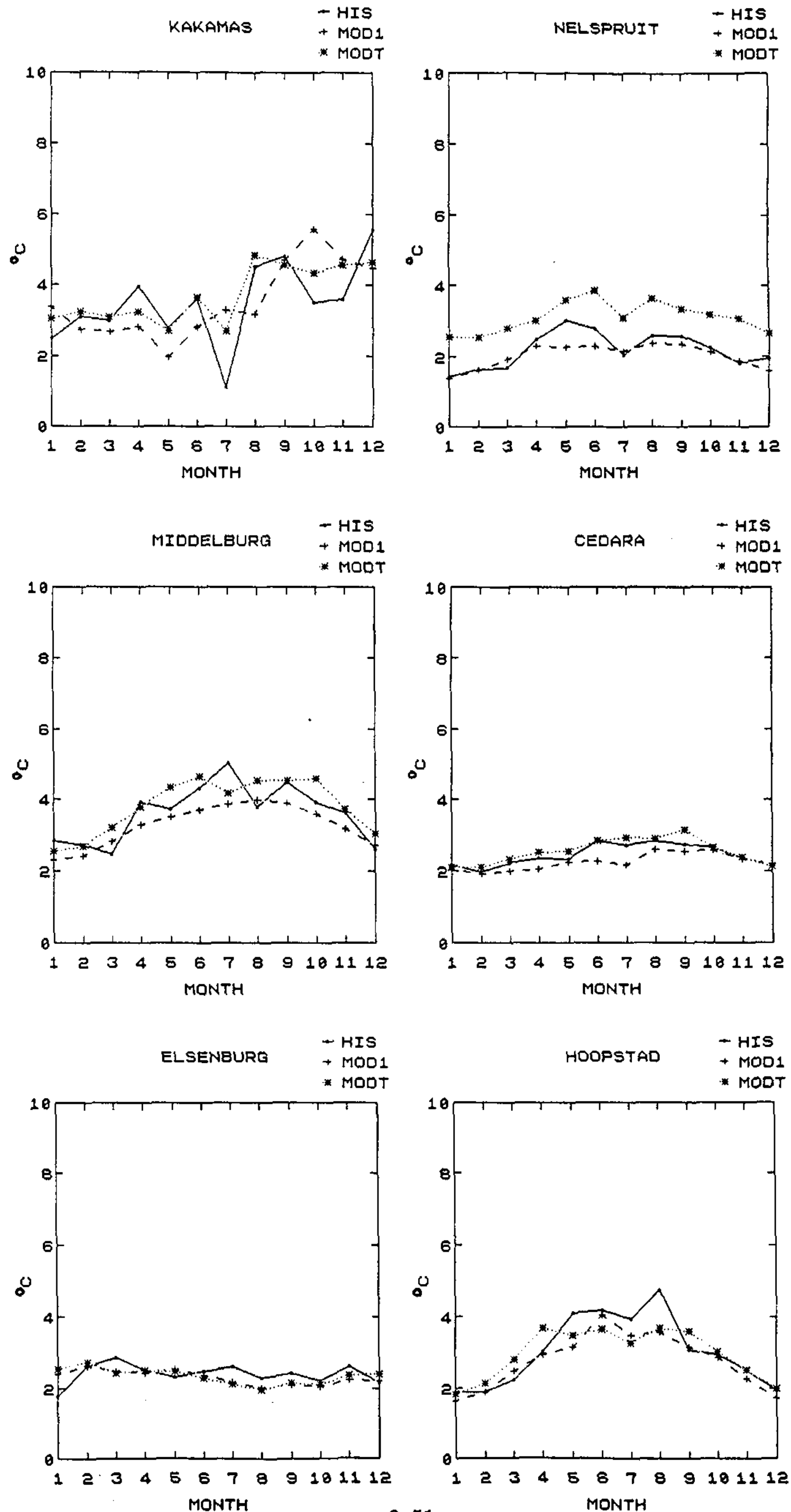
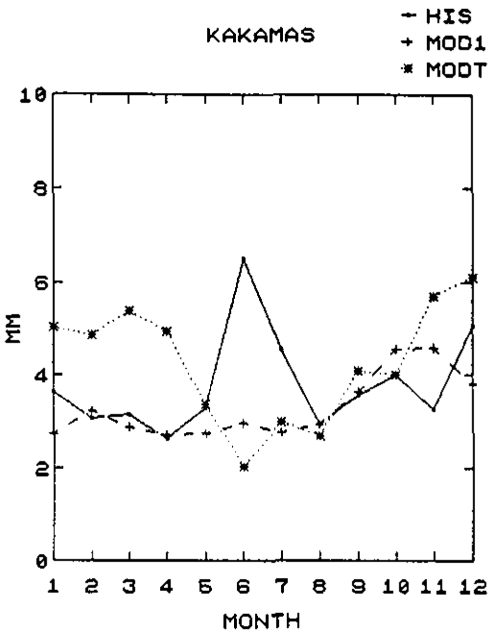
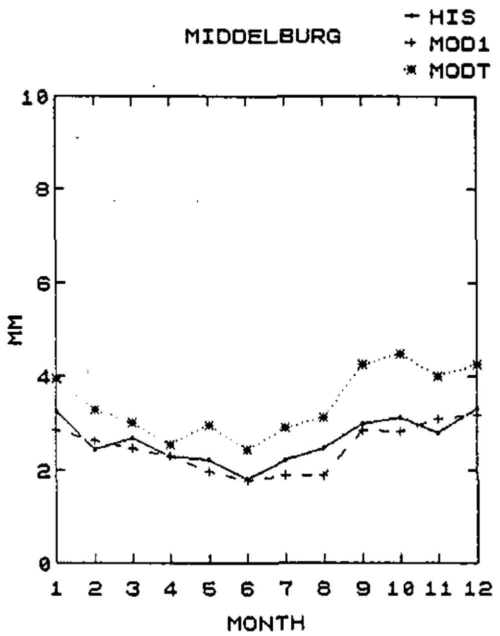


FIGURE 6.38 Monthly standard deviations for evaporation for wet days





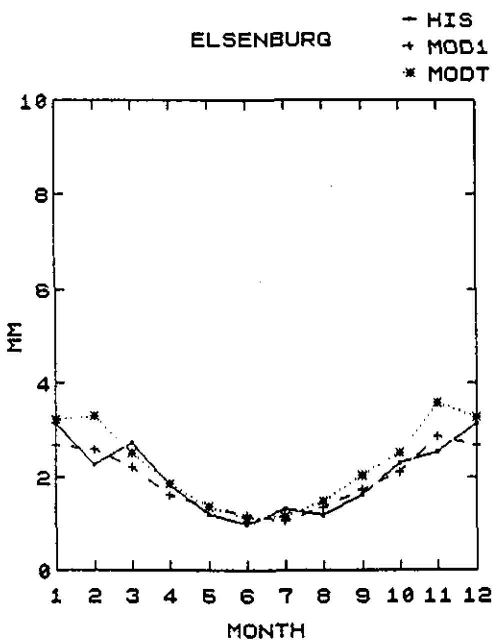


FIGURE 6.39 Monthly standard deviations for sunshine duration for wet days

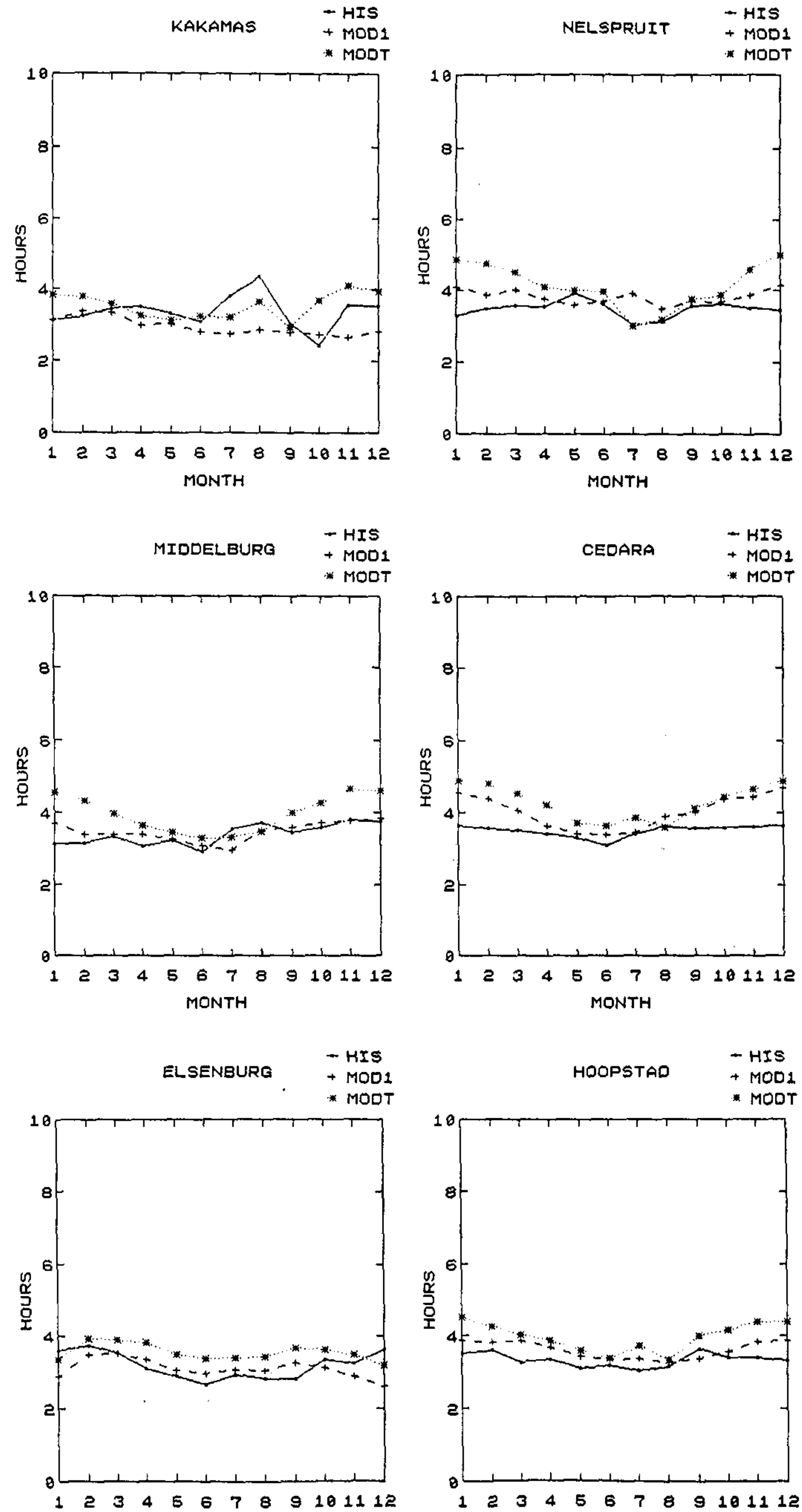


FIGURE 6.40 Monthly standard deviations for wind run for wet days

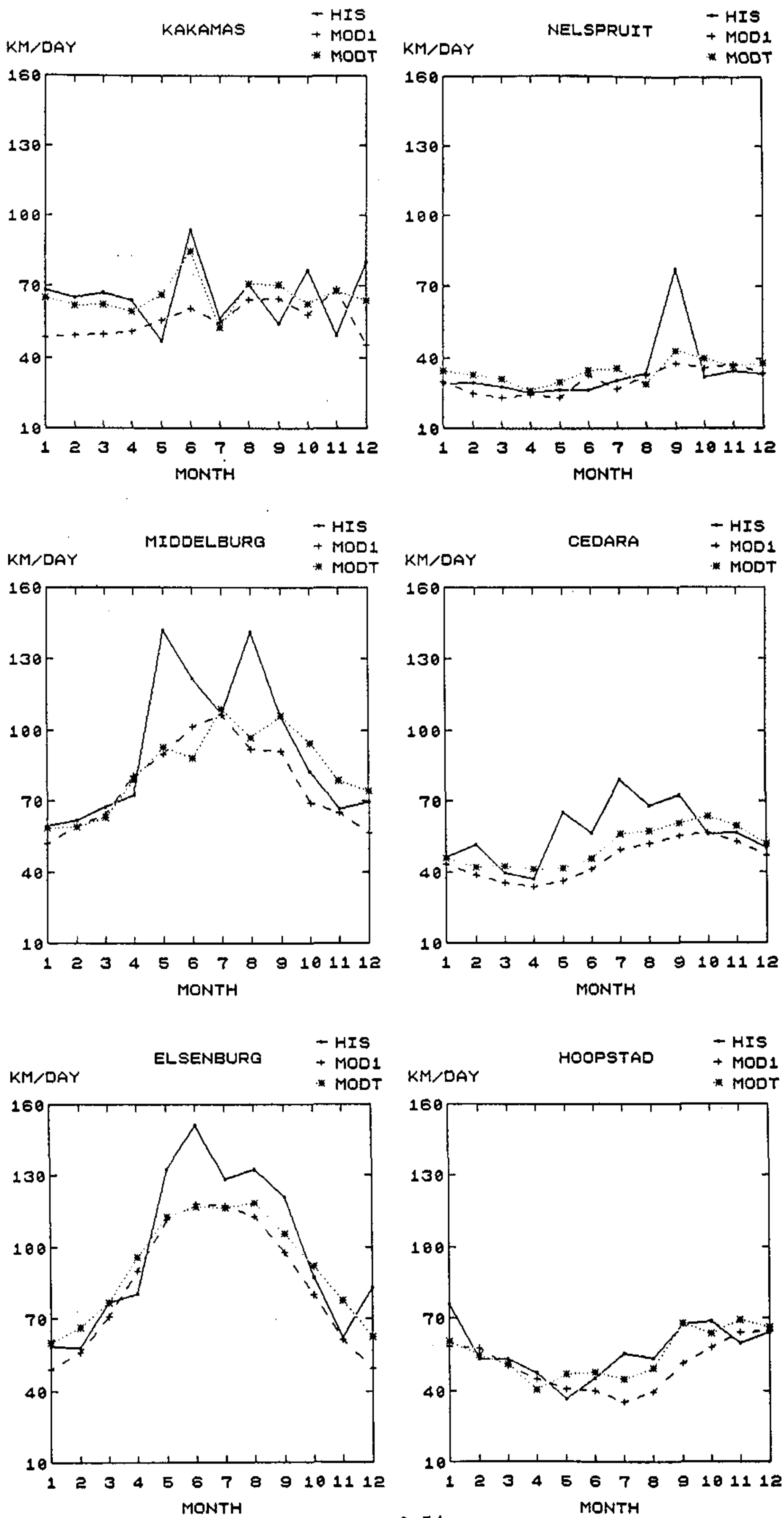


FIGURE 6.41 Monthly standard deviations for maximum humidity for wet days

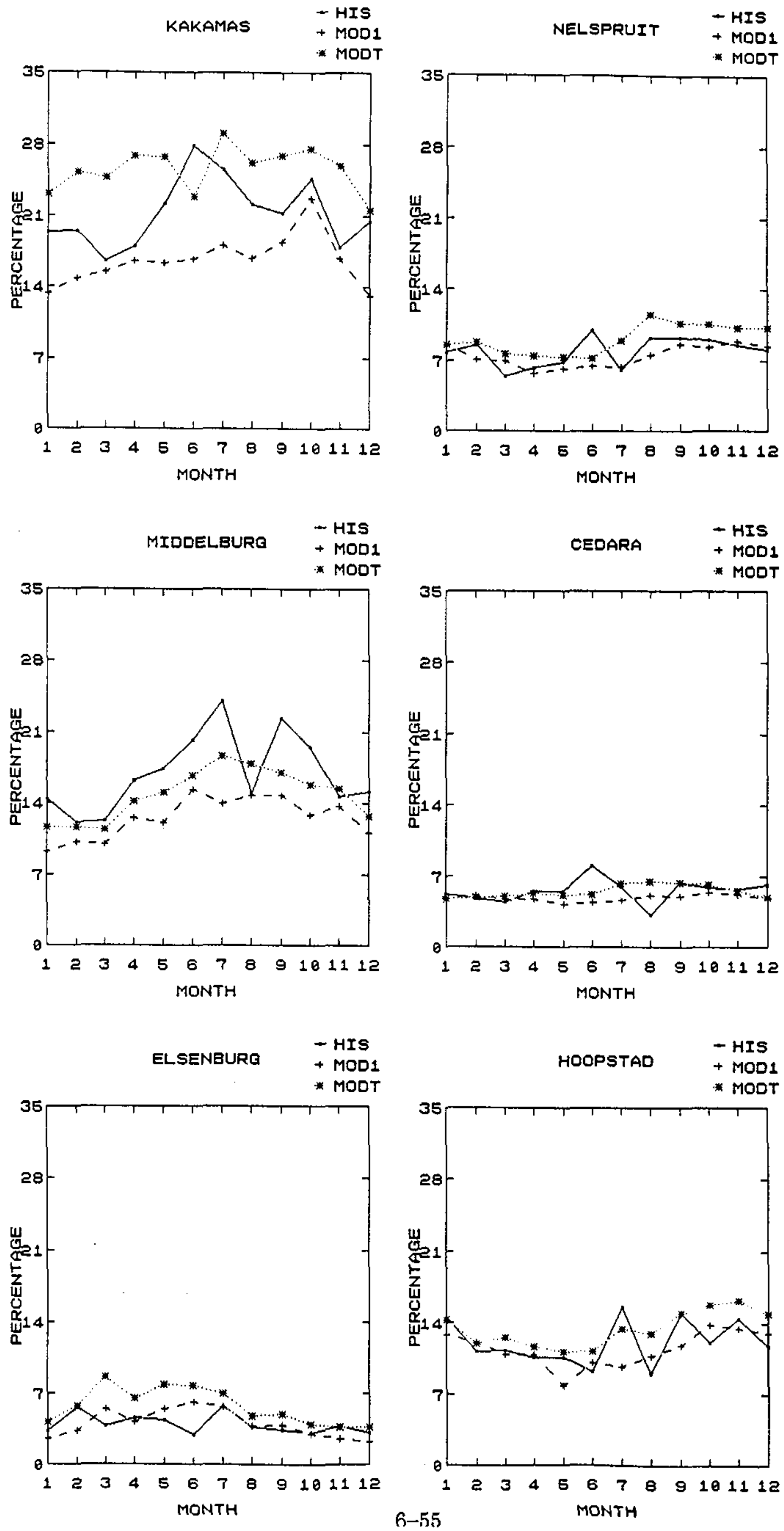


FIGURE 6.42 Monthly standard deviations for minimum humidity for wet days

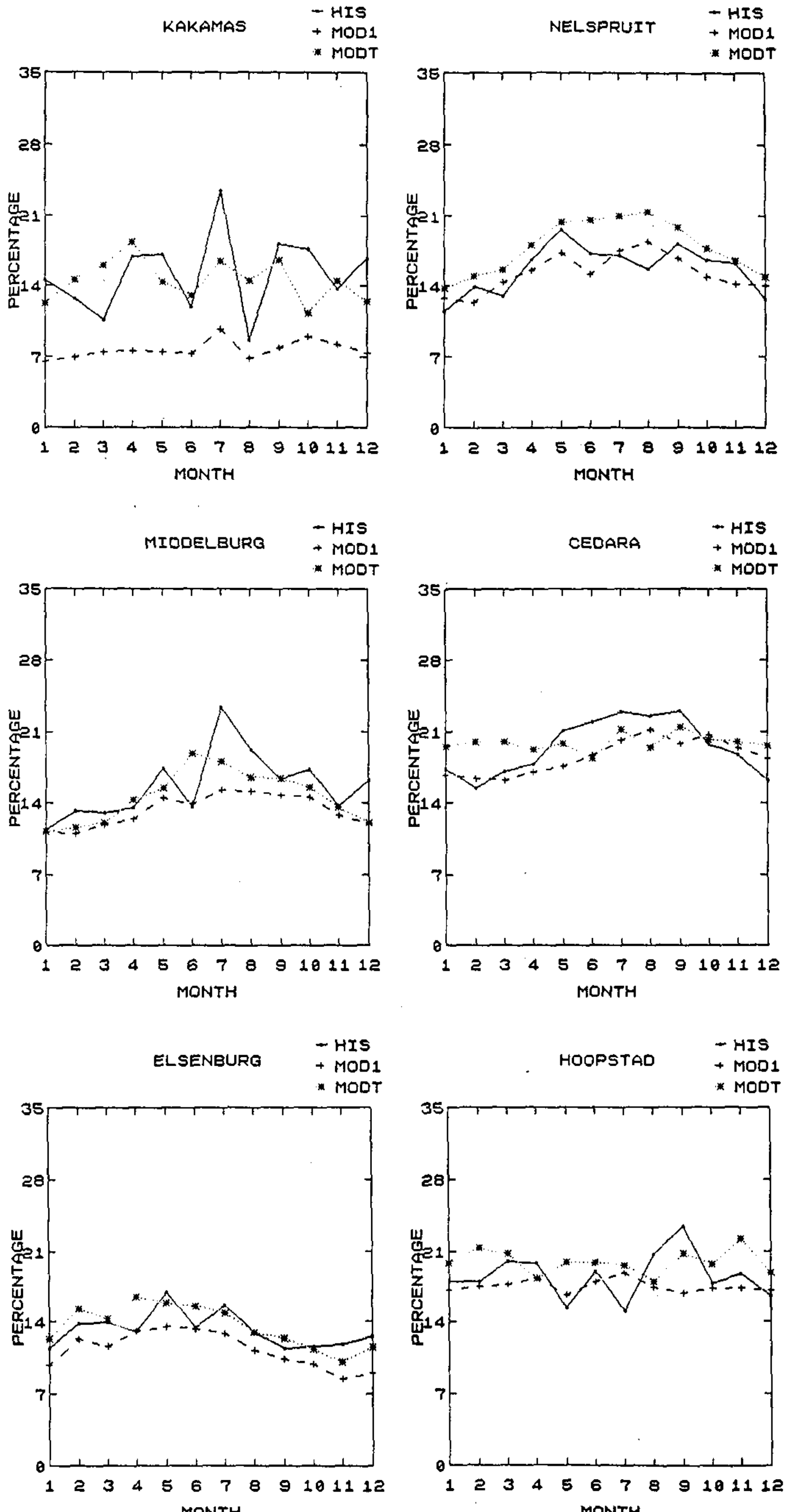


FIGURE 6.43 Monthly standard deviations for maximum temperature for dry days

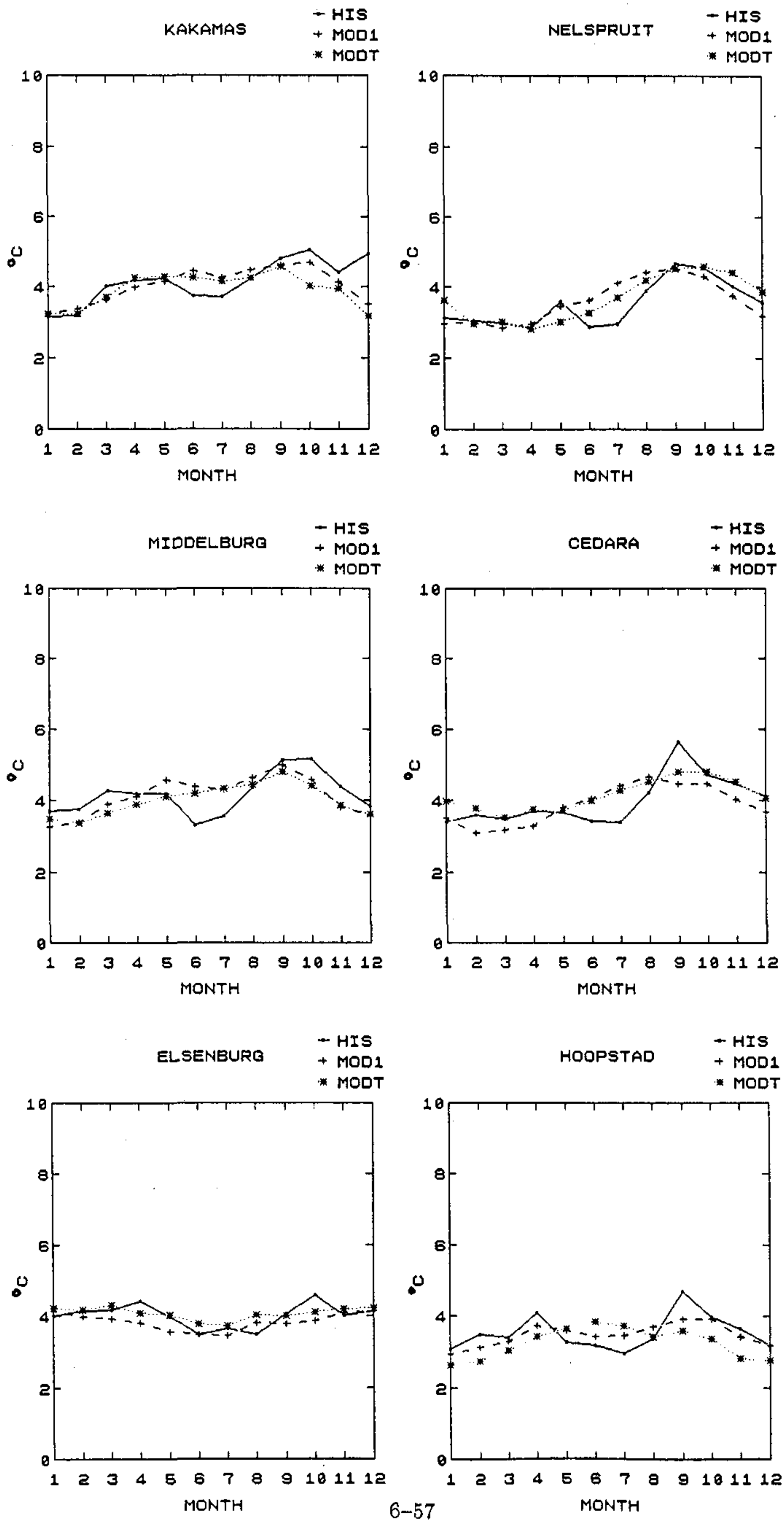


FIGURE 6.44 Monthly standard deviations for minimum temperature for dry days

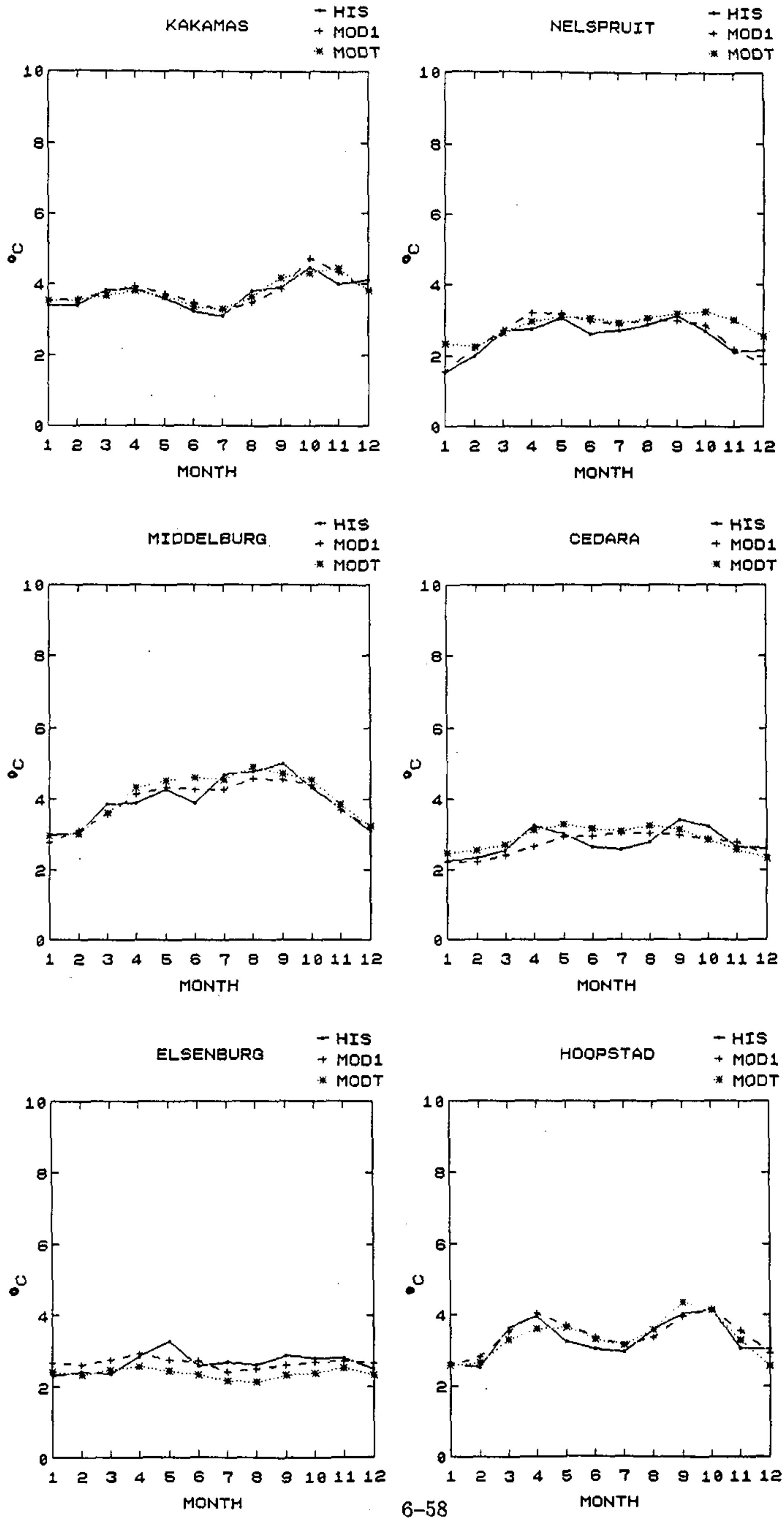
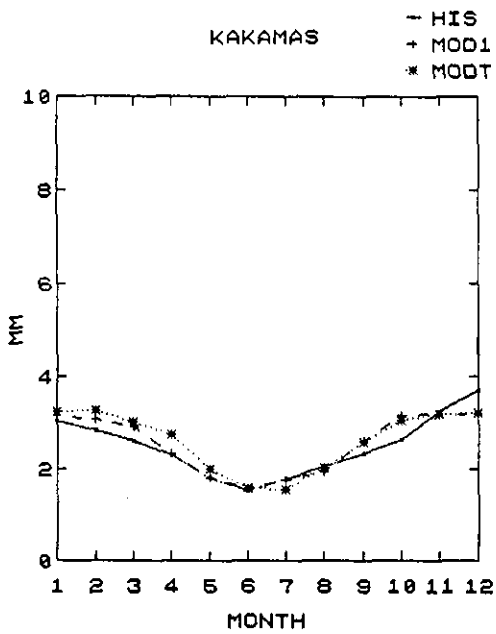
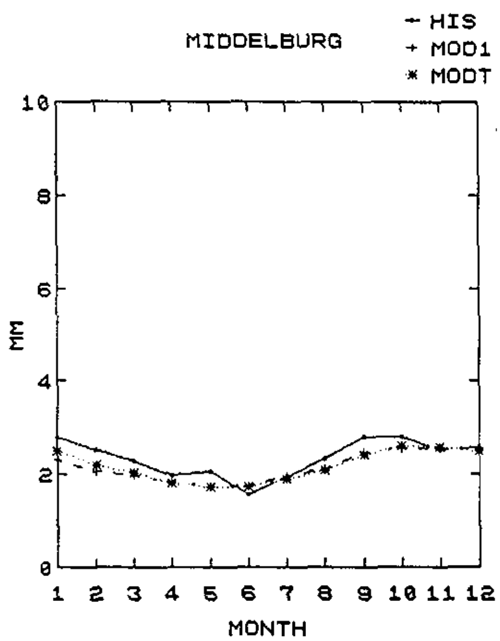


FIGURE 6.45 Monthly standard deviations for evaporation for dry days





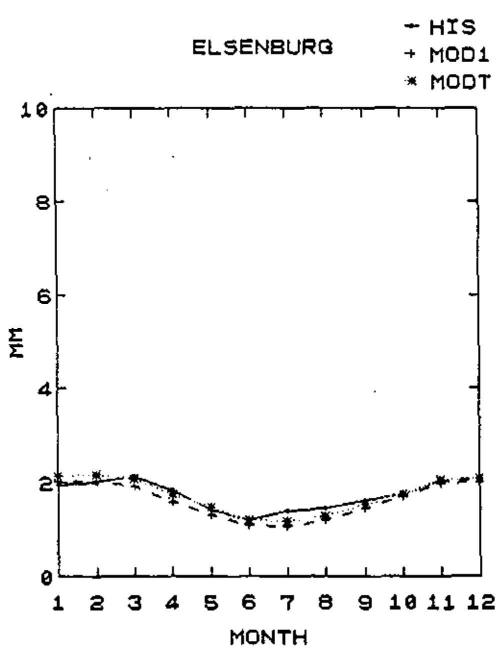


FIGURE 6.46 Monthly standard deviations for sunshine duration for dry days

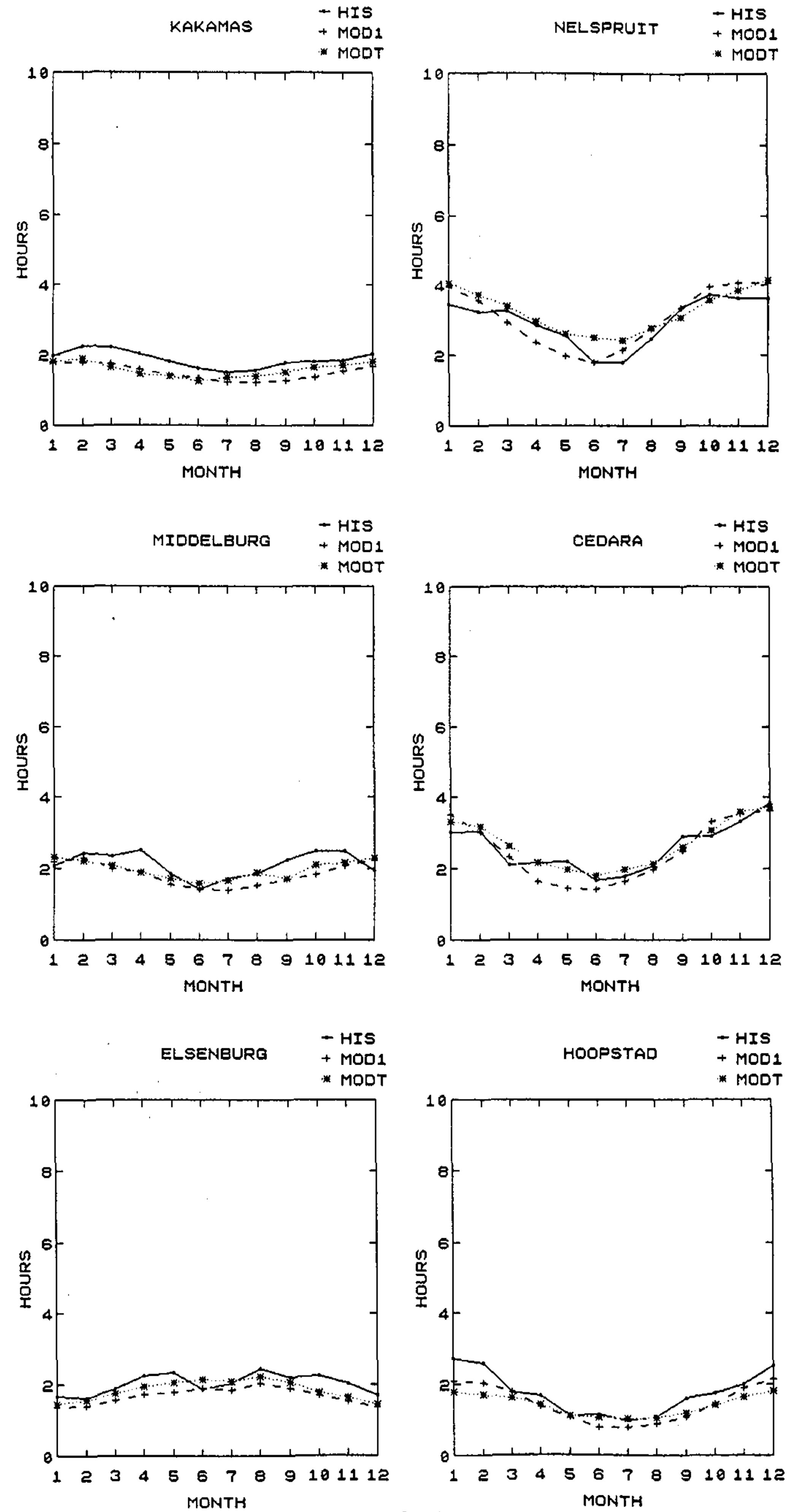


FIGURE 6.47 Monthly standard deviations for wind run for dry days

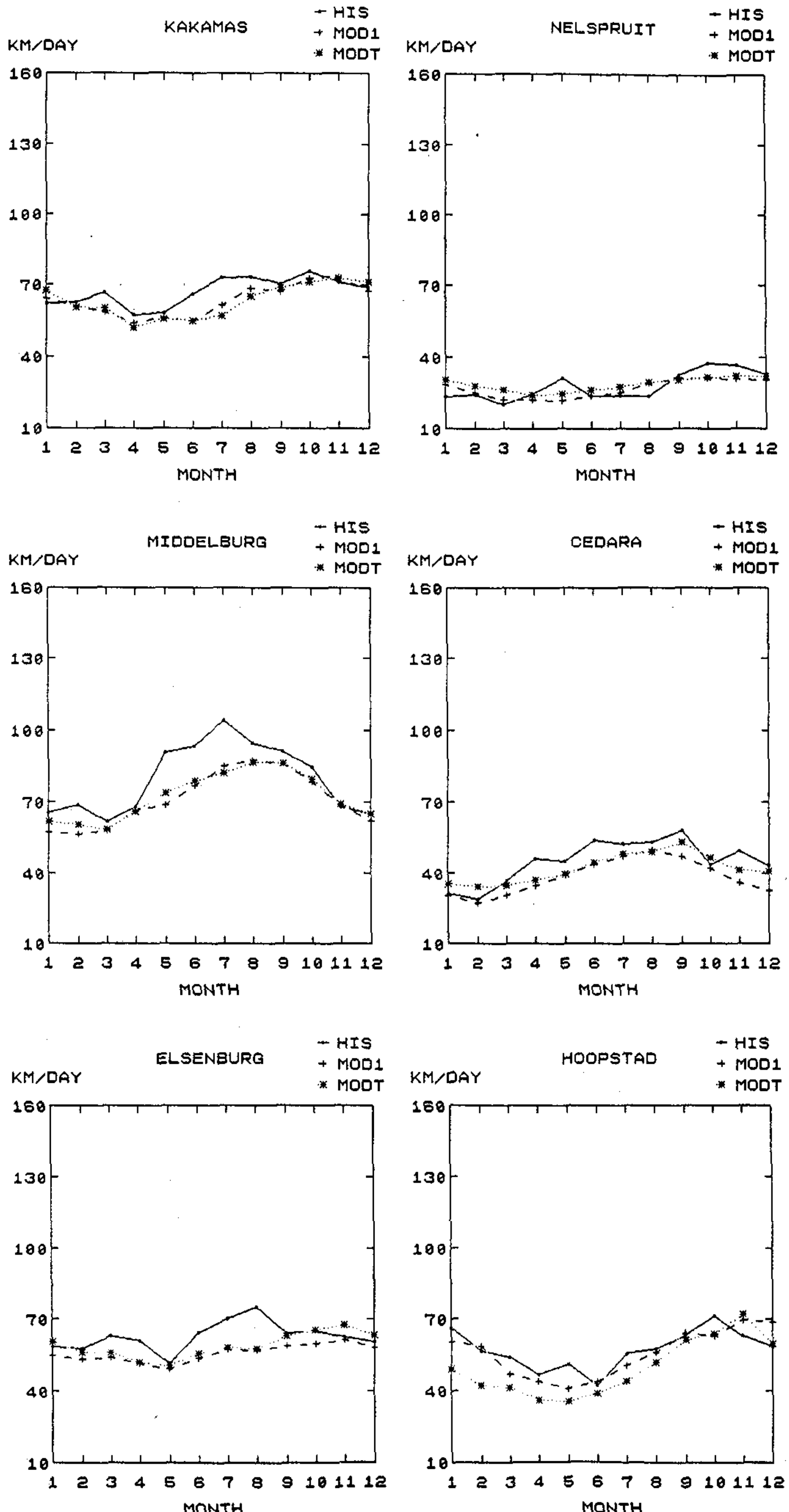


FIGURE 6.48 Monthly standard deviations for maximum humidity for dry days

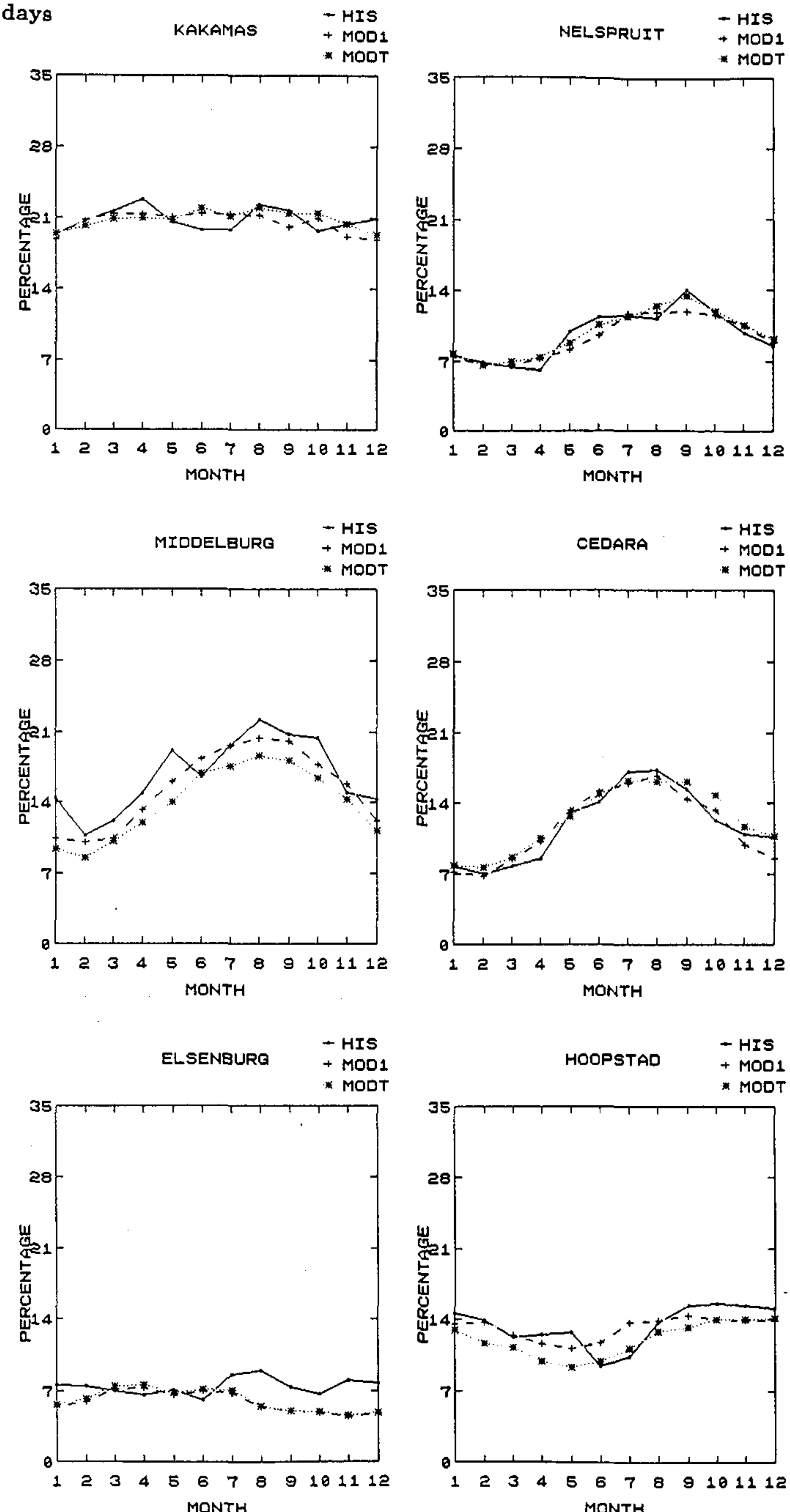
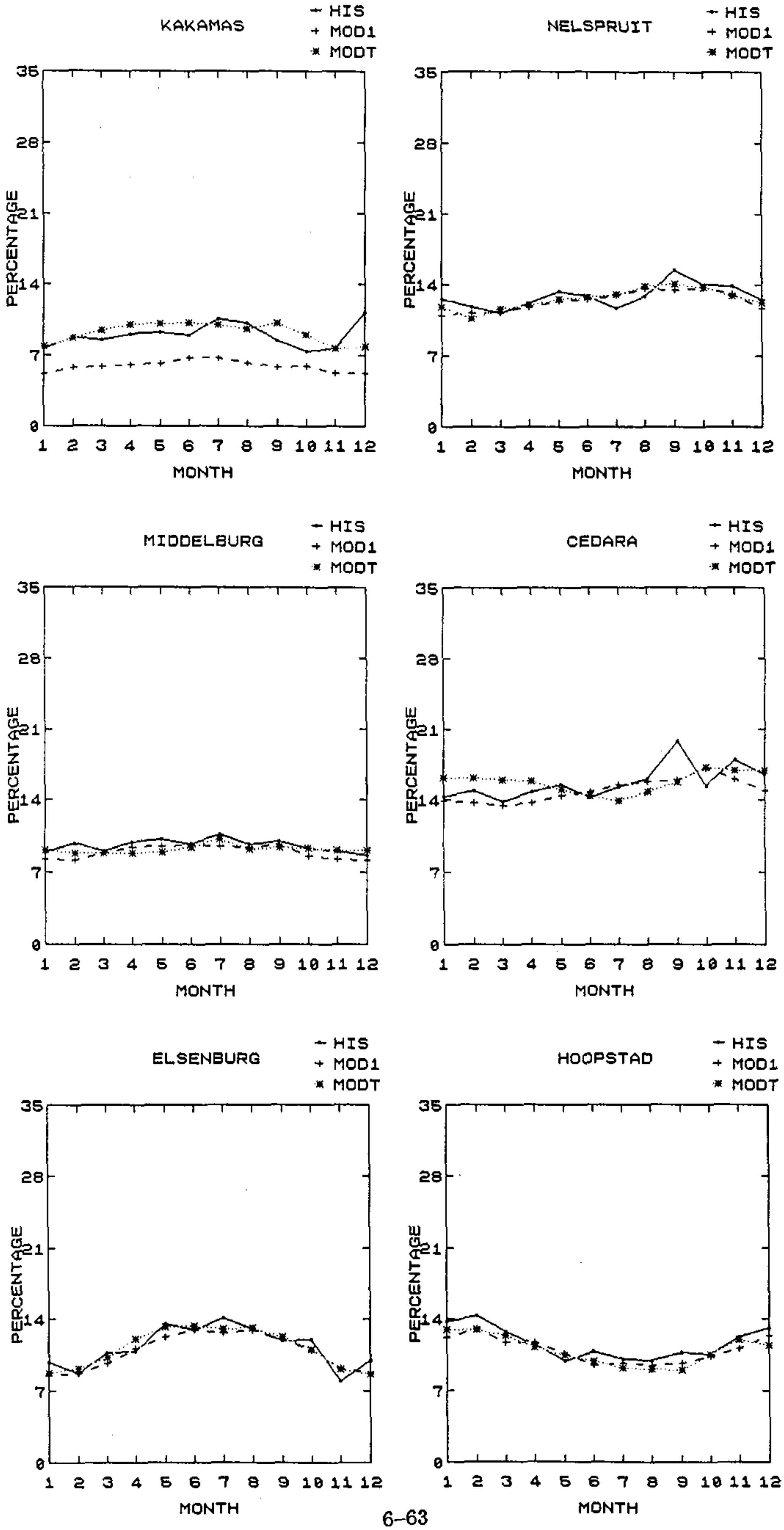


FIGURE 6.49 Monthly standard deviations for minimum humidity for dry days



Autocorrelation property

One of the properties exhibited by the historical record is that of autocorrelation, that is, within each climate variable there is a short–term persistence. For example, the temperature observed on a given day is statistically related to the temperature observed on the previous day.

To determine whether the models were successful in reproducing the autocorrelation structure that is present in the historical data, the autocorrelation coefficients (of up to a lag of four days) of each variable in the simulated sequence were compared to those of the historical record (Figures 6.50 – 6.57). From these comparisons it can be concluded that both models have described the autocorrelation property very well. Any differences observed between the simulated and historical sequences are mostly within 0.1 of the historical record. The variables that show these differences are generally sunshine, maximum humidity and minimum humidity. It must be noted here that the models assume an autoregressive process of order 1, that is, a lag of one day, and the bigger differences observed occur for lags of two or more days. Models with a higher autoregressive order might describe the autocorrelation structure of these variables, but this would mean increasing the complexity and the number of parameters in the models.

The autocorrelation coefficients of the simulated sequences were compared with those of the historical data, both for wet (Figures 6.57 – 6.63) and for dry sequences (Figures 6.64 – 6.70). The plots show that the autocorrelation structure in the simulated sequences closely resembles that of the observed data. Again the differences that are observed between the simulated and the historical sequences are mostly within 0.1 of the historical record.

Cross–correlation property

The cross–correlation coefficients for lag -1, 0 and 1 were used in the simulation technique. Therefore, it is necessary that the models should maintain this property. Figures 6.71 – 6.91 show the comparison of the historical and simulated cross–correlation coefficients for all climate variables. Generally, the models have successfully preserved the cross–correlation coefficients, in particular the lag 0 cross–correlation. The only exceptions to this are the cross–correlation coefficients of the simulated sequence of Model T, between the variables maximum and minimum humidity and the other climate variables, in particular for the stations Kakamas, Middelburg and Hoopstad. The cross–correlation of other variables

FIGURE 6.50 Autocorrelation coefficients for maximum temperature

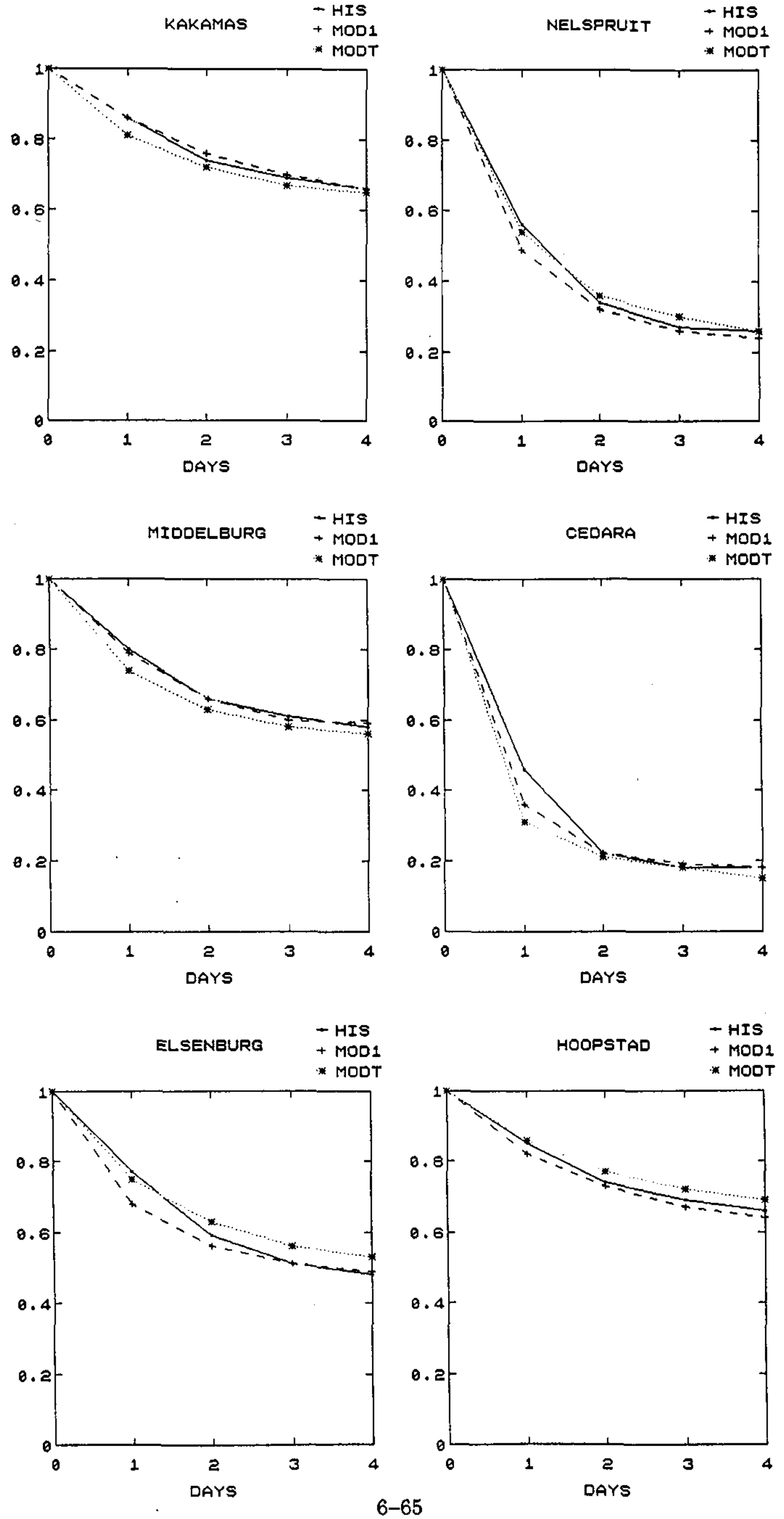


FIGURE 6.51 Autocorrelation coefficients for minimum temperature

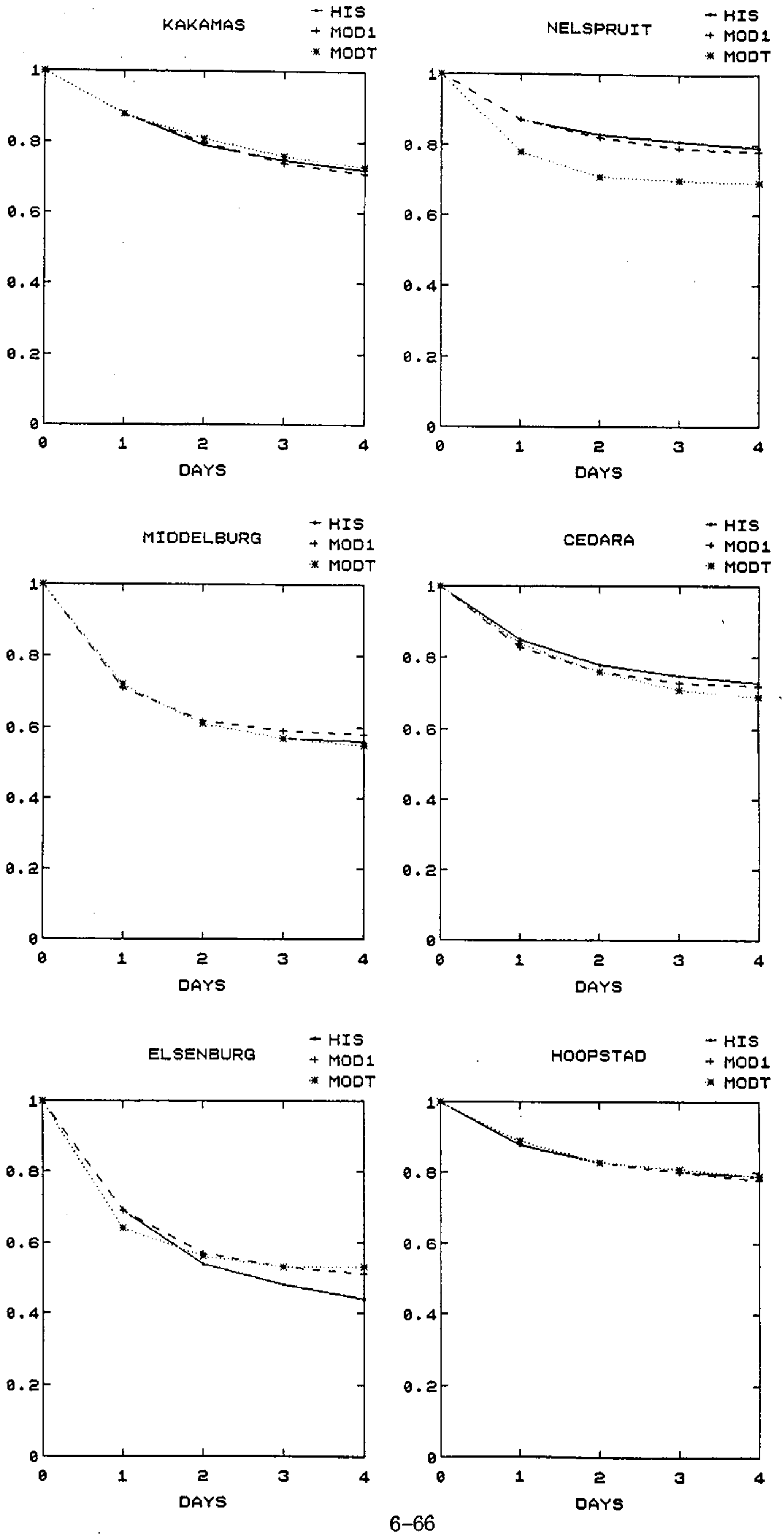


FIGURE 6.52 Autocorrelation coefficients for evaporation

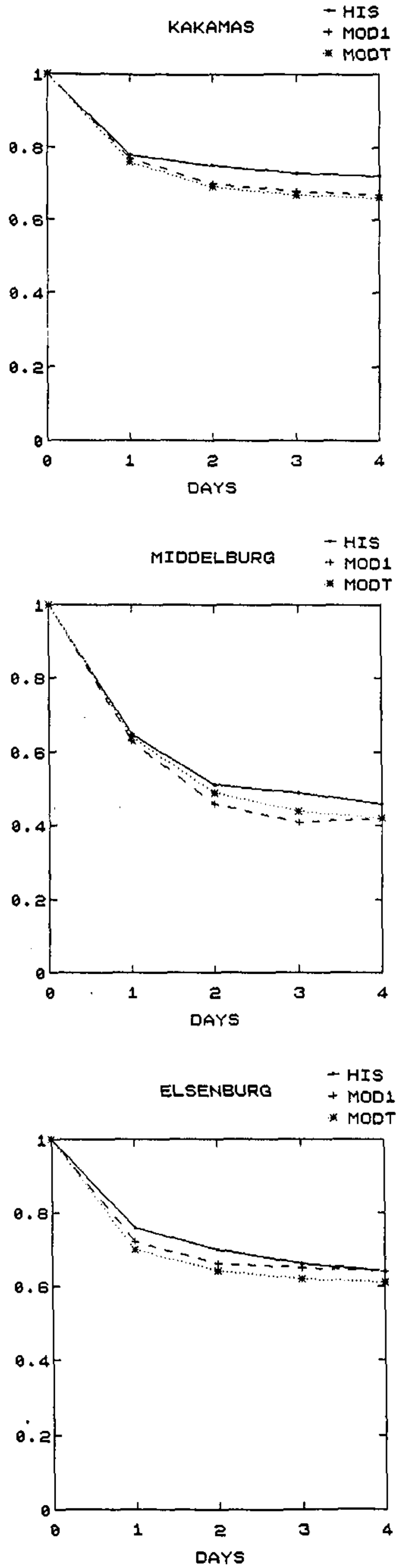


FIGURE 6.53 Autocorrelation coefficients for sunshine duration

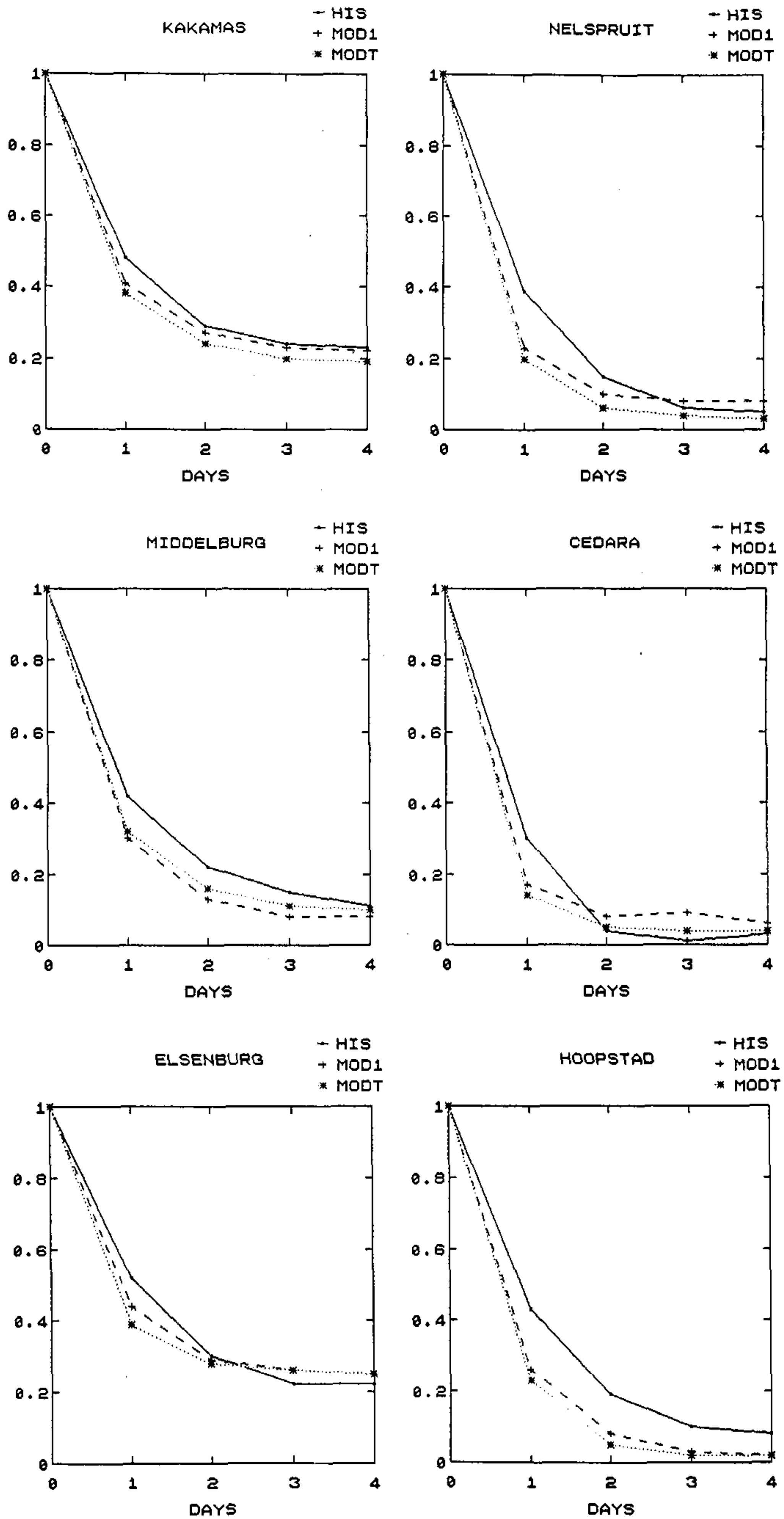


FIGURE 6.54 Autocorrelation coefficients for wind run

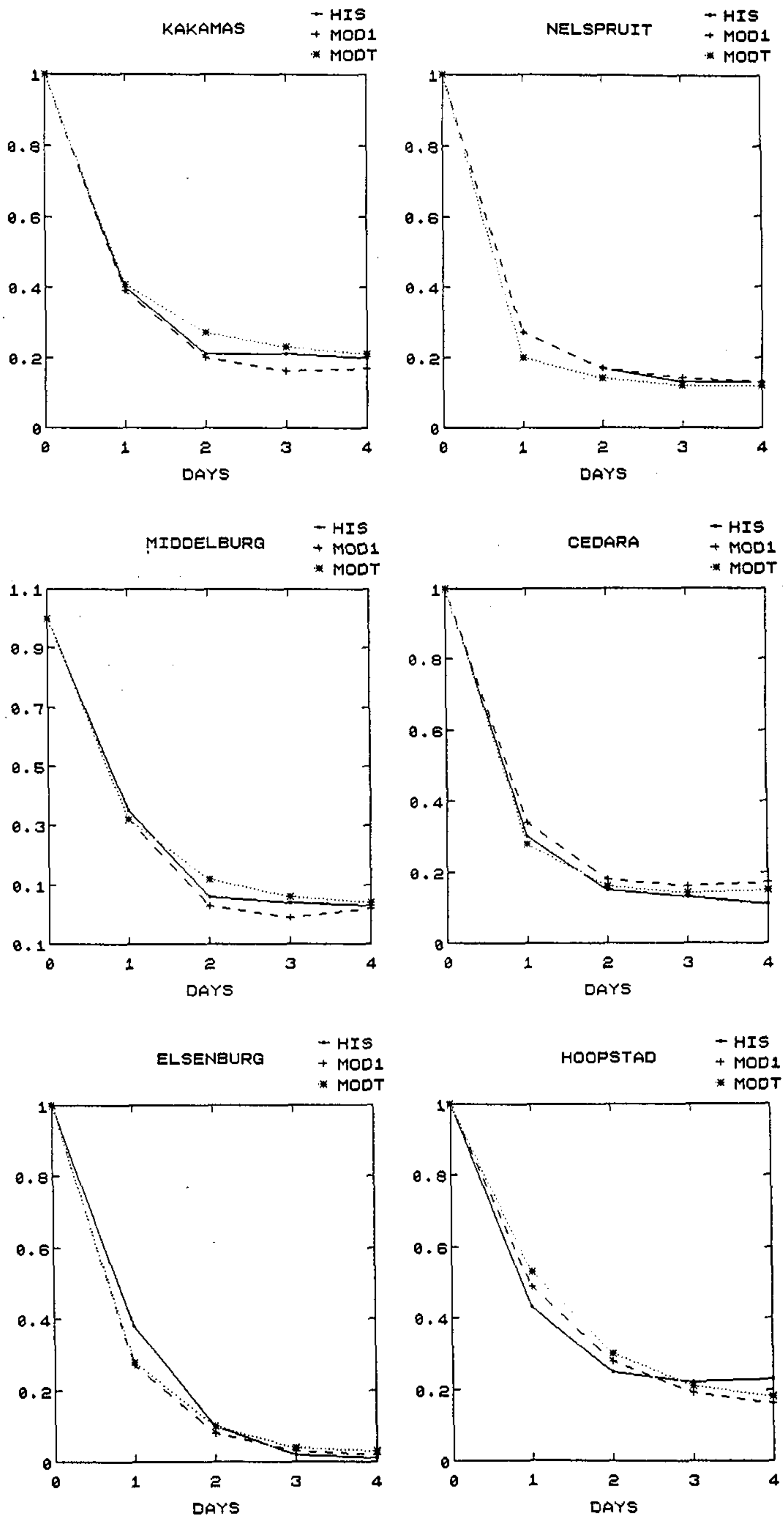


FIGURE 6.55 Autocorrelation coefficients for maximum humidity

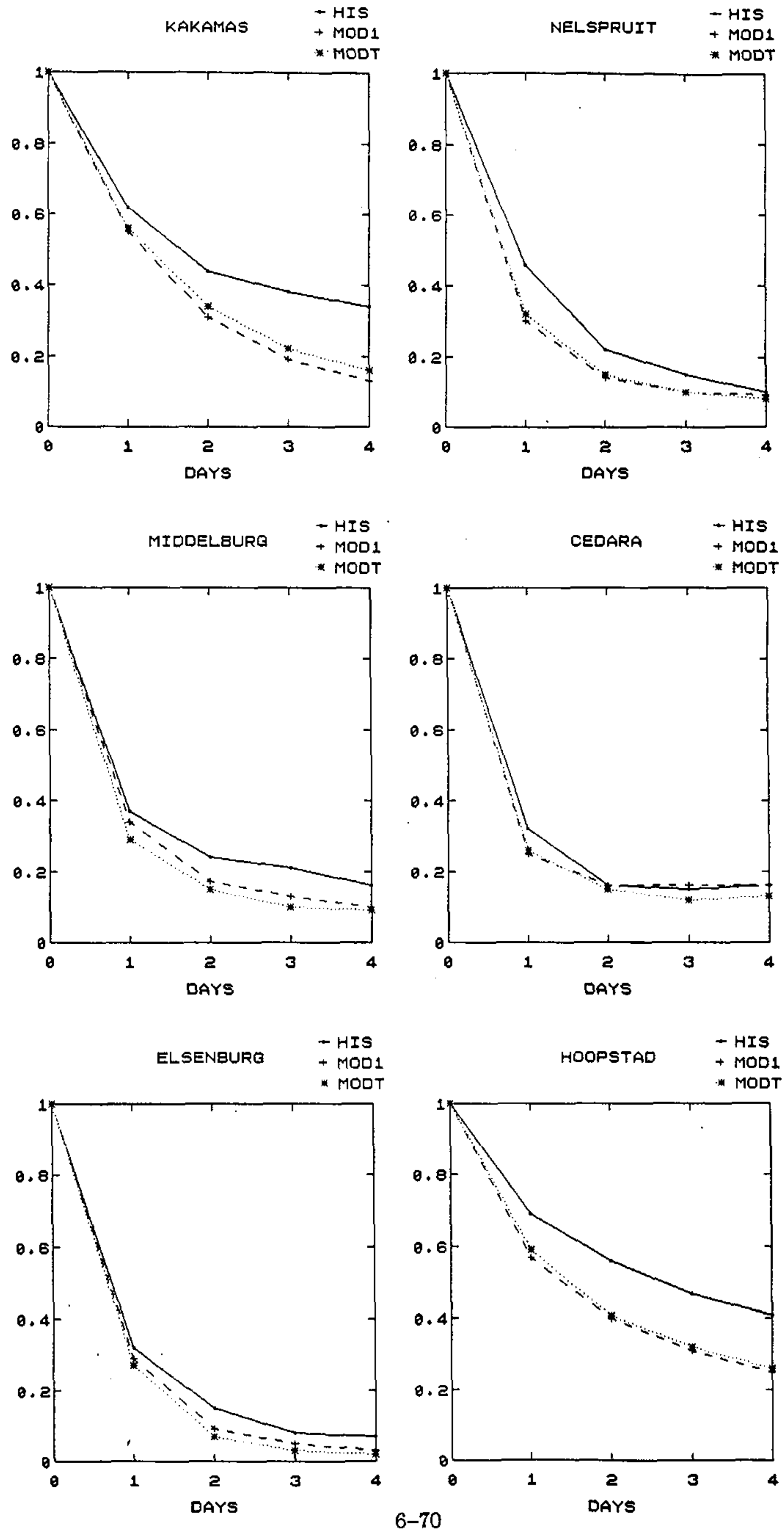


FIGURE 6.56 Autocorrelation coefficients for minimum humidity

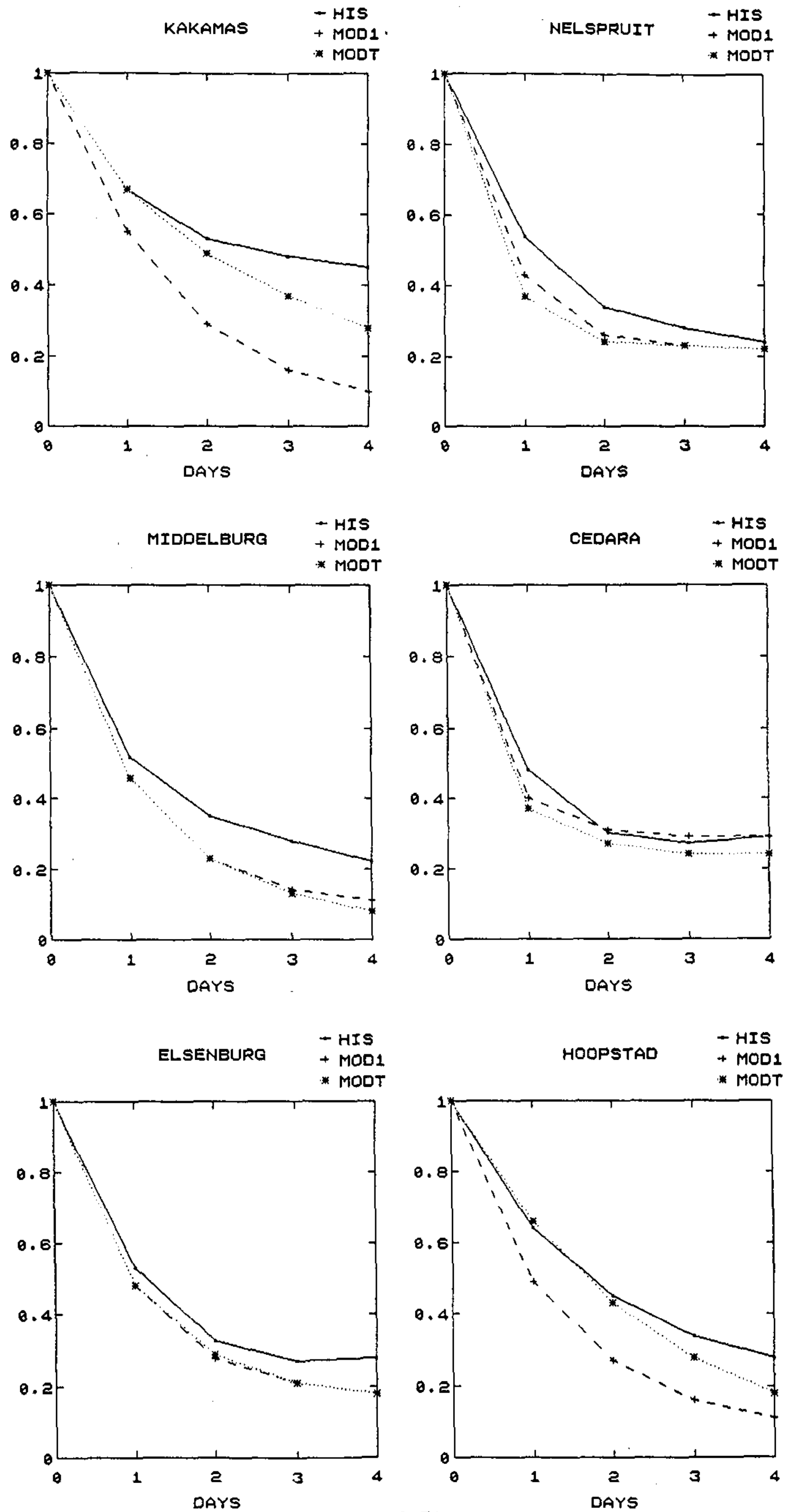


FIGURE 6.57 Autocorrelation coefficients for maximum temperature for wet days

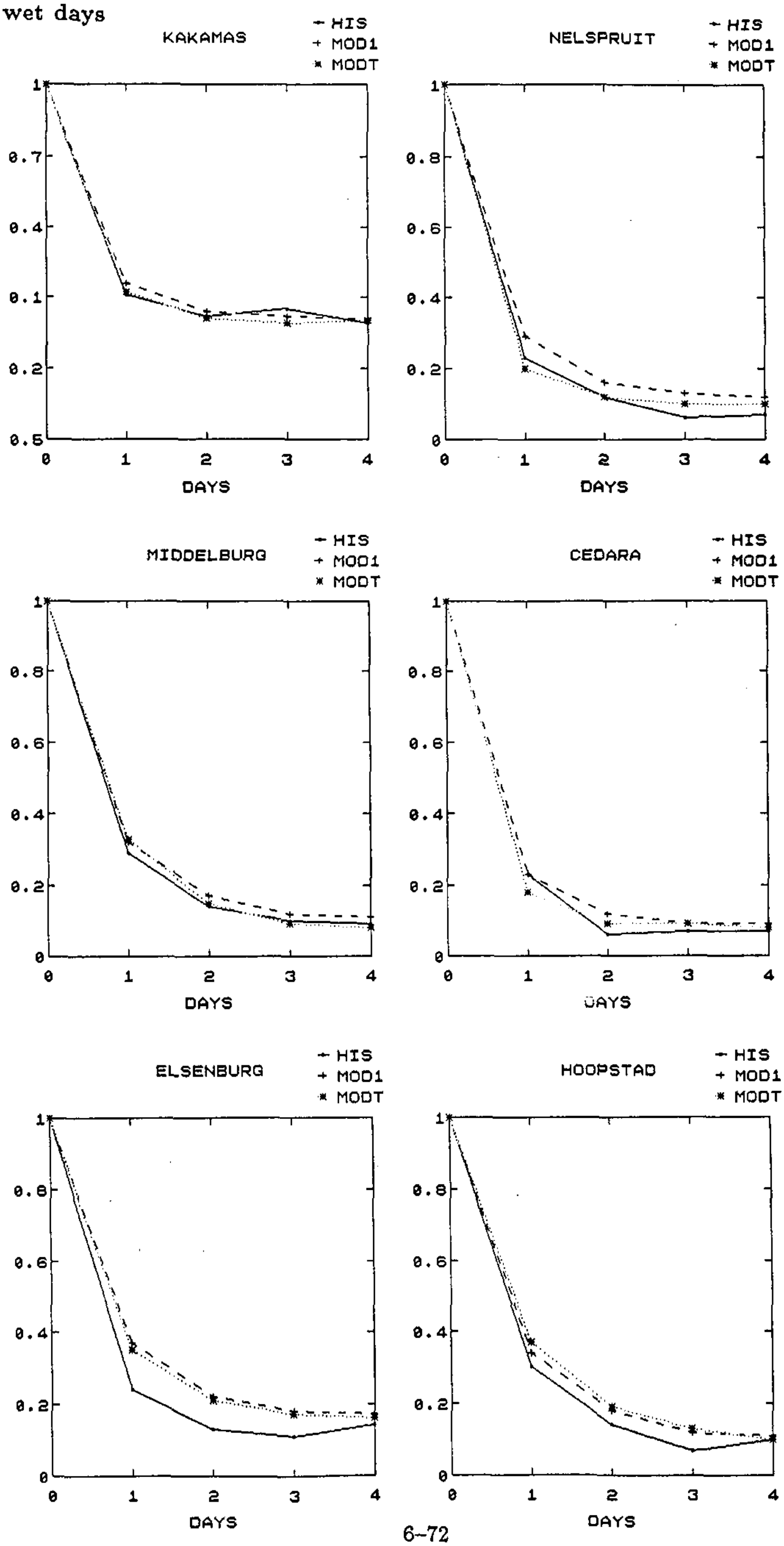


FIGURE 6.58 Autocorrelation coefficients for minimum temperature for wet days

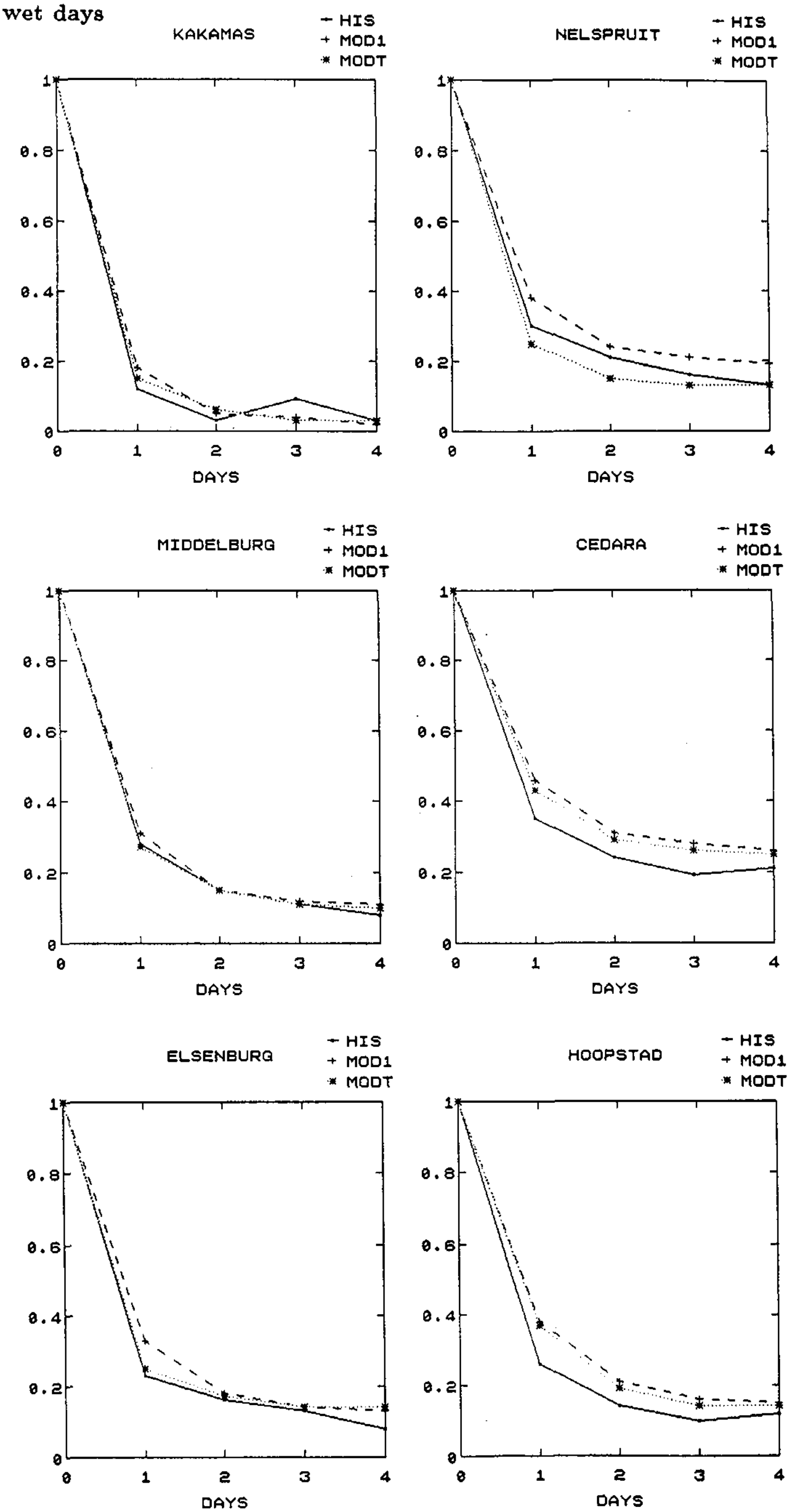


FIGURE 6.59 Autocorrelation coefficients for evaporation for wet days

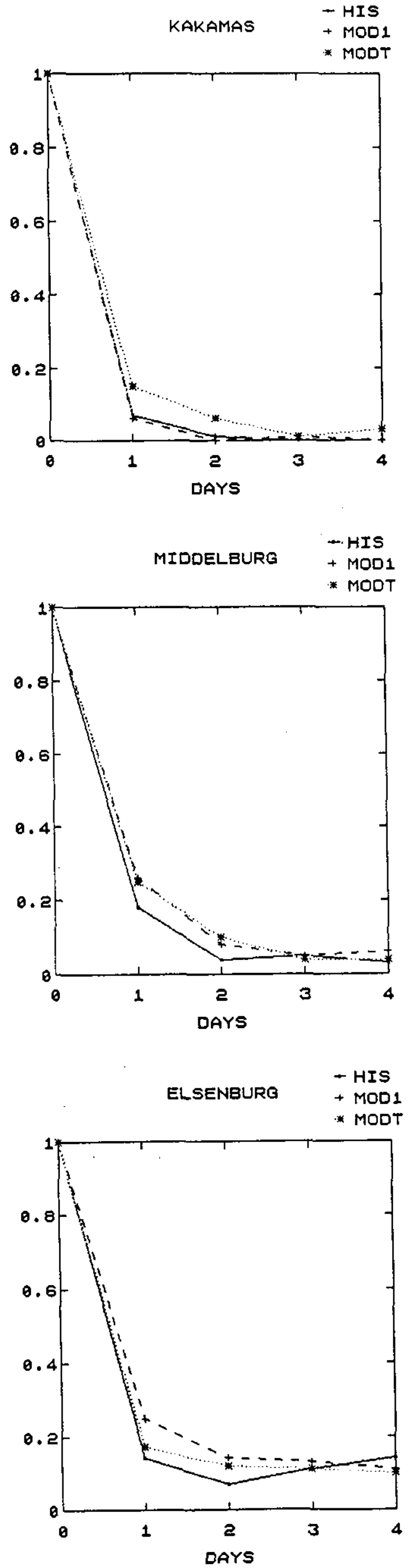


FIGURE 6.60 Autocorrelation coefficients for sunshine duration for wet days

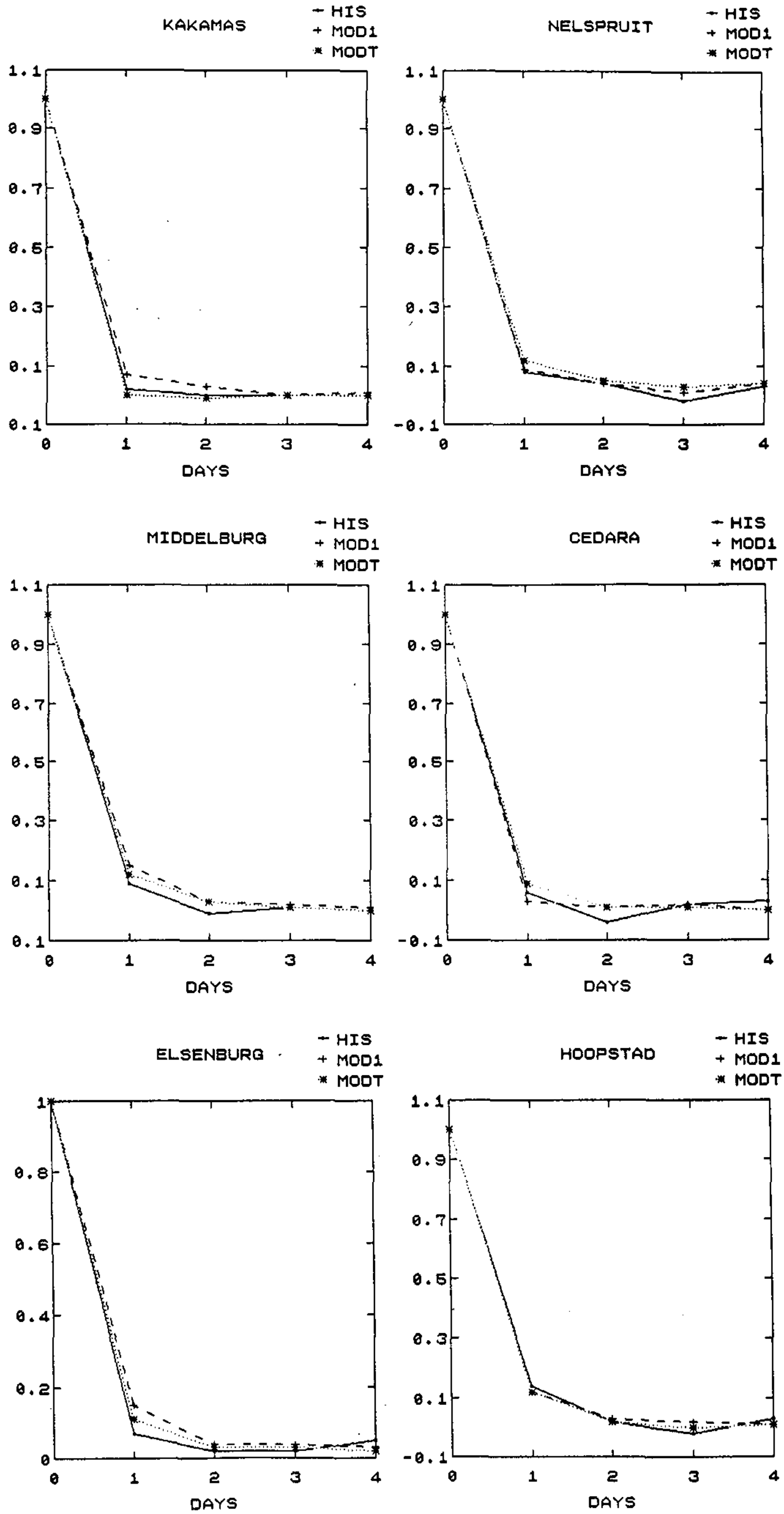


FIGURE 6.61 Autocorrelation coefficients for wind run for wet days

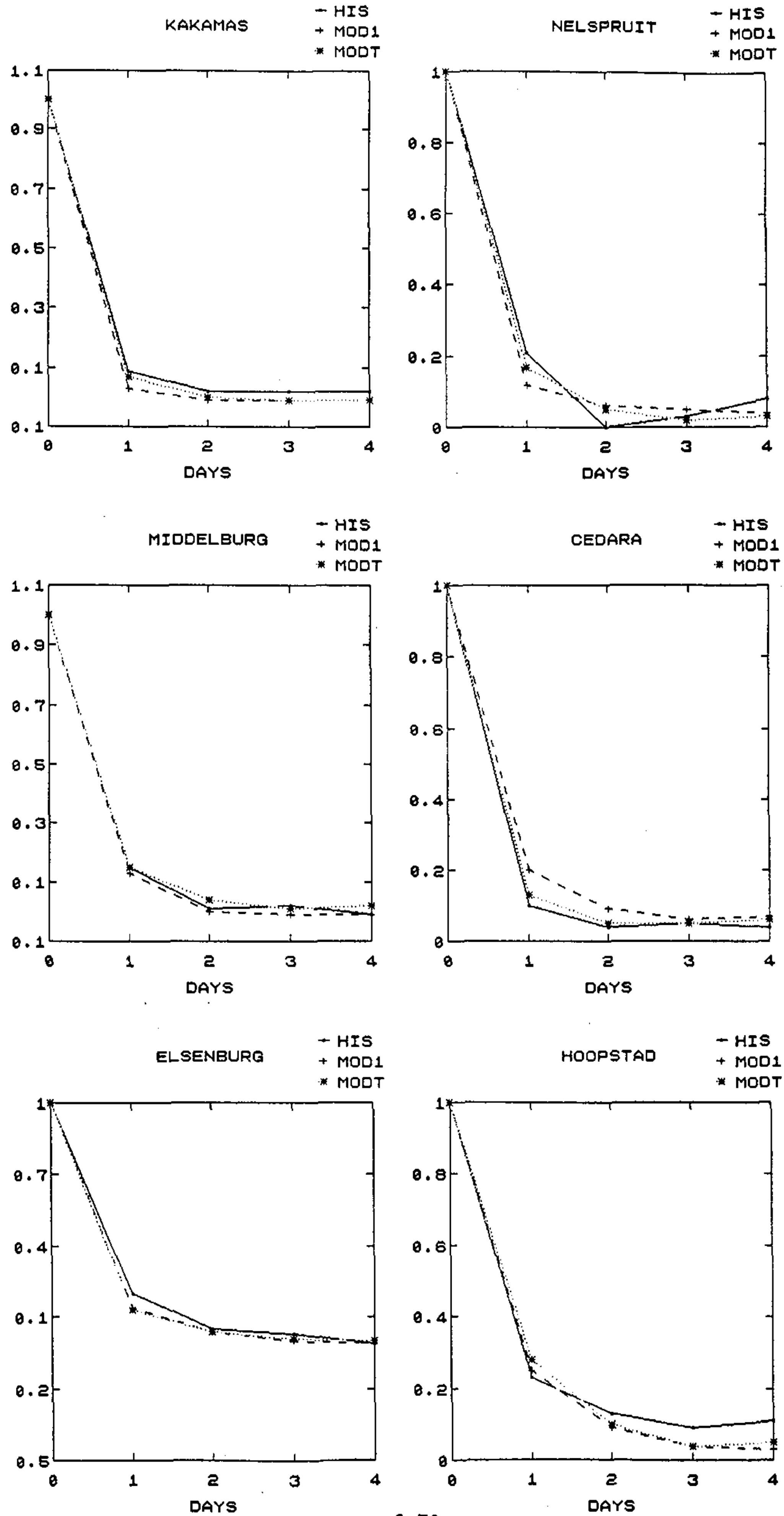


FIGURE 6.62 Autocorrelation coefficients for maximum humidity for wet days

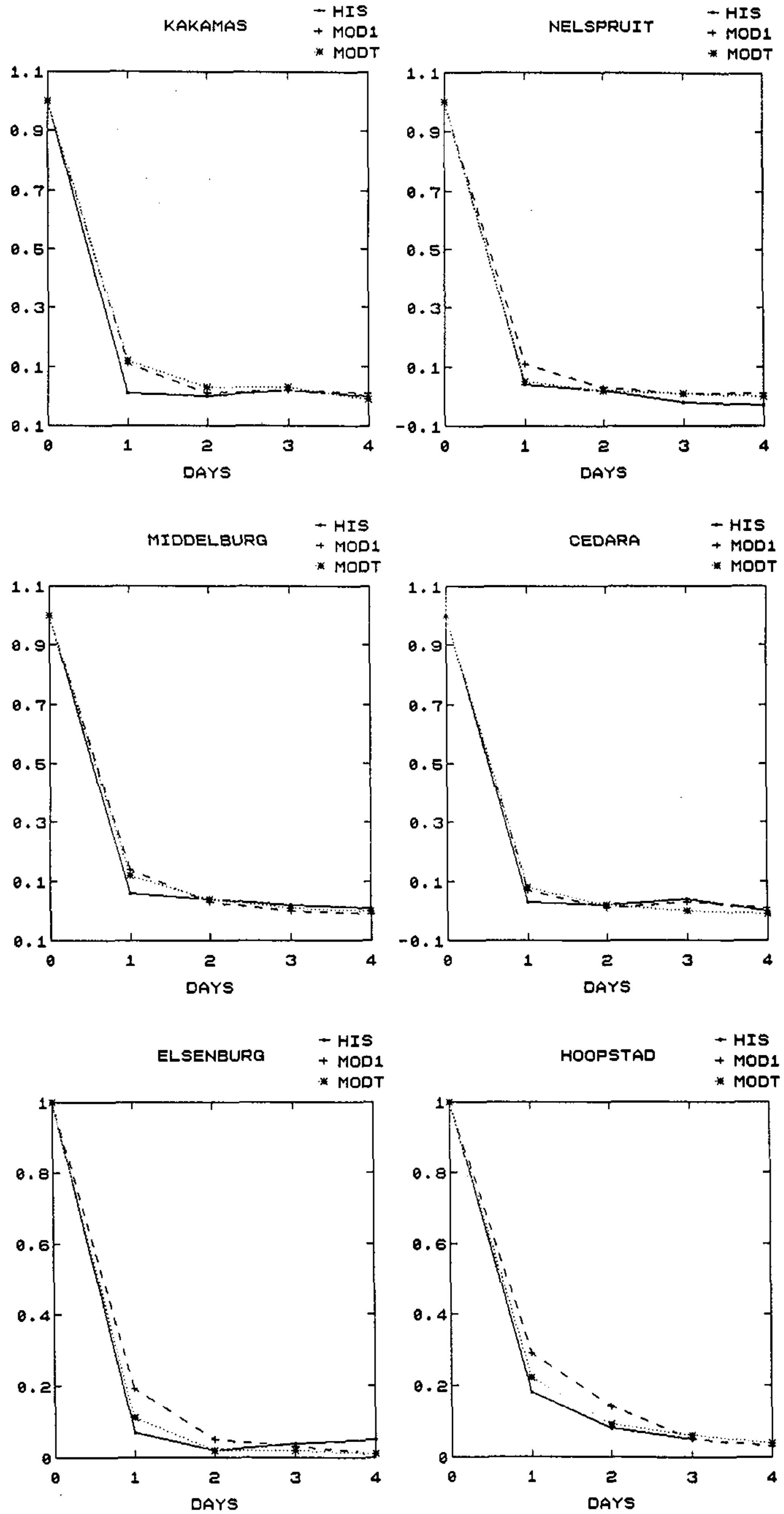


FIGURE 6.63 Autocorrelation coefficients for minimum humidity for wet days

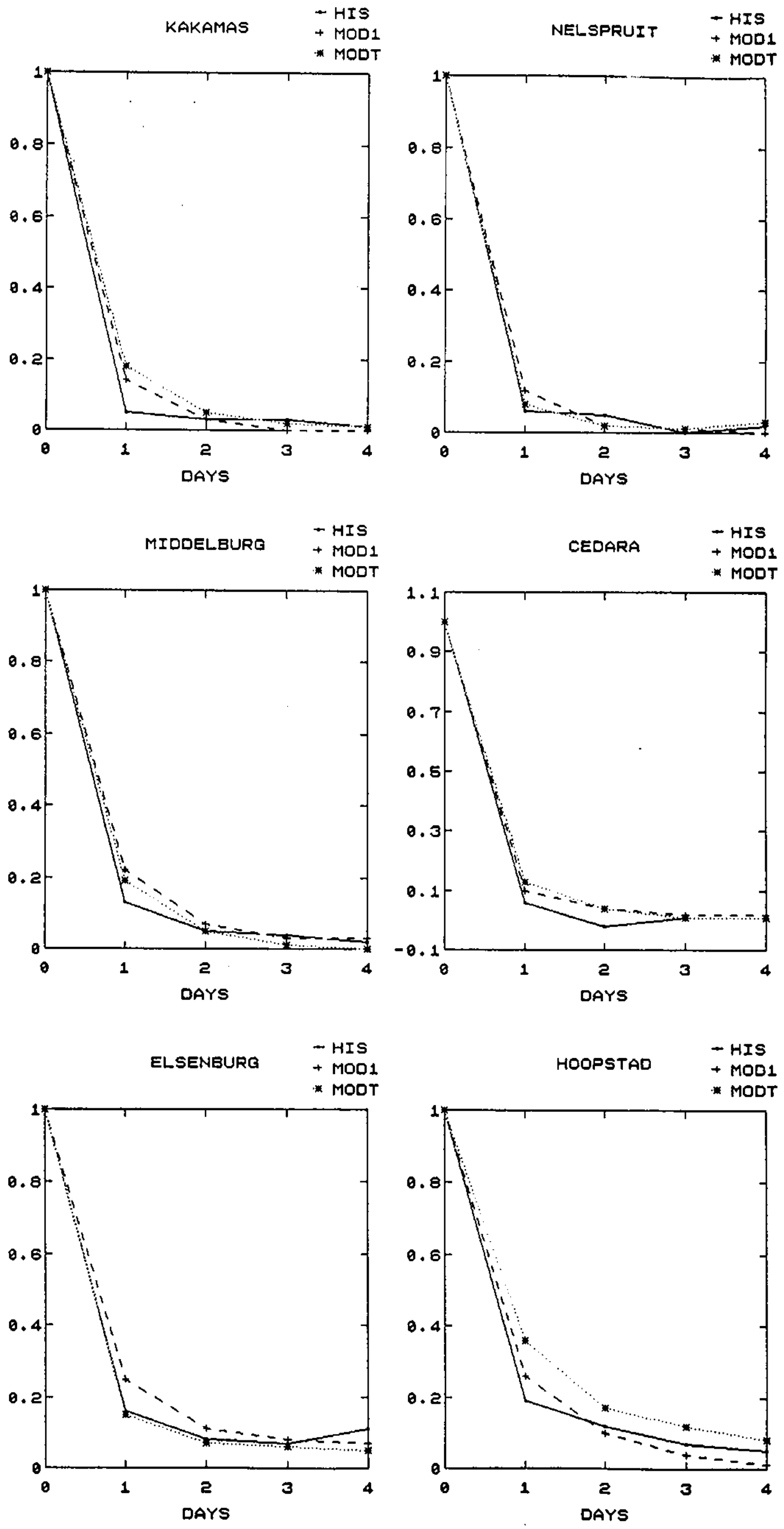


FIGURE 6.64 Autocorrelation coefficients for maximum temperature for dry days

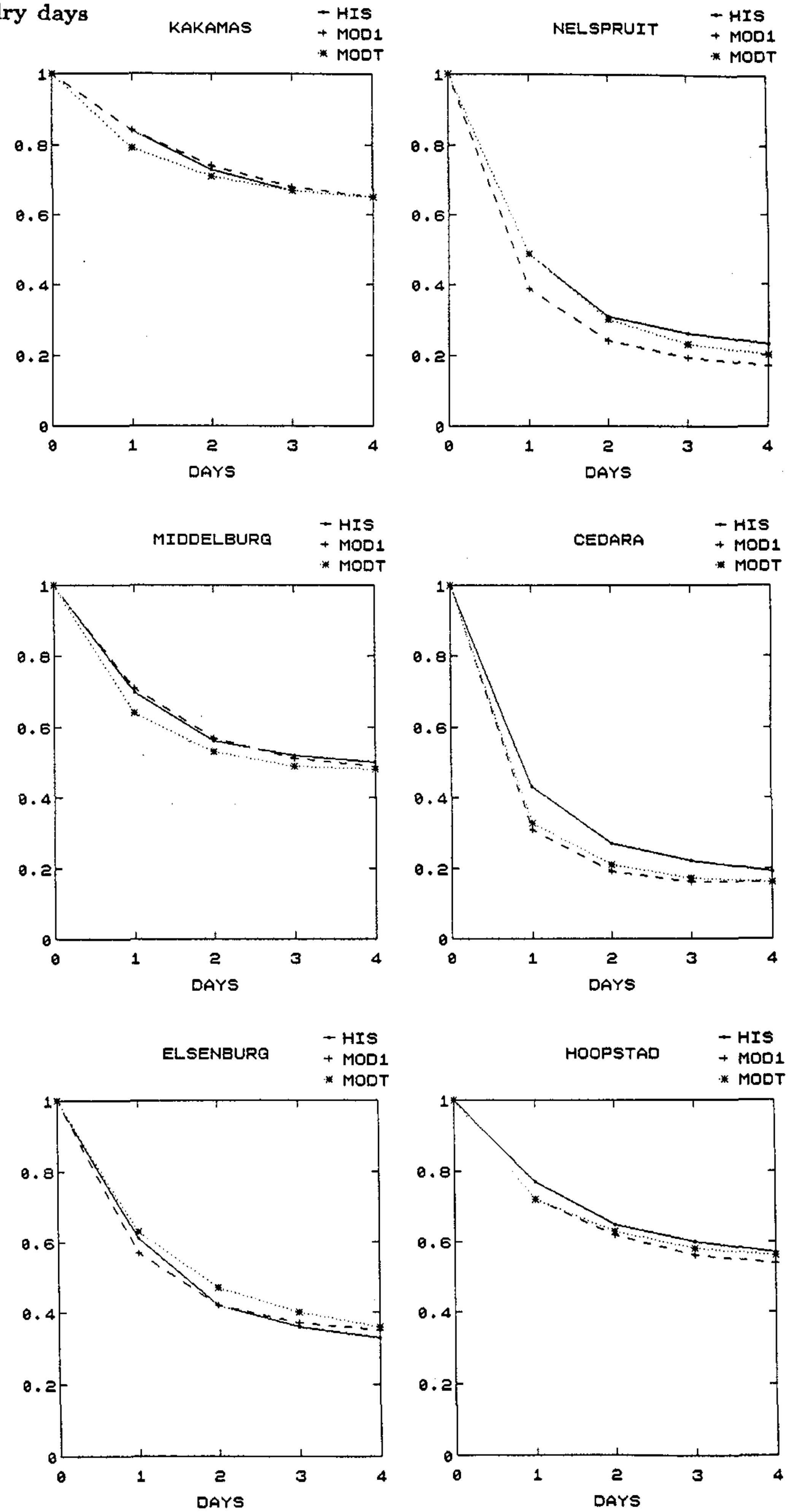


FIGURE 6.65 Autocorrelation coefficients for minimum temperature for dry days

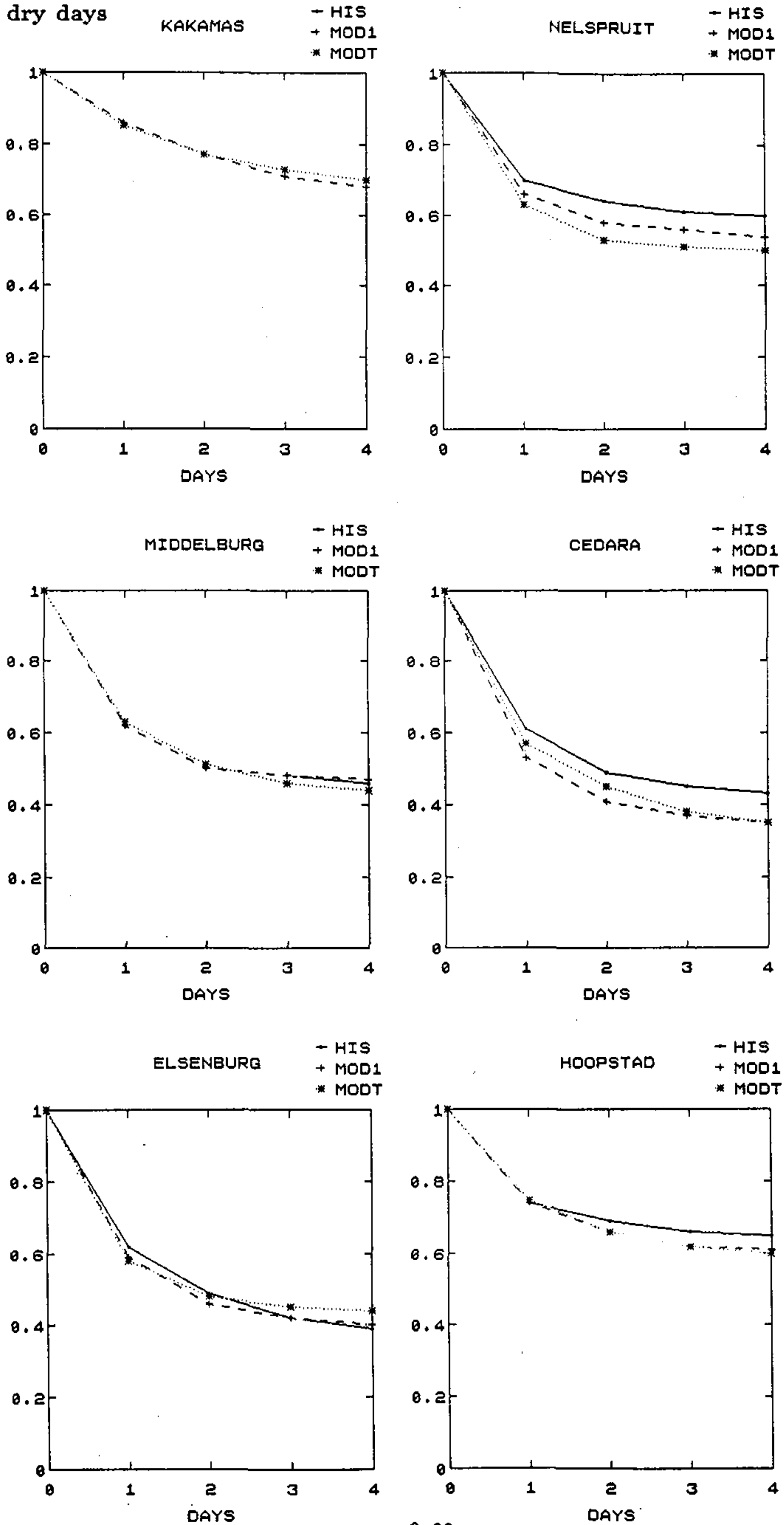


FIGURE 6.66 Autocorrelation coefficients for evaporation for dry days

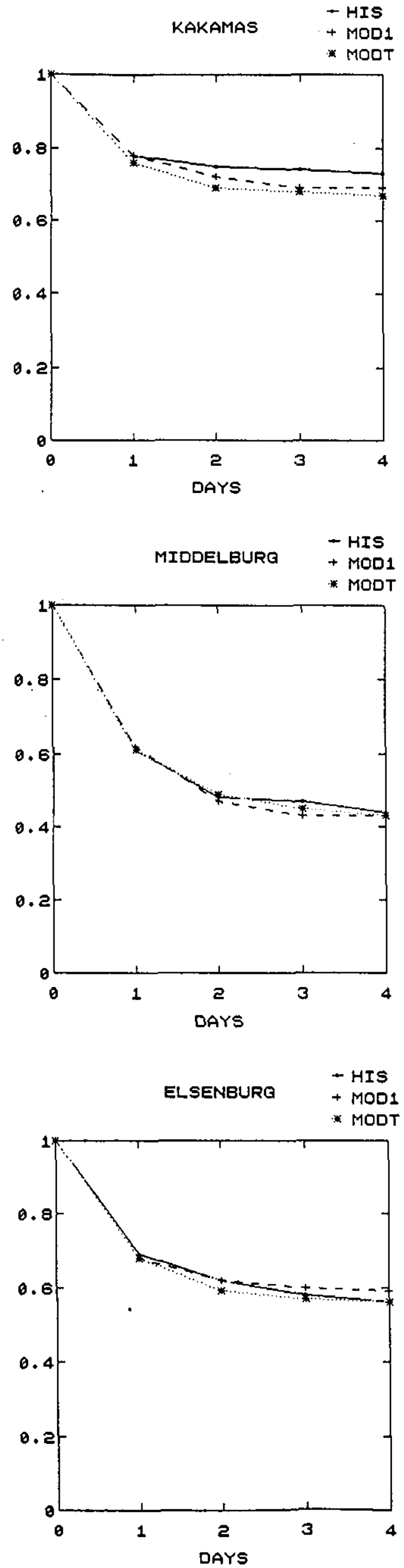


FIGURE 6.67 Autocorrelation coefficients for sunshine duration for dry days

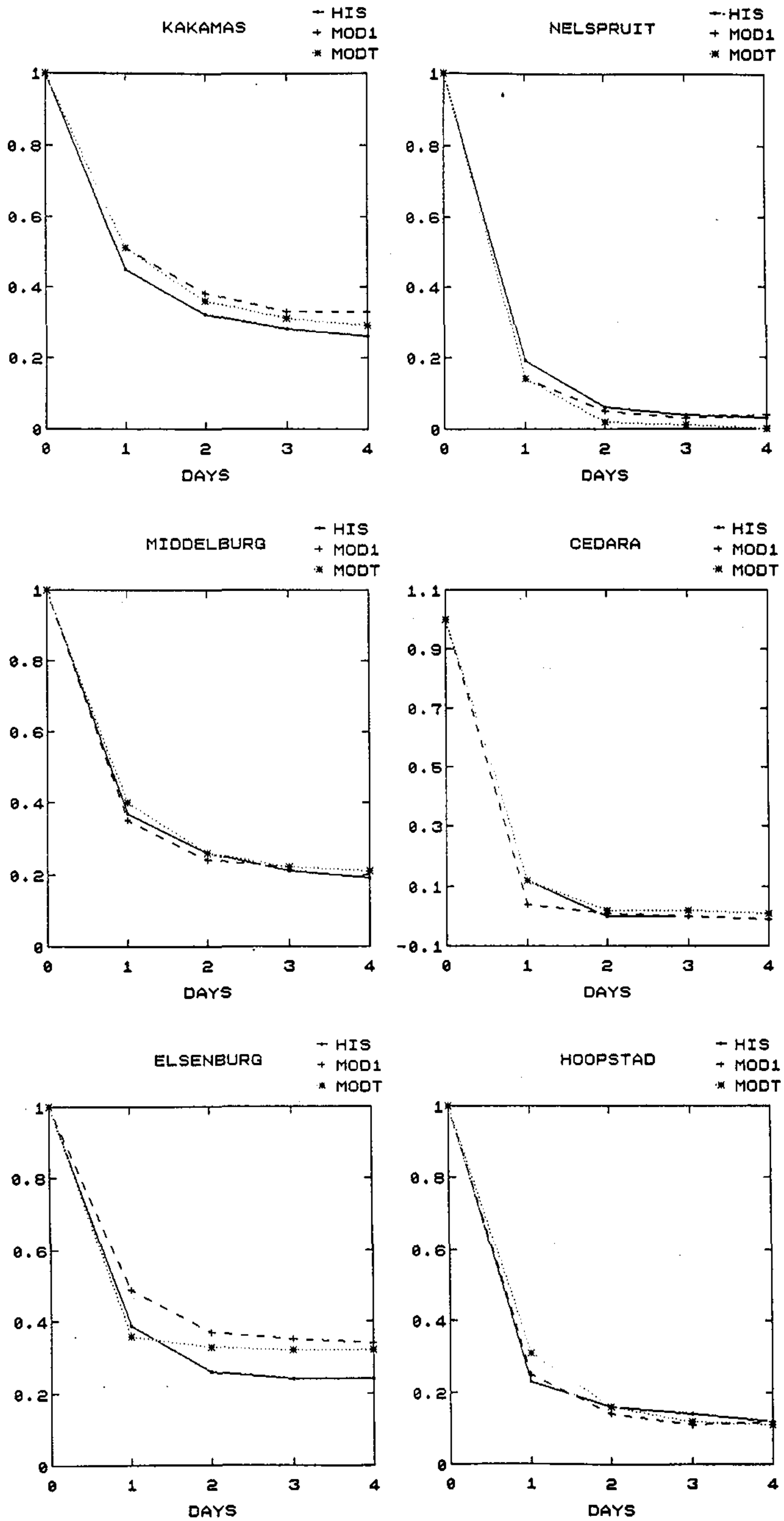


FIGURE 6.68 Autocorrelation coefficients for wind run for dry days

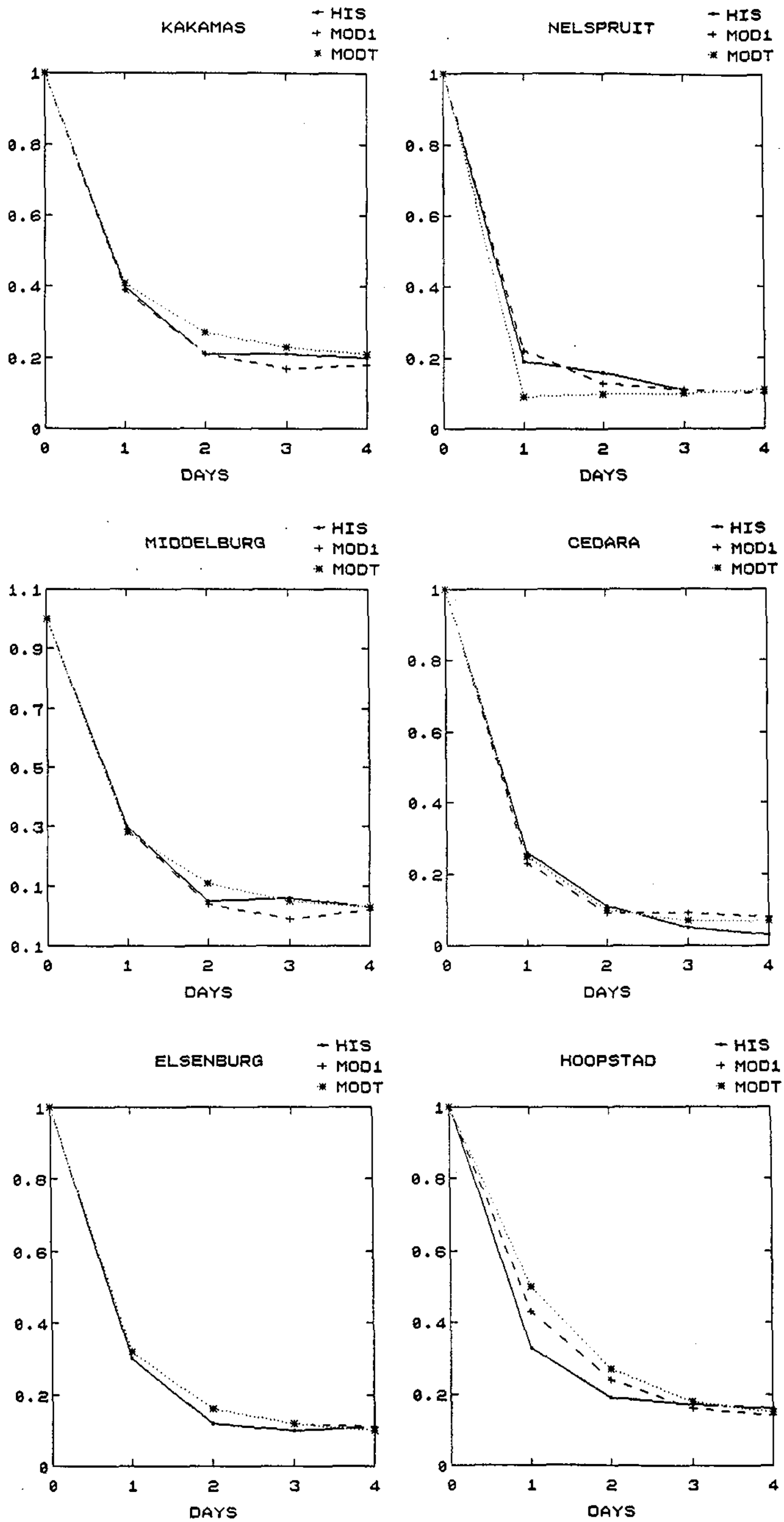


FIGURE 6.69 Autocorrelation coefficients for maximum humidity for dry days

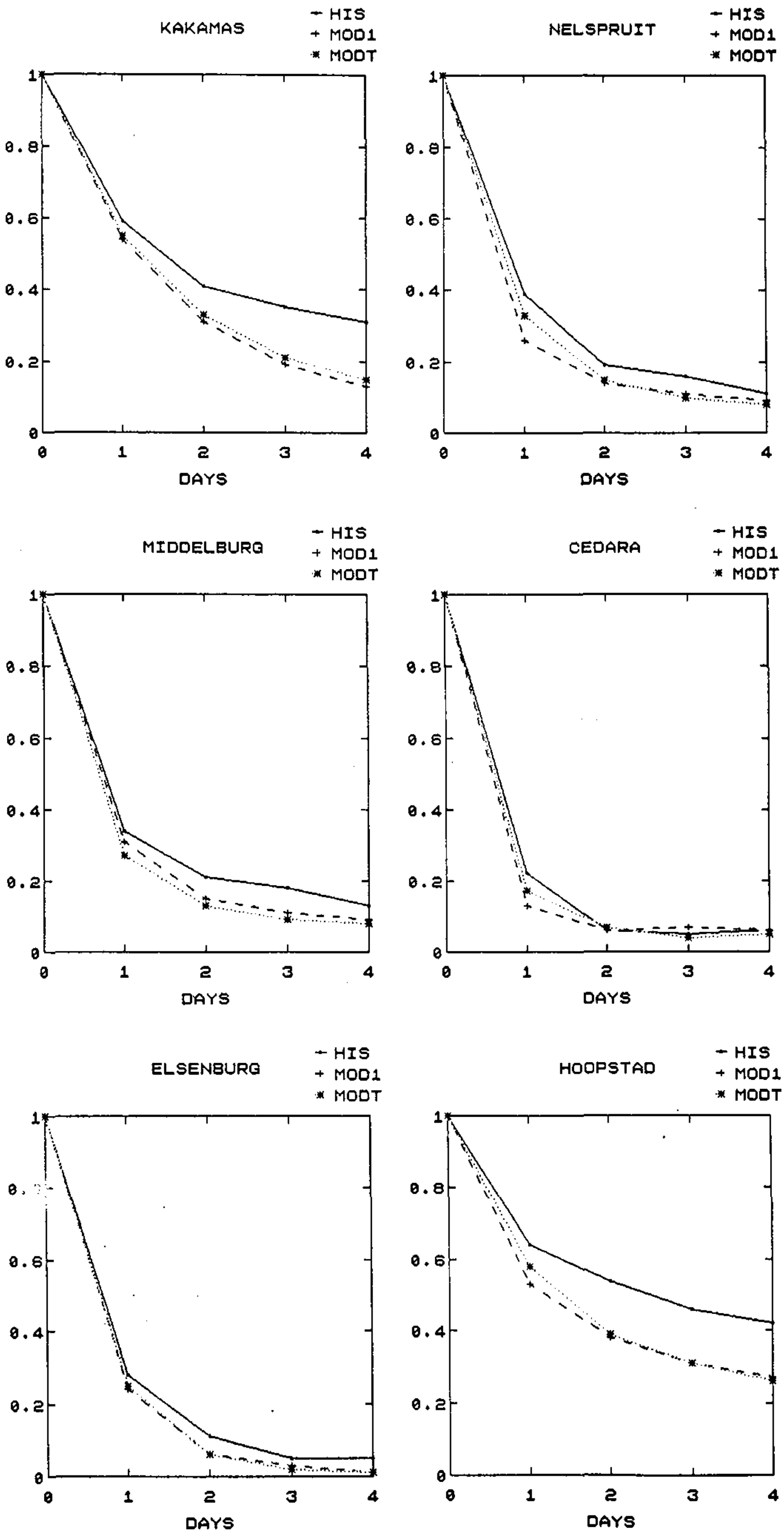


FIGURE 6.70 Autocorrelation coefficients for minimum humidity for dry days

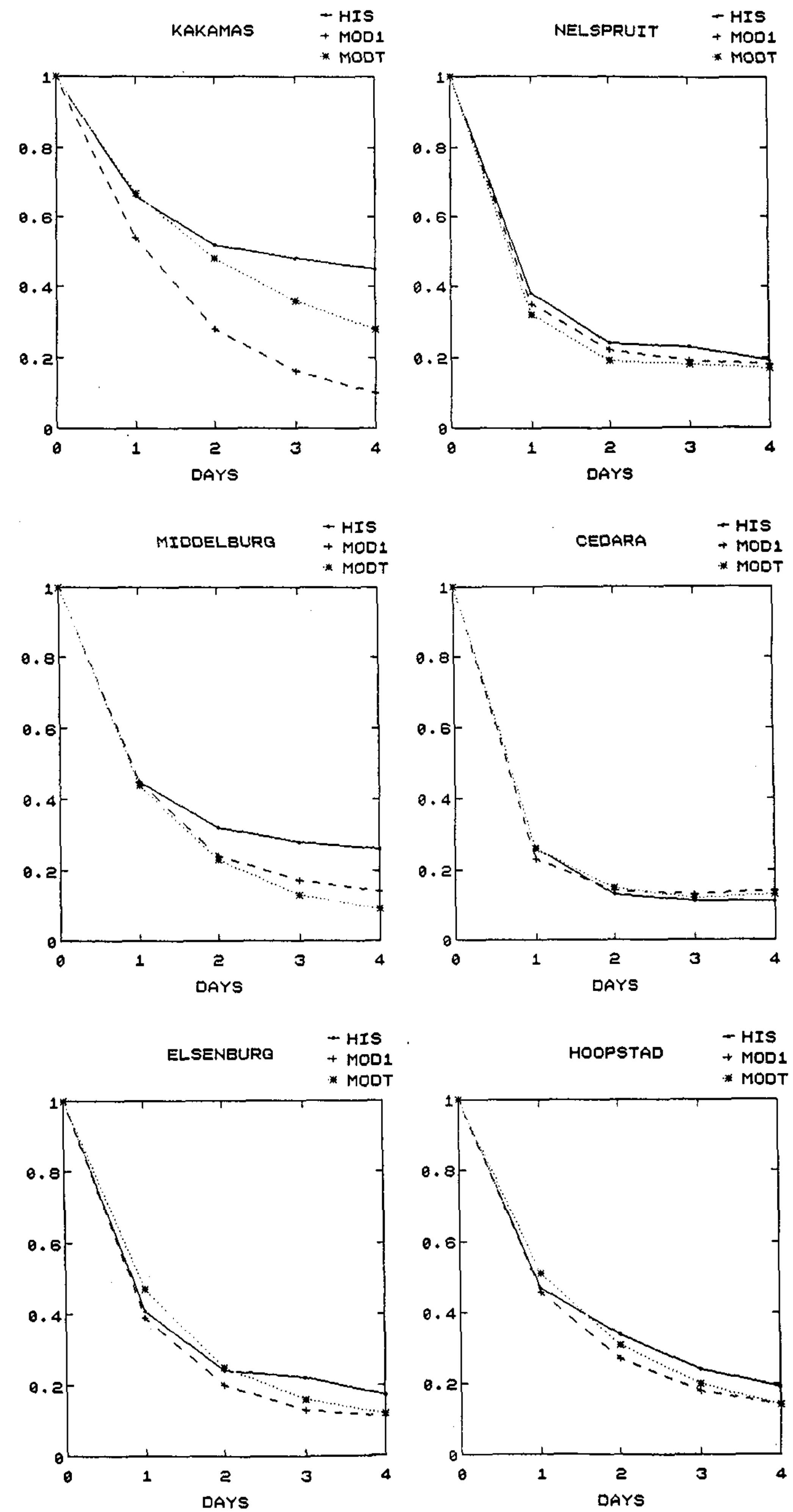


FIGURE 6.71 Cross-correlation coefficients for maximum temperature and minimum temperature

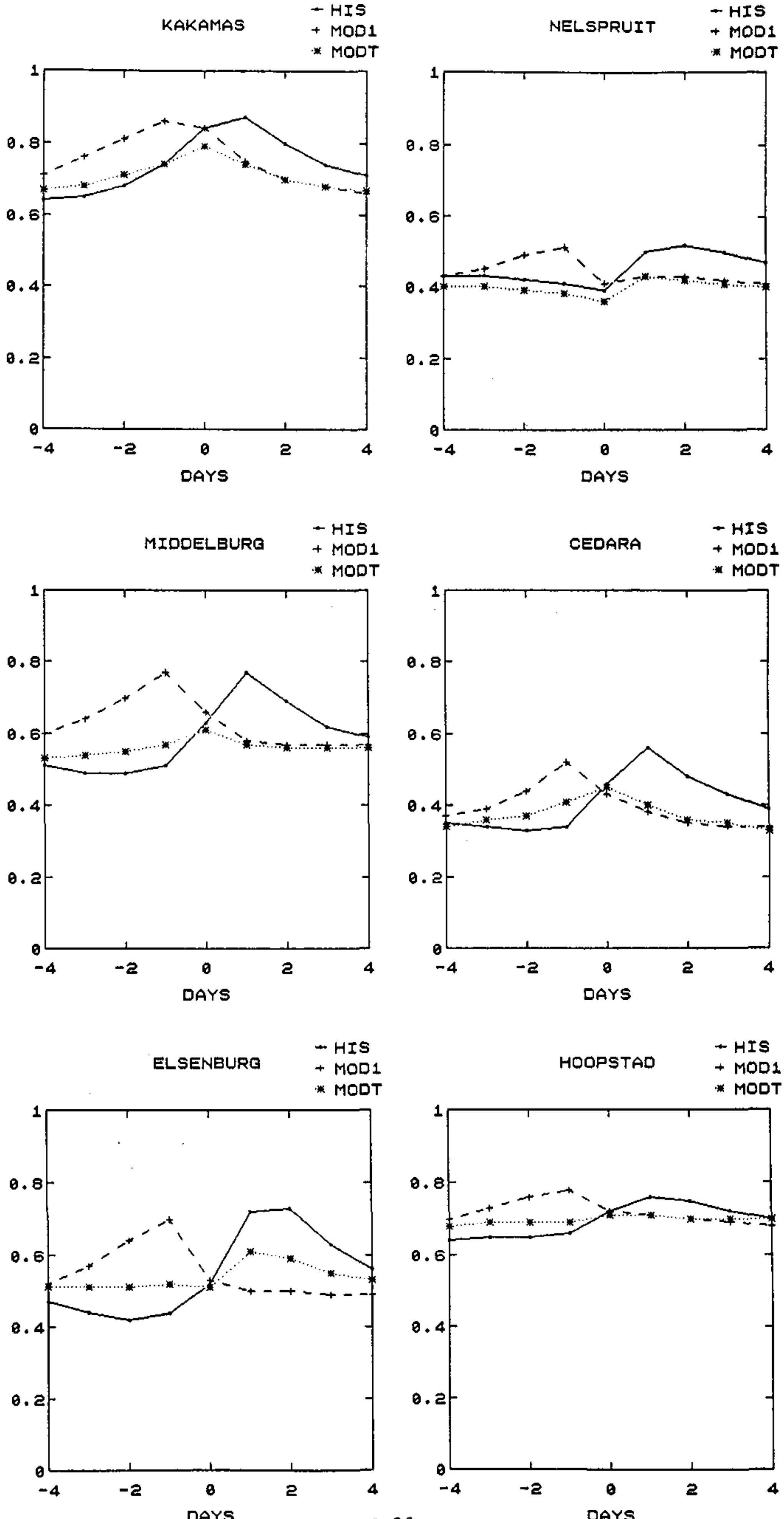


FIGURE 6.72 Cross-correlation coefficients for maximum temperature and evaporation

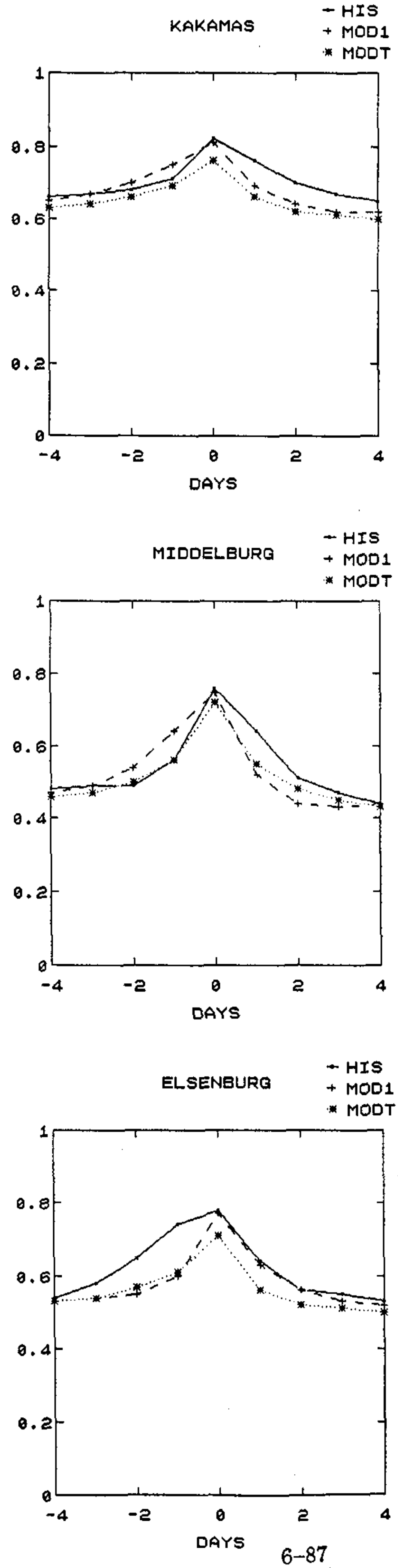


FIGURE 6.73 Cross–correlation coefficients for maximum temperature and sunshine duration

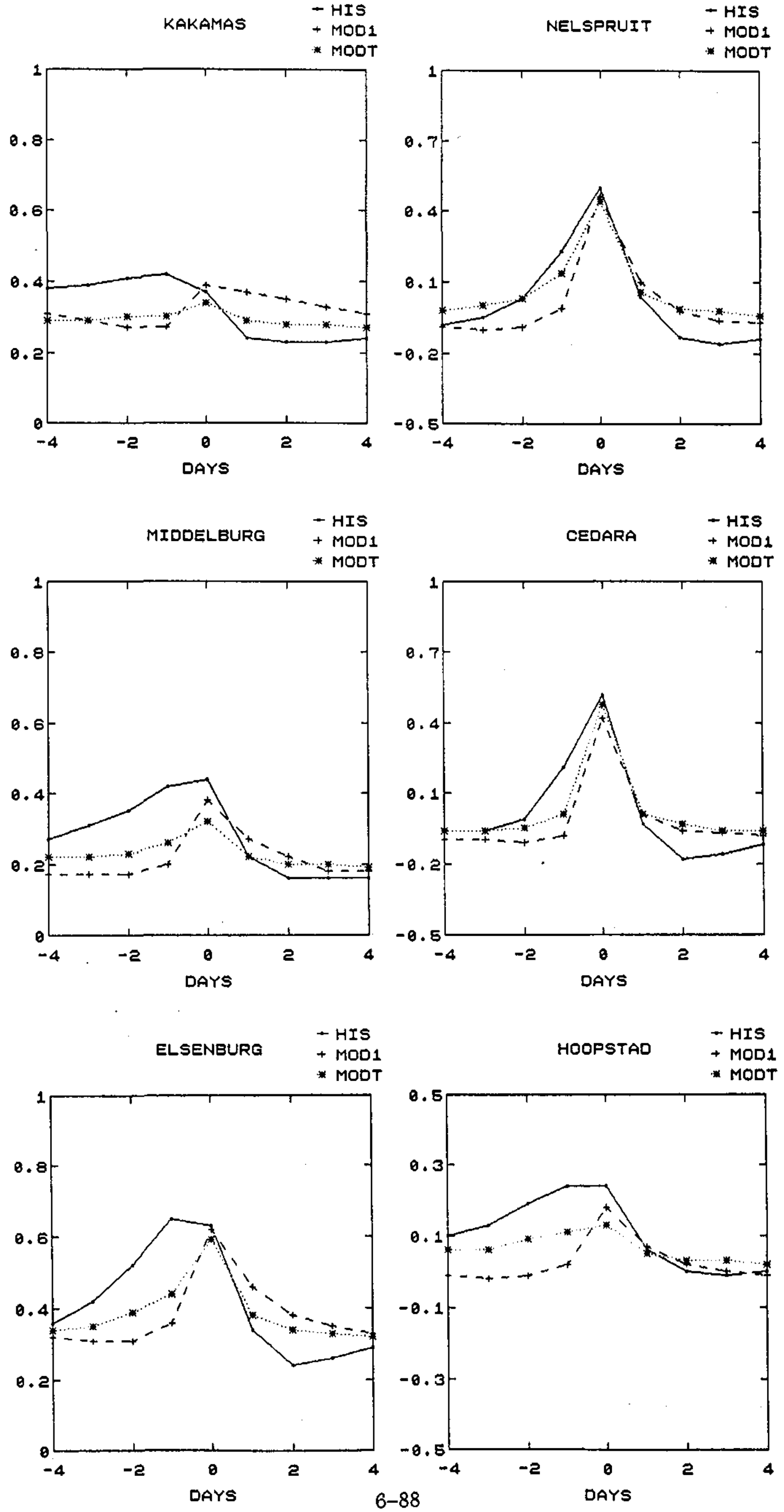


FIGURE 6.74 Cross-correlation coefficients for maximum temperature and wind run

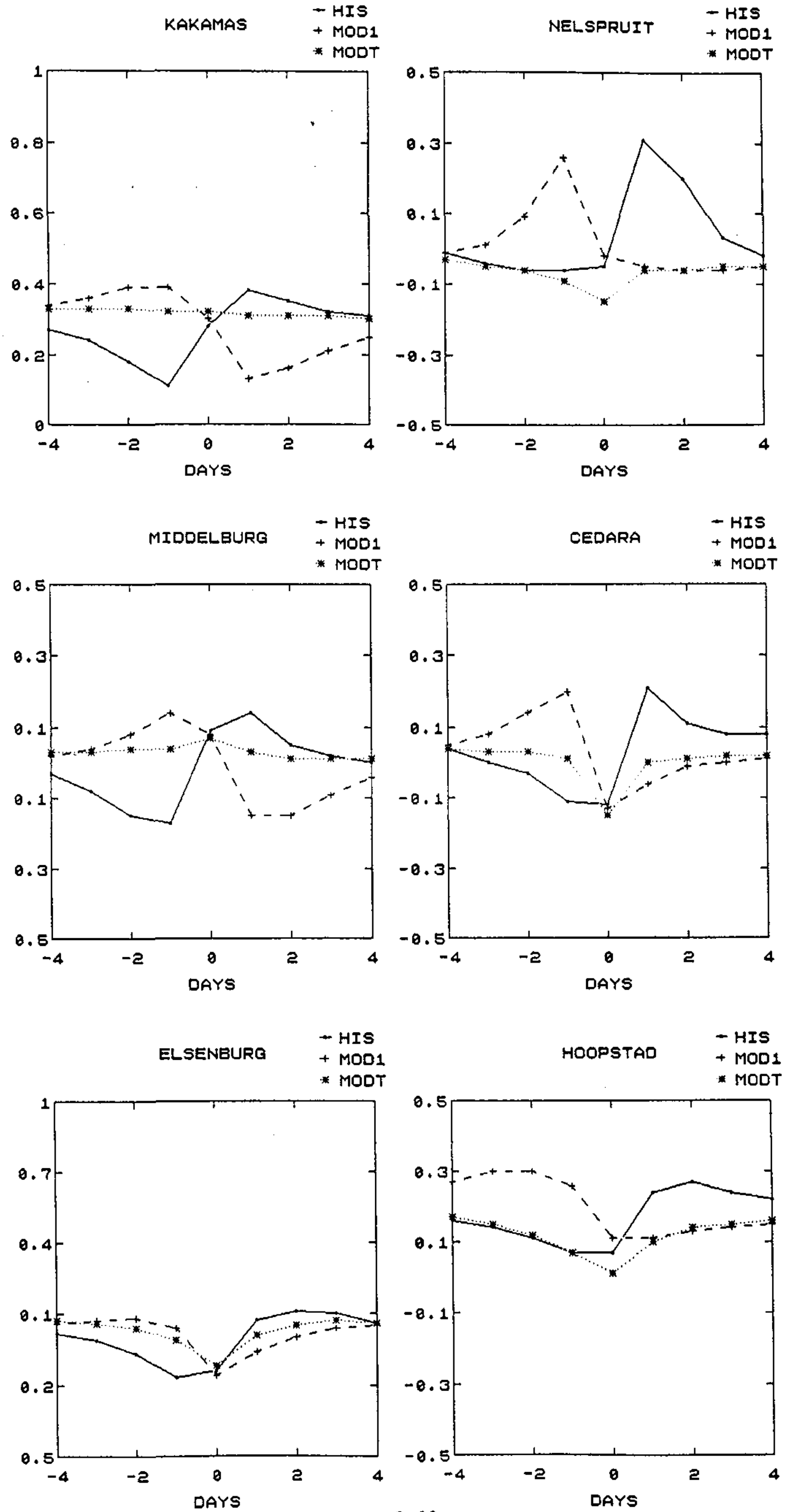


FIGURE 6.75 Cross-correlation coefficients for maximum temperature and maximum humidity

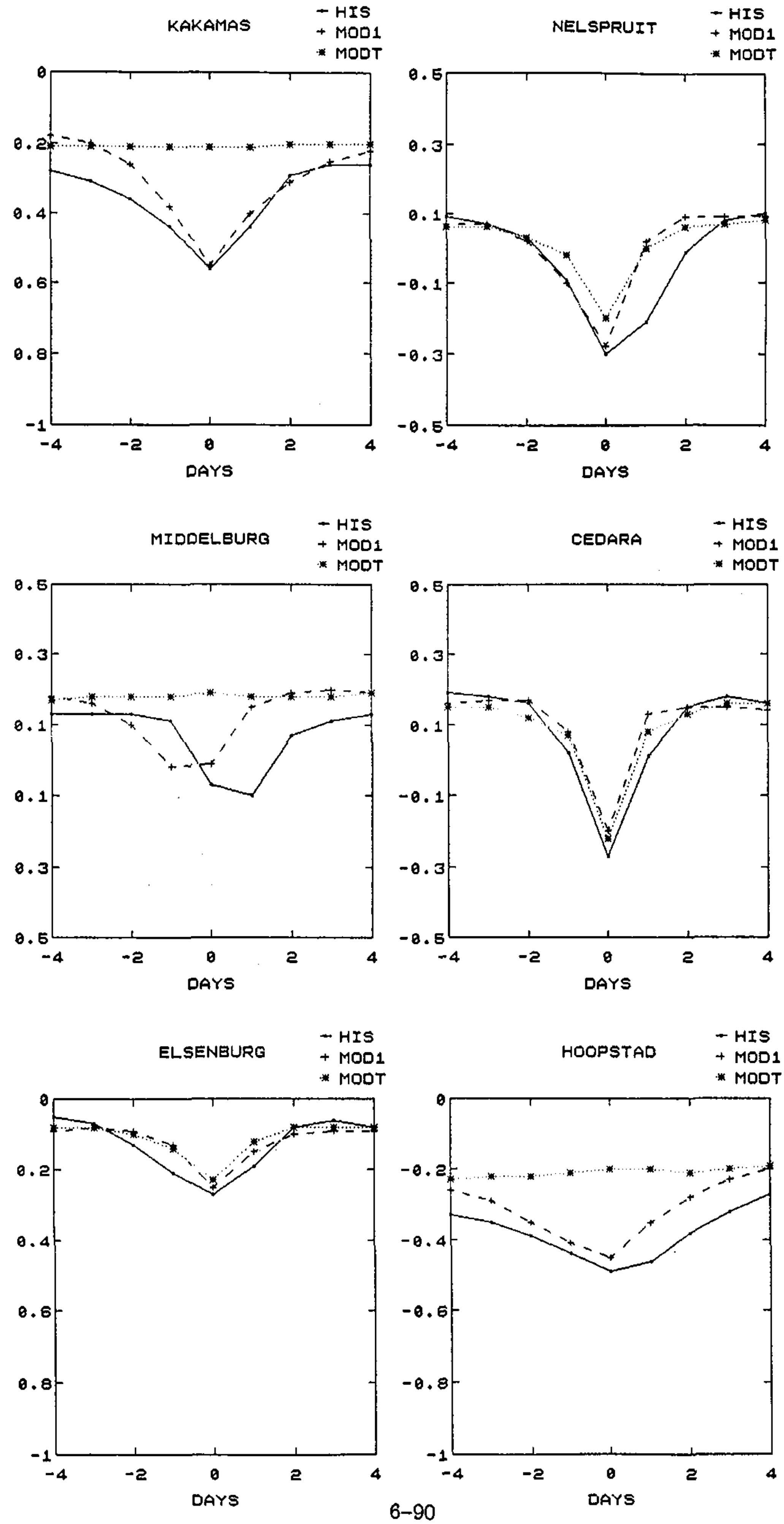


FIGURE 6.76 Cross-correlation coefficients for maximum temperature and minimum humidity

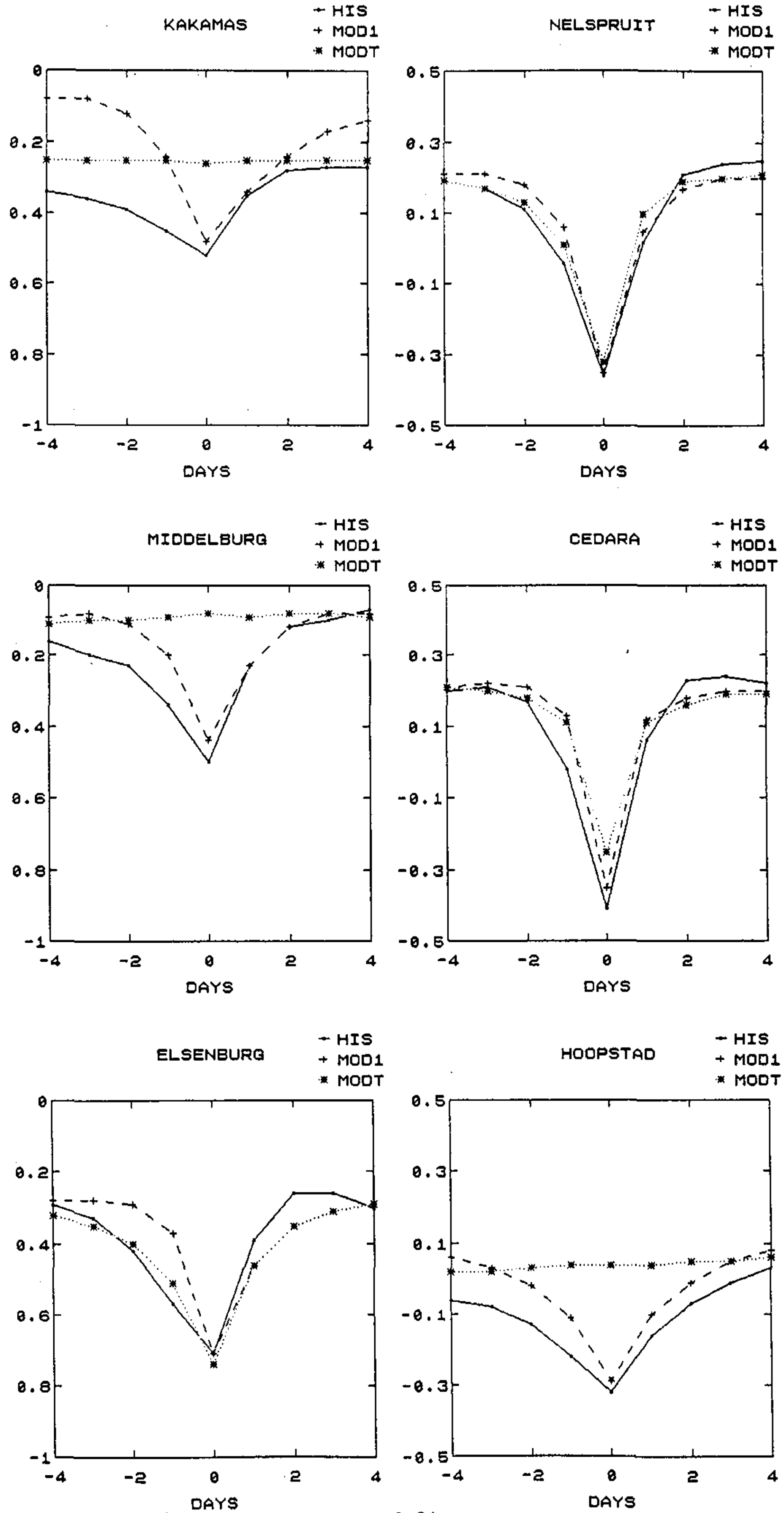


FIGURE 6.77 Cross-correlation coefficients for minimum temperature and evaporation

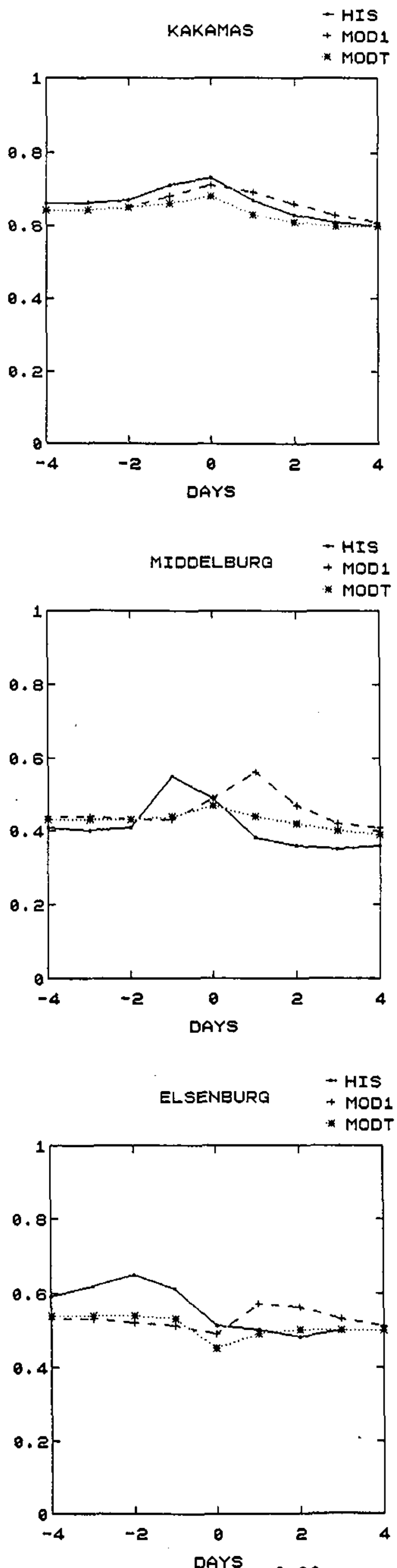


FIGURE 6.78 Cross–correlation coefficients for minimum temperature and sunshine duration

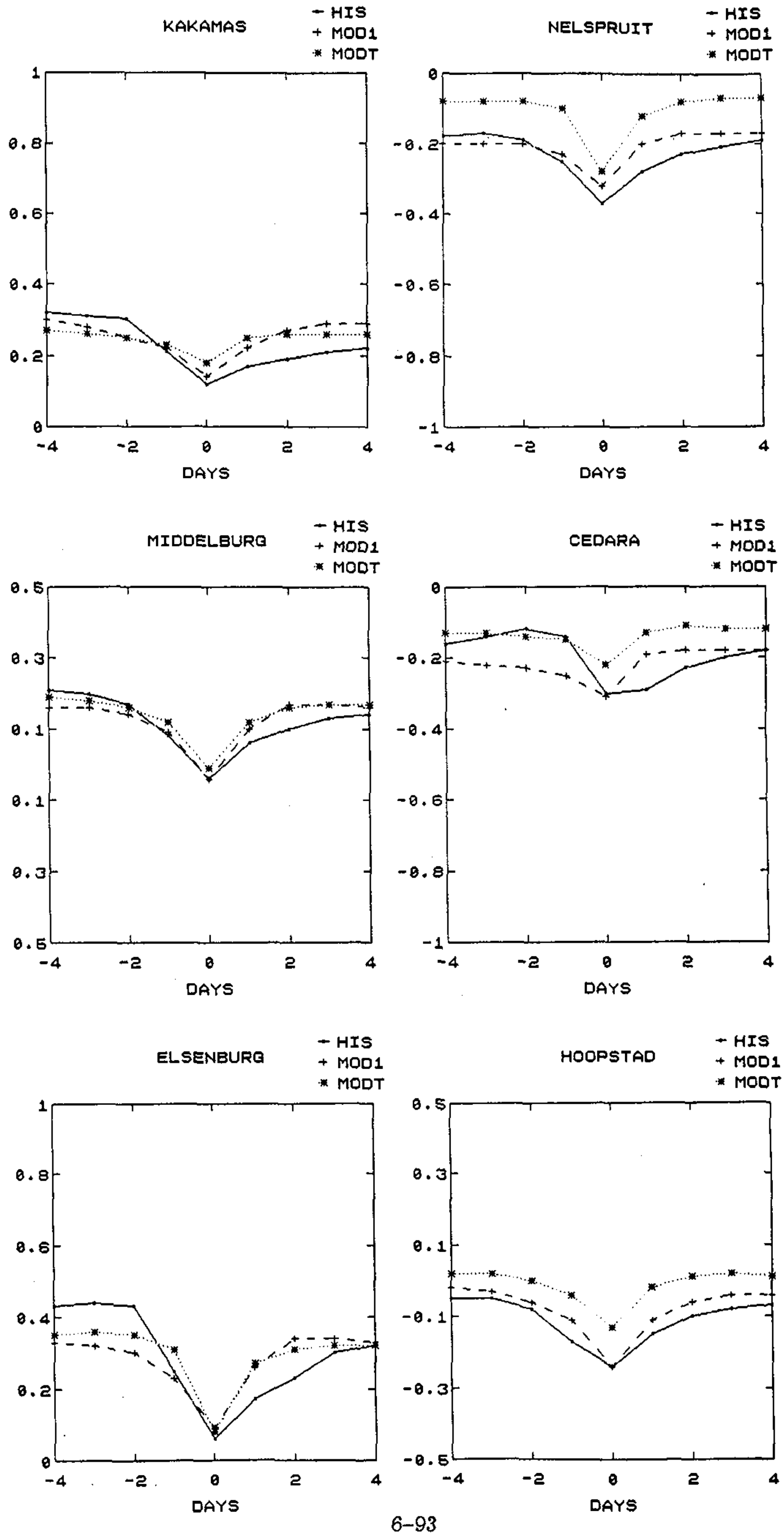


FIGURE 6.79 Cross–correlation coefficients for minimum temperature and wind run

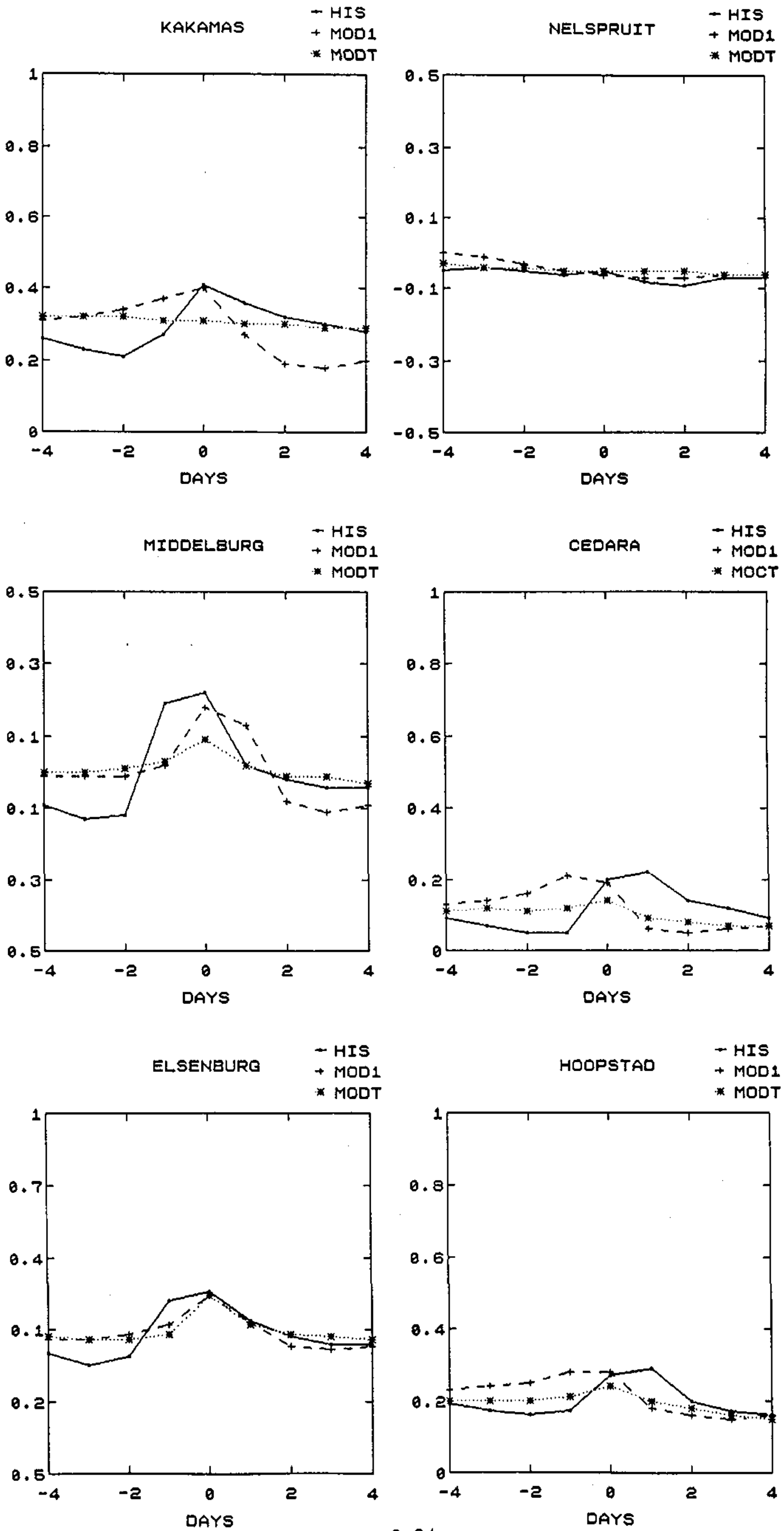


FIGURE 6.80 Cross-correlation coefficients for minimum temperature and maximum humidity

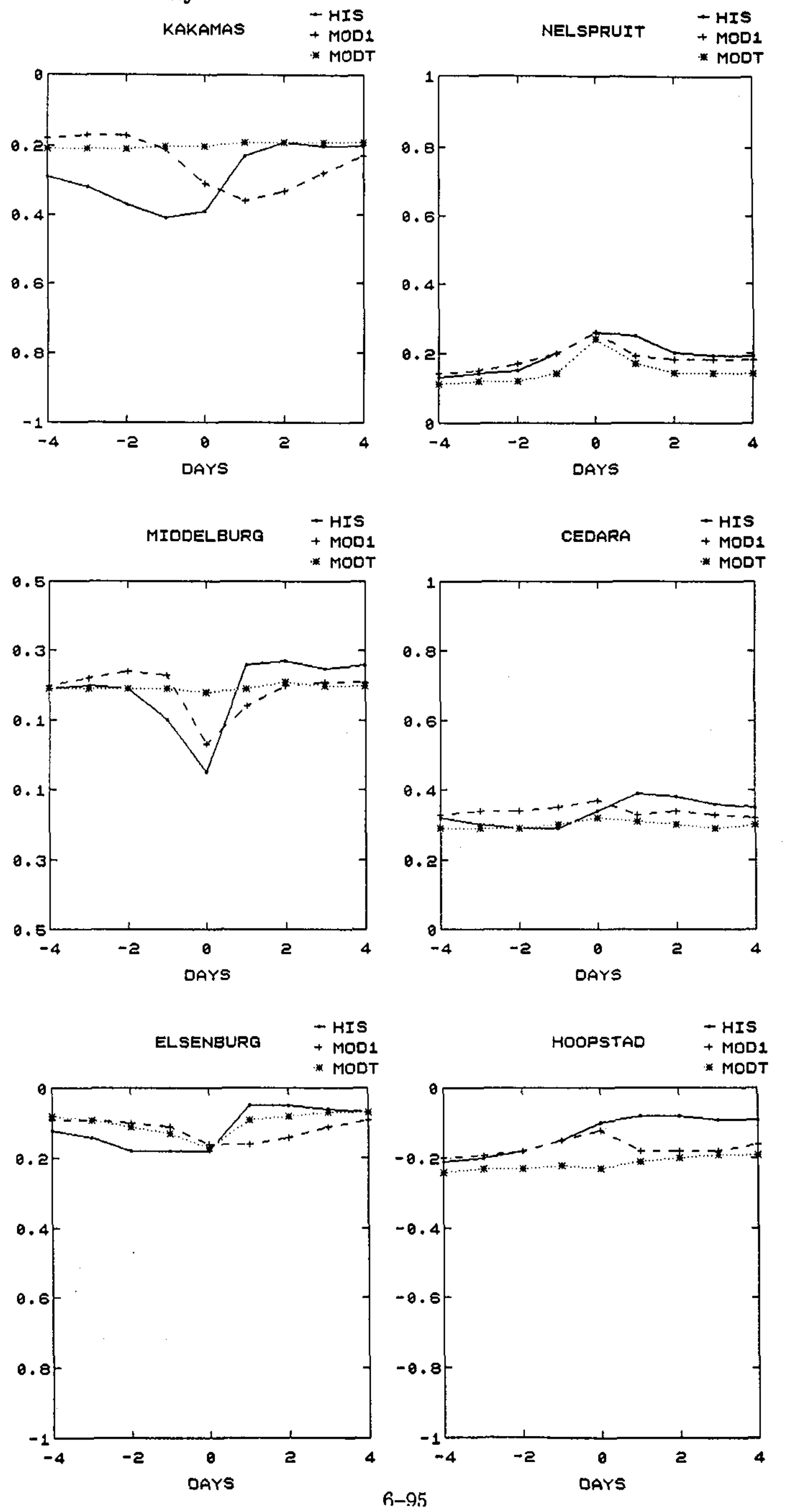


FIGURE 6.81 Cross-correlation coefficients for minimum temperature and minimum humidity

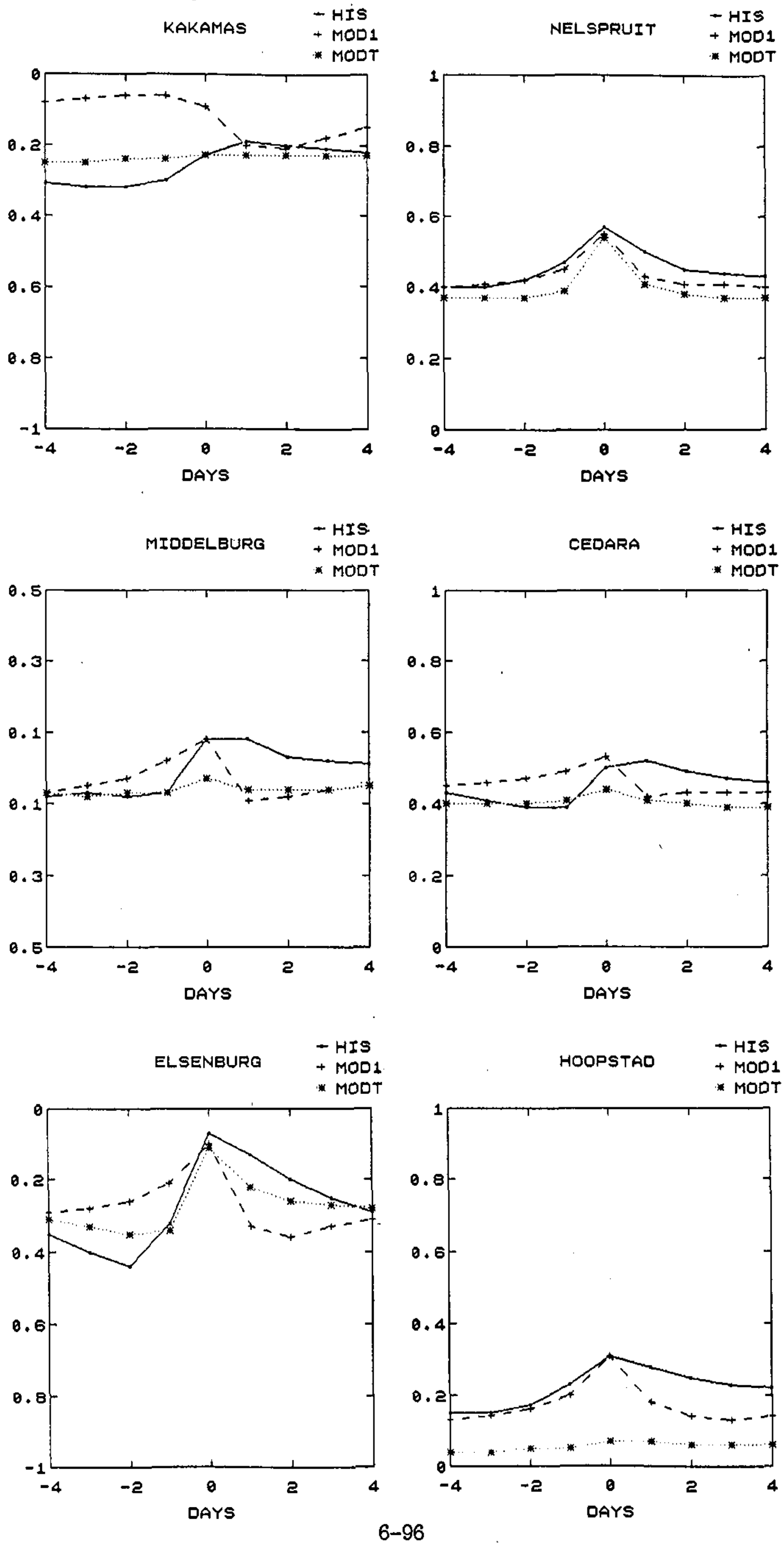


FIGURE 6.82 Cross-correlation coefficients for evaporation and sunshine duration

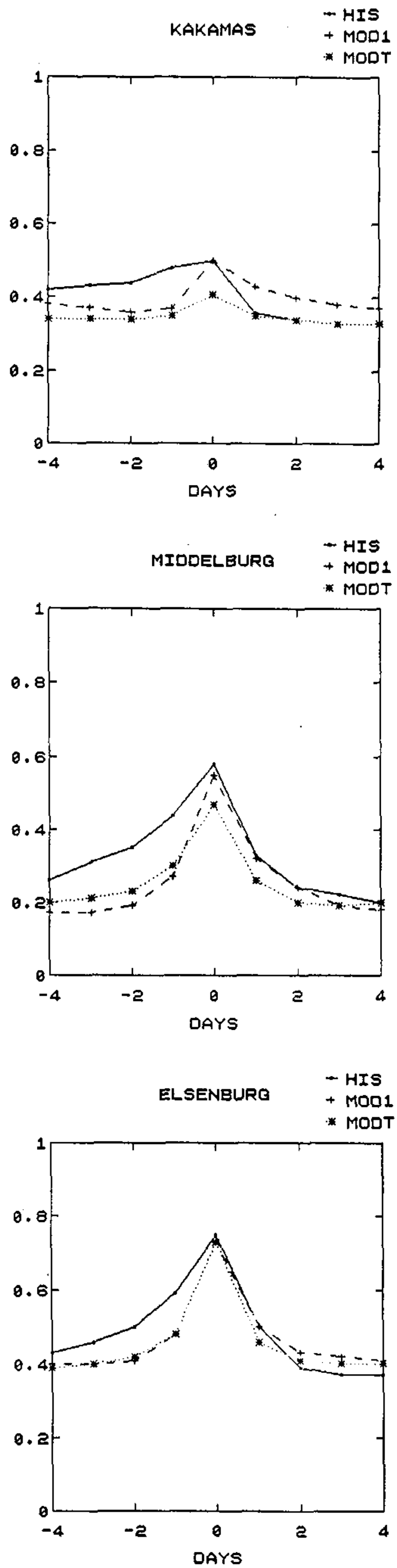


FIGURE 6.83 Cross-correlation coefficients for evaporation and wind run

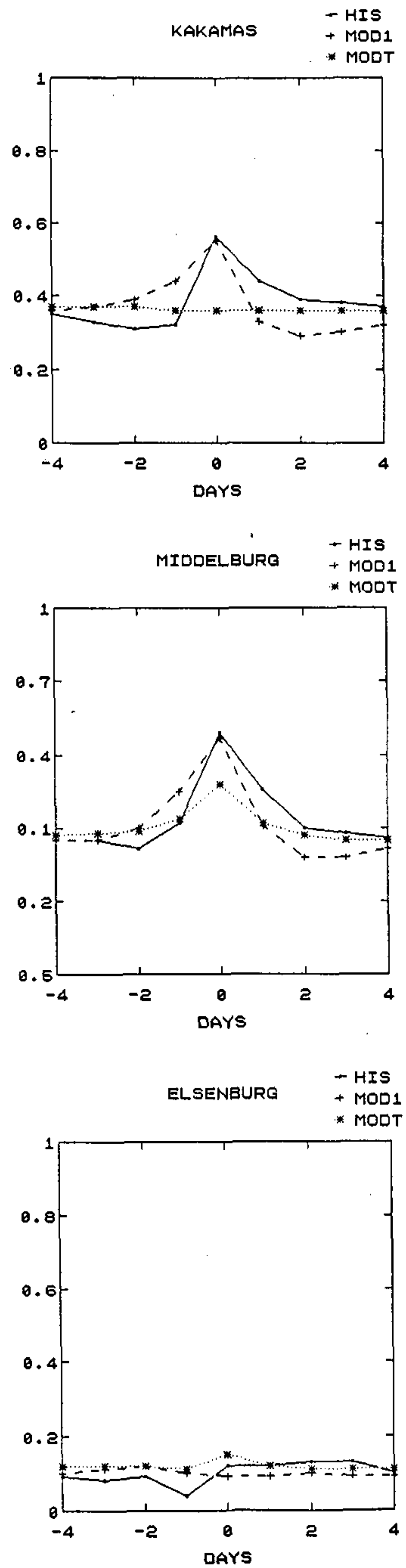
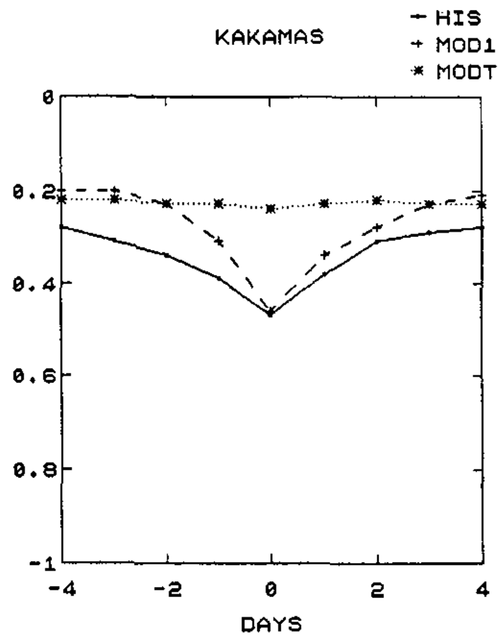
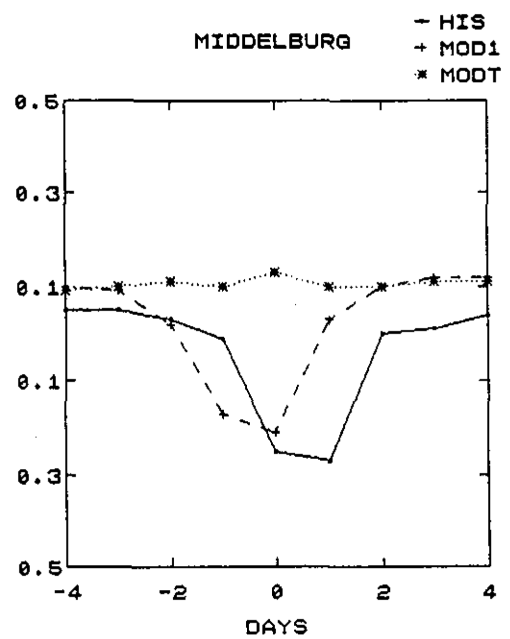


FIGURE 6.84 Cross-correlation coefficients for evaporation and maximum humidity





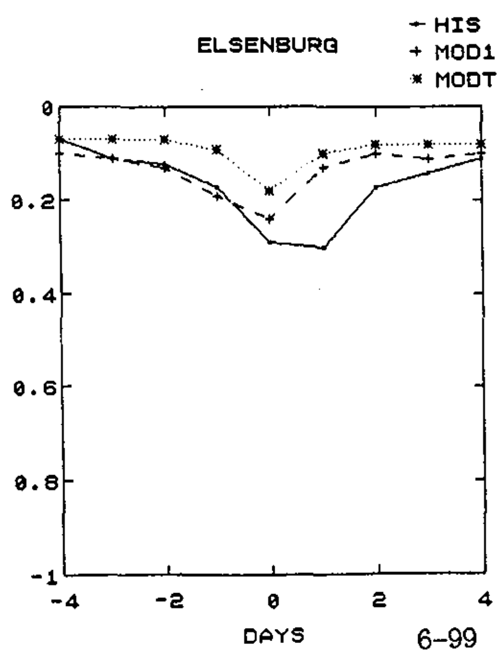


FIGURE 6.85 Cross–correlation coefficients for evaporation and minimum humidity

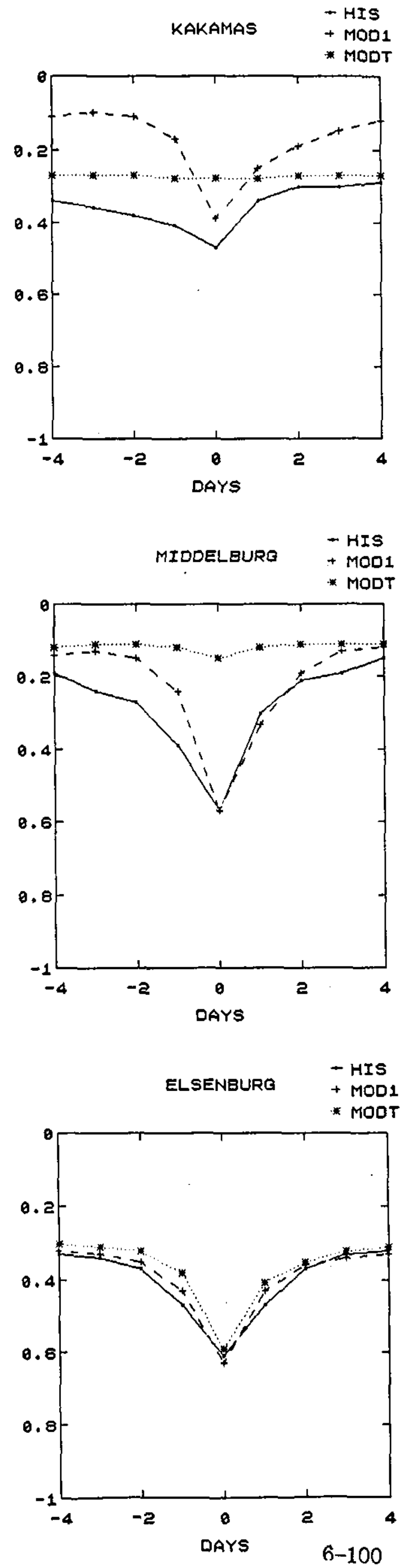


FIGURE 6.86 Cross-correlation coefficients for sunshine duration and wind run

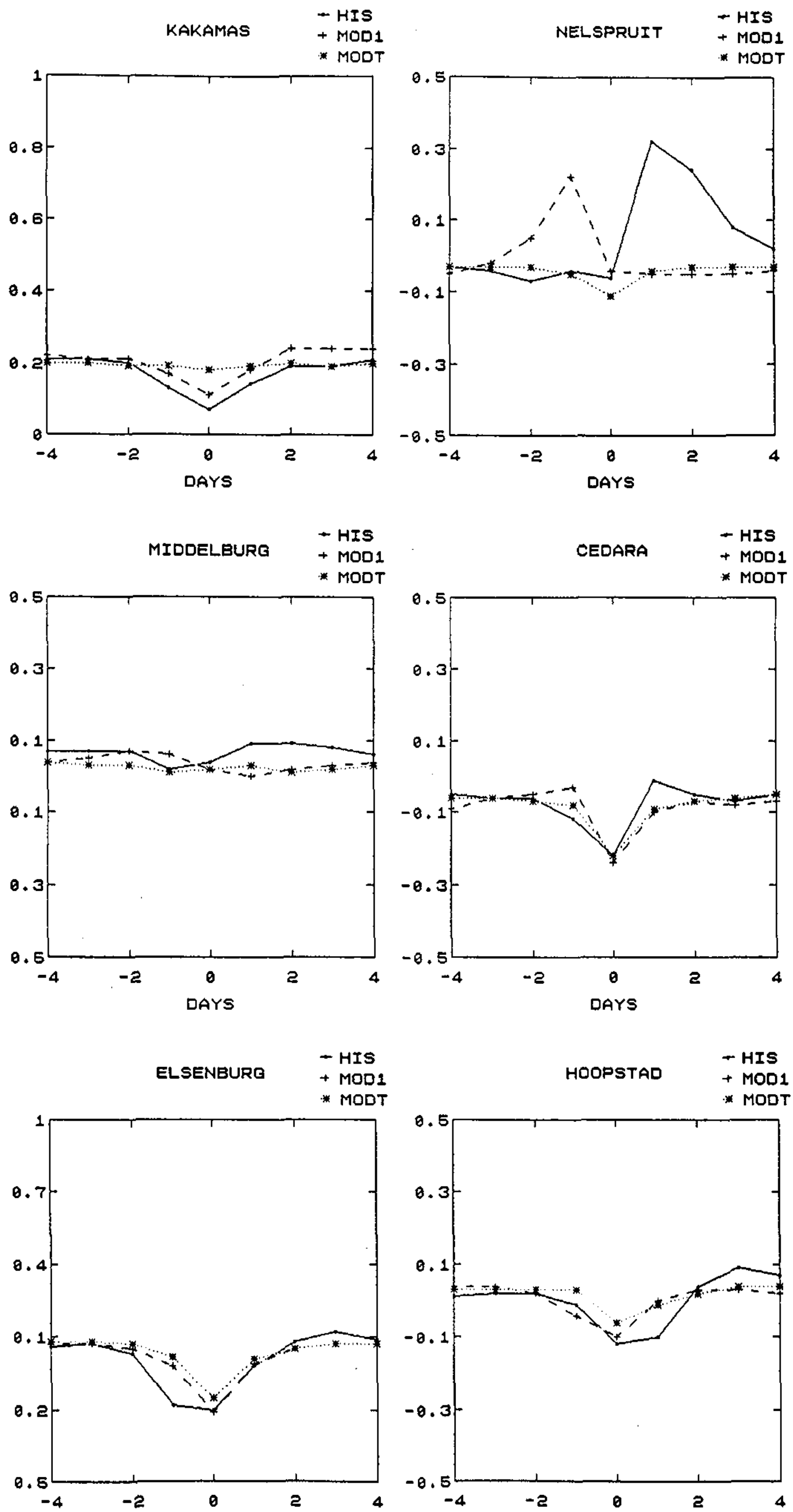


FIGURE 6.87 Cross–correlation coefficients for sunshine duration and maximum humidity

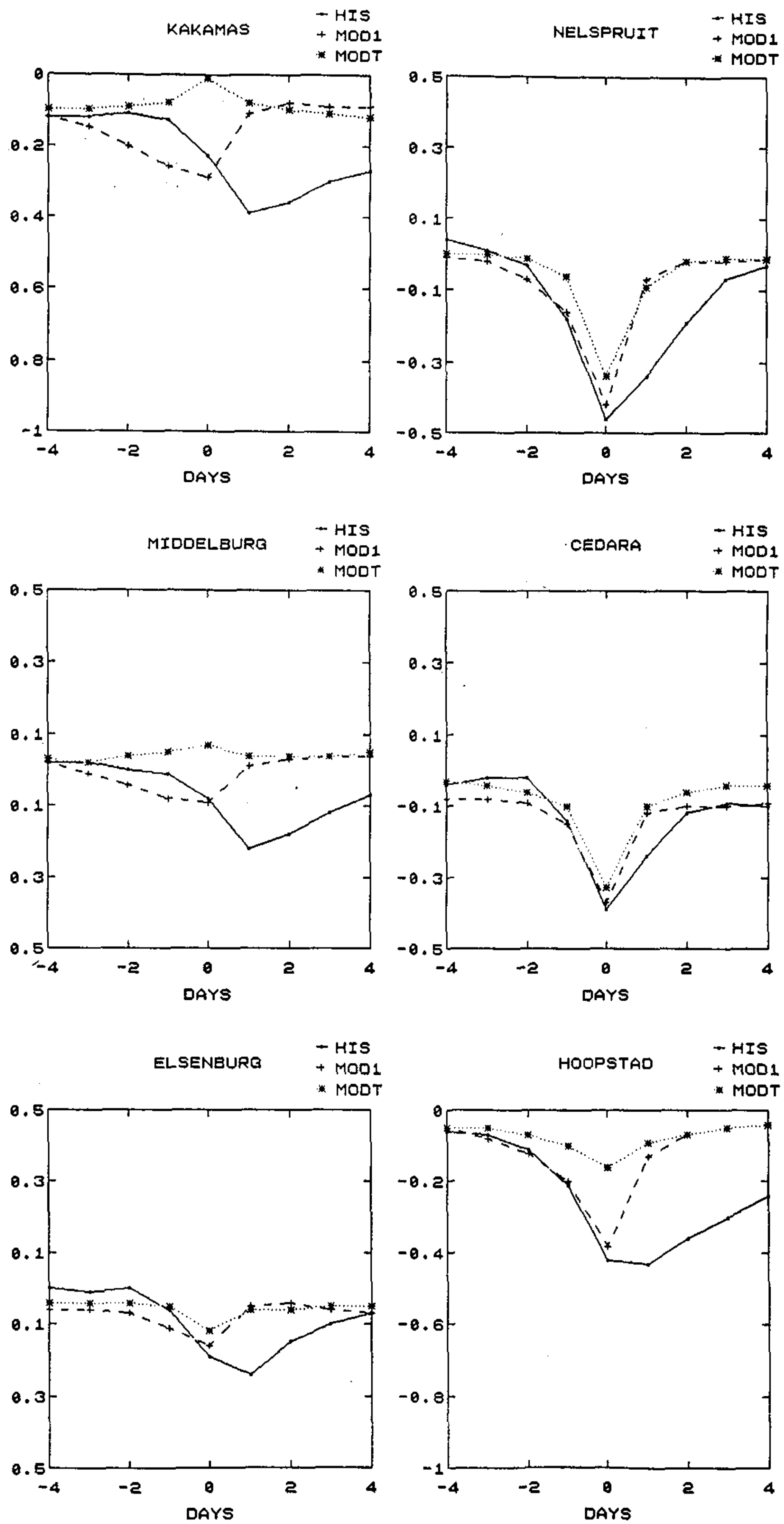


FIGURE 6.88 Cross–correlation coefficients for sunshine duration and minimum humidity

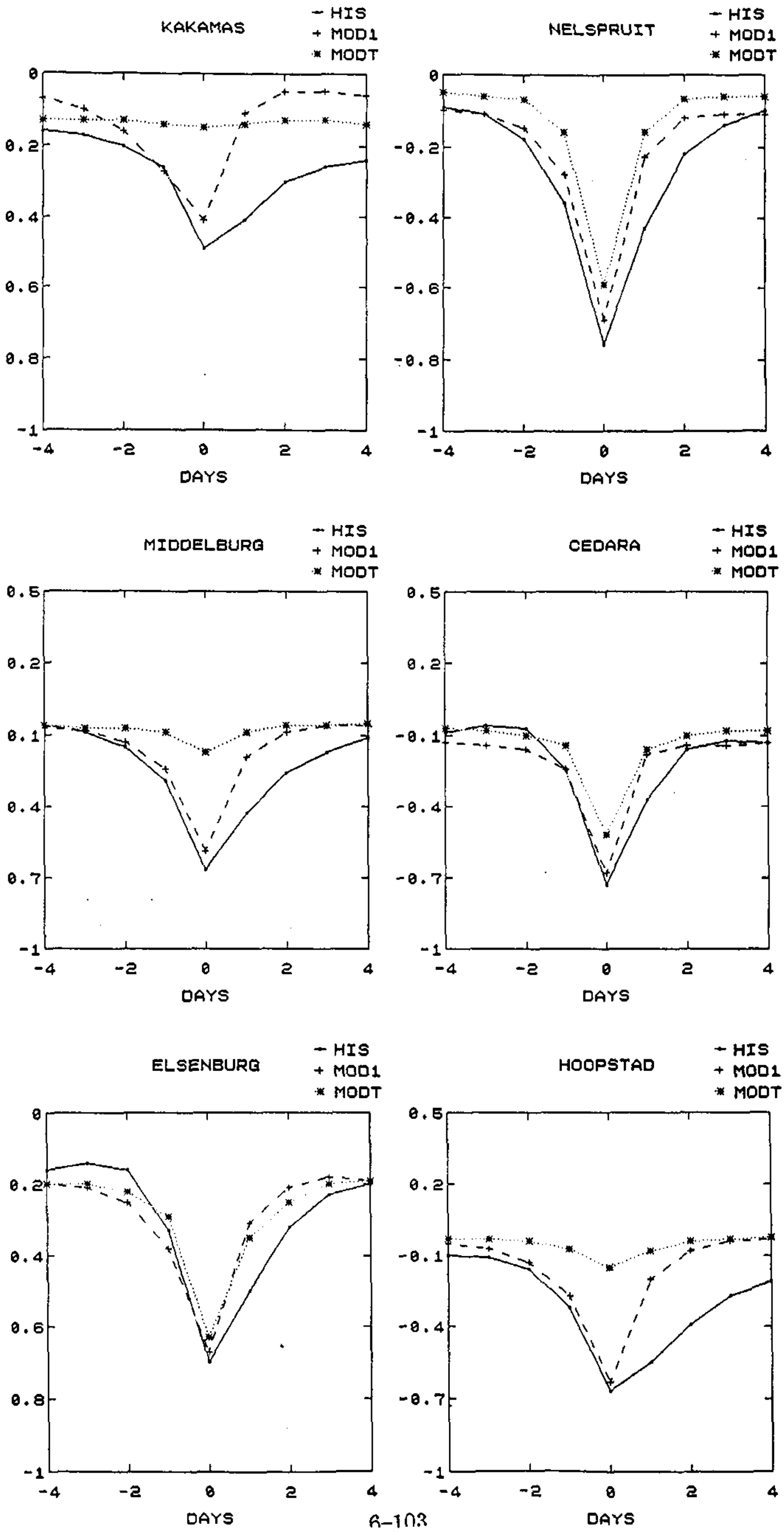


FIGURE 6.89 Cross-correlation coefficients for wind run and maximum humidity

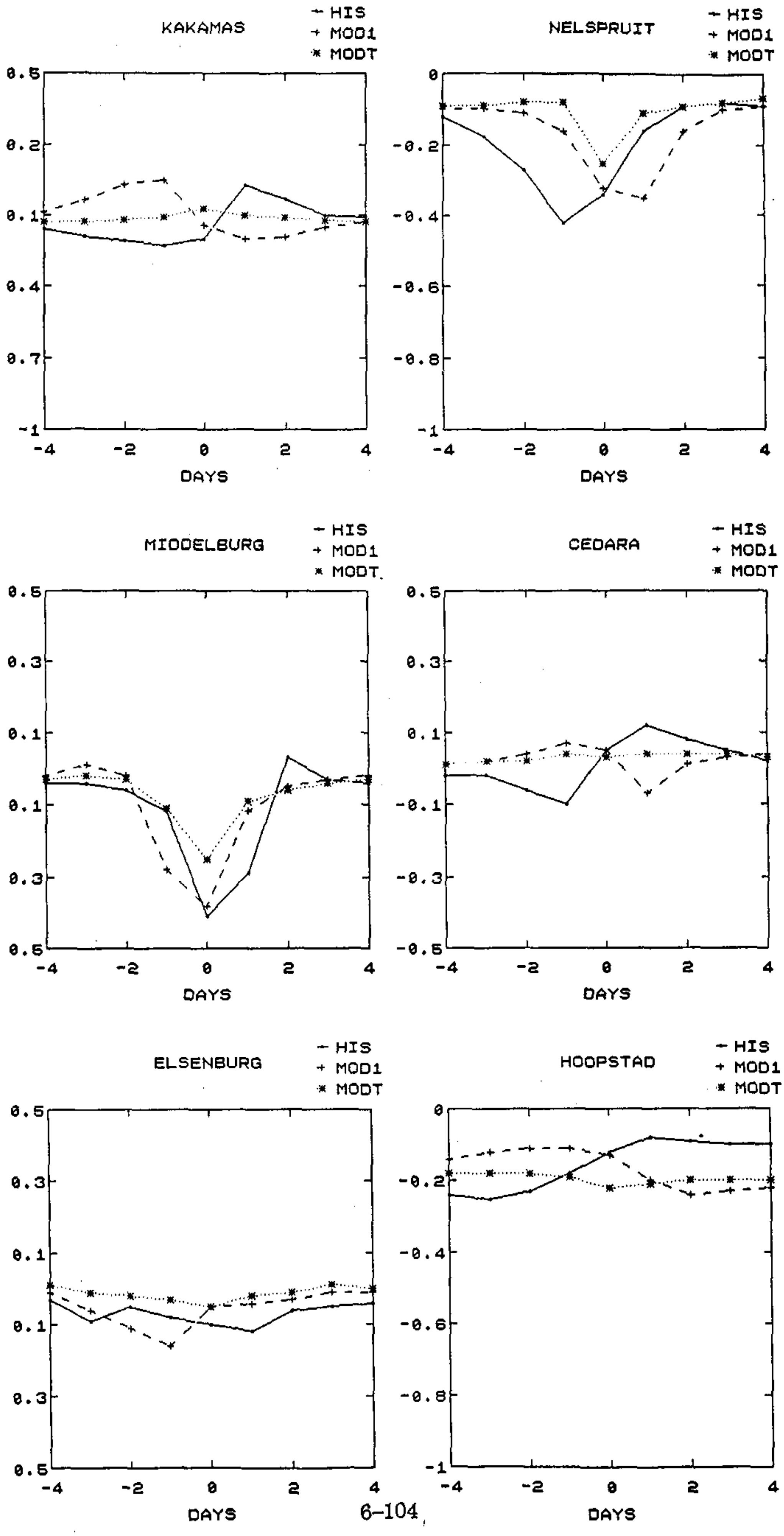


FIGURE 6.90 Cross-correlation coefficients for wind run and minimum humidity

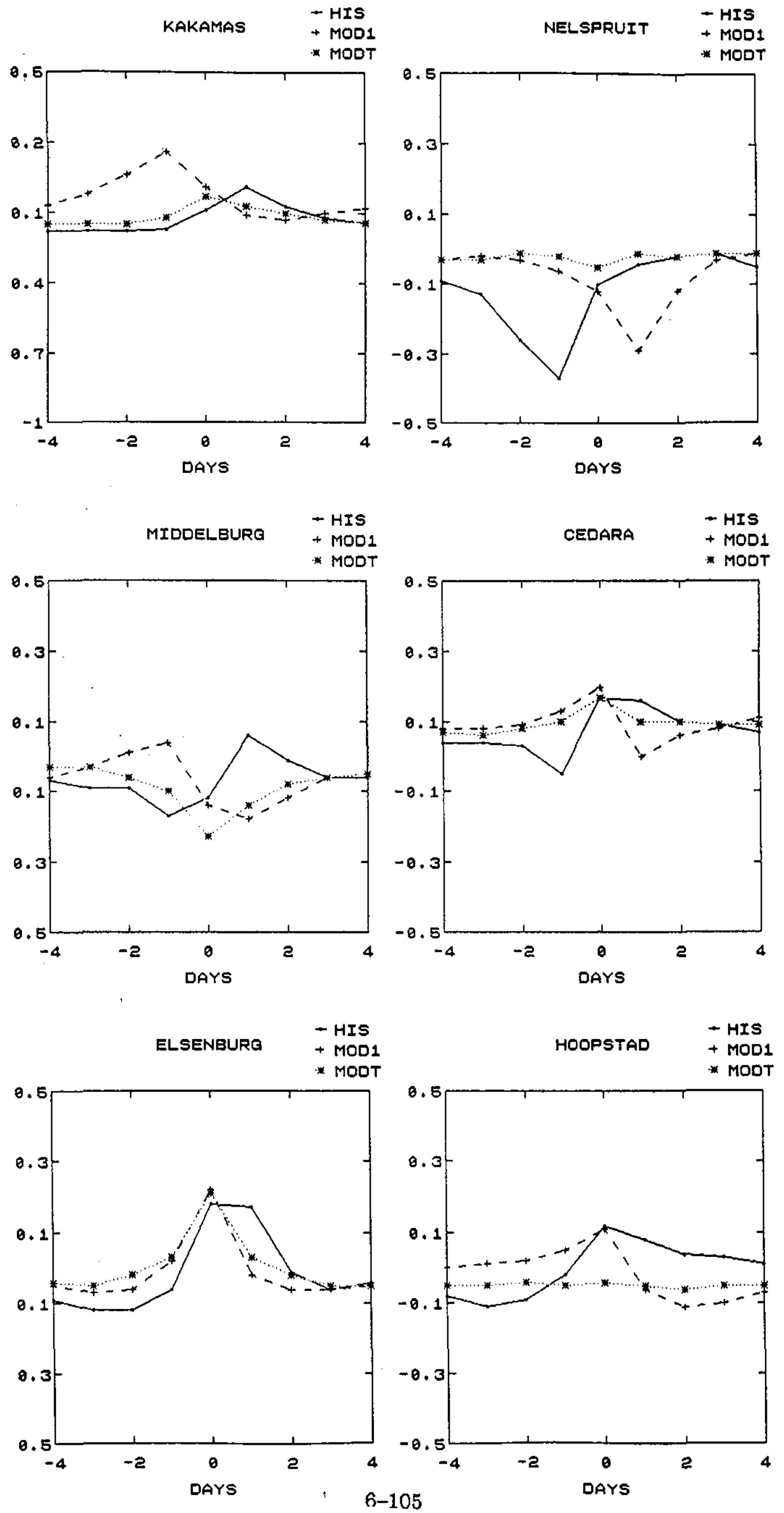
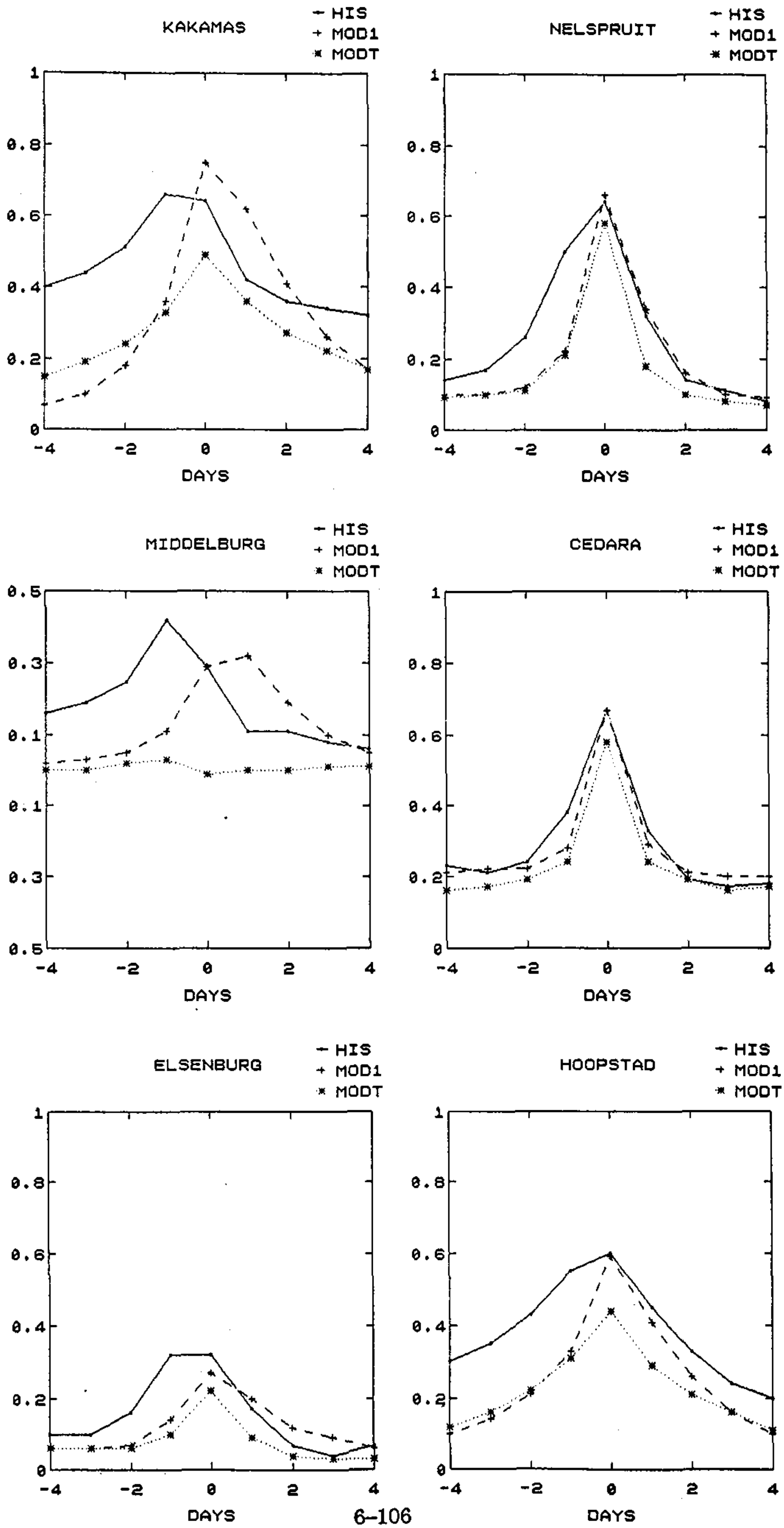


FIGURE 6.91 Cross–correlation coefficients for maximum humidity and minimum humidity



sometimes also differ from the historical cross-correlation coefficients for lags 1 and -1, but generally this difference is quite small.

Summary

If one reflects on the complexity of the climate process and in particular on the large number of properties which the models are required to preserve, it can be reasonably concluded that the models perform remarkably well. All but a few of the relevant parameter functions and cross-correlations are preserved faithfully by the models. One other factor that one must keep in mind when evaluating the performance of the models is the quality and quantity of the historical record. For the model parameters to truly reflect the properties of climate variables there must be enough data records to have a representative sample of the climate variables. It is difficult to be specific about how long a record needs to be in order to be representative. Very roughly, and based on our experience, we would recommend the minimum of 20 years of record before one can feel confidence in the results.

There are a number of weaknesses displayed by the models. The most important are:

(i) Model T does not retain the property of the monthly means in the variable minimum humidity for Hoopstad. It also does not preserve the monthly means on wet days for the variables maximum humidity and minimum humidity for most stations, in particular those that have relatively few rainfall observations.

(ii) Model 1 does not retain the property of monthly standard deviations in the variable minimum humidity for Kakamas. It also does not preserve the monthly standard deviations on wet days for the variables maximum humidity and minimum humidity for some stations.

(iii) Model T shows a weakness in maintaining the cross-correlation coefficients of maximum humidity and minimum humidity and the other variables in particular for the stations Kakamas, Middelburg and Hoopstad.

A choice of model at this point is not straightforward as the performance of the models is neither perfect nor totally without merit, but each model shows strengths and weaknesses. A criterion to base our preference on any particular model can be based on factors such as:

(a) Implementation costs, that is derivation of theory for parameter estimation, complexity of model in terms of number of parameters needed and the computational simplicity aspect of the model.

(b) Preservation of climate variable properties by the model.

Model T is the more complex of the models in terms of computational difficulty. The model parameters are estimated iteratively and therefore more time consuming. It is also the most flexible of the models in that each variable is allowed to be modelled by a model that "best" describes it. In Model 1 all climate variables have the same structure.

Generally Model 1 appears to perform as well and sometimes better than Model T in describing some aspects of the climate variables. However, Model T cannot be simply dismissed as it does perform better than Model 1 in some aspects and one must bear in mind that in Model T the climate sequences are separated into four, while Model 1 only separates them into two. Thus fewer observations are available for the estimation of some of the parameters of Model T. An increase in the length of the historical record may therefore result in an improved performance by Model T.

CHAPTER 7

SUMMARY AND RECOMMENDATIONS

This chapter gives a brief summary of the study performed followed by the main research findings and finally by recommendations.

Summary

Five stochastic models to describe daily climate sequences of South Africa were considered. The climate variables included in these models are rainfall, maximum and minimum temperature, maximum and minimum humidity, evaporation (when records available), sunshine duration and wind run. Except for rainfall, which is an essential component of all the models, this list of variables may be either reduced or augmented. Thus the modelling procedure which has been developed is not restricted to this particular set of climate variables.

The models are required to preserve important properties exhibited by the daily climate sequences. These properties are seasonality, wet/dry day effect, autocorrelation, cross–correlation and boundedness. Suitable transformations need to be applied to the climate variables at the start of the modelling procedure to take care of the property of boundedness.

The technique employed was firstly to model rainfall using a first–order Markov chain with seasonal parameters to describe the occurrence of wet and dry days, while the Weibul distribution was used to describe the rainfall depth of wet days. The rainfall mean was allowed to vary seasonally. This model provides synthetic sequences of wet and dry days. Finally, the remaining climate variables were modelled according to the wet or dry status of each day.

The first model considered was proposed by Richardson (1981), where a stationary residual series is obtained by subtracting the seasonal mean and dividing by the standard deviation of each climate variable, each of the functions conditioned on the wet and dry status of the day. A weakly stationary process suggested by Matalas (1967) is used to model the residual series. It is assumed that the residual series is normally distributed and that the serial correlation of each variable can be described by a first–order autoregressive process.

Three models, referred to as Model 3, 4 and 5 (Model 2 was developed as a prototype to the others) were developed to incorporate additional flexibility in the autocorrelation structure of Model 1. That is, the autocorrelation structure is allowed to depend on the wet and dry status of the day as well as that of the previous day.

The three new models which were developed form a compatible family of models of varying degrees of complexity. This feature leads to a number of advantages. It allows one to select relatively simple models for sites where historical records are short (as is presently the case at practically all sites in South Africa) and to change the selection to a more complex model when the records become sufficiently long. In addition it is possible to assemble the final multivariate model for a site from components from any of the three model types. Thus, for example, it is possible to apply Model 3 to wind run, Model 4 to minimum and maximum temperature and Model 5 to the remaining variables.

The problem of deciding which model or model combination is most appropriate for a particular site can be determined objectively. The Akaike Information Criterion was found to be suitable for this purpose and has been incorporated in the software package which was developed for this project.

Apart from the mathematical development of the new models, one of the most difficult obstacles that had to be overcome in the course of the project was that of controlling the range of extreme values generated by the models. In part this problem arises because some of the climate variables must fall within fixed boundaries and, in addition, some of these variables (for example maximum humidity) exhibit a high frequency of occurrence on or near their boundaries. Suitable transformations had to be found to ensure that the generated value would remain within their appropriate bounds. The results of our validation tests indicate that this type of difficulty can be successfully overcome.

A second problem that had to be solved related to the presence of gaps in the historical records. As well as the gaps that were present in the original record one has to add the gaps which are created by filtering out observations that are clearly incorrect (for example, that fall outside their permissible range). The number of missing values in the historical records used in this report ranged between 1% and 13% of the data. The serial correlation and cross–correlation structure of climate variables does not allow one to simply ignore missing values. Special methods had to be developed to deal with this problem. We found that a procedure based on the EM algorithm can be used to satisfactorily estimate the missing values thereby filling the gaps in the historical records.

The results of model validation for the rainfall model confirm the findings of Zucchini and Adamson (1984), namely that the assumptions regarding the characteristics of daily rainfall sequences, the rationale of model structure and the parameter estimation techniques are particularly successful in providing a model that can adequately reproduce the properties of daily rainfall sequences.

The requirement for an accurate simulation of the occurrence of wet and dry days is very important in the present study as these simulated sequences are used to determine the generation procedure to be adopted by the other climate variables. This component of the rainfall model was found to be very successful in preserving the characteristics of the occurrence of wet and dry days.

As already mentioned, the multivariate models for climate data were required to meet certain specifications. Namely, they had to preserve properties such as seasonality, wet/dry day effect, autocorrelation, cross–correlation as well as annual, monthly and daily properties, in particular the mean and the standard deviation functions. Tests of the multivariate models for climate data showed that the models were capable of representing almost all the characteristics exhibited by the historical data.

Whenever the models showed differences between the simulated and historical sequences, it was noted that it usually was for the variables wind run, maximum humidity and minimum humidity and mainly for the case where these sequences were conditioned on wet days. Further investigation revealed that the stations where these differences occurred were those for which relatively few rainfall records are observed.

When evaluating the performance of the climate models one must bear in mind the fact that the length of the historical records determines in a way the performance of the models. A relatively short historical record leads to three problems. Firstly, one is estimating a large number of parameters with very few data values thus decreasing the precision of the estimates. Secondly, because the models separate the sequences into wet and dry sequences, the effective record length for the conditioned estimates is further reduced, in particular for the wet sequence as rainfall events in some parts of South Africa are relatively rare. Thus long records of climate observations are needed to compensate for the lack of rainfall events. Thirdly, the fact that the record lengths of the stations in this study are quite small, combined with the fact that there are missing observations in the records means that the historical data might not wholly be representative of the long term climate for that particular location.

The climate variables investigated in this study have a very complex joint distribution. Each variable exhibits a number of distinctive features and in addition the variables are interdependent. Any model which is to usefully describe climate sequences must preserve these properties. This study has shown that it is feasible to model climate on a daily basis and that there are at least four models which can be used to do so. Either Model 1 or a combination of Models 3, 4 and 5 can be used. A choice between Model 1 and Model T is not clear. Both models show some weaknesses and some merits. Model 1 does appear to perform better than Model T for those stations that have few rainfall observations, such as Kakamas. It is also less time consuming in parameter estimation as it does not estimate them iteratively. However, Model T (that is, a combination of Models 3, 4 and 5) does outperform Model 1 in some instances and we would expect that this will become increasingly the case as the historical records become longer.

The software which was developed in this project covers both the parameter estimation and the generating of artificial sequences for all the models that have been described in this report. The programs were coded in FORTRAN and make no use of licensed software. In addition they can be implemented on micro computers thereby making the methodology easily accessible to a wide range of users.

Recommendations

Quality of historical records

The main obstacle to the application of the techniques described in this report on a large scale is the lack of suitable historical records. This refers to both the quantity and the quality of available data. The records which were used for this report represent some of the best available in South Africa. Nevertheless, for the purpose of modelling daily climate, they are barely adequate. Although there is little that can be done to increase the length of records except to wait for more data to be collected, it should be possible to improve the quality of historical records. In particular it would be useful if some measure of the reliability of the observations were also recorded on a regular basis. As we have repeatedly pointed out in the body of the report, one of the problems which we encountered was that of identifying incorrect observations. This task would be considerably simplified ifone had some index of reliability associated with (ideally) each recording or set of recordings.

Transfer of technology

For the methods developed in this project to realise their full potential it will be necessary to calibrate the models at many more sites. As was pointed out in the report, no special training is required to use the programs for generating climate sequences once the parameters of the model have been estimated. However, some training is required to use the program to prepare the data for estimation and to carry out the estimation for a new site. We estimate that, with instruction, it would require two to three weeks for a competent programmer to learn how to use the methodology.

We recommend that the Computing Centre for Water Research (CCWR) be approached to acquire the expertise to implement the estimation techniques and with the help of users, gradually build up a data base of estimates of the model parameters for as many sites as possible in South Africa. The CCWR already offer a similar data product, namely the parameter estimates of a daily rainfall model for 2550 sites in South Africa. These arose from a previous Water Research Commission project (Zucchini and Adamson (1984)). The CCWR also offer the artificial rainfall generating program which can be applied to any of these sites. Thus the programs developed in the course of this project constitute a logical extension of a service that the CCWR already offer.

REFERENCES

ACTON, F.S. (1970). *Numerical methods that work.* Harper and Row, New York.

BARRY, R.G. and CHARLEY, R.J. (1968). *Atmosphere, weather and climate.* Methuen and Co. Ltd., London.

BOX, G.E.P. and JENKINS, G.M. (1970). *Time series analysis: forecasting and control.* Holden Day, New York.

BRUHN, J.A., FRY, W.E. and FICK, G.W. (1980). Simulation of daily weather data using theoretical probability distributions. *Journal of Applied Meteorology,* **19**, 1029–1036.

BUISHAND, T.A. (1977). *Stochastic modelling of daily rainfall sequences.* Meded. Landbouwhagesch, Wageningen, 77–83.

BUISHAND, T.A. (1978). Some remarks on the use of daily rainfall models. *Journal of Hydrology,* **36**, 295–308.

CASKEY, J.E. (1963). A Markov chain model for the probability of precipitation occurrence in intervals of various length. *Monthly Weather Review,* **91**, 298–301.

CHATFIELD, C. (1980). *The analysis of time series: An introduction.* (2nd ed.) Chapman and Hall, New York.

COCHRANE, D. and ORCUTT, G.H. (1949). Application of least squares regression to relationships containing autocorrelated error terms. *Journal of the American Statistical Association,* **44**, 32–61.

DEMPSTER, A.P., LAIRD, N.M. and RUBIN, D.B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society,* **B39**, 1–38.

DOBSON, A.J. (1983). *An introduction to statistical modelling.* Chapman and Hall, New York.

GABRIEL, K.R. and NEUMANN, J. (1962). A Markov chain model for daily rainfall occurrence at Tel Aviv. *Quarterly Journal of the Royal Meteorological Society,* **88**, 90–95.

GOH, T.N. and TAN, K.J. (1977). Stochastic modelling and forecasting of solar radiation data. *Solar Energy,* **19**, 755–757.

HAAN, C.T., ALLEN, D.M. and STREET, J.O. (1976). A Markov chain model of daily rainfall. *Water Resources Research*, **12**, 443–449.

HARTLEY, H.O. (1958). Maximum likelihood estimation from incomplete data. *Biometrics*, **14**, 174–194.

HOPKINS, J.W. and ROBILLARD, P. (1964). Some statistics of daily rainfall occurrences for the Canadian prairie provinces. *Journal of Applied Meteorology*, **3**, 600–602.

ISON, N.T., FEYERHERM, A.M. and BARK, L.D. (1971). Wet period precipitation and the gamma distribution. *Journal of Applied Meteorology*, **10**, 658–665.

LARSEN, G.A. and PENSE, R.B. (1982). Stochastic simulation of daily climatic data for agronomic models. *Agronomy Journal*, **74**, 510–514.

LINHART, H. and ZUCCHINI, W. (1982). A method for selecting the covariates in analysis of covariates. *South African Statistical Journal*, **16**, 97–112.

LITTLE, R.J.A. and RUBIN, D.B. (1987). *Statistical analysis with missing data.* John Wiley, New York.

MAKHUVHA, T. (1988). *The estimates of missing values in hydrological records using the EM algorithm and regression methods.* Unpublished Masters thesis, Department of Mathematical Statistics, University of Cape Town, Cape Town.

MATALAS, N.C. (1967). Mathematical assessment of synthetic hydrology. *Water Resources Research*, **3**, 937–945.

MIELKE, P.W. (1973). Another family of distributions for describing and analyzing precipitation data. *Journal of Applied Meteorology*, **10**, 275–280.

NELDER, J.A. and WEDDERBURN, R.W. M. (1972). Generalized linear models. *Journal of the Royal Statistical Society*, **A135**, 370–384.

NICKS, A.D. and HARP, J.F. (1980). Stochastic generation of temperature and solar radiation data. *Journal of Hydrology*, **48**, 1–17.

RICHARDSON, C.W. (1981). Stochastic simulation of daily precipitation, temperature and solar radiation. *Water Resources Research*, **17**, 182–190.

ROLDAN, J. and WOOLHISER, D.A. (1982). Stochastic daily precipitation models: (1) A comparison of occurrence processes. *Water Resources Research*, **18**, 1451–1459.

SMITH, R.E. and SCHREIBER, H.A. (1973). Point processes of seasonal thunderstorm

APPENDIX A

The choice of the Fourier approximation, L

The order of approximation of the Fourier representation of a function, $\lambda(t)$, is always taken to be an odd integer. This restriction is made partly for programming convenience and partly for the following reason:

If we rewrite the Fourier representation of $\lambda(t,L)$ by its polar form, we get

$$\lambda(t,L)=\begin{cases}\alpha_0+\sum_{i=1}^{p}\alpha_i\cos\left(\frac{2\pi i}{NT}((t-1)-\beta_i)\right), & L \text{ odd}\\ \alpha_0+\sum_{i=1}^{p}\alpha_i\cos\left(\frac{2\pi i}{NT}((t-1)-\beta_i)\right)+\alpha_p\cos\frac{2\pi p(t-1)}{NT}, & L \text{ even}\end{cases}$$

where

$$\alpha_0=\gamma_1$$
$$\alpha_i=\left(\gamma_{2i}^2+\gamma_{2i+1}^2\right)^{\frac{1}{2}}, \qquad i=1,2,\ldots,p$$
$$\beta_i=\frac{NT}{2\pi i}\arctan\left(\frac{\gamma_{2i+1}}{\gamma_{2i}}\right), \qquad i=1,2,\ldots,p$$

and p is the integer part of $\frac{L-1}{2}$. The α_i is called the amplitude and β_i is called the phase of the ith harmonic.

If L is even, then the highest harmonic does not have a phase parameter. Thus the quality of the fit of the model depends on the time origin selected. If L is odd we obtain the same degree of approximation for all time origins.

rainfall, I, Distributions of rainfall events. *Water Resources Research*, **9**, 871–884.

STERN, R.D. and COE, R. (1983). A model fitting analysis of daily rainfall data. *Journal of the Royal Statistical Society*, **A147**, 1–34.

TODOROVIC, P. and WOOLHISER, D.A. (1975). A stochastic model of n-day precipitation. *Journal of Applied Meteorology*, **14**, 17–24.

WEISS, L.L. (1964). Sequences of wet or dry days described by a Markov chain probability model. *Monthly Water Review*, **92**, 169–176.

WOOLHISER, D.A. and PEGRAM, G.G.S. (1979). Maximum likelihood estimation of Fourier coefficients to describe seasonal variations of parameters in stochastic daily precipitation models. *Journal of Applied Meteorology*, **18**, 34–42.

WOOLHISER, D.A. and ROLDAN, J. (1982). Stochastic daily precipitation models: (2) A comparison of distributions of amounts. *Water Resources Research*, **18**, 1461–1468.

ZUCCHINI, W. and ADAMSON, P.T. (1984a). The occurrence and severity of droughts in South Africa. *WRC Report* No. 91/1/84, Water Research Commission, Pretoria.

ZUCCHINI, W. and ADAMSON, P.T. (1984b). The occurrence and severity of droughts in South Africa : Appendix 6. *WRC Report* No. 91/1/84(A), Water Research Commission, Pretoria.

APPENDIX B

Properties of the Fourier series approximation

We have used the Fourier representation of $\lambda(t)$ as the basis for obtaining approximations. Other representations are also feasible, e.g. polynomials or rational functions. There are several reasons for selecting the Fourier representation rather than other possibilities. Firstly, $\lambda(t)$ is known to be approximately sinusoidal in shape and consequently we can expect that even for small values of L, the approximation $\lambda(t, L) \approx \lambda(t)$ will be reasonably accurate. Secondly, $\lambda(t, L)$ is periodic, which is a property that $\lambda(t)$ is known to have. Thirdly, the individual components in the representation are orthogonal, which is a convenient mathematical property.

APPENDIX C

The Cholesky decomposition

For A an $(n \times n)$ symmetric, positive definite matrix, there exists a unique lower triangular matrix F with positive diagonal elements such that

$$A = FF^T.$$

This is known as the Cholesky decomposition. An algorithm to reduce a matrix A to its Cholesky decomposition is given below.

Notation

f_{ij} is the ijth element of the matrix F.

a_{ij} is the ijth element of the matrix A.

Algorithm

Step 1: Set $f_{11} = \sqrt{a_{11}}$

Step 2: For $j = 2, 3, \ldots, n$

Set $f_{j1} = \frac{a_{1j}}{f_{11}}$

Next j.

Step 3: For $i = 2, 3, \ldots, n-1$

$$Set\ f_{ii} = \sqrt{a_{ii} - \sum_{j=1}^{i-1} f_{ij}^2}$$

$$For\ j = i+1, i+2, \ldots, n.$$

$$Set\ f_{ji} = \frac{a_{ij} - \sum_{k=1}^{i-1} f_{ik} f_{jk}}{f_{ii}}$$

Next j.

Next i.

Step 4:

$$\text{Set } f_{nn} = \sqrt{a_{nn} - \sum_{j=1}^{n-1} f_{nj}^2}.$$

APPENDIX D

A listing of the FORTRAN programs referred to in Chapter 5 are obtainable from the CCWR. The address is:

Computing Centre for Water Research

c/o University of Natal

P O Box 375

Pietermaaritzburg

3200

Tel. (0331) 63320 ext. 177/178

Fax (0331) 61896

Step 5: End.

The elements of F above the main diagonal are defined to be zero. The above algorithm does not set them to zero, so if necessary the following step should be inserted immediately preceding "Next j" in Step 3:

$$\text{Set } f_{ij} = 0.$$

APPENDIX E

The EM algorithm

In any data record collected over a long period of time, one would expect to encounter gaps, where the number of gaps usually increases proportionally with the size of the data set.

Factors which contribute to the occurrence of these gaps may be, for example, loss of records, temporary absence of observers, breakdown of measuring devices or simply incorrect recordings noted. Whatever the reason for their occurrence, gaps in climate variables are problematic for the following reasons:

Firstly, the cross–correlation structure present in the multivariate time series will be destroyed if there are missing values present. Secondly, the autocorrelation structure breaks down when gaps occur and finally, the seasonal structure disappears if the data is not complete.

An effective way of dealing with incomplete data sets is to "fill" these gaps with data. A recent method known as the EM algorithm has been shown to work very satisfactorily when estimating missing values in rainfall data (Makhuvha, 1988). In fact, out of the several methods investigated, the EM algorithm was chosen as the most efficient method for estimation of missing rainfall records, and it performs at least as well as the other methods in terms of accuracy.

The theory and definition of the EM algorithm given here has been extracted from Makhuvha 1988. The same terminology has been adhered to, with only slight changes to suit it to the present problem.

Literature focuses attention on estimating model parameters in the presence of missing observations. However, we are interested in the missing values themselves. Thus the convergence criteria is based on the estimated missing values rather than on the successive parameter estimates.

General description of the EM algorithm

The EM algorithm is a method which iteratively computes maximum likelihood estimates when some observations are missing. Let Z be a complete data set matrix of n observations on k climate variables, where $k \geq 2$ and $n \geq k+2$. We assume that the data is generated by a model described by a density function $f(Z|\phi)$ indexed by unknown parameter ϕ . Given the model and parameter vector $\phi, f(Z|\phi)$ is a function of Z , that is, of the observations.

Definition: The likelihood function $L(\phi|Z)$ is any function of ϕ which is proportional to $f(Z|\phi)$ when given the data value Z .

We denote the log–likelihood function by

$$\ell(\phi, Z) = \ell n L(\phi, Z).$$

Let $Z = (Z_{\text{obs}}, Z_{\text{mis}})$ where Z_{obs} denotes the observed values of Z and Z_{mis} denotes the missing values of Z . Write

$$Z_{\text{obs}} = (Z_{\text{obs},1}, Z_{\text{obs},2}, \ldots, Z_{\text{obs,n}})$$

where $Z_{\text{obs,i}}$ represents the set of climate variables having observation at i, $i = 1, 2, \ldots, n$.

Let $f(Z|\phi) = f(Z_{\text{obs}}, Z_{\text{mis}}|\phi)$ denote the density function of the joint distribution of Z_{obs} and Z_{mis} . To obtain the marginal probability density of Z_{obs} , the missing data Z_{mis} is integrated out. That is,

$$f(Z_{\text{obs}}|\phi) = \int f(Z_{\text{obs}}, Z_{\text{mis}}|\phi) dZ_{\text{mis}}. \tag{1}$$

The likelihood function of ϕ based on Z_{obs} is defined to be any function of ϕ proportional to $f(Z_{\text{obs}}|\phi)$:

$$L(\phi, Z_{\text{obs}}) \propto f(Z_{\text{obs}}|\phi).$$

In situations where values are missing at random, $L(\phi, Z_{\text{obs}})$ is called the true likelihood of ϕ based on the observed data Z_{obs} . By making use of the complete data specification $f(Z|\phi)$, the EM algorithm is used to estimate the parameter ϕ which maximizes $f(Z_{\text{obs}}|\phi)$. In other words, we try to maximize the likelihood function

$$L(\phi, Z_{\text{obs}}) = \int f(Z_{\text{obs}}, Z_{\text{mis}}|\phi) dZ_{\text{mis}} \tag{2}$$

with respect of ϕ .

Definition of the EM algorithm

The EM algorithm has a useful and simple interpretation when the complete data Z has a distribution from the regular exponential family defined by

$$f(Z|\phi) = \frac{b(Z)\ \exp(\phi t(Z)^T)}{a(\phi)} \tag{3}$$

where

ϕ denotes a $(1 \times R)$ vector of parameters,

$t(Z)$ denotes a $(1 \times R)$ vector of complete data sufficient statistics, and

a and b are functios of ϕ and Z respectively.

The parameterization of ϕ in (3) is unique up to an arbitrary non–singular $(R \times R)$ linear transformation, as is the corresponding choice of $t(Z)$.

We restrict our attention to only one class of the exponential type of distribution, namely, the Multivariate Normal distribution. We say that a distribution is Multivariate Normal if its density function is given by:

$$f(Z|\mu, \Sigma) = (2\pi)^{-k/2}|\Sigma|^{-\frac{1}{2}}\exp[-\frac{1}{2}(Z-\mu)^T\Sigma^{-1}(Z-\mu)] \tag{4}$$

where

$$Z^T = (Z_1\ Z_2\ \ldots Z_k),$$

$$\mu = (\mu_1\ \mu_2\ \ldots \mu_k),$$

$$\Sigma = \begin{bmatrix} \sigma_1^2 & \cdots & & \cdots & \sigma_{1k} \\ \vdots & & \vdots & & \vdots \\ \vdots & \cdots & \sigma_{ij} & \cdots & \vdots \\ \sigma_{k1} & \cdots & \vdots & \cdots & \sigma_k^2 \end{bmatrix} \tag{5}$$

where σ_{ij} is the covariance of the ith and jth component of Z .

Suppose we are dealing with more than one set of observations, that is, we have a matrix of n sets of observations such that

$$Z = \begin{bmatrix} Z_{11} & Z_{12} & \ldots & Z_{1k} \\ Z_{21} & Z_{22} & \ldots & Z_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ Z_{n1} & Z_{n2} & \ldots & Z_{nk} \end{bmatrix}. \tag{6}$$

The likelihood of the observations (6) is

$$L(\mu, \Sigma|Z) = (2\pi)^{\frac{-nk}{2}}|\Sigma|^{\frac{-n}{2}}\exp\left[-\frac{1}{2}\sum_{i=1}^{n}(Z_i-\mu)^T\Sigma^{-1}(Z_i-\mu)\right] \tag{7}$$

Using (7) we can find the sufficient statistics for the parameters.

$$L(\mu,\Sigma|Z) = (2\pi)^{\frac{-nk}{2}}|\Sigma|^{\frac{-n}{2}}\exp\left[-\frac{1}{2}n\ \text{tr}(\mu^T\mu\Sigma^{-1})\right]\exp\left[-\frac{1}{2}\sum_{i=1}^{n}tr(Z_iZ_iZ_i^T)\begin{pmatrix}-2\Sigma^{-1}\\ \Sigma^{-1}\end{pmatrix}\right]$$

$$= (2\pi)^{\frac{-nk}{2}}|\Sigma|^{\frac{-n}{2}}\exp\left[-\frac{1}{2}n\ \text{tr}(\mu^T\mu\Sigma^{-1})\right]$$

$$\exp\left[-\frac{1}{2}\sum_{i=1}^{n}\left(\sum_{j=1}^{k}\sum_{\ell=1}^{k}[Z_{ij}\mu_{\ell}\sigma_{j\ell}+Z_{ij}Z_{i\ell}\sigma_{j\ell}]\right)\right]$$

$$= (2\pi)^{\frac{-nk}{2}}|\Sigma|^{\frac{-n}{2}}\exp\left[-\frac{1}{2}n\ \text{tr}(\mu^T\mu\Sigma^{-1})\right]$$

$$\exp\left[-\frac{1}{2}\sum_{i=1}^{n}(1_n\otimes Z_i)^T\begin{pmatrix}\mu_1\underset{\sim}{\sigma}_1\\ \vdots\\ \mu_k\underset{\sim}{\sigma}_k\end{pmatrix}-\frac{1}{2}\sum_{i=1}^{n}(Z_i\otimes Z_i)^T\sigma_i\right]$$

Therefore

$$t(Z)=\begin{pmatrix}\sum_{i=1}^{n}\ 1_n\otimes Z_i\\ \sum_{i=1}^{n}\ Z_i\otimes Z_i\end{pmatrix} \tag{8}$$

$$\phi=\begin{bmatrix}\mu_1\underset{\sim}{\sigma}_1\\ \vdots\\ \mu_k\underset{\sim}{\sigma}_k\\ -\frac{1}{2}\underset{\sim}{\sigma}_1\\ \vdots\\ -\frac{1}{2}\underset{\sim}{\sigma}_k\end{bmatrix} \tag{9}$$

where

ϕ is a vector of parameters, and

$t(Z)$ is the sufficient statistics for ϕ since it does not depend on any parameter.

Since the statistics $t(Z)$ is sufficient for the parameter ϕ, it therefore has all the relevant information contained in Z for inference about the parameter.

The E step and the M step of EM.

Each iteration of the EM algorithm involves two steps which are called the expectation step (E step) and the maximization step (M step). The steps given below may be applied if equation (7) satisfies the conditions of it being a class of the exponential type of distribution.

Suppose that $\phi^{(p)}$ denotes the current value of ϕ after p cycles of the algorithm. The next cycle involves the following two steps:

E step: At the $(p+1)$ cycle, the E step is the computation of the conditional expectation of the complete data sufficient statistics given:

i) the observed data $Z_{\text{obs}} = (Z_{\text{obs,1}}, \ldots, Z_{\text{obs,n}})$ and

ii) the estimated value of the parameter from the pth cycle.

That is, we compute

$$t^{(p)} = E[t(Z)|Z_{\text{obs}}, \phi^{(p)}]. \tag{10}$$

M step: At the $(p+1)$ cycle, the M step is the maximization of the complete data likelihood function in which the complete data sufficient statistics $t(Z)$ has been replaced by its conditional expectation obtained in the E step. We set the derivatives of the complete data likelihood function to zero and determine $\phi^{(p+1)}$, i.e. as the solution of the equation

$$E(t(Z)|\phi) = t^{(p)} \tag{11}$$

which defines the maximum likelihood estimator of ϕ under the assumption that (7) is a class of the exponential family.

We now show how the E and M steps of the EM algorithm are obtained under the assumption that the distribution is multivariate normal.

If, at the pth iteration, $\phi^{(p)}$ denotes the current estimates of the parameters, then the E step of the algorithm consists of calculating:

$$E\left(\sum_{i=1}^{n} Z_{ij}|Z_{\text{obs}}, \phi^{(p)}\right) = \sum_{i=1}^{n} Z_{ij}^{(p)}, \qquad j = 1, 2, \ldots, k.$$

$$E\left(\sum_{i=1}^{n} Z_{ij}Z_{i\ell}|Z_{\text{obs}}, \phi^{(p)}\right) = \sum_{i=1}^{n} Z_{ij}^{(p)}Z_{i\ell}^{(p)} + C_{j\ell i}^{(p)}, \qquad j, \ell = 1, \ldots, k,$$

where

$$\begin{aligned} Z_{ij}^{(p)} &= Z_{ij} && \text{if } Z_{ij} \text{ is observed} \\ &= E(Z_{ij}|Z_{\text{obs,i}}, \phi^{(p)}) && \text{if } Z_{ij} \text{ is missing} \end{aligned}$$

and

$$\begin{aligned} C_{j\ell i}^{(p)} &= 0 && \text{if } Z_{ij} \text{ or } Z_{i\ell} \text{ are observed} \\ &= \text{cov}(Z_{ij}, Z_{i\ell}|Z_{\text{obs},i}, \phi^{(p)}) && \text{if } Z_{ij} \text{ or } Z_{i\ell} \text{ are missing} \end{aligned}$$

Missing values Z_{ij} are therefore replaced by the conditional mean of Z_{ij} given the set of values $Z_{\text{obs},i}$ observed for that observation.

Similarly, the maximization step (M step) is found from equation (8). The new estimates $\phi^{(p+1)}$ of the parameters are estimated as follows:

$$\mu_j^{(p+1)} = \frac{1}{n}\sum_{i=1}^{n} Z_{ij}^{(p)}, \qquad j = 1, 2, \ldots, k$$

$$\sigma_{j\ell}^{(p+1)} = \frac{1}{n}\, E\left(\sum_{i=1}^{n} Z_{ij} Z_{i\ell} | Z_{\text{obs}}\right) - \mu_j^{(p+1)} \mu_\ell^{(p+1)}$$

$$= \frac{1}{n}\sum_{i=1}^{n}\left[(Z_{ij}^{(p)} - \mu_j^{(p+1)})(Z_{i\ell}^{(p)} - \mu_\ell^{(p+1)}) + C_{j\ell i}^{(p)}\right], \qquad j, \ell = 1, 2, \ldots, k.$$

Estimation of missing values using the EM

The method described here estimates the missing data point, say y_ℓ, by using all the records, i.e. estimated and real records. In this method all the observations are utilized after the initial estimation stage.

Algorithm

Suppose we are considering a climate variables matrix Z of dimension $n \times (k+1)$. Partition the Z matrix into a vector of observations in the target variable, y, of dimension $(n \times 1)$ and a matrix of observations in the control variables, X, of dimension $(n \times k)$. Note that any variable in the Z matrix can be regarded as the target variable, depending on which variable's missing values we are currently estimating.

Suppose we wish to estimate the missing value y_ℓ:

Cycle 0

Step 1

Construct the vector y^* from y and the matrix X^* from the $(n \times k)$ matrix X, by eliminating from both all the rows which contain one or more missing observations in either. For example, because y_ℓ is one of the missing observations then the ℓth row in both y and X is eliminated. Suppose that y^* ends up with n^* entries, then X^* is an $(n^* \times k)$ matrix. The vector y^* and matrix X^* should now contain no missing observations. Check that there is sufficient data to regress y^* on X^*. If there is not then some of the control variables will have to be removed and one must begin again.

Step 2

Calculate the least squares estimates of the regression parameters using the target variable vector y^* and the matrix of control variables X^*. That is find:

$$\widehat{\beta}^{(0)} = (X^{*T}X^{*})^{-1}X^{*T}y^{*}$$

and

$$\widehat{\beta}_0^{(0)} = \overline{y}^{*} - \overline{X}^{*}\widehat{\beta}^{(0)}$$

where

$$\overline{y}^{*} = \frac{1}{n^{*}} \sum_{i=1}^{n*} y_i^{*},$$

and

$$\overline{X}^{*} = \frac{1}{n*} \sum_{i=1}^{n*} x_{ij}^{*} \qquad j = 1, 2, \ldots, k$$

and where the superscript (0) represents the initial estimation cycle.

Estimate the missing record y_ℓ using the regression model:

$$y_\ell^{(0)} = \beta_0^{(0)} + \sum_{i=1}^{k} x_{\ell j}\widehat{\beta}_j^{(0)}$$

where $x_{\ell j}$, $(j = 1, 2, \ldots, k)$ are the observed values from the control variables matrix X .

Step 3

After all the missing values in matrix Z have been estimated, create a new "data" matrix, say $Z^{(0)}$ containing estimates obtained in place of missing values.

Cycle b

Step 1

Suppose that the new data matrix created in the previous cycle is $Z^{(b-1)}$, then partition $Z^{(b-1)}$ into a matrix of control variables $X^{(b-1)}$ and a vector of the target variable $y^{(b-1)}$.

Step 2

Calculate the least squares estimates by using the new target variable vector $y^{(b-1)}$ and the matrix of control variables $X^{(b-1)}$, where superscript $(b-1)$ represents the previous cycle. That is find:

$$\widehat{\beta}^{(b)} = (X^{(b-1)T}X^{(b-1)})^{-1}X^{(b-1)T}y^{(b-1)}$$

and

$$\widehat{\beta}_{(0)}^{(b)} = \overline{y}^{(b-1)} - \overline{X}^{(b-1)}\widehat{\beta}^{(b)}$$

where

$$\overline{y}^{(b-1)} = \frac{1}{n}\sum_{i=1}^{n} y_i^{(b-1)},$$

and

$$\overline{X}^{(b-1)} = \frac{1}{n}\sum_{i=1}^{n} x_{ij}^{(b-1)}, \qquad j = 1, 2, \ldots, k.$$

Re–estimate the missing record y_ℓ :

$$y_\ell^{(b)} = \widehat{\beta}_{(0)}^{(b)} + \sum_{j=1}^{k} x_{\ell j}^{(b-1)}\widehat{\beta}_j^{(b)}.$$

Let

$$\text{Conv}_\ell = \frac{y_\ell^{(b)} - y_\ell^{(b-1)}}{y_\ell^{(b)}}$$

where b represents the current iteration; $b-1$ represents the previous cycle.

Step 3

If all the missing values in the current variable have been estimated, then check for convergence by using the following criterion:

$$\text{Crit } = \sum_{j=1}^{k} \sum_{\ell=1}^{n-n_j*} \text{conv}_{\ell j}$$

where n_j^* is the number of observed values in the current variable.

Step 4

If Crit $\leq F, F$ a small number close to zero, then y_ℓ is considered the required estimate of the missing value and the re–estimation if discontinued, otherwise repeat Steps 1, 2 and 3.

APPENDIX D

PROGRAMS

In this appendix, we give a listing of the FORTRAN programs which were developed to implement the methodology discussed in this report. The programs listed here are set for 7 climate variables, excluding rainfall, and for 12 years of historical records.

PROGRAM 1

```
C        --------------------------------------------------------
C        ...... PROGRAM TO COMPUTE VECTORS REQUIRED FOR PARAMETER
C               ESTIMATION OF RAINFALL MODEL.
C        --------------------------------------------------------

C        NWW(T) = THE NUMBER OF TIMES IT WAS WET IN PERIOD T-1 AND
C                 WET IN PERIOD T.

C        NDW(T) = THE NUMBER OF TIMES IT WAS DRY IN PERIOD T-1 AND
C                 WET IN PERIOD T.

C        R(I,T) = THE Ith NON-ZERO RAINFALL DEPTH IN PERIOD T,
C                 I = 1, 2, ........., NR(T); T = 1, 2, ........., NT.

C        NR(T) = THE NUMBER OF TIMES IT WAS WET (NON-ZERO RAIN) IN
C                PERIOD T.

C        NW(T) = THE NUMBER OF TIMES IT WAS WET IN PERIOD T-1 AND THERE
C                WAS AN OBSERVATION IN PERIOD T (i.e. THERE WAS NOT A
C                GAP IN PERIOD T).

C        ND(T) = THE NUMBER OF TIMES IT WAS DRY (ZERO RAIN) IN PERIOD
C                T-1 AND THERE WAS AN OBSERVATION IN PERIOD T.

C        FOR EACH T = 1,2,...,NT (WHERE NT = THE NUMBER OF PERIODS IN
C        THE YEAR):
C        THE ABOVE ARRAYS ARE REQUIRED BY THE ESTIMATION ALGORITHMS
C        AS FOLLOWS:-
C                i) NW( ) AND NWW( ) ARE REQUIRED TO ESTIMATE THE
C                   PARAMETER FOR THE PROBABILITY THAT A WET PERIOD
C                   FOLLOWS A WET PERIOD.
C               ii) ND( ) AND NDW( ) ARE REQUIRED TO ESTIMATE THE
C                   PARAMETERS FOR THE PROBABILITY THAT A WET PERIOD
C                   FOLLOWS A DRY PERIOD.
C              iii) NR( ) AND R( , ) ARE REQUIRED TO FIT THE PARAMETERS
C                   OF THE MEAN RAINFALL RAIN IN A WET PERIOD AND THE
C                   COEFFICIENT OF VARIATION.

C        THE OUTPUT OF THIS PROGRAM IS GIVEN IN TWO PARTS.  NAMELY, THE
C        FIRST PART:
C        T (T=1,2,....,NT), NW, NWW, ND, NDW, N, NR
C        THE SECOND PART:
C        R(I,T) (I=1,2,....,NR(T); T=1,2,....,NT).
```

```
C       MAIN PROGRAM
C       ------------

C       ...... NT = THE NUMBER OF PERIODS IN THE YEAR (e.g. 365 FOR
C                  DAILY DATA)
C       ...... NY = THE NUMBER OF YEARS OF DATA (INCLUDING THE MISSING
C                  VALUES)
C       ...... NRT = THE MAXIMUM VALUE GIVEN FOR THE DIMENSION OF THE
C                   ARRAY R(I,T)
C       ...... RAIN = THE ARRAY THAT CONTAINS THE DATA
C       ...... IND = AN INDICATOR OF THE STATUS OF THE PREVIOUS PERIOD
C                      -1 => PREVIOUS OBSERVATION MISSING
C                       0 => PREVIOUS PERIOD WAS DRY
C                       1 => PREVIOUS PERIOD WAS WET

        INTEGER         NT,NY,NRT,IND,T
        PARAMETER       (NT=365)
        PARAMETER       (NY=12)
        PARAMETER       (NRT=50)
        INTEGER         N(NT)
        INTEGER         NR(NT)
        INTEGER         NW(NT)
        INTEGER         NWW(NT)
        INTEGER         ND(NT)
        INTEGER         NDW(NT)
        REAL            R(NRT,NT)

  15    FORMAT (7 (I4))
  25    FORMAT (8 (F9.2))

        OPEN(UNIT=12,FILE='\WATER\DATA\CLIMA.DAT',STATUS='OLD')
        OPEN(UNIT=10,FILE='\WATER\DATA\COUNTS.DAT',STATUS='UNKNOWN')
        OPEN(UNIT=20,FILE='\WATER\DATA\RAIN.DAT',STATUS='UNKNOWN')


C       ...... THE REQUIRED VECTORS ARE COMPUTED.

        DO 10, I = 1, NT
           N(I) = 0
           NR(I) = 0
           NW(I) = 0
           NWW(I) = 0
           ND(I) = 0
           NDW(I) = 0
           DO 20, J = 1, NRT
              R(J,I) = 0
  20       CONTINUE
  10    CONTINUE
        IND = -1
        DO 30, J = 1, NY
           DO 40, I = 1, NT
              READ (12,*) RAIN
              IF (RAIN .EQ. 0) THEN
```

```
                  N(I) = N(I) + 1
                  IF (IND .EQ. 0) THEN
                     ND(I) = ND(I) + 1
                  ELSEIF (IND .EQ. 1) THEN
                     NW(I) = NW(I) + 1
                     IND = 0
                  ELSEIF (IND .EQ. -1) THEN
                     IND = 0
                  ENDIF
               ELSEIF (RAIN .GT. 0) THEN
                  NR(I) = NR(I) + 1
                  R(NR(I),I) = RAIN
                  IF (IND .EQ. 0) THEN
                     NDW(I) = NDW(I) + 1
                     IND = 1
                  ELSEIF (IND .EQ. 1) THEN
                     NWW(I) = NWW(I) + 1
                  ELSEIF (IND .EQ. -1) THEN
                     IND = 1
                  ENDIF
               ELSEIF (RAIN .LT. 0) THEN
                  IND = -1
               ENDIF
 40         CONTINUE
 30      CONTINUE
         DO 50, I = 1, NT
            N(I) = N(I) + NR(I)
            ND(I) = ND(I) + NDW(I)
            NW(I) = NW(I) + NWW(I)
 50      CONTINUE
C
C        ...... THE VECTORS COMPUTED ARE WRITTEN OUT
C
         DO 60, I = 1, NT
            WRITE (10,15) I, NW(I), NWW(I), ND(I), NDW(I), N(I), NR(I)
 60      CONTINUE
         DO 70, T = 1, NT
            WRITE (20,25) (R(I,T), I = 1, NR(T))
 70      CONTINUE

         STOP
         END
```

PROGRAM 2

```
C        ------------------------------------------------------------
C        ...... PROGRAM TO COMPUTE PARAMETER ESTIMATES FOR RAINFALL
C               MODEL - PROBABILITIES OF WET & DRY SEQUENCES
C        ------------------------------------------------------------

C        ...... THIS PROGRAM USES THE GENERIC NOTATION MM(T) & M(T)
C               TO REPRESENT THE RELEVANT ARRAYS AS FOLLOWS:
C                 i) WHEN WE ARE ESTIMATING THE PROBABILITY THAT A
C                    WET PERIOD FOLLOWS A WET PERIOD, THEN
C                        MM(T) = NW(T)
C                        M(T) = NWW(T)
C                ii) WHEN WE ARE ESTIMATING THE PROBABILITY THAT A
C                    WET PERIOD FOLLOWS A DRY PERIOD, THEN
C                        MM(T) = ND(T)
C                        M(T) = NDW(T)
C               iii) WHEN WE ARE ESTIMATING THE PROBABILITY THAT PERIOD
C                    T IS WET, THEN
C                        MM(T) = N(T)
C                        M(T) = NR(T)
C
C               NP = NUMBER OF PARAMETERS TO BE FITTED
C               THETA(NP) = VECTOR OF PARAMETERS ESTIMATED
C               AM(O:K) = CORRESPONDING AMPLITUDES, K = (NP-1)/2
C               PH(K) = CORRESPONDING PHASES
C               P(NT) = CURRENT ESTIMATES OF PROBABILITIES
C               L(NT) = CURRENT ESTIMATES OF LOGITS
C               DER(NT) = VECTOR OF 1ST PARTIAL DERIVATIVES
C               DER2(NT,NT) = MATRIX OF 2ND PARTIAL DERIVATIVES
C               PHI(NP,NT) = MATRIX OF FOURIER TERMS
C               DELTA = CONVERGENCE CRITERION

        INTEGER         NP,K,NT,ITER,T,MAXITER,NPMAX,KMAX
        PARAMETER       (NPMAX=13)
        PARAMETER       (KMAX=(NPMAX-1)/2)
        PARAMETER       (NT=365)
        REAL            MM(NT)
        REAL            M(NT)
        REAL            PI, LOGIT
        PARAMETER       (PI=3.141593)
        REAL            AM(O:KMAX)
        REAL            PH(KMAX)
        REAL            P(NT)
        REAL            L(NT)
        REAL            DER(NPMAX)
        REAL            DER2(NPMAX,NPMAX)
        REAL            THETA(NPMAX)
        REAL            PHI(NT,NPMAX)
```

```
C  5     FORMAT (4X,2F4.0)  ----- PROB (W/W)
C  5     FORMAT (12X,FF4.0) ----- PROB (W/D)
C  5     FORMAT (20X,2F4.0) ----- PROB (W)

   5     FORMAT (4X,2F4.0)
C  5     FORMAT (12X,2F4.0)
C  5     FORMAT (20X,2F4.0)
  15     FORMAT (' EPS, MAXITER = ')
  25     FORMAT (' ...... DID NOT CONVERGE')
  35     FORMAT (/, ' ......', I3, ' ITERATION',/)
  45     FORMAT (/, ' AMPLITUDE: ')
  55     FORMAT (/, ' PHASE: ')
  65     FORMAT (9F8.3)
  75     FORMAT (' OPTIMAL PARAMETERS TO BE FITTED: ', I4)
  85     FORMAT (' INITIAL ESTIMATES: ', F10.4)

         OPEN (UNIT=4,FILE='CON')
         OPEN (UNIT=9,FILE='LPT1')
         OPEN (UNIT=10,FILE='\WATER\DATA\COUNTS.DAT',STATUS='OLD')

C        ...... INPUT DATA

         PRINT 15
         READ (4,*) EPS, MAXITER

         DO 10, T = 1, NT
            READ (10,5) MM(T), M(T)
  10     CONTINUE

         CALL TRIG (PHI,NPMAX,NT)

C        ...... DIFFERENT AMOUNT OF PARAMETERS FITTED AT A TIME.
C               PROGRAM STOPS ONCE OPTIMAL NO. OF PARAMETERS ARE FITTED

         CRITO = 10 ** 10
         DO 300, NP = 1, NPMAX, 2
            WRITE (9,*) 'NO. OF PARAMETERS FITTED = ', NP

C        ...... COMPUTE INITIAL ESTIMATES OF THE PROBABILITIES
C               AND LOGITS

            DO 20, T = 1, NT
               IF (MM(T) .GT. 0) THEN
                  P(T) = M(T) / MM(T)
               ELSE
                  P(T) = -1
                  GOTO 20
               ENDIF
               IF (M(T) .EQ. 0) THEN
                  L(T) = -5
               ELSEIF (M(T) .EQ. MM(T)) THEN
                  L(T) = 5
               ELSEIF ((M(T) .GT. 0.00001).AND.(M(T) .NE. MM(T))) THEN
```

```
               L(T) = LOG (P(T) / (1-P(T)))
            ENDIF
   20    CONTINUE

         DO 30, I = 1, NP
            TO = 0
            T1 = 0
            DO 40, T = 1, NT
               IF (MM(T) .GT. 0.00001) THEN
                  TO = TO + L(T) * PHI (T,I)
                  T1 = T1 + PHI (T,I) ** 2
               ENDIF
   40       CONTINUE
            THETA (I) = TO / T1
            WRITE (9,85) THETA (I)
   30    CONTINUE

C      ...... ITERATIVE ESTIMATION OF PARAMETERS

         IC = 0
         DO 200, ITER = 1, MAXITER
            WRITE (9,35) ITER

C      ...... COMPUTE 1ST AND 2ND DERIVATIVES

            DO 50, I = 1, NP
               DER (I) = 0
               DO 60, J = 1, NP
                  DER2(I,J) = 0
   60          CONTINUE
   50       CONTINUE
            DELTA = 0
            DO 70, T = 1, NT
               LOGIT = THETA(1)
               DO 80, I = 2, NP
                  LOGIT = LOGIT + THETA(I) * PHI(T,I)
   80          CONTINUE
               TO = EXP (LOGIT)
               PROB = TO / (1+TO)
               T1 = M(T) - MM(T) * PROB
               T2 = MM(T) * PROB / (1+TO)
               DELTA = DELTA + ABS (P(T) - PROB)
               P(T) = PROB
               DO 90, I = 1, NP
                  DER(I) = DER(I) + T1 * PHI(T,I)
                  DO 100, J = 1, I
                     DER2(I,J) = DER2(I,J)-T2*PHI(T,I)*PHI(T,J)
  100             CONTINUE
   90          CONTINUE
   70       CONTINUE
            DO 110, I = 1, NP
               DO 120, J = I+1, NP
                  DER2(I,J) = DER2(J,I)
  120          CONTINUE
```

```
 110           CONTINUE

               CALL LINEAR (NPMAX,NP,DER,DER2,THETA)

C        ...... TESTING FOR CONVERGENCE

               IF (DELTA .GT. EPS) THEN
                  IC = 0
               ELSE
                  IC = 1
               ENDIF
               IF (IC) 200, 200, 400
 200        CONTINUE
            WRITE (9,25)

C        ...... TRANSFORMING PARAMETERS TO THEIR AMPLITUDE AND PHASE
C               REPRESENTATION

 400        K = (NP-1) / 2
            CALL AMPHA (AM,PH,THETA,NPMAX,KMAX,K,PI,NT)

C        ...... MODEL SELECTION

            LLK = 0
            DO 210, T = 1, NT
               IF (MM(T) .GT. 0.000001) THEN
                  LLK = LLK+M(T)*LOG(P(T))+(MM(T)-M(T))*LOG(1-P(T))
               ENDIF
 210        CONTINUE
            CRIT = -LLK + NP
            IF (CRIT .LT. CRITO) THEN
               LO = NP
               CRITO = CRIT
            ELSE
               WRITE (9,75) LO
               STOP
            ENDIF
 300     CONTINUE

         STOP
         END
```

PROGRAM 3

```
C        ----------------------------------------------------------
C        ...... PROGRAM TO COMPUTE PARAMETER ESTIMATES FOR THE
C               DISTRIBUTION OF RAINFALL DEPTHS.
C        ----------------------------------------------------------

C        ...... THE FOLLOWING NOTATION IS USED:
C               NP = NUMBER OF PARAMETERS IN THE MEAN FUNCTION
C               R(,) = MATRIX OF RAINFALL DEPTHS OBSERVED
C               DER() = VECTOR OF 1ST PARTIAL DERIVATIVES
C               DER2() = MATRIX OF 2ND PARTIAL DERIVATIVES
C               Q() = VECTOR OF AVERAGE OBSERVED RAINFALL IN EACH PERIOD
C               THETA() = VECTOR OF PARAMETER ESTIMATES
C               AM() = CORRESPONDING AMPLITUDES
C               PH() = CORRESPONDING PHASES
C               F() = CURRENT ESTIMATE OF THE MEAN
C               SO() = OBSERVED DAILY STANDARD DEVIATIONS
C               SF() = FITTED DAILY STANDARD DEVIATIONS
C               DELTA = CONVERGENCE CRITERION

      INTEGER          NP,NT,T,ITER,K,MAXITER
      PARAMETER        (NP=3)
      PARAMETER        (NT=365)
      PARAMETER        (K=(NP-1)/2)
      INTEGER          NR(NT)
      REAL             R(50,NT)
      REAL             PI,DENOM,NUM
      PARAMETER        (PI=3.141593)
      REAL             COEFF,DELTA
      REAL             DER(NP),SOLN(NP)
      REAL             DER2(NP,NP)
      REAL             PHI(NT,NP)
      REAL             Q(NT),F(NT),SO(NT),SF(NT)
      REAL             THETA(NP)
      REAL             AM(0:K)
      REAL             PH(K)

    5 FORMAT (6(4X), I4)
   15 FORMAT (14(I5))
   25 FORMAT (' EPS, MAXITER = ')
   35 FORMAT (' ...... DID NOT CONVERGE')
   45 FORMAT (/, ' ......', I3, ' ITERATION',/)
   55 FORMAT (/, ' AMPLITUDE: ')
   65 FORMAT (/, ' PHASE: ')
   75 FORMAT (9F8.3)
   85 FORMAT (' OPTIMAL PARAMETERS TO BE FITTED: ', I4)
   95 FORMAT (' INITIAL ESTIMATES: ', F10.4)
  105 FORMAT (' COEFFICIENT OF VARIATION: ', F10.4)

      OPEN (UNIT=4,FILE='CON')
      OPEN (UNIT=9,FILE='LPT1')
      OPEN (UNIT=10,FILE='\WATER\DATA\RAIN.DAT',STATUS='OLD')
```

```
      OPEN (UNIT=20,FILE='\WATER\DATA\COUNTS.DAT',STATUS='OLD')
C     ...... INPUT DATA
      PRINT 25
      READ (4,*) EPS, MAXITER

      DO 10, T = 1, NT
         READ (20,5) NR(T)
 10   CONTINUE
      DO 20, T = 1, NT
         READ (10,*) (R(I,T), I = 1, NR(T))
 20   CONTINUE

      CALL TRIG (PHI,NP,NT)

C     ...... COMPUTE INITIAL ESTIMATES

      DO 30, T = 1, NT
         IF (NR(T) .GT. 0) THEN
            TERM0 = 0
            DO 40, I = 1, NR(T)
               TERM0 = TERM0 + R(I,T)
 40         CONTINUE
            Q(T) = TERM0 / NR(T)
         ENDIF
 30   CONTINUE
      DO 50, I = 1, NP
         TERM0 = 0
         TERM1 = 0
         DO 60, T = 1, NT
            IF (NR(T) .NE. 0) THEN
               TERM0 = TERM0 + Q(T) * PHI(T,I)
               TERM1 = TERM1 + PHI(T,I) ** 2
            ENDIF
 60      CONTINUE
         THETA (I) = TERM0 / TERM1
         WRITE (9,95) THETA (I)
 50   CONTINUE

C     ...... ITERATIVE PARAMETER ESTIMATION

      IC = 0
      DO 100, ITER = 1, MAXITER
         WRITE (9,45) ITER

C     ...... COMPUTE 1ST & 2ND PARTIAL DERIVATIVES

         DO 70, I = 1, NP
            DER(I) = 0
            DO 80, J = 1, I
               DER2(I,J) = 0
 80         CONTINUE
 70      CONTINUE
```

```
          DO 90, T = 1, NT
             TERMO = THETA(1)
             DO 180, I = 2, NP
                TERMO = TERMO + THETA(I) * PHI(T,I)
  180        CONTINUE
             F(T) = TERMO
   90     CONTINUE
          DO 110, T = 1, NT
             IF (NR(T) .GT. 0) THEN
                DO 120, I = 1, NP
                   DER(I) = DER(I) - NR(T) * (Q(T) - F(T)) * PHI(T,I)
                   DO 130, J = 1, I
                      DER2(I,J) = DER2(I,J)+NR(T)*PHI(T,I)*PHI(T,J)
  130              CONTINUE
  120           CONTINUE
             ENDIF
  110     CONTINUE
          DO 140, I = 1, NP
             DO 150, J = I+1, NP
                DER2(I,J) = DER2(J,I)
  150        CONTINUE
  140     CONTINUE

          CALL LINEAR (NP,NP,DER,DER2,THETA)

C       ...... CONVERGENCE TEST

          DELTA = 0
          DO 170, I = 1, NP
             DELTA = DELTA + ABS (DER(I))
  170     CONTINUE
          IF (DELTA .GT. EPS) THEN
             IC = 0
          ELSE
             IC = 1
          ENDIF
          IF (IC) 100,100,2
  100  CONTINUE
       WRITE (9,35)

    2  DO 200, T = 1, NT
          TERMO = THETA(1)
          DO 220, I = 2, NP
             TERMO = TERMO + THETA(I) * PHI(T,I)
  220     CONTINUE

C       ...... COMPUTE FITTED VALUES

          F(T) = TERMO
  200  CONTINUE

C       ...... OUTPUT OBSERVED AND FITTED DAILY MEANS

       DO 300, T = 1, NT
```

```
         PRINT *, T, Q(T), F(T)
 300  CONTINUE
C     ...... COMPUTE THE COEFFICIENT OF VARIATION
      DENOM = 0
      NUM = 0
      DO 510, T = 1, NT
         DO 520, I = 1, NR(T)
            NUM = NUM + (R(I,T) - F(T)) ** 2
 520     CONTINUE
         DENOM = DENOM + NR(T) * (F(T) ** 2)
 510  CONTINUE
      COEFF = SQRT (NUM / DENOM)
      WRITE (9,105) COEFF
C     ...... COMPUTE THE AMPLITUDE AND PHASE REPRESENTATION
      CALL AMPHA (AM,PH,THETA,NP,K,K,PI,NT)
C     ...... COMPUTE THE FITTED AND OBSERVED STANDARD DEVIATIONS
      STOP
      END
```

PROGRAM 4

```
C         ---------------------------------------------
C         ...... PROGRAM TO GENERATE RAINFALL SEQUENCES
C         ---------------------------------------------

          INTEGER          NT,NY,NP,PSTATE,STATE

C         ...... PSTATE = PRESENT STATE OF DAY
C         ...... STATE = PREVIOUS STATE OF DAY

          PARAMETER        (NT=365)
          PARAMETER        (NY=51)
          PARAMETER        (NP=3)

C         ...... NT = £ OBSERVATIONS PER YEAR
C         ...... NV = £ VARIABLES
C         ...... NY = £ YEARS TO BE GENERATED
C         ...... NP = £ PARAMETERS IN SEASONAL MODEL

          INTEGER          SEED
          REAL             RAIN
          REAL             GAM (2,NP)
          REAL             PHI (NP,0:NT)
          REAL             AMP (0:NP)
          REAL             PHASE (NP)

          COMMON           IDUM1,IDUM2

   15     FORMAT (F9.2)
   25     FORMAT (' GIVE 2 -VE Nos. TO INITIALIZE RANDOM GENERATOR',/)

          OPEN (UNIT=9,FILE='LPT1')
          OPEN (UNIT=12,FILE='\WATER\DATA\EST.DAT',STATUS='OLD')
          OPEN (UNIT=10,FILE='\WATER\DATA\SIMU.DAT',STATUS='UNKNOWN')
          OPEN (UNIT=22,FILE='CON')

C         ...... COMPUTE THE FOURIER SERIES TERMS

          CALL COSSIN (PHI,NP,NT)
          DO 60, I = 1, NP
             PHI (I,0) = PHI (I,NT)
   60     CONTINUE

          PI=3.14159

C         ...... READING PARAMETER ESTIMATES

          READ (12, *) (GAM (1,J), J = 1, NP)
          READ (12, *) (GAM (2,J), J = 1, NP)
          READ (12, *) (AMP (I), I = 0, 1)
          READ (12, *) (PHASE (I), I = 1, 1)
          READ (12, *) CV
```

```
C       ...... INPUT SEED TO START RANDOM NUMBER GENERATOR.  MUST BE
C              NEGATIVE NUMBER.

        PRINT 25
        READ (22, *) SEED (1), SEED(2)

        IDUM1 = SEED (1)
        IDUM2 = SEED (2)

C       ...... COMPUTE PARAMETERS NEEDED FOR COMPUTATION OF RAINFALL
C              DEPTH

        CALL CALBET (BETA,CV)
        ALPH = 1 + 1 / BETA
        GAMM = GAMMA (ALPH)
        BI = 1 /BETA
        W = 0.01721421

C       ...... SET INITIAL STATE OF DAY TO BE DRY
C       ...... SET INITIAL CLIMATE VALUE TO ZERO

C       ...... STATE = 1 ==> DRY
C       ...... STATE = 2 ==> WET

        STATE = 1

        DO 30, I = 1, NY
           DO 40, J = 1, NT

C       ...... GENERATE RAINFALL VALUE
C              ----------------------

C       ...... COMPUTE PROBABILITY THAT A WET DAY FOLLOWS A WET DAY, OR
C              THE PROBABILITY THAT A WET DAY FOLLOWS A DRY DAY.

              CALL PIEST (NP,GAM,STATE,J,PHI,PI,NT)

C       ...... GENERATE A UNIFORM RANDOM NUMBER BETWEEN 0 AND 1.

              UNIFOR = URAN (IDUM)
              IF (UNIFOR .LT. PI) THEN
                 PSTATE = 2
              ELSE
                 PSTATE = 1
              ENDIF

C       ...... GENERATE RAINFALL DEPTH
C              ----------------------

                CALL DEPTH3 (IDUM2,NP,RAIN,J,AMP,PHASE,GAMM,BI,W)

C       ...... OUTPUT GENERATED SEQUENCES
```

```
            IF (I .NE. 1) THEN
               WRITE (10,15) RAIN
            ENDIF
C        ...... UPDATE THE STATE OF THE PREVIOUS DAY
            IF (PSTATE .NE. STATE) THEN
               STATE = PSTATE
            ENDIF
   40       CONTINUE
   30    CONTINUE
         STOP
         END
```

PROGRAM 5

```
C       ------------------------------------------------------------------
C       ...... PROGRAM TO CONDITION DATA SET ACCORDING TO TH WET &
C              DRY STATUS OF THE DAY
C       ------------------------------------------------------------------

        INTEGER         NY,NT
        PARAMETER       (NY=12)
        PARAMETER       (NT=365)
        INTEGER         SEQ(2,NY,NT)
        INTEGER         COUNT(2,NY)
        REAL            OBSN

   15   FORMAT (14(I5))
   25   FORMAT (I5)

        DO 20, J = 1, 2
           DO 40, I = 1, NY
              COUNT (J, I) = 0
   40      CONTINUE
   20   CONTINUE

        OPEN (UNIT=8,FILE='\WATER\DATA\CLIMA.DAT',STATUS='OLD')

        DO 10, J = 1, NY
           DO 50, I = 1, NT
              READ (8, *) OBSN
              IF (OBSN .EQ. 0) THEN
                 COUNT (1, J) = COUNT (1, J) + 1
                 SEQ (1, J, COUNT (1, J)) = I
              ELSEIF (OBSN .GT. 0) THEN
                 COUNT (2, J) = COUNT (2, J) + 1
                 SEQ (2, J, COUNT (2, J)) = I
              ENDIF
   50      CONTINUE
   10   CONTINUE

        OPEN (UNIT=10,FILE='\WATER\DATA\SEQ2.DAT',STATUS='UNKNOWN')

        DO 60, I = 1, NY
           DO 30, J = 1, 2
              WRITE (10, 25) COUNT (J, I)
              WRITE (10, 15) (SEQ (J, I, K), K = 1, COUNT (J, I))
   30      CONTINUE
   60   CONTINUE

        STOP
        END
```

PROGRAM 6

```
C         --------------------------------------
C         ...... PROGRAM TO COMPUTE MEAN VECTOR
C         --------------------------------------

          INTEGER           NV,NY,NT,NPARM
          PARAMETER         (NV=7)
          PARAMETER         (NY=12)
          PARAMETER         (NT=365)
          PARAMETER         (NPARM=11)
          INTEGER           SEQ (2,NY,NT)
          INTEGER           COUNT (2,NY)
          INTEGER           DENOM (NT)
          REAL              MU (2,NT)
          REAL              PHI (NT,NPARM)
          REAL              THETA,OMEGA,PI
          PARAMETER         (PI = 3.14159265)
          DIMENSION         CLIMA[HUGE] (NY,NT)

    5     FORMAT (11F6.2)
   15     FORMAT (14 (I5))
   25     FORMAT (I5)
   35     FORMAT (18X,F9.2)
   45     FORMAT (27X,F9.2)
   55     FORMAT (36X,F9.2)
   65     FORMAT (45X,F10.2)
   75     FORMAT (55X,F10.2)
   85     FORMAT (65X,F9.2)
   95     FORMAT (9X,F9.2)

          OPEN (UNIT=30,FILE='\WATER\DATA\MEAND.DAT',STATUS='UNKNOWN')
          OPEN (UNIT=20,FILE='\WATER\DATA\PHID.DAT',STATUS='UNKNOWN')
          OPEN (UNIT=32,FILE='\WATER\DATA\MEANW.DAT',STATUS='UNKNOWN')
          OPEN (UNIT=22,FILE='\WATER\DATA\PHIW.DAT',STATUS='UNKNOWN')
          OPEN (UNIT=24,FILE='\WATER\DATA\SEQ2.DAT',STATUS='OLD')
          OPEN (UNIT=18,FILE='\WATER\DATA\CLIMA.DAT',STATUS='OLD')

C         ...... INPUT SEQ OF DRY & WET DAYS

          DO 50, I = 1, NY
             DO 60, J = 1, 2
                READ (24, 25) COUNT (J, I)
                READ (24, 15) (SEQ (J, I, K), K = 1, COUNT (J, I))
   60        CONTINUE
   50     CONTINUE

          DO 30, K = 1, NV
             DO 10, I = 1, NY
                DO 20, J = 1, NT

C         ...... INPUT ONE VARIABLE AT A TIME
```

```
            IF (K .EQ. 1) THEN
               READ (18,95) CLIMA (I, J)
            ELSEIF (K .EQ. 2) THEN
               READ (18,35) CLIMA (I, J)
            ELSEIF (K .EQ. 3) THEN
               READ (18,45) CLIMA (I, J)
            ELSEIF (K .EQ. 4) THEN
               READ (18,55) CLIMA (I, J)
            ELSEIF (K .EQ. 5) THEN
               READ (18,65) CLIMA (I, J)
            ELSEIF (K .EQ. 6) THEN
               READ (18,75) CLIMA (I, J)
            ELSEIF (K .EQ. 7) THEN
               READ (18,85) CLIMA (I, J)
            ENDIF

 20      CONTINUE
 10   CONTINUE

C     ...... COMPUTE MEAN VECTOR FOR WET & DRY DAYS

      DO 310, M = 1, 2
         DO 320, J = 1, NT
            DENOM (J) = 0
            MU (M,J) = -999.0
320      CONTINUE
         DO 330, I = 1, NY
            DO 370, J = 1, COUNT (M,I)
               L = SEQ (M, I, J)
               IF (CLIMA (I,L) .NE. -999) THEN
                  IF (MU (M,L) .LE. -900) THEN
                     MU (M,L) = 0.0
                  ENDIF
                  MU (M,L) = MU (M,L) + CLIMA (I,L)
                  DENOM (L) = DENOM (L) + 1
               ENDIF
370         CONTINUE
330      CONTINUE
         DO 380, J = 1, NT
            IF (MU (M,J) .NE. -999) THEN
               MU (M,J) = MU (M,J) / DENOM (J)
            ENDIF
380      CONTINUE
310   CONTINUE

      DO 130, M = 1, 2

      OMEGA = 2 * PI / NT
      KK = (NPARM - 1) / 2
      DO 510, T = 1, NT
         PHI (T,1) = 1
510   CONTINUE
      DO 520, J = 1, KK
         J1 = 2 * J
```

```
               J2 = J1 + 1
               THETA = OMEGA * J
               A = 2 * COS (THETA)
               PHI (1,J1) = 1
               PHI (2,J1) = A / 2
               PHI (1,J2) = 0
               PHI (2,J2) = SIN (THETA)
               DO 530, T = 3, NT
                  PHI (T,J1) = A * PHI (T-1,J1) - PHI (T-2,J1)
                  PHI (T,J2) = A * PHI (T-1,J2) - PHI (T-2,J2)
 530           CONTINUE
 520        CONTINUE

C       ...... SHRINK MEAN & FOURIER VECTOR BY OMITTING MISSING OBSNS.

               NC = 0
               DO 120, I = 1, NT
                  IF (MU (M,I) .NE. -999) THEN
                     MU (M,I-NC) = MU (M,I)
                     DO 140, L = 1, NPARM
                        PHI (I-NC,L) = PHI (I,L)
  140                CONTINUE
                  ELSE
                     NC = NC + 1
                  ENDIF
  120          CONTINUE

C       ...... OUTPUT OF MEAN & FOURIER VECTORS FOR DRY & WET DAYS

               IF (M .EQ. 1) THEN
                  PRINT *, 'NO. OBNS ON DRY DAYS: ', NT - NC
                  DO 80, I = 1, NT - NC
                     WRITE (30, *) MU (M,I)
                     WRITE (20, 5) (PHI (I,L), L = 1, NPARM)
   80             CONTINUE
               ELSE
                  PRINT *, 'NO. OBNS ON WET DAYS: ', NT - NC
                  DO 40, I = 1, NT - NC
                     WRITE (32, *) MU (M,I)
                     WRITE (22, 5) (PHI (I,L), L = 1, NPARM)
   40             CONTINUE
               ENDIF
  130       CONTINUE
            REWIND 18
   30    CONTINUE

         STOP
         END
```

PROGRAM 7

```
C       ---------------------------------------------------------
C       ...... PROGRAM TO ESTIMATE PARAMETERS FOR THE MEAN AND
C              STANDARD DEVIATION FUNCTIONS
C              --> NB THIS PROGRAM IS DESIGNED TO ESTIMATE
                   PARAMETERS FOR A DRY SEQUENCE. IT CAN ALSO
                   BE USED FOR WET SEQUENCES BY READING THE
                   APPROPRIATE INPUT DATA FILES
C       ---------------------------------------------------------

        INTEGER          NT,NV,NPARM,PI
        PARAMETER        (NT=365)
        PARAMETER        (NV=7)
        PARAMETER        (NPARM=11)
        PARAMETER        (PI=3.141593)
        REAL             LLK
        REAL             MU (NT,1)
        REAL             MEAN (NT)
        REAL             PHI (NT,NPARM)
        REAL             TRSP (NPARM,NT)
        REAL             SOLN (NPARM,NPARM)
        REAL             RESULT (NPARM,NT)
        REAL             BETA (NPARM,1)

 15     FORMAT (/,/)
 25     FORMAT (F10.3)
 35     FORMAT (A13,I4,A11)

        OPEN (UNIT=9,FILE='LPT1')
        OPEN (UNIT=6,FILE='CON')
        OPEN (UNIT=12,FILE='\WATER\DATA\MEAND.DAT',STATUS='OLD')
        OPEN (UNIT=14,FILE='\WATER\DATA\PHID.DAT',STATUS='OLD')

        WRITE (9,*) 'INITIAL ESTIMATES FOR MEAN (DRY) FUNCTION'
        WRITE (9,15)

        WRITE (6,*) ' MAXIMUM NUMBER OF PARAMETERS TO BE FITTED'
        READ (6,*) NPT

        DO 30, K = 1, NV
           CRITO = 10**10
           DO 20, I = 1, NT
              READ (12,*) MU (I,1)
              READ (14,*) (PHI (I,L), L =1, NPT)
 20        CONTINUE

           DO 100, NP = 1, NPT, 2
              CALL TRNSP (PHI,NP,NT,TRSP,NPARM,NT)
              CALL XNP (TRSP,PHI,SOLN,NP,NT,NT,NP,NPARM,NT,NT,NPARM)
              CALL INVNP (NP,SOLN,NPARM)
              CALL XNP (SOLN,TRSP,RESULT,NP,NP,NP,NT,NPARM,NPARM,
     &                 NPARM,NT)
```

```
          CALL XNP (RESULT,MU,BETA,NP,NT,NT,1,NPARM,NT,NT,1)
C       ...... OUTPUT OF PARAMETER ESTIMATES
          WRITE (9,*) 'BETA ESTIMATES FOR VARIABLE: ', K
          DO 10, I = 1, NP
             WRITE (9,*) BETA (I,1)
 10       CONTINUE

          DO 50, I = 1, NT
             MEAN (I) = 0.0
             DO 60, L = 1, NP
                MEAN (I) = MEAN (I) + BETA(L,1) * PHI(I,L)
 60          CONTINUE
 50       CONTINUE
          LLK = 0
          DO 40, I = 1, NT
             LLK = LLK + (MU(I,1) - MEAN (I))**2
 40       CONTINUE
          LLK = -LLK/2 - (NT/2) * LOG (2*PI)
          CRIT = -LLK + NP
          WRITE (9,*) ' AKAIKE"S INFO CRITERION FOR VARIABLE ',K
          WRITE (9,*) CRIT
          WRITE (9,35) ' WHEN FITTING' ,NP, ' PARAMETERS'
          WRITE (9,15)
          IF (CRIT .LT. CRITO) THEN
             LL = NP
             CRITO = CRIT
          ELSE
             GOTO 200
          ENDIF
 100    CONTINUE
 200    WRITE (9,*) ' NUMBER OF PARAMETERS CHOSEN: ', LL
 30   CONTINUE

      STOP
      END
```

PROGRAM 8

```
C        --------------------------------------------------------
C        ...... PROGRAM TO COMPUTE THE STANDARD DEVIATION VECTOR
C        --------------------------------------------------------

         INTEGER              NV,NY,NT,NPARM,NP
         PARAMETER            (NV=7)
         PARAMETER            (NY=12)
         PARAMETER            (NT=365)
         PARAMETER            (NPARM=3)
         PARAMETER            (NP=11)
         INTEGER              SEQ (2,NY,NT)
         INTEGER              COUNT (2,NY)
         REAL                 MU (2,NT)
         REAL                 SIGMA (2,NT)
         REAL                 PHI (NT,NP)
         REAL                 ALPHA (NV,2,NPARM)
         DIMENSION            CLIMA[HUGE] (NY,NT)
         REAL                 DENOM (NT)

   5     FORMAT (11F6.2)
  15     FORMAT (14(I5))
  25     FORMAT (I5)
  35     FORMAT (18X,F9.2)
  45     FORMAT (27X,F9.2)
  55     FORMAT (36X,F9.2)
  65     FORMAT (45X,F10.2)
  75     FORMAT (55X,F10.2)
  85     FORMAT (65X,F9.2)
  95     FORMAT (9X,F9.2)

         OPEN (UNIT=40,FILE='\WATER\DATA\SIGMAD.DAT',STATUS='UNKNOWN')
         OPEN (UNIT=20,FILE='\WATER\DATA\PHD.DAT',STATUS='UNKNOWN')
         OPEN (UNIT=42,FILE='\WATER\DATA\SIGMAW.DAT',STATUS='UNKNOWN')
         OPEN (UNIT=22,FILE='\WATER\DATA\PHW.DAT',STATUS='UNKNOWN')
         OPEN (UNIT=24,FILE='\WATER\DATA\SEQ2.DAT',STATUS='OLD')
         OPEN (UNIT=18,FILE='\WATER\DATA\CLIMA.DAT',STATUS='OLD')
         OPEN (UNIT=10,FILE='\WATER\DATA\EST-M.DAT',STATUS='OLD')

C        ...... INPUT OF PARAMETER ESTIMATES FOR MEAN FUNCTION

         DO 170, K = 1, NV
            DO 40, M= 1, 2
               READ (10,*) (ALPHA (K,M,I), I = 1, NPARM)
  40        CONTINUE
 170     CONTINUE

C        ...... INPUT SEQ OF DRY & WET DAYS

         DO 50, I = 1, NY
            DO 60, J = 1, 2
               READ (24,25) COUNT (J,I)
```

```
            READ (24,15) (SEQ (J,I,K), K = 1, COUNT (J,I))
   60    CONTINUE
   50 CONTINUE

      DO 30, K = 1, NV
         DO 10, I = 1, NY
            DO 20, J = 1, NT

C     ...... INPUT ONE VARIABLE AT A TIME

               IF (K .EQ. 1) THEN
                  READ (18,95) CLIMA (I, J)
               ELSEIF (K .EQ. 2) THEN
                  READ (18,35) CLIMA (I, J)
               ELSEIF (K .EQ. 3) THEN
                  READ (18,45) CLIMA (I, J)
               ELSEIF (K .EQ. 4) THEN
                  READ (18,55) CLIMA (I, J)
               ELSEIF (K .EQ. 5) THEN
                  READ (18,65) CLIMA (I, J)
               ELSEIF (K .EQ. 6) THEN
                  READ (18,75) CLIMA (I, J)
               ELSEIF (K .EQ. 7) THEN
                  READ (18,85) CLIMA (I, J)
               ENDIF

   20       CONTINUE
   10    CONTINUE
         CALL TRIG (PHI,NP,NT)

C     ...... GENERATE MEAN VECTOR

         CALL GMEAN (MU,PHI,NT,NPARM,ALPHA,K,NV)

C     ...... COMPUTE STD DEV VECTOR FOR WET & DRY DAYS

         DO 330, M = 1, 2
            DO 310, I = 1, NT
               DENOM (I) = 0
               SIGMA (M,I) = -999.0
  310       CONTINUE
            DO 320, J = 1, NY
               DO 370, I = 1, COUNT (M,J)
                  L = SEQ (M,J,I)
                  IF (CLIMA (J,L) .NE. -999) THEN
                     IF (SIGMA (M,L) .EQ. -999) THEN
                        SIGMA (M,L) = 0.0
                     ENDIF
                     SIGMA(M,L) = SIGMA(M,L)+(CLIMA(J,L)-MU(M,L))**2
                     DENOM (L) = DENOM (L) + 1
                  ENDIF
  370          CONTINUE
  320       CONTINUE
            DO 380, I = 1, NT
```

```
                IF (SIGMA (M,I) .NE. -999) THEN
                   SIGMA (M,I) = SQRT(SIGMA (M,I) / DENOM (I))
                ENDIF
 380         CONTINUE
 330       CONTINUE

C     ...... SHRINK STD DEV VECTOR & FOURIER VECTOR BY OMITTING MISSING
C              OBSERVATIONS

           DO 130, M = 1, 2
              NC = 0
              DO 120, I = 1, NT
                 IF (SIGMA (M,I) .NE. -999) THEN
                    SIGMA (M,I-NC) = SIGMA (M,I)
                    DO 140, L = 1, NP
                       PHI (I-NC,L) = PHI (I,L)
  140               CONTINUE
                 ELSE
                    NC = NC + 1
                 ENDIF
  120         CONTINUE

C     ...... OUTPUT OF STDE DEV & FOURIER VECTORS FOR WET & DRY DAYS

              IF (M .EQ. 1) THEN
                 PRINT *, 'NO. OBNS ON DRY DAYS: ', NT - NC
                 DO 70, I = 1, NT - NC
                    WRITE (40, *) SIGMA (M,I)
                    WRITE (20, 5) (PHI (I,L), L = 1, NP)
   70            CONTINUE
              ELSE
                 PRINT *, 'NO. OBNS ON WET DAYS: ', NT - NC
                 DO 80, I = 1, NT - NC
                    WRITE (42, *) SIGMA (M,I)
                    WRITE (22, 5) (PHI (I,L), L = 1, NP)
   80            CONTINUE
              ENDIF
              CALL TRIG (PHI,NP,NT)
  130      CONTINUE
           REWIND 18
   30   CONTINUE

        STOP
        END
```

PROGRAM 9

```
C          ------------------------------------------------------
C          ...... PROGRAM TO STANDARDIZE RESIDUAL TIME SERIES
C          ------------------------------------------------------

           INTEGER           NV,NY,NT,NPARM
           PARAMETER         (NV=7)
           PARAMETER         (NY=12)
           PARAMETER         (NT=365)
           PARAMETER         (NPARM=3)
           REAL              CLIMA (0:NV)
           REAL              MU (2,NV,NT)
           REAL              SIGMA (2,NV,NT)
           REAL              PHI (NT,NPARM)
           REAL              PSI (NV,2,NPARM)
           REAL              ALPHA (NV,2,NPARM)

    5      FORMAT (3F 10.6)
   35      FORMAT (5F9.2,2F10.2,F9.2)

           OPEN (UNIT=12,FILE='\WATER\DATA\CLIMAR.DAT',STATUS='UNKNOWN')
           OPEN (UNIT=18,FILE='\WATER\DATA\CLIMA.DAT',STATUS='OLD')
           OPEN (UNIT=10,FILE='\WATER\DATA\EST-M.DAT',STATUS='OLD')
           OPEN (UNIT=40,FILE='\WATER\DATA\EST-S.DAT',STATUS='OLD')

C          ...... INPUT OF PARAMETER ESTIMATES FOR MEAN AND STANDARD
C                 DEVIATION FUNCTION

           DO 10, K = 1, NV
              DO 20, M= 1, 2
                 READ (10,*) (ALPHA (K,M,I), I = 1, NPARM)
   20         CONTINUE
   10      CONTINUE

           DO 60, K = 1, NV
              DO 70, M= 1, 2
                 READ (40,*) (PSI (K,M,I), I = 1, NPARM)
   70         CONTINUE
   60      CONTINUE

              CALL TRIG (PHI,NPARM,NT)

C          ...... GENERATE MEAN AND STANDARD DEVIATION VECTORS

              CALL GAVSTD (MU,PHI,NT,NPARM,ALPHA,NV,PSI,SIGMA)

              DO 30, I = 1, NY
                 DO 40, J = 1, NT
                    READ (18,35) (CLIMA (K), K = 0, NV)
                    IF (CLIMA(0) .EQ. 0) THEN
                       M = 1
                    ELSE
```

```
                    M = 2
                 ENDIF
                 DO 50 , K = 1, NV
                    IF (CLIMA (K) .NE. -999) THEN
                       CLIMA(K)=(CLIMA(K)-MU(M,K,J))/SIGMA(M,K,J)
                    ENDIF
   50            CONTINUE

C        ...... OUTPUT OF STANDARDIZED TIME SERIES

                 WRITE (12,35) (CLIMA (K), K = 0, NV)
   40          CONTINUE
   30        CONTINUE

         STOP
         END
```

PROGRAM 10

```
C        ---------------------------------------------------------
C        ...... PROGRAM TO COMPUTE CROSS-CORRELATION COEFFICIENTS
C               FOR LAGO AND LAG1.
C        ---------------------------------------------------------

         INTEGER          NY, NT, NV
         PARAMETER        (NY=12)
         PARAMETER        (NT=365)
         PARAMETER        (NV=7)
         INTEGER          DENOM(0:1)
         REAL             CLIMA(NV,NY*NT)
         REAL             CROSS(0:1)
         REAL             AVEG(NV),DEV(NV)
         REAL             CLAGO(NV,NV)
         REAL             CLAG1(NV,NV)

   25    FORMAT (9F8.4)
   35    FORMAT (9X,4F9.2,2F10.2,F9.2)
   45    FORMAT (7F9.3)

         OPEN (UNIT=9,FILE='LPT1')
         OPEN (UNIT=10,FILE='\WATER\DATA\CLIMAR.DAT',STATUS='OLD')
         OPEN (UNIT=20,FILE='\WATER\DATA\LAGO.DAT',STATUS='UNKNOWN')
         OPEN (UNIT=30,FILE='\WATER\DATA\LAG1.DAT',STATUS='UNKNOWN')
         OPEN (UNIT=4,FILE='CON')
```

```
      NTIME = NY * NT
      DO 10, I = 1, NTIME
         READ (10,35) (CLIMA (K,I), K = 1, NV)
10    CONTINUE

      CALL AVSTD3 (CLIMA,AVEG,DEV,NY,NT,NV)

      DO 20, K = 1, NV
         DO 30, KK = 1, NV
            DO 40, I = 0, 1
               CROSS (I) = 0.0
               DENOM (I) = 0.0
40          CONTINUE

            DO 50, I = 0, 1
               DO 60, J = 1, NTIME-I
                  IF ((CLIMA(K,J).GT.-900).AND.(CLIMA(KK,J+I)
     &               .GT.-900)) THEN
                     CROSS(I)=CROSS(I)+((CLIMA(K,J)-AVEG(K))*
     &                        (CLIMA(KK,J+I)-AVEG(KK)))
                     DENOM(I) = DENOM(I) + 1
                  ENDIF
60             CONTINUE
               IF (DENOM(I).GT.0) THEN
                  CROSS(I)=(CROSS(I)/DENOM(I))/(DEV(K)*DEV(KK))
               ENDIF
50          CONTINUE

            CLAGO(K,KK) = CROSS(0)
            CLAG1(K,KK) = CROSS(1)

30       CONTINUE
20    CONTINUE

      DO 90, K = 1, NV
         WRITE (20,45) (CLAGO(K,KK), KK = 1, NV)
         WRITE (30,45) (CLAG1(K,KK), KK = 1, NV)
90    CONTINUE

      STOP
      END
```

PROGRAM 11

```
C       ------------------------------------------------------------
C       ...... PROGRAM TO COMPUTE THE MATRICES A & B FOR MODEL1
C       ------------------------------------------------------------

        INTEGER         NV
        PARAMETER       (NV=7)
        REAL            CLAGO(NV,NV)
        REAL            CLAG1(NV,NV)
        REAL            A(NV,NV)
        REAL            B(NV,NV)
        REAL            INV(NV,NV)
        REAL            TRSP(NV,NV)
        REAL            TERM(NV,NV)

 15     FORMAT (7F9.3)

        OPEN (UNIT=9,FILE='LPT1')
        OPEN (UNIT=20,FILE='\WATER\DATA\LAGO.DAT',STATUS='OLD')
        OPEN (UNIT=30,FILE='\WATER\DATA\LAG1.DAT',STATUS='OLD')
        OPEN (UNIT=40,FILE='\WATER\DATA\A.DAT',STATUS='UNKNOWN')
        OPEN (UNIT=50,FILE='\WATER\DATA\B.DAT',STATUS='UNKNOWN')

        DO 10, K = 1, NV
           READ (20,15) (CLAGO (K,KK), KK = 1, NV)
 10     CONTINUE
        DO 20, K = 1, NV
           READ (30,15) (CLAG1 (K,KK), KK = 1, NV)
 20     CONTINUE

        CALL INVT (CLAGO,INV,NV)
        CALL MULT (CLAG1,INV,A,NV,NV,NV,NV)

        DO 30, K = 1, NV
           WRITE (40,15) (A(K,KK), KK = 1, NV)
 30     CONTINUE

        CALL TRANSP (CLAG1,NV,NV,TRSP)
        CALL MULT (A,TRSP,TERM,NV,NV,NV,NV)
        CALL SUBTR (CLAGO,TERM,NV)
        CALL CHOLKY (B,TERM,NV)

        DO 40, K = 1, NV
           WRITE (50,15) (B(K,KK), KK = 1, NV)
 40     CONTINUE

        STOP
        END
```

PROGRAM 12

```
C        ---------------------------------------------------------------
C        ...... PROGRAM TO GENERATE CLIMATE SEQUENCES ACCORDING TO
C               MODEL 1.
C        ---------------------------------------------------------------


C        ...... INTEGER VARIABLES

         INTEGER          NT,NV,NY,NP,PSTATE,STATE

C        ...... PSTATE = PRESENT STATE OF DAY
C        ...... STATE = PREVIOUS STATE OF DAY

C        ...... PARAMETER STATEMENTS

         PARAMETER        (NT=365)
         PARAMETER        (NY=51)
         PARAMETER        (NV=7)
         PARAMETER        (NP=3)

C        ...... NT = £ OBSERVATIONS PER YEAR
C        ...... NV = £ VARIABLES
C        ...... NY = £ YEARS TO BE GENERATED
C        ...... NP = £ PARAMETERS IN SEASONAL MODEL

         INTEGER          SEED (9)
         REAL             RAIN
         REAL             GAM (2,NP)
         REAL             PHI (NP,0:NT)
         REAL             RAND (NV,1)
         REAL             SIGMA  (2,NV,0:NT)
         REAL             MU (2,NV,0:NT)
         REAL             OBSN (NV,1)
         REAL             RES (NV,1)
         REAL             TEMP (NV)
         REAL             AMP (0:NP)
         REAL             PHASE (NP)
         REAL             A (NV,NV)
         REAL             B (NV,NV)
         REAL             C (NT)
         COMMON           IDUM1,IDUM2,IDUM3,IDUM4,IDUM5,IDUM6,IDUM7

  15     FORMAT (5F9.2, 2F10.2, F9.2)
  25     FORMAT (' GIVE 9 -VE Nos. TO INITIALIZE RANDOM GENERATOR',/)

         OPEN (UNIT=9,FILE='LPT1')
         OPEN (UNIT=10,FILE='\WATER\DATA\SIMU.DAT',STATUS='UNKNOWN')
         OPEN (UNIT=22,FILE='CON')

C        ...... COMPUTE THE FOURIER SERIES TERMS
```

```
      CALL COSSIN (PHI,NP,NT)
      DO 60, I = 1, NP
         PHI (I,0) = PHI (I,NT)
60    CONTINUE

      PI=3.14159
      SMAX=135
      SMIN=110
      AVE=(SMAX+SMIN)/2
      AMPS=SMAX-SMIN
      DO 330, I = 1, NT
         C(I) = AVE+(AMPS/2)*COS((2*PI/NT)*(I+11))
330   CONTINUE

C     ...... READING PARAMETER ESTIMATES

      CALL DATA1 (GAM,MU,SIGMA,NP,NV,AMP,PHASE,CV,PHI,A,B,NT)

C     ...... INPUT SEEDS TO START RANDOM NUMBER GENERATOR.  MUST BE
C            NEGATIVE NUMBER.

      PRINT 25
      DO 50, II = 1, 9
         READ (22, *) SEED (II)
50    CONTINUE
      IDUM1 = SEED (1)
      IDUM2 = SEED (2)
      IDUM3 = SEED (3)
      IDUM4 = SEED (4)
      IDUM5 = SEED (5)
      IDUM6 = SEED (6)
      IDUM7 = SEED (7)
      IDUM8 = SEED (8)
      IDUM9 = SEED (9)

C     ...... COMPUTE PARAMETERS NEEDED FOR COMPUTATION OF RAINFALL
C            DEPTH

      CALL CALBET (BETA,CV)
      ALPH = 1 + 1 / BETA
      GAMM = GAMMA (ALPH)
      BI = 1 /BETA
      W = 0.01721421

C     ...... SET INITIAL STATE OF DAY TO BE DRY
C     ...... SET INITIAL CLIMATE VALUE TO IT'S MEAN AT TIME ZERO

C     ...... STATE = 1 ==> DRY
C     ...... STATE = 2 ==> WET

      STATE = 1
      DO 10, I = 1, NV
         OBSN (I,1) = 0.0
10    CONTINUE
```

```
      DO 30, I = 1, NY
         DO 40, J = 1, NT

C     ...... GENERATE RAINFALL VALUE
C            ----------------------

C     ...... COMPUTE PROBABILITY THAT A WET DAY FOLLOWS A WET DAY, OR
C            THE PROBABILITY THAT A WET DAY FOLLOWS A DRY DAY.

            CALL PIEST (NP,GAM,STATE,J,PHI,PI,NT)

C     ...... GENERATE A UNIFORM RANDOM NUMBER BETWEEN 0 AND 1.

            UNIFOR = URAN8 (IDUM8)
            IF (UNIFOR .LT. PI) THEN
               PSTATE = 2
            ELSE
               PSTATE = 1
            ENDIF

C     ...... GENERATE A NORMAL RANDOM NUMBER

            CALL GRAND2 (RAND,NV)

C     ...... GENERATE CLIMATE SEQUENCES
C            -------------------------

            CALL MODEL1 (RAND,NV,SIGMA,MU,J,OBSN,PSTATE,NT,A,B,RES)

C     ...... DETERMINE WHETHER IT RAINED AND SET RAIN VALUE

C     ...... RAIN = 0  ==> DID NOT RAIN
C     ...... RAIN = 1  ==> RAINED

            IF (PSTATE .EQ. 1) THEN
               RAIN = 0
            ELSE
               RAIN = 1
            ENDIF

C     ...... GENERATE RAINFALL DEPTH IF IT RAINED
C            ------------------------------------

            IF (RAIN .EQ. 1) THEN
               CALL DEPTH3 (IDUM9,NP,RAIN,J,AMP,PHASE,GAMM,BI,W)
            ENDIF

C     ...... TRANSFORM VARIABLES TO THE ORIGINAL UNITS

            TEMP(2)=(230-100*EXP(RES(2,1)))/(EXP(RES(2,1))+1)
            TEMP(1)=(410+TEMP(2)*EXP(RES(1,1)))/(EXP(RES(1,1))+1)
            TEMP(3)=(C(J)-0.01-(0.01*EXP(RES(3,1))))/(EXP(RES(3,1))+1)
            TEMP(4)=(10000/(EXP(RES(4,1))+1))-0.01
```

```
            TEMP(5)=101/(EXP(RES(5,1))+1)
            TEMP(6)=TEMP(5)/(EXP(RES(6,1))+1)

C        ...... OUTPUT GENERATED SEQUENCES

            IF (I.NE.1) THEN
               WRITE (10,15) RAIN, (TEMP (K), K = 1, NV)
            ENDIF

C        ...... UPDATE THE STATE OF THE PREVIOUS DAY

            IF (PSTATE .NE. STATE) THEN
               STATE = PSTATE
            ENDIF

 40        CONTINUE
 30     CONTINUE

        STOP
        END
```

PROGRAM 13

```
C       --------------------------------------------------------
C       ..... PROGRAM TO PREPARE DATA SETS OF POSSIBLE WET/DRY
C             SEQUENCES
C       --------------------------------------------------------

        INTEGER           NY,NT,PREV,RAIN
        PARAMETER         (NY=12)
        PARAMETER         (NT=365)
        INTEGER           SEQ (4,NY,NT)
        INTEGER           COUNT (4,NY)
        REAL              CLIMA

  5     FORMAT (F9.2)
 15     FORMAT (14(I5))
 25     FORMAT (I5)

        PREV = 0
        DO 20, J = 1, 4
           DO 40, I = 1, NY
              COUNT(J,I) = 0
 40        CONTINUE
 20     CONTINUE

        OPEN (UNIT=8,FILE='\WATER\DATA\CLIMA.DAT',STATUS='OLD')
```

```
      DO 10, J = 1, NY
         DO 50, I = 1, NT
            READ (8,5) CLIMA
            IF (CLIMA .EQ. 0) THEN
               RAIN = 0
            ELSEIF (CLIMA .GT. 0) THEN
               RAIN = 1
            ELSEIF (CLIMA .EQ. -999) THEN
               RAIN = 2
            ENDIF
            IF ((RAIN .NE. 2) .AND. (PREV .NE. 2)) THEN
               IF (RAIN .EQ. PREV) THEN
                  IF (RAIN .EQ. 0) THEN
                     COUNT(1,J) = COUNT(1,J) + 1
                     SEQ(1,J,COUNT(1,J)) = I
                  ELSE
                     COUNT(2,J) = COUNT(2,J) + 1
                     SEQ (2,J,COUNT(2,J)) = I
                  ENDIF
               ELSE
                  IF (RAIN .EQ. 0) THEN
                     COUNT(4,J) = COUNT(4,J) + 1
                     SEQ(4,J,COUNT(4,J)) = I
                  ELSE
                     COUNT(3,J) = COUNT(3,J) + 1
                     SEQ (3,J,COUNT(3,J)) = I
                  ENDIF
               ENDIF
            ENDIF
            PREV = RAIN
50       CONTINUE
10    CONTINUE

      OPEN (UNIT=10,FILE='\WATER\DATA\SEQ.DAT',STATUS='UNKNOWN')

      DO 60, I = 1, NY
         DO 30, J = 1, 4
            WRITE (10, 25) COUNT(J,I)
            WRITE (10, 15) (SEQ(J,I,K), K = 1, COUNT(J,I))
30       CONTINUE
60    CONTINUE

      STOP
      END
```

PROGRAM 14

```
C         -----------------------------------------------------------
C         ...... PROGRAM TO COMPUTE THE AUTOCORRELATION COEFFICIENT
C                CONDITIONED ON WET/DRY STATUS OF THE DAY
C         -----------------------------------------------------------

          INTEGER             NY,NV,NT,NPARM,P,PP,NRAU
          PARAMETER           (NRAU=4)
          PARAMETER           (NY=12)
          PARAMETER           (NV=7)
          PARAMETER           (NT=365)
          PARAMETER           (NPARM=3)
          INTEGER             COUNT (NRAU,NY)
          INTEGER             SEQ (NRAU,NY,NT)
          INTEGER             SUM (NRAU)
          REAL                CLIMA (NY,NT)
          REAL                MU (2,NT)
          REAL                PHI (NT,NPARM)
          REAL                ALPHA (NV,2,NPARM)
          REAL                RAU (NRAU)

   5      FORMAT (/)
  25      FORMAT (9X,F9.2)
  35      FORMAT (18X,F9.2)
  45      FORMAT (27X,F9.2)
  55      FORMAT (36X,F9.2)
  65      FORMAT (45X,F10.2)
  75      FORMAT (55X,F10.2)
  85      FORMAT (65X,F9.2)
  95      FORMAT (I5)
 105      FORMAT (14I5)

          OPEN (UNIT=9,FILE='LPT1')
          OPEN (UNIT=18,FILE='\WATER\DATA\CLIMA.DAT',STATUS='OLD')
          OPEN (UNIT=10,FILE='\WATER\DATA\EST-M.DAT',STATUS='OLD')
          OPEN (UNIT=12,FILE='\WATER\DATA\SEQ.DAT',STATUS='OLD')

C         ...... INPUT SEQUENCE OF DRY/WET DAYS

          DO 140, J = 1, NY
             DO 150, I = 1, 4
                READ (12,95) COUNT (I,J)
                READ (12,105) (SEQ (I,J,K), K = 1, COUNT (I,J))
 150         CONTINUE
 140      CONTINUE

C         ...... INPUT OF PARAMETER ESTIMATES FOR THE MEAN FUNCTION

          DO 170, K = 1, NV
             DO 90, M= 1, 2
                READ (10,*) (ALPHA (K,M,I), I = 1, NPARM)
  90         CONTINUE
```

```
 170    CONTINUE

        CALL TRIG (PHI,NPARM,NT)
        DO 220, JJ = 1, NRAU
           DO 230, I = 1, NY
              SUM (JJ) = SUM (JJ) + COUNT (JJ,I)
 230       CONTINUE
 220    CONTINUE

        DO 30, K = 1, NV
           DO 10, I = 1, NY
              DO 20, J = 1, NT

C       ...... INPUT ONE VARIABLE AT A TIME

                 IF (K .EQ. 1) THEN
                    READ (18,25) CLIMA (I,J)
                 ELSEIF (K .EQ. 2) THEN
                    READ (18,35) CLIMA (I,J)
                 ELSEIF (K .EQ. 3) THEN
                    READ (18,45) CLIMA (I,J)
                 ELSEIF (K .EQ. 4) THEN
                    READ (18,55) CLIMA (I,J)
                 ELSEIF (K .EQ. 5) THEN
                    READ (18,65) CLIMA (I,J)
                 ELSEIF (K .EQ. 6) THEN
                    READ (18,75) CLIMA (I,J)
                 ELSEIF (K .EQ. 7) THEN
                    READ (18,85) CLIMA (I,J)
                 ENDIF
  20          CONTINUE
  10       CONTINUE

C       ...... GENERATE MEAN VECTOR

           CALL GMEAN (MU,PHI,NT,NPARM,ALPHA,K,NV)

C       ...... COMPUTE AUTOCORRELATION

           WRITE (9,*) 'INITIAL ESTIMATES FOR RAU OF VARIABLE: ', K

           DO 120, JJ = 1, 4

              NUM = 0
              DENOM = 0
              CNT = 0
              CNT2 = 0

              IF (JJ.EQ. 1) THEN
                 M = 1
                 L = 1
              ELSEIF (JJ .EQ. 2) THEN
                 M = 2
                 L = 2
```

```
               ELSEIF (JJ .EQ. 3) THEN
                  M = 2
                  L = 1
               ELSEIF (JJ .EQ. 4) THEN
                  M = 1
                  L = 2
               ENDIF
               DO 110, I = 1, NY
                  DO 130, J = 1, COUNT (JJ,I)
                     P = SEQ (JJ,I,J)
                     N = 0
                     IF ((P .EQ. 1) .AND. (I .EQ. 1)) THEN
                        GOTO 130
                     ENDIF
                     IF ((P .EQ. 1) .AND. (I .GT. 1)) THEN
                        N = 1
                        PP = 365
                     ELSE
                        PP = SEQ (JJ,I,J) - 1
                        N = 0
                     ENDIF
                     IF ((CLIMA(I,P).NE.-999).AND.(CLIMA(I-N,PP).NE.
     &                  -999)) THEN
                        NUM = NUM+(CLIMA(I,P)-MU(M,P))*(CLIMA(I-N,PP)-
     &                        MU(L,PP))
                     ELSE
                        CNT = CNT + 1
                     ENDIF
                     IF (CLIMA(I-N,PP).NE.-999) THEN
                        DENOM = DENOM+(CLIMA(I-N,PP)-MU(L,PP))**2
                     ELSE
                        CNT2 = CNT2 + 1
                     ENDIF
 130              CONTINUE
 110           CONTINUE

 200           NUM = NUM/(SUM(JJ)-1-CNT)
               DENOM = DENOM/(SUM(JJ)-CNT2)
               RAU (JJ) = NUM / DENOM
               WRITE (9,*) RAU (JJ)
 120        CONTINUE
            WRITE (9,5)
            REWIND 18
  30     CONTINUE

         STOP
         END
```

PROGRAM 15

```
C        ---------------------------------------------------------------
C        ...... PROGRAM TO ESTIMATE INITIAL STANDARD DEVIATION FUNCTION
C               -- MEAN VECTOR GENERATED
C        ---------------------------------------------------------------

         INTEGER          NY,NV,NT,NPARM,P,PP,NRAU
         PARAMETER        (NRAU=4)
         PARAMETER        (NY=12)
         PARAMETER        (NV=7)
         PARAMETER        (NT=365)
         PARAMETER        (NPARM=3)
         INTEGER          COUNT (NRAU,NY)
         INTEGER          SEQ (NRAU,NY,NT)
         INTEGER          SUM (NRAU)
         REAL             SIGMA (NRAU)
         REAL             CLIMA (NY,NT)
         REAL             MU (2,NT)
         REAL             PHI (NT,NPARM)
         REAL             ALPHA (NV,2,NPARM)
         REAL             RAU (NRAU,NV)

  5      FORMAT (/)
 15      FORMAT (F9.2)
 25      FORMAT (9X,F9.2)
 35      FORMAT (18X,F9.2)
 45      FORMAT (27X,F9.2)
 55      FORMAT (36X,F9.2)
 65      FORMAT (45X,F10.2)
 75      FORMAT (55X,F10.2)
 85      FORMAT (65X,F9.2)
 95      FORMAT (I5)
 105     FORMAT (14I5)

         OPEN (UNIT=9,FILE='LPT1')
         OPEN (UNIT=18,FILE='\WATER\DATA\CLIMA.DAT',STATUS='OLD')
         OPEN (UNIT=10,FILE='\WATER\DATA\EST-M.DAT',STATUS='OLD')
         OPEN (UNIT=14,FILE='\WATER\DATA\RAU-M.DAT',STATUS='OLD')
         OPEN (UNIT=12,FILE='\WATER\DATA\SEQ.DAT',STATUS='OLD')

C        ...... INPUT SEQUENCE OF DRY/WET DAYS

         DO 140, J = 1, NY
            DO 150, I = 1, 4
               READ (12,95) COUNT (I,J)
               READ (12,105) (SEQ (I,J,K), K = 1, COUNT (I,J))
 150        CONTINUE
 140     CONTINUE

C        ...... INPUT OF PARAMETER ESTIMATES FOR THE MEAN FUNCTION

         DO 170, K = 1, NV
```

```
      DO 90, M= 1, 2
         READ (10,*) (ALPHA (K,M,I), I = 1, NPARM)
 90   CONTINUE
170   CONTINUE

C     ...... INPUT OF PARAMETER ESTIMATES FOR RAU

      DO 470, K = 1, NV
         READ (14,*) (RAU (I,K), I = 1, NRAU)
470   CONTINUE

      CALL TRIG (PHI,NPARM,NT)
      DO 220, JJ = 1, 4
         DO 230, I = 1, NY
            SUM (JJ) = SUM (JJ) + COUNT (JJ,I)
230      CONTINUE
220   CONTINUE

      DO 30, K = 1, NV
         DO 10, I = 1, NY
            DO 20, J = 1, NT

C     ...... INPUT ONE VARIABLE AT A TIME

               IF (K .EQ. 1) THEN
                  READ (18,25) CLIMA (I, J)
               ELSEIF (K .EQ. 2) THEN
                  READ (18,35) CLIMA (I, J)
               ELSEIF (K .EQ. 3) THEN
                  READ (18,45) CLIMA (I, J)
               ELSEIF (K .EQ. 4) THEN
                  READ (18,55) CLIMA (I, J)
               ELSEIF (K .EQ. 5) THEN
                  READ (18,65) CLIMA (I, J)
               ELSEIF (K .EQ. 6) THEN
                  READ (18,75) CLIMA (I, J)
               ELSEIF (K .EQ. 7) THEN
                  READ (18,85) CLIMA (I, J)
               ENDIF
 20         CONTINUE
 10      CONTINUE

C     ...... GENERATE MEAN VECTOR

         CALL GMEAN (MU,PHI,NT,NPARM,ALPHA,K,NV)

C     ...... COMPUTE STANDARD DEVIATIONS

         WRITE (9,*) 'INITIAL ESTIMATES FOR SIGMA OF VARIABLE: ', K
         DO 120, JJ = 1, 4
            IF (JJ.EQ. 1) THEN
               M = 1
               L = 1
            ELSEIF (JJ .EQ. 2) THEN
```

```
            M = 2
            L = 2
         ELSEIF (JJ .EQ. 3) THEN
            M = 2
            L = 1
         ELSEIF (JJ .EQ. 4) THEN
            M = 1
            L = 2
         ENDIF
         NUM = 0
         CNT = 0
         DO 110, I = 1, NY
            DO 130, J = 1, COUNT (JJ,I)
               P = SEQ (JJ,I,J)
               N = 0
               IF ((P .EQ. 1) .AND. (I .EQ. 1)) THEN
                  CNT = CNT + 1
                  GOTO 130
               ENDIF
               IF ((P .EQ. 1) .AND. (I .GT. 1)) THEN
                  N = 1
                  PP = 365
               ELSE
                  PP = SEQ (JJ,I,J)-1
                  N = 0
               ENDIF
               IF ((CLIMA(I,P).NE.-999).AND.(CLIMA(I-N,PP).NE.
     &            -999)) THEN
                  NUM = NUM+(CLIMA(I,P)-MU(M,P)-RAU(JJ,K)*(CLIMA
     &                  (I-N,PP)-MU(L,PP)))**2
               ELSE
                  CNT = CNT + 1
               ENDIF
130         CONTINUE
110      CONTINUE

200      SIGMA (JJ) = SQRT(NUM/(SUM(JJ)-1-CNT))
         WRITE (9,*) SIGMA (JJ)
120   CONTINUE
      WRITE (9,5)
      REWIND 18
 30   CONTINUE

      STOP
      END
```

PROGRAM 16

```
C       ----------------------------------------------------------------
C       ...... PROGRAM TO COMPUTE PARAMETER ESTIMATES FOR MODEL 3
C       ----------------------------------------------------------------

        INTEGER         NV,NY,NT,NP,NPARM,NRAU,CONVG,T
        PARAMETER       (NV=7)
        PARAMETER       (NY=12)
        PARAMETER       (NT=365)
        PARAMETER       (NP=14)
        PARAMETER       (NPARM=3)
        PARAMETER       (NRAU=4)
        INTEGER         COUNT (4,NY)
        INTEGER         SEQ (4,NY,NT)
        REAL            RESID (NV,NY,NT)
        REAL            MU (2,0:NT)
        REAL            LNLIKE,AKAIKE,PI
        PARAMETER       (PI=3.141593)
        REAL            CLIMA (NY,0:NT)
        REAL            ALPHA (2,NV,NPARM)
        REAL            SIGMA (NRAU,NV)
        REAL            DER (NP)
        REAL            DER2 (NP,NP)
        REAL            PHI (NPARM,0:NT)
        REAL            RAU (NRAU,NV)
        REAL            THETA (NP)
        REAL            A (NP,0:NP)

   5    FORMAT (/)
  15    FORMAT (' ESTIMATES OF MEAN FOR DRY DAYS:', 3F10.4)
  25    FORMAT (' ESTIMATES OF MEAN FOR WET DAYS:', 3F10.4)
  35    FORMAT (' ESTIMATES OF STANDARD DEVIATIONS:', 4F10.4)
  55    FORMAT (' ESTIMATES OF AUTOCORRELATION:', 4F10.4)
  65    FORMAT (7F10.4)
  75    FORMAT (' AKAIKE"S CRITERION FOR VARIABLE:', I4, ' IS:', F30.4)
  85    FORMAT (9X,F9.2)
  95    FORMAT (18X,F9.2)
 105    FORMAT (27X,F9.2)
 115    FORMAT (36X,F9.2)
 125    FORMAT (45X,F10.2)
 135    FORMAT (55X,F10.2)
 145    FORMAT (65X,F9.2)
 155    FORMAT (I5)
 165    FORMAT (14I5)

        OPEN (UNIT=18,FILE='\WATER\DATA\CLIMA.DAT',STATUS='OLD')
        OPEN (UNIT=12,FILE='\WATER\DATA\SEQ.DAT',STATUS='OLD')
        OPEN (UNIT=14,FILE='\WATER\DATA\RESI.DAT',STATUS='UNKNOWN')
        OPEN (UNIT=9,FILE='LPT1')
        OPEN (UNIT=6,FILE='CON')

        CALL COSSIN (PHI,NPARM,NT)
```

```
         DO 10, I = 1, NPARM
            PHI (I,0) = PHI (I,NT)
   10    CONTINUE

C        ...... INPUT OF TIME SERIES AND INITIAL PARAMETER ESTIMATES

         CALL INTAL3 (EPS,MAXITER,ALPHA,SIGMA,RAU,NPARM,NV,NRAU)
         DO 20, K = 1, NV
            CONVG = 1
            DO 130, I = 1, NY
               DO 140, J = 1, NT
                  IF (K.EQ.1) THEN
                     READ (18,85) CLIMA (I, J)
                  ELSEIF (K.EQ.2) THEN
                     READ (18,95) CLIMA (I, J)
                  ELSEIF (K.EQ.3) THEN
                     READ (18,105) CLIMA (I, J)
                  ELSEIF (K.EQ.4) THEN
                     READ (18,115) CLIMA (I, J)
                  ELSEIF (K.EQ.5) THEN
                     READ (18,125) CLIMA (I, J)
                  ELSEIF (K.EQ.6) THEN
                     READ (18,135) CLIMA (I, J)
                  ELSEIF (K.EQ.7) THEN
                     READ (18,145) CLIMA (I, J)
                  ENDIF
  140          CONTINUE
               IF (I.EQ.1) THEN
                  CLIMA (I,0) = CLIMA (I,1) - 0.5
               ELSEIF (I.NE.1) THEN
                  CLIMA(I,0) = CLIMA(I-1,NT)
               ENDIF
  130       CONTINUE

            REWIND 18
            IF (K .EQ. 1) THEN
               DO 150, KK = 1, NY
                  DO 160, I = 1, 4
                     READ (12,155) COUNT (I,KK)
                     READ (12,165) (SEQ (I,KK,J), J = 1, COUNT (I,KK))
  160             CONTINUE
  150          CONTINUE
            ENDIF

C        ...... ITERATIVE ESTIMATION OF PARAMETERS

            CALL NEWT3 (ALPHA,SIGMA,RAU,NPARM,MAXITER,NT,NY,CLIMA,SEQ,
     &              COUNT,DER,DER2,PHI,EPS,NP,NV,K,A,THETA,NRAU,CONVG)

C        ...... OUTPUT OF FINAL PARAMETER ESTIMATES

            WRITE (9,5)
            WRITE (9,15) (ALPHA (1,K,L), L = 1, NPARM)
            WRITE (9,25) (ALPHA (2,K,L), L = 1, NPARM)
```

```
            WRITE (9,35) (SIGMA (J,K), J = 1, NRAU)
            WRITE (9,55) (RAU (J,K), J = 1, NRAU)
            WRITE (9,5)

C     ...... COMPUTE RESIDUAL MATRIX

            IF (CONVG.EQ.1) THEN
               DO 30, M = 1, 2
                  DO 40, I = 0, NT
                     MU (M,I) = 0.0
                     DO 50, L = 1, NPARM
                        MU(M,I) = MU(M,I)+ALPHA(M,K,L)*PHI(L,I)
 50                  CONTINUE
 40               CONTINUE
 30            CONTINUE

               DO 60, I = 1, NY
                  DO 70, J = 1, NT
                     RESID (K,I,J) = -999.00
 70               CONTINUE
 60            CONTINUE
               LNLIKE = 0
               TERM = 0

               DO 80, J = 1, 4
                  IF (J .EQ. 1) THEN
                     M = 1
                     L = 1
                  ELSEIF (J .EQ. 2) THEN
                     M = 2
                     L = 2
                  ELSEIF (J .EQ. 3) THEN
                     M = 2
                     L = 1
                  ELSEIF (J .EQ. 4) THEN
                     M = 1
                     L = 2
                  ENDIF
                  DO 90, I = 1, NY
                     DO 100, KK = 1, COUNT (J,I)
                        T = SEQ (J,I,KK)
                        IF ((CLIMA(I,T).NE.-999).AND.(CLIMA(I,T-1)
     &                     .NE.-999)) THEN
                           RESID(K,I,T)=(CLIMA(I,T)-MU(M,T))/SIGMA(J,K)
     &                        -RAU(J,K)*((CLIMA(I,T-1)-MU(L,T-1))/
     &                        SIGMA(J,K))
                           LNLIKE = LNLIKE + (RESID(K,I,T))**2
                        ENDIF
                        TERM = TERM + LOG(SIGMA(J,K))
100                  CONTINUE
 90               CONTINUE
 80            CONTINUE

               LNLIKE = -((NY*NT)/2)*LOG(2*PI)-TERM-LNLIKE/2
```

```
            AKAIKE = -2*LNLIKE+2*NP

            WRITE (9,75) K, AKAIKE
         ENDIF
 20   CONTINUE

      DO 110, I = 1, NY
         DO 120, T = 1, NT
            WRITE (14,65) (RESID (K,I,T), K = 1, NV)
120      CONTINUE
110   CONTINUE

      STOP
      END
```

PROGRAM 17

```
C         ------------------------------------------------------------
C         ...... PROGRAM TO COMPUTE PARAMETER ESTIMATES FOR MODEL 3
C                USING CONJUGATE GRADIENT METHODS IN MULTIDIMENSIONS
C         ------------------------------------------------------------

          INTEGER         NV,NY,NT,NP,NPARM,NRAU,T
          PARAMETER       (NV=7)
          PARAMETER       (NY=12)
          PARAMETER       (NT=365)
          PARAMETER       (NP=14)
          PARAMETER       (NPARM=3)
          PARAMETER       (NRAU=4)
          REAL            THETA (NP)
          REAL            LNLIKE,AKAIKE,PI
          PARAMETER       (PI=3.141593)
          REAL            MU (2,0:NT)
          REAL            RESID (NV,NY,NT)

          COMMON          K,ICOUNT(NRAU,NY),ISEQ(NRAU,NY,NT),CLIMA(NY,0:NT),
     &                    ALPHA(2,NV,NPARM),SIGMA(NRAU,NV),PHI(NPARM,0:NT),
     &                    RAU(NRAU,NV),ISCALE(3,NV)

    5     FORMAT (/)
   15     FORMAT (' ESTIMATES OF MEAN FOR DRY DAYS:', 3F10.4)
   25     FORMAT (' ESTIMATES OF MEAN FOR WET DAYS:', 3F10.4)
   35     FORMAT (' ESTIMATES OF STANDARD DEVIATIONS:', 4F10.4)
   55     FORMAT (' ESTIMATES OF AUTOCORRELATION:', 4F10.4)
   65     FORMAT (' PARAMETER ESTIMATES FOR VARIABLE: ', I4)
   75     FORMAT (' CONVERGE ACHIEVED IN ', I4, ' ITERATIONS')
  115     FORMAT (3X, F5.0)
  125     FORMAT (9X,F9.2)
  135     FORMAT (18X,F9.2)
  145     FORMAT (27X,F9.2)
  155     FORMAT (36X,F9.2)
  165     FORMAT (45X,F10.2)
  175     FORMAT (55X,F10.2)
  185     FORMAT (65X,F9.2)
  195     FORMAT (I5)
  205     FORMAT (14I5)
  215     FORMAT (7F10.4)
  315     FORMAT (' AKAIKE"S CRITERION FOR VARIABLE:', I4, ' IS:', F30.4)

          OPEN (UNIT=14,FILE='\WATER\DATA\RESIT.DAT',STATUS='UNKNOWN')
          OPEN (UNIT=18,FILE='\WATER\DATA\CLIMA.DAT',STATUS='OLD')
          OPEN (UNIT=12,FILE='\WATER\DATA\SEQ.DAT',STATUS='OLD')
          OPEN (UNIT=9,FILE='LPT1')
          OPEN (UNIT=6,FILE='CON')
```

```
      CALL COSSIN (PHI,NPARM,NT)
      DO 10, I = 1, NPARM
         PHI (I,0) = PHI (I,NT)
 10   CONTINUE

C     ...... INPUT OF TIME SERIES AND INITIAL PARAMETER ESTIMATES

      CALL INT3 (ALPHA,SIGMA,RAU,NPARM,NV,NRAU,ISCALE)

      PRINT *, 'WHICH VARIABLE TO BE ESTIMATED?'
      READ (6,*) K

      DO 20, I = 1, NY
         DO 30, J = 1, NT
            IF (K .EQ. 1) THEN
               READ (18,125) CLIMA (I, J)
            ELSEIF (K .EQ. 2) THEN
               READ (18,135) CLIMA (I, J)
            ELSEIF (K .EQ. 3) THEN
               READ (18,145) CLIMA (I, J)
            ELSEIF (K .EQ. 4) THEN
               READ (18,155) CLIMA (I, J)
            ELSEIF (K .EQ. 5) THEN
               READ (18,165) CLIMA (I, J)
            ELSEIF (K .EQ. 6) THEN
               READ (18,175) CLIMA (I, J)
            ELSEIF (K .EQ. 7) THEN
               READ (18,185) CLIMA (I, J)
            ENDIF
 30      CONTINUE
         CLIMA (I,0) = CLIMA (I,1) - 0.5
 20   CONTINUE

      DO 40, KK = 1, NY
         DO 50, I = 1, 4
            READ (12,195) ICOUNT (I,KK)
            READ (12,205) (ISEQ (I,KK,J), J = 1, ICOUNT (I,KK))
 50      CONTINUE
 40   CONTINUE

C     ...... ITERATIVE ESTIMATION OF PARAMETERS

      WRITE (9,65) K

C     ...... TRANSFORMING THE SEPARATE PARAMETER ARRAYS INTO ONE
C            VECTOR

      DO 60, J = 1, NPARM
         THETA (J) = ALPHA (1,K,J)*ISCALE(3,K)
         THETA (J+3) = ALPHA (2,K,J)*ISCALE(3,K)
 60   CONTINUE
      DO 70, J = 1, NRAU
         THETA (J+6) = SIGMA (J,K)*ISCALE(2,K)
         THETA (J+10) = RAU (J,K)*ISCALE(1,K)
```

```
   70      CONTINUE
           CALL POLRIB (THETA,NP,TOL,ITER,FMIN)

C          ...... UPDATE PARAMETER ESTIMATES

           DO 80 J = 1, NPARM
              ALPHA (1,K,J) = THETA (J)/ISCALE(3,K)
              ALPHA (2,K,J) = THETA (J+3)/ISCALE(3,K)
   80      CONTINUE

           DO 90, J = 1, NRAU
              SIGMA (J,K) = THETA (J+6)/ISCALE(2,K)
              RAU (J,K) = THETA (J+10)/ISCALE(1,K)
   90      CONTINUE

           WRITE (9,75) ITER

C          ...... OUTPUT OF FINAL PARAMETER ESTIMATES

              WRITE (9,5)
              WRITE (9,15) (ALPHA (1,K,L), L = 1, NPARM)
              WRITE (9,25) (ALPHA (2,K,L), L = 1, NPARM)
              WRITE (9,35) (SIGMA (J,K), J = 1, NRAU)
              WRITE (9,55) (RAU (J,K), J = 1, NRAU)
              WRITE (9,5)

C          ...... COMPUTE RESIDUAL MATRIX

           DO 100, M = 1, 2
              DO 120, I = 0, NT
                 MU (M,I) = 0.0
                 DO 130, L = 1, NPARM
                    MU (M,I) = MU (M,I) + ALPHA (M,K,L) * PHI (L,I)
  130            CONTINUE
  120         CONTINUE
  100      CONTINUE

           DO 180, I = 1, NY
              DO 190, J = 1, NT
                 RESID (K,I,J) = -999.00
  190         CONTINUE
  180      CONTINUE
           LNLIKE = 0
           TERM = 0

           DO 140, J = 1, 4
              IF (J .EQ. 1) THEN
                 M = 1
                 L = 1
              ELSEIF (J .EQ. 2) THEN
                 M = 2
```

```
            L = 2
         ELSEIF (J .EQ. 3) THEN
            M = 2
            L = 1
         ELSEIF (J .EQ. 4) THEN
            M = 1
            L = 2
         ENDIF
         DO 150, I = 1, NY
            DO 160, KK = 1, ICOUNT (J,I)
               T = ISEQ (J,I,KK)
               IF ((CLIMA(I,T).NE.-999).AND.(CLIMA(I,T-1).NE.-999))
     &            THEN
                 RESID(K,I,T)=(CLIMA(I,T)-MU(M,T))/SIGMA(J,K)-RAU(J,K)
     &                        *((CLIMA(I,T-1)-MU(L,T-1))/SIGMA(J,K))
                 LNLIKE = LNLIKE + (RESID(K,I,T))**2
               ENDIF
               TERM = TERM + LOG(SIGMA(J,K))
160         CONTINUE
150      CONTINUE
140   CONTINUE

      LNLIKE = -((NY*NT)/2)*LOG(2*PI)-TERM-LNLIKE/2
      AKAIKE = -2*LNLIKE+2*NP

      WRITE (9,315) K, AKAIKE

      DO 200, I = 1, NY
         DO 170, T = 1, NT
            WRITE (14,215) (RESID (K,I,T), K = 1, NV)
170      CONTINUE
200   CONTINUE

      STOP
      END
```

PROGRAM 18

```
C        -----------------------------------------------------------
C        ...... PROGRAM TO COMPUTE THE AUTOCORRELATION COEFFICIENT
C               -- FOR UNCONDITIONED DATA SET
C        -----------------------------------------------------------

         INTEGER          NY,NV,NT,NPARM
         PARAMETER        (NY=12)
         PARAMETER        (NV=7)
         PARAMETER        (NT=365)
         PARAMETER        (NPARM=3)
         REAL             RAIN (NY*NT)
         REAL             CLIMA (NY*NT)
         REAL             MU (2,NT)
         REAL             PHI (NT,NPARM)
         REAL             ALPHA (NV,2,NPARM)

  5      FORMAT (/)
 15      FORMAT (F9.2)
 25      FORMAT (9X,F9.2)
 35      FORMAT (18X,F9.2)
 45      FORMAT (27X,F9.2)
 55      FORMAT (36X,F9.2)
 65      FORMAT (45X,F10.2)
 75      FORMAT (55X,F10.2)
 85      FORMAT (65X,F9.2)

         OPEN (UNIT=9,FILE='LPT1')
         OPEN (UNIT=18,FILE='\WATER\DATA\CLIMA.DAT',STATUS='OLD')
         OPEN (UNIT=10,FILE='\WATER\DATA\EST-M.DAT',STATUS='OLD')

C        ...... INPUT OF RAINFALL DATA

         DO 40, J = 1, NY*NT
            READ (18, 15) RAIN(J)
  40     CONTINUE
         REWIND 18

C        ...... INPUT OF PARAMETER ESTIMATES FOR THE MEAN FUNCTION

         DO 170, K = 1, NV
            DO 90, M= 1, 2
               READ (10,*) (ALPHA (K,M,I), I = 1, NPARM)
  90        CONTINUE
 170     CONTINUE

         CALL TRIG (PHI,NPARM,NT)
         DO 30, K = 1, NV
            DO 10, I = 1, NY*NT

C        ...... INPUT ONE VARIABLE AT A TIME
```

```
              IF (K .EQ. 1) THEN
                 READ (18,25) CLIMA (I)
              ELSEIF (K .EQ. 2) THEN
                 READ (18,35) CLIMA (I)
              ELSEIF (K .EQ. 3) THEN
                 READ (18,45) CLIMA (I)
              ELSEIF (K .EQ. 4) THEN
                 READ (18,55) CLIMA (I)
              ELSEIF (K .EQ. 5) THEN
                 READ (18,65) CLIMA (I)
              ELSEIF (K .EQ. 6) THEN
                 READ (18,75) CLIMA (I)
              ELSEIF (K .EQ. 7) THEN
                 READ (18,85) CLIMA (I)
              ENDIF
 10         CONTINUE

C        ...... GENERATE MEAN VECTOR

            CALL GMEAN (MU,PHI,NT,NPARM,ALPHA,K,NV)

C        ...... COMPUTE AUTOCORRELATION

            CNT = 0
            NUM = 0
            DENOM = 0
            COUNT = 0
            COUNT2 = 0
            DO 20, J = 2, NY*NT
               IF (RAIN(J).EQ.0) THEN
                  M = 1
               ELSE
                  M = 2
               ENDIF
               IF (J.GT.NT*(CNT+1)) THEN
                  CNT = CNT + 1
               ENDIF
               I = J-NT*CNT
               IF (I .EQ. 1) THEN
                  II = 365
               ELSE
                  II = I-1
               ENDIF
               IF (RAIN (J-1).EQ.0) THEN
                  L = 1
               ELSE
                  L = 2
               ENDIF
               IF ((CLIMA(J).NE.-999).AND.(CLIMA(J-1).NE.-999)) THEN
                  NUM = NUM+(CLIMA(J)-MU(M,I))*(CLIMA(J-1)-MU(L,II))
               ELSE
                  COUNT = COUNT + 1
               ENDIF
               IF (CLIMA(J-1).NE.-999) THEN
```

```
               DENOM = DENOM+(CLIMA(J-1)-MU(L,II))**2
            ELSE
               COUNT2 = COUNT2 + 1
            ENDIF

20       CONTINUE

         IF (RAIN (NY*NT).EQ.0) THEN
            L = 1
         ELSE
            L = 2
         ENDIF

         IF (CLIMA(NY*NT).NE.-999) THEN
            DENOM = DENOM+(CLIMA(NY*NT)-MU(L,NT))**2
         ELSE
            COUNT2 = COUNT2 + 1
         ENDIF

         NUM = NUM/(NY*NT-1-COUNT)
         DENOM = DENOM/(NY*NT-COUNT2)
         RAU = NUM / DENOM
         WRITE (9,*) 'INITIAL ESTIMATE FOR RAU OF VARIABLE: ', K
         WRITE (9,*) RAU
         WRITE (9,5)
         REWIND 18
 30   CONTINUE

      STOP
      END
```

PROGRAM 19

```
C        ------------------------------------------------------------
C        ...... PROGRAM TO COMPUTE PARAMETER ESTIMATES FOR MODEL 4
C        ------------------------------------------------------------

         INTEGER        NV,NY,NT,NP,NPARM,NRAU,CONVG,T
         PARAMETER      (NV=7)
         PARAMETER      (NY=12)
         PARAMETER      (NT=365)
         PARAMETER      (NP=13)
         PARAMETER      (NPARM=3)
         PARAMETER      (NRAU=1)
         INTEGER        COUNT (4,NY)
         INTEGER        SEQ (4,NY,NT)
         REAL           CLIMA (NY,0:NT)
         REAL           ALPHA (2,NV,NPARM)
         REAL           PSI (2,NV,NPARM)
         REAL           DER (NP)
         REAL           DER2 (NP,NP)
         REAL           PHI (NPARM,0:NT)
         REAL           RAU (NRAU,NV)
         REAL           THETA (NP)
         REAL           A (NP,0:NP)

    5    FORMAT (/)
   15    FORMAT (' ESTIMATES OF MEAN FOR DRY DAYS:', 3F10.4)
   25    FORMAT (' ESTIMATES OF MEAN FOR WET DAYS:', 3F10.4)
   35    FORMAT (' ESTIMATES OF VAR FOR DRY DAYS:', 3F10.4)
   45    FORMAT (' ESTIMATES OF VAR FOR WET DAYS:', 3F10.4)
   55    FORMAT (' ESTIMATE OF AUTOCORRELATION:', F10.4)
   65    FORMAT (9X,F9.2)
   75    FORMAT (18X,F9.2)
   85    FORMAT (27X,F9.2)
   95    FORMAT (36X,F9.2)
  105    FORMAT (45X,F10.2)
  115    FORMAT (55X,F10.2)
  125    FORMAT (65X,F9.2)
  135    FORMAT (I5)
  145    FORMAT (14I5)

         OPEN (UNIT=18,FILE='\WATER\DATA\CLIMA.DAT',STATUS='OLD')
         OPEN (UNIT=12,FILE='\WATER\DATA\SEQ.DAT',STATUS='OLD')
         OPEN (UNIT=9,FILE='LPT1')
         OPEN (UNIT=6,FILE='CON')

         CALL COSSIN (PHI,NPARM,NT)
         DO 10, I = 1, NPARM
            PHI (I,0) = PHI (I,NT)
   10    CONTINUE

C        ...... INPUT OF TIME SERIES AND INITIAL PARAMETER ESTIMATES
```

```
      CALL INITIAL (EPS,MAXITER,ALPHA,PSI,RAU,NPARM,NV,NRAU)

      DO 20, K = 1, NV
         CONVG = 1
         DO 30, I = 1, NY
            DO 40, J = 1, NT
               IF (K.EQ.1) THEN
                  READ (18,65) CLIMA (I, J)
               ELSEIF (K.EQ.2) THEN
                  READ (18,75) CLIMA (I, J)
               ELSEIF (K.EQ.3) THEN
                  READ (18,85) CLIMA (I, J)
               ELSEIF (K.EQ.4) THEN
                  READ (18,95) CLIMA (I, J)
               ELSEIF (K.EQ.5) THEN
                  READ (18,105) CLIMA (I, J)
               ELSEIF (K.EQ.6) THEN
                  READ (18,115) CLIMA (I, J)
               ELSEIF (K.EQ.7) THEN
                  READ (18,125) CLIMA (I, J)
               ENDIF
 40         CONTINUE
            IF (I.EQ.1) THEN
               CLIMA (I,0) = CLIMA (I,1) - 0.5
            ELSEIF (I.NE.1) THEN
               CLIMA(I,0) = CLIMA(I-1,NT)
            ENDIF
 30      CONTINUE

         REWIND 18
         IF (K .EQ. 1) THEN
            DO 50, KK = 1, NY
               DO 60, I = 1, 4
                  READ (12,135) COUNT (I,KK)
                  READ (12,145) (SEQ (I,KK,J), J = 1, COUNT (I,KK))
 60            CONTINUE
 50         CONTINUE
         ENDIF

C     ...... ITERATIVE ESTIMATION OF PARAMETERS

         CALL NEWT4 (ALPHA,PSI,RAU,NPARM,MAXITER,NT,NY,CLIMA,SEQ,
     &               COUNT,DER,DER2,PHI,EPS,NP,NV,K,A,THETA,NRAU,CONVG)

C     ...... OUTPUT OF FINAL PARAMETER ESTIMATES

         WRITE (9,5)
         WRITE (9,15) (ALPHA (1,K,L), L = 1, NPARM)
         WRITE (9,25) (ALPHA (2,K,L), L = 1, NPARM)
         WRITE (9,35) (PSI (1,K,L), L = 1, NPARM)
         WRITE (9,45) (PSI (2,K,L), L = 1, NPARM)
         WRITE (9,55) (RAU (J,K), J = 1, NRAU)
         WRITE (9,5)
```

```
C         ...... COMPUTE RESIDUAL MATRIX

             IF ((CONVG.EQ.1).OR.(K.EQ.7)) THEN
                CALL M4RES (RAU,ALPHA,PSI,PHI,COUNT,SEQ,CLIMA,NT,NY,
     &                      NPARM,NV,K,NRAU,NP,CONVG)
             ENDIF
   20     CONTINUE

          STOP
          END
```

PROGRAM 20

```
C       ----------------------------------------------------------
C       ...... PROGRAM TO COMPUTE PARAMETER ESTIMATES FOR MODEL 4
C              USING CONJUGATE GRADIENT METHODS IN MULTIDIMENSIONS
C       ----------------------------------------------------------

        INTEGER         NV,NY,NT,NP,NPARM,NRAU,T
        PARAMETER       (NV=7)
        PARAMETER       (NY=12)
        PARAMETER       (NT=365)
        PARAMETER       (NP=13)
        PARAMETER       (NPARM=3)
        PARAMETER       (NRAU=1)
        REAL            THETA (NP)
        REAL            AKAIKE,LNLIKE,PI
        PARAMETER       (PI=3.141593)
        REAL            MU (2,0:NT)
        REAL            SIGMA (2,0:NT)
        REAL            RESID (NV,NY,NT)

        COMMON          K,ICOUNT(4,NY),ISEQ(4,NY,NT),CLIMA(NY,0:NT),
     &                  ALPHA(2,NV,NPARM),PSI(2,NV,NPARM),PHI(NPARM,0:NT),
     &                  RAU(NRAU,NV),ISCALE(3,NV)
```

```
  5     FORMAT (/)
 15     FORMAT (' ESTIMATES OF MEAN FOR DRY DAYS:', 3F10.4)
 25     FORMAT (' ESTIMATES OF MEAN FOR WET DAYS:', 3F10.4)
 35     FORMAT (' ESTIMATES OF VAR FOR DRY DAYS:', 3F10.4)
 45     FORMAT (' ESTIMATES OF VAR FOR WET DAYS:', 3F10.4)
 55     FORMAT (' ESTIMATE OF AUTOCORRELATION:', F10.4)
 65     FORMAT (' PARAMETER ESTIMATES FOR VARIABLE: ', I4)
 75     FORMAT (' CONVERGE ACHIEVED IN ', I4, ' ITERATIONS')
115     FORMAT (3X, F5.0)
205     FORMAT (9X,F9.2)
305     FORMAT (18X,F9.2)
405     FORMAT (27X,F9.2)
505     FORMAT (36X,F9.2)
605     FORMAT (45X,F10.2)
705     FORMAT (55X,F10.2)
805     FORMAT (65X,F9.2)
905     FORMAT (I5)
105     FORMAT (14I5)
125     FORMAT (7F10.4)
135     FORMAT (' AKAIKE"S CRITERION FOR VARIABLE:', I4, ' IS:', F30.4)

        OPEN (UNIT=18,FILE='\WATER\DATA\CLIMA.DAT',STATUS='OLD')
        OPEN (UNIT=12,FILE='\WATER\DATA\SEQ.DAT',STATUS='OLD')
        OPEN (UNIT=9,FILE='LPT1')
        OPEN (UNIT=6,FILE='CON')
        OPEN (UNIT=14,FILE='\WATER\DATA\RESIT.DAT',STATUS='UNKNOWN')

         TOL = 0.0000000001
        CALL COSSIN (PHI,NPARM,NT)
        DO 30, I = 1, NPARM
           PHI (I,0) = PHI (I,NT)
 30     CONTINUE

C       ...... INPUT OF TIME SERIES AND INITIAL PARAMETER ESTIMATES

        CALL INT4 (ALPHA,PSI,RAU,NPARM,NV,NRAU,ISCALE)

        PRINT *, 'WHICH VARIABLE TO BE ESTIMATED?'
        READ (6,*) K

        DO 10, I = 1, NY
           DO 220, J = 1, NT
              IF (K .EQ. 1) THEN
                 READ (18,205) CLIMA (I, J)
              ELSEIF (K .EQ. 2) THEN
                 READ (18,305) CLIMA (I, J)
              ELSEIF (K .EQ. 3) THEN
                 READ (18,405) CLIMA (I, J)
              ELSEIF (K .EQ. 4) THEN
                 READ (18,505) CLIMA (I, J)
              ELSEIF (K .EQ. 5) THEN
                 READ (18,605) CLIMA (I, J)
              ELSEIF (K .EQ. 6) THEN
                 READ (18,705) CLIMA (I, J)
              ELSEIF (K .EQ. 7) THEN
                 READ (18,805) CLIMA (I, J)
              ENDIF
220        CONTINUE
           CLIMA (I,0) = CLIMA (I,1) - 0.5
 10     CONTINUE
```

```
      REWIND 18
      DO 440, KK = 1, NY
         DO 330, I = 1, 4
            READ (12,905) ICOUNT (I,KK)
            READ (12,105) (ISEQ (I,KK,J), J = 1, ICOUNT (I,KK))
 330     CONTINUE
 440  CONTINUE

C     ...... ITERATIVE ESTIMATION OF PARAMETERS

      WRITE (9,65) K

C     ...... TRANSFORMING THE SEPARATE PARAMETER ARRAYS INTO ONE
C            VECTOR

      DO 20, J = 1, NPARM
         THETA (J) = ALPHA (1,K,J)*ISCALE(3,K)
         THETA (J+3) = ALPHA (2,K,J)*ISCALE(3,K)
         THETA (J+6) = PSI (1,K,J)*ISCALE(2,K)
         THETA (J+9) = PSI (2,K,J)*ISCALE(2,K)
 20   CONTINUE
      DO 70, J = 1, NRAU
         THETA (J+12) = RAU (J,K)*ISCALE(1,K)
 70   CONTINUE

      CALL POLRIB (THETA,NP,TOL,ITER,FMIN)

C     ...... UPDATE PARAMETER ESTIMATES

      DO 40 J = 1, NPARM
         ALPHA (1,K,J) = THETA (J)/ISCALE(3,K)
         ALPHA (2,K,J) = THETA (J+3)/ISCALE(3,K)
         PSI (1,K,J) = THETA (J+6)/ISCALE(2,K)
         PSI (2,K,J) = THETA (J+9)/ISCALE(2,K)
 40   CONTINUE

      DO 80, J = 1, NRAU
         RAU (J,K) = THETA (J+12)/ISCALE(1,K)
 80   CONTINUE

      WRITE (9,75) ITER

C     ...... OUTPUT OF FINAL PARAMETER ESTIMATES

         WRITE (9,5)
         WRITE (9,15) (ALPHA (1,K,L), L = 1, NPARM)
         WRITE (9,25) (ALPHA (2,K,L), L = 1, NPARM)
         WRITE (9,35) (PSI (1,K,L), L = 1, NPARM)
         WRITE (9,45) (PSI (2,K,L), L = 1, NPARM)
         WRITE (9,55) (RAU (J,K), J = 1, NRAU)
         WRITE (9,5)
```

```
C        ...... COMPUTE RESIDUAL MATRIX

         DO 50, M = 1, 2
            DO 60, I = 0, NT
               MU (M,I) = 0.0
               SIGMA (M,I) = 0.0
               DO 90, L = 1, NPARM
                  MU (M,I) = MU (M,I) + ALPHA (M,K,L) * PHI (L,I)
                  SIGMA (M,I) = SIGMA (M,I) + PSI (M,K,L) * PHI (L,I)
   90          CONTINUE
   60       CONTINUE
   50    CONTINUE

         DO 100, I = 1, NY
            DO 110, J = 1, NT
               RESID (K,I,J) = -999.00
  110       CONTINUE
  100    CONTINUE
         LNLIKE = 0
         TERM = 0
      DO 120  J = 1, 4
         IF (J .EQ. 1) THEN
            M = 1
            L = 1
         ELSEIF (J .EQ. 2) THEN
            M = 2
            L = 2
         ELSEIF (J .EQ. 3) THEN
            M = 2
            L = 1
         ELSEIF (J .EQ. 4) THEN
            M = 1
            L = 2
         ENDIF
         DO 130, I = 1, NY
            DO 140, KK = 1, ICOUNT (J,I)
               T = ISEQ (J,I,KK)
               IF ((CLIMA(I,T).NE.-999).AND.(CLIMA(I,T-1).NE.-999))
   &                THEN
               RESID(K,I,T) = (CLIMA(I,T)-MU(M,T))/SIGMA(M,T)-RAU(1,K)
   &                          *((CLIMA(I,T-1)-MU(L,T-1))/SIGMA(L,T-1))
               LNLIKE = LNLIKE + (RESID(K,I,T))**2
               ENDIF
               TERM = TERM + LOG(SIGMA(M,T))
140         CONTINUE
130      CONTINUE
120   CONTINUE

      LNLIKE = -((NY*NT)/2)*LOG(2*PI)-TERM-LNLIKE/2
      AKAIKE = -2*LNLIKE+2*NP

      WRITE (9,135) K, AKAIKE

      DO 150, I = 1, NY
         DO 160, T = 1, NT
            WRITE (14,125) (RESID (K,I,T), K = 1, NV)
160      CONTINUE
150   CONTINUE

      STOP
      END
```

PROGRAM 21

```
C        --------------------------------------------------------
C        ...... PROGRAM TO COMPUTE PARAMETER ESTIMATES FOR MODEL 5
C        --------------------------------------------------------


C        PROGRAM EST-M5
C        --------------


         INTEGER         NV,NY,NT,NP,NPARM,NRAU,CONVG,T
         PARAMETER       (NV=7)
         PARAMETER       (NY=12)
         PARAMETER       (NT=365)
         PARAMETER       (NP=16)
         PARAMETER       (NPARM=3)
         PARAMETER       (NRAU=4)
         INTEGER         COUNT (4,NY)
         INTEGER         SEQ (4,NY,NT)
         REAL            CLIMA (NY,0:NT)
         REAL            ALPHA (2,NV,NPARM)
         REAL            PSI (2,NV,NPARM)
         REAL            DER (NP)
         REAL            DER2 (NP,NP)
         REAL            PHI (NPARM,0:NT)
         REAL            RAU (NRAU,NV)
         REAL            THETA (NP)
         REAL            A (NP,0:NP)

    5    FORMAT (/)
   15    FORMAT (' ESTIMATES OF MEAN FOR DRY DAYS:', 3F10.4)
   25    FORMAT (' ESTIMATES OF MEAN FOR WET DAYS:', 3F10.4)
   35    FORMAT (' ESTIMATES OF VAR FOR DRY DAYS:', 3F10.4)
   45    FORMAT (' ESTIMATES OF VAR FOR WET DAYS:', 3F10.4)
   55    FORMAT (' ESTIMATE OF AUTOCORRELATION:', 4F10.4)
   75    FORMAT (9X,F9.2)
   85    FORMAT (18X,F9.2)
   95    FORMAT (27X,F9.2)
  105    FORMAT (36X,F9.2)
  115    FORMAT (45X,F10.2)
  125    FORMAT (55X,F10.2)
  135    FORMAT (65X,F9.2)
  145    FORMAT (I5)
  155    FORMAT (14I5)

         OPEN (UNIT=18,FILE='\WATER\DATA\CLIMA.DAT',STATUS='OLD')
         OPEN (UNIT=12,FILE='\WATER\DATA\SEQ.DAT',STATUS='OLD')
         OPEN (UNIT=9,FILE='LPT1')

         CALL COSSIN (PHI,NPARM,NT)
         DO 10, I = 1, NPARM
            PHI (I,0) = PHI (I,NT)
```

```
   10     CONTINUE

C         ...... INPUT OF TIME SERIES AND INITIAL PARAMETER ESTIMATES

          CALL INITIAL (EPS,MAXITER,ALPHA,PSI,RAU,NPARM,NV,NRAU)
          DO 20, K = 1, NV
             CONVG = 1
             DO 30, I = 1, NY
                DO 40, J = 1, NT
                   IF (K.EQ.1) THEN
                      READ (18,75) CLIMA (I, J)
                   ELSEIF (K.EQ.2) THEN
                      READ (18,85) CLIMA (I, J)
                   ELSEIF (K.EQ.3) THEN
                      READ (18,95) CLIMA (I, J)
                   ELSEIF (K.EQ.4) THEN
                      READ (18,105) CLIMA (I, J)
                   ELSEIF (K.EQ.5) THEN
                      READ (18,115) CLIMA (I, J)
                   ELSEIF (K.EQ.6) THEN
                      READ (18,125) CLIMA (I, J)
                   ELSEIF (K.EQ.7) THEN
                      READ (18,135) CLIMA (I, J)
                   ENDIF
   40           CONTINUE
                IF (I.EQ.1) THEN
                   CLIMA (I,0) = CLIMA (I,1) - 0.5
                ELSEIF (I.NE.1) THEN
                   CLIMA(I,0) = CLIMA(I-1,NT)
                ENDIF
   30        CONTINUE

             REWIND 18
             IF (K .EQ. 1) THEN
                DO 50, KK = 1, NY
                   DO 60, I = 1, 4
                      READ (12,145) COUNT (I,KK)
                      READ (12,155) (SEQ (I,KK,J), J = 1, COUNT (I,KK))
   60              CONTINUE
   50           CONTINUE
             ENDIF

C         ...... ITERATIVE ESTIMATION OF PARAMETERS

             CALL NEWT5 (ALPHA,PSI,RAU,NPARM,MAXITER,NT,NY,CLIMA,SEQ,
     &                COUNT,DER,DER2,PHI,EPS,NP,NV,K,A,THETA,NRAU,CONVG)

C         ...... OUTPUT OF FINAL PARAMETER ESTIMATES

             WRITE (9,5)
             WRITE (9,15) (ALPHA (1,K,L), L = 1, NPARM)
             WRITE (9,25) (ALPHA (2,K,L), L = 1, NPARM)
             WRITE (9,35) (PSI (1,K,L), L = 1, NPARM)
             WRITE (9,45) (PSI (2,K,L), L = 1, NPARM)
```

```
            WRITE (9,55) (RAU (J,K), J = 1, NRAU)
            WRITE (9,5)

C        ...... COMPUTE RESIDUAL MATRIX

            IF ((CONVG.EQ.1).OR.(K.EQ.7)) THEN
               CALL M5RES (RAU,ALPHA,PSI,PHI,COUNT,SEQ,CLIMA,NT,NY,
     &                    NPARM,NV,K,NRAU,NP,CONVG)
            ENDIF

   20    CONTINUE

         STOP
         END
```

PROGRAM 22

```
C        ----------------------------------------------------------------
C        ...... PROGRAM TO COMPUTE PARAMETER ESTIMATES FOR MODEL 5
C               USING CONJUGATE GRADIENT METHODS IN MULTIDIMENSIONS
C        ----------------------------------------------------------------


         INTEGER          NV,NY,NT,NP,NPARM,NRAU,T
         PARAMETER        (NV=7)
         PARAMETER        (NY=12)
         PARAMETER        (NT=365)
         PARAMETER        (NP=16)
         PARAMETER        (NPARM=3)
         PARAMETER        (NRAU=4)
         REAL             THETA (NP)
         REAL             LNLIKE,AKAIKE,PI
         PARAMETER        (PI=3.141593)
         REAL             MU (2,0:NT)
         REAL             SIGMA (2,0:NT)
         REAL             RESID (NV,NY,NT)

         COMMON           K,ICOUNT(4,NY),ISEQ(4,NY,NT),CLIMA(NY,0:NT),
     &                    ALPHA(2,NV,NPARM),PSI(2,NV,NPARM),PHI(NPARM,0:NT),
     &                    RAU(NRAU,NV),ISCALE(3,NV)


   5     FORMAT (/)
  15     FORMAT (' ESTIMATES OF MEAN FOR DRY DAYS:', 3F10.4)
  25     FORMAT (' ESTIMATES OF MEAN FOR WET DAYS:', 3F10.4)
  35     FORMAT (' ESTIMATES OF VAR FOR DRY DAYS:', 3F10.4)
  45     FORMAT (' ESTIMATES OF VAR FOR WET DAYS:', 3F10.4)
  55     FORMAT (' ESTIMATES OF AUTOCORRELATION:', 4F10.4)
  65     FORMAT (' PARAMETER ESTIMATES FOR VARIABLE: ', I4)
  75     FORMAT (' CONVERGE ACHIEVED IN ', I4, ' ITERATIONS')
 105     FORMAT (3X, F5.0)
 205     FORMAT (9X,F9.2)
 305      FORMAT (18X,F9.2)
 405      FORMAT (27X,F9.2)
 505      FORMAT (36X,F9.2)
 605      FORMAT (45X,F10.2)
 705      FORMAT (55X,F10.2)
 805      FORMAT (65X,F9.2)
 905      FORMAT (I5)
 115     FORMAT (14I5)
 315     FORMAT (7F10.4)
 415     FORMAT (' AKAIKE"S CRITERION FOR VARIABLE:', I4, ' IS:', F10.4)

         OPEN (UNIT=14,FILE='\WATER\DATA\RESIT.DAT',STATUS='UNKNOWN')
         OPEN (UNIT=18,FILE='\WATER\DATA\CLIMA.DAT',STATUS='OLD')
         OPEN (UNIT=12,FILE='\WATER\DATA\SEQ.DAT',STATUS='OLD')
         OPEN (UNIT=9,FILE='LPT1')
         OPEN (UNIT=6,FILE='CON')
```

```
      TOL = 0.0000000001

      CALL COSSIN (PHI,NPARM,NT)
      DO 10, I = 1, NPARM
         PHI (I,0) = PHI (I,NT)
10    CONTINUE

C     ...... INPUT OF TIME SERIES AND INITIAL PARAMETER ESTIMATES

      CALL INT4 (ALPHA,PSI,RAU,NPARM,NV,NRAU,ISCALE)

      PRINT *, 'WHICH VARIABLE TO BE ESTIMATED?'
      READ (6,*) K


      DO 20, I = 1, NY
         DO 30, J = 1, NT
            IF (K .EQ. 1) THEN
               READ (18,205) CLIMA (I, J)
            ELSEIF (K .EQ. 2) THEN
               READ (18,305) CLIMA (I, J)
            ELSEIF (K .EQ. 3) THEN
               READ (18,405) CLIMA (I, J)
            ELSEIF (K .EQ. 4) THEN
               READ (18,505) CLIMA (I, J)
            ELSEIF (K .EQ. 5) THEN
               READ (18,605) CLIMA (I, J)
            ELSEIF (K .EQ. 6) THEN
               READ (18,705) CLIMA (I, J)
            ELSEIF (K .EQ. 7) THEN
               READ (18,805) CLIMA (I, J)
            ENDIF
30       CONTINUE
         CLIMA (I,0) = CLIMA (I,1) - 0.5
20    CONTINUE

      REWIND 18
      DO 40, KK = 1, NY
         DO 50, I = 1, NRAU
            READ (12,905) ICOUNT (I,KK)
            READ (12,115) (ISEQ (I,KK,J), J = 1, ICOUNT (I,KK))
50       CONTINUE
40    CONTINUE

C     ...... ITERATIVE ESTIMATION OF PARAMETERS

      WRITE (9,65) K

C     ...... TRANSFORMING THE SEPARATE PARAMETER ARRAYS INTO ONE
C            VECTOR

      DO 60, J = 1, NPARM
         THETA (J) = ALPHA (1,K,J)*ISCALE(3,K)
```

```
            THETA (J+3) = ALPHA (2,K,J)*ISCALE(3,K)
            THETA (J+6) = PSI (1,K,J)*ISCALE(2,K)
            THETA (J+9) = PSI (2,K,J)*ISCALE(2,K)
   60    CONTINUE

         DO 70, J = 1, NRAU
            THETA (J+12) = RAU (J,K)*ISCALE(1,K)
   70    CONTINUE

         CALL POLRIB (THETA,NP,TOL,ITER,FMIN)

C        ...... UPDATE PARAMETER ESTIMATES

         DO 80 J = 1, NPARM
            ALPHA (1,K,J) = THETA (J)/ISCALE(3,K)
            ALPHA (2,K,J) = THETA (J+3)/ISCALE(3,K)
            PSI (1,K,J) = THETA (J+6)/ISCALE(2,K)
            PSI (2,K,J) = THETA (J+9)/ISCALE(2,K)
   80    CONTINUE

         DO 90, J = 1, NRAU
            RAU (J,K) = THETA (J+12)/ISCALE(1,K)
   90    CONTINUE

         WRITE (9,75) ITER

C        ...... OUTPUT OF FINAL PARAMETER ESTIMATES

            WRITE (9,5)
            WRITE (9,15) (ALPHA (1,K,L), L = 1, NPARM)
            WRITE (9,25) (ALPHA (2,K,L), L = 1, NPARM)
            WRITE (9,35) (PSI (1,K,L), L = 1, NPARM)
            WRITE (9,45) (PSI (2,K,L), L = 1, NPARM)
            WRITE (9,55) (RAU (J,K), J = 1, NRAU)
            WRITE (9,5)

C        ...... COMPUTE RESIDUAL MATRIX

         DO 100, M = 1, 2
            DO 120, I = 0, NT
               MU (M,I) = 0.0
               SIGMA (M,I) = 0.0
               DO 130, L = 1, NPARM
                  MU (M,I) = MU (M,I) + ALPHA (M,K,L) * PHI (L,I)
                  SIGMA (M,I) = SIGMA (M,I) + PSI (M,K,L) * PHI (L,I)
  130          CONTINUE
  120       CONTINUE
  100    CONTINUE

         DO 140, I = 1, NY
```

```
          DO 150, J = 1, NT
             RESID (K,I,J) = -999.00
150       CONTINUE
140    CONTINUE
       LNLIKE = 0
       TERM = 0

       DO 160, J = 1, 4
          IF (J .EQ. 1) THEN
             M = 1
             L = 1
          ELSEIF (J .EQ. 2) THEN
             M = 2
             L = 2
          ELSEIF (J .EQ. 3) THEN
             M = 2
             L = 1
          ELSEIF (J .EQ. 4) THEN
             M = 1
             L = 2
          ENDIF

          DO 170, I = 1, NY
             DO 180, KK = 1, ICOUNT (J,I)
                T = ISEQ (J,I,KK)
                IF ((CLIMA(I,T).NE.-999).AND.(CLIMA(I,T-1).NE.-999))
    &                THEN
                RESID(K,I,T) = (CLIMA(I,T)-MU(M,T))/SIGMA(M,T)-
    &                          RAU(J,K)*((CLIMA(I,T-1)-MU(L,T-1))/
    &                          SIGMA(L,T-1))
                LNLIKE = LNLIKE + (RESID(K,I,T))**2
                ENDIF
                TERM = TERM + LOG(SIGMA(M,T))
180          CONTINUE
170       CONTINUE
160    CONTINUE
       LNLIKE = -((NY*NT)/2)*LOG(2*PI)-TERM-LNLIKE/2
       AKAIKE = -2*LNLIKE+2*NP
       WRITE (9,415) K, AKAIKE


       DO 190, I = 1, NY
          DO 200, T = 1, NT
             WRITE (14,315) (RESID (K,I,T), K = 1, NV)
200       CONTINUE
190    CONTINUE

       STOP
       END
```

PROGRAM 23

```
C         ----------------------------------------------------------------
C         ...... PROGRAM TO RECORD TIME PERIODS FOR WHICH A MISSING
C                OBSERVATION OCCURS
C         ----------------------------------------------------------------

          INTEGER             NV,TIME,BOUND
          PARAMETER           (NV=7)
          PARAMETER           (TIME=4380)
          PARAMETER           (BOUND=500)
          INTEGER             SEQMISS (NV,BOUND)
          INTEGER             COUNT (NV)
          REAL                CLIMA (NV)

   15     FORMAT (14(I5))
   25     FORMAT (I5)

          OPEN (UNIT=10,FILE='\WATER\DATA\RESIDU.DAT',STATUS='OLD')
          OPEN (UNIT=8,FILE='\WATER\DATA\SEQM.DAT',STATUS='UNKNOWN')
          DO 20, K = 1, NV
             COUNT (K) = 0
   20     CONTINUE
          DO 10, J = 1, TIME
             READ (10, *) (CLIMA (K), K = 1, NV)
             DO 30, K = 1, NV
                IF (CLIMA (K) .LT. -900) THEN
                   COUNT (K) = COUNT (K) + 1
                   SEQMISS (K, COUNT (K)) = J
                ENDIF
   30        CONTINUE
   10     CONTINUE
          DO 60, K = 1, NV
             WRITE (8, 25) COUNT (K)
             WRITE (8, 15) (SEQMISS (K, I), I = 1, COUNT (K))
   60     CONTINUE

          STOP
          END
```

PROGRAM 24

```
C      -----------------------------------------------------------
C      ...... A PROGRAM TO PATCH THE MISSING OBSERVATIONS IN A
C             GIVEN DATA SET USING THE EM-ALGORITHM
C      --------------------------------------------------------------
C
C      In this program there are missing observations in almost all the
c      the variables.
C
C      The variables are read as one big matrix which consists of a
C      column of the dependent variable - which should always be the
C      first column, and the of the columns being the matrix of the
C      independent variables.
C
C      Each row of data represents an observation and where a "-999" is
C      encountered, that would be representing a missing observation.
C
C      The data is stored in a matrix called the Z-matrix, and that is
C      subdivided into :
C                Y-matrix = A matrix of the dependent variable
C                X-matrix = A matrix of the independent variables
C
C      The maximum dimensions of the matrices are:
C                Dependent variable        :  1
C                Independent variables     :  25
C                Observations              :  100
C
C      Note that only one Y-variable can be patched at a time, and
C      therefore we can only have one Y-variable at a time.   If
C      there are missing observations in more than one variable, then
C      it is therefore necessary to swop the variable s columns  so
C      that the variable which needs to be patched is always in the
C      first column of the matrix.
C
C      Note again that most of the routines which are in this program
C      were copied from the programs written by Dr Ross Sparks.

C      ...... VARIABLES DECLARATION

       INTEGER NOBS, NSTAT, IV, DV
       PARAMETER(NOBS=4380)
       PARAMETER(NSTAT=7)
       PARAMETER(IV=12)
       PARAMETER(DV=1)
       PARAMETER(NI=500)

C      ...... NOBS = Number of all the records
C             NSTAT = Number of all the stations i.e. target & control
C             IV = Number of control stations
C             DV = Number of target stations

       REAL Z(NOBS,NSTAT)
```

```
      REAL TMAT (NOBS,NSTAT)
      REAL ZCEN(NOBS,NSTAT), PATCH(NOBS)
      REAL MEANZ(NSTAT,DV), MEANZZ(DV,NSTAT)
      REAL ZTZ(NSTAT,NSTAT)
      REAL TEMP1(NSTAT), TEMPO, TEMP2
      REAL MEAN1(NSTAT), MEAN2(NSTAT)
      REAL BHAT(NSTAT,DV), BETA(7,NI), CONV(DV,DV)
      INTEGER COUNT (NSTAT)
      INTEGER SEQMISS (NSTAT,500)
      INTEGER ROW, COL, ROUND, NROW
      INTEGER NROUND

    1 FORMAT(9F8.0)
    2 FORMAT(20F6.0)

      OPEN (UNIT=9,FILE='LPT1')
      OPEN (UNIT=10,FILE='\WATER\DATA\RESIDU.DAT',STATUS='OLD')
      OPEN (UNIT=20,FILE='\WATER\DATA\SEQM.DAT',STATUS='OLD')

C     ...... This DO-LOOP reads a matrix of all the rainfall
C            stations and all the observations in a MATRIX Z.

      DO 10 ROW = 1, NOBS

         READ(10,*) (Z(ROW,COL), COL = 1, NSTAT)

   10 CONTINUE

C     ...... This DO-LOOP reads a vector of the amount of missing
C            values for each variable in MATRIX COUNT and a matrix of
C            the specific times when missing values occur for each of
C            the variables in a MATRIX SEQMISS.

      DO 20, K = 1, NSTAT
         READ (20,*) COUNT (K)
         READ (20,*) (SEQMISS (K,I), I = 1, COUNT (K))
   20 CONTINUE

C     ...... FIND THE MEANS OF THE DIFFERENT COLS, I.E. FIND THE MEAN
C            OF ALL THE OBSERVATIONS IN COL 1, COL2, ETC.

      DO 110 COL = 1, NSTAT
         MEANZ(COL,1) = 0.0
         MEAN1(COL) = 0.0
         DO 100 ROW = 1, NOBS
            IF (Z(ROW,COL) .NE. -999) THEN
               MEAN1(COL) = MEAN1(COL) + Z(ROW,COL)
            ENDIF
  100    CONTINUE
         MEANZ(COL,1) = MEAN1(COL) / (NOBS - COUNT (COL))
  110 CONTINUE

C     ...... SUBSTITUTE THE MISSING OBSERVATIONS BY THE
             CALCULATED MEANS
```

```
      DO 130 COL = 1, NSTAT
         DO 120 K = 1, COUNT (COL)
            ROW = SEQMISS (COL,K)
            Z(ROW,COL) = MEANZ(COL,1)
 120     CONTINUE
 130  CONTINUE


      NROUND = 1


12121 CALL CNTRAL(ZCEN,NOBS,NSTAT,Z,NOBS,NSTAT,NOBS,NSTAT)
      CALL TMULT(ZCEN,NOBS,NSTAT,ZTZ,NSTAT,NSTAT,NSTAT,NOBS,NSTAT)
      CALL INV(ZTZ,NSTAT,NSTAT)

      DO 167 COL = 1, NSTAT
         MEAN2(COL) = 0.0
         DO 163 K = 1, COUNT (COL)
            ROW = SEQMISS (COL,K)
            MEAN2(COL) = MEAN2(COL) + Z(ROW,COL)
 163     CONTINUE
         MEANZZ(1,COL) = (MEAN1(COL) + MEAN2(COL)) / NOBS
 167  CONTINUE

      ROUND = 1

13131 DO 810 ROW = 1, NSTAT
            TEMP1(ROW) = (-1.0) * ZTZ(ROW,ROUND) / ZTZ(ROUND,ROUND)
 810  CONTINUE

      TEMP2 = 0.0
      DO 830 ROW = 1, NSTAT
         IF (ROW .NE. ROUND) THEN
            TEMP2 = TEMP2 + MEANZZ(1,ROW) * TEMP1(ROW)
         ENDIF
 830  CONTINUE

      TEMPO = MEANZZ(1,ROUND) - TEMP2

      TEMP1(ROUND) = TEMPO

      DO 450 ROW = 1, NSTAT
         BETA(ROW,NROUND) = TEMP1(ROW)
 450  CONTINUE

      IF (NROUND .GT. 1) THEN

         DO 460 ROW = 1, NSTAT
            BHAT(ROW,1) = BETA(ROW,NROUND) - BETA(ROW,NROUND-1)
 460     CONTINUE

         CALL TMULT(BHAT,NSTAT,DV,CONV,DV,DV,DV,NSTAT,DV)
```

```
      ENDIF

C     ...... PATCH THE MISSING OBSERVATIONS

      DO 210 ROW = 1, NOBS
         PATCH (ROW) = Z (ROW,ROUND)
 210  CONTINUE
      DO 200 K = 1, COUNT (ROUND)
         ROW = SEQMISS (ROUND,K)
         PATCH(ROW) = 0.0
         DO 192 COL = 1, NSTAT
            IF (COL .EQ. ROUND) THEN
               GO TO 192
            ENDIF
            PATCH(ROW) = PATCH(ROW) + Z(ROW,COL) * TEMP1(COL)
 192     CONTINUE
         PATCH(ROW) = TEMP1(ROUND) + PATCH(ROW)
 200  CONTINUE

      DO 220, ROW = 1, NOBS
         TMAT (ROW,ROUND) = PATCH (ROW)
 220  CONTINUE

      IF (NROUND .GT. 1) THEN

         IF (CONV(1,1) .LT. 0.0000001) THEN

            WRITE (9,*) 'VALUES PATCHED AFTER ', NROUND, ' ALTERATIONS.'
            CALL PPMAT (BETA,NSTAT,NROUND,NSTAT, NROUND)
            CALL PMAT (TMAT,NOBS,NSTAT,NOBS,NSTAT)
            CALL PPMAT (TEMP1,NSTAT,DV,NSTAT,DV)
            GO TO 998

         ENDIF
      ENDIF

      ROUND = ROUND + 1

      IF (ROUND .GT. NSTAT) THEN
         NROUND = NROUND + 1
         CALL COPY (TMAT,NOBS,NSTAT,Z,NOBS,NSTAT,NOBS,NSTAT)
         IF (NROUND .GT. NI) THEN
            WRITE (9,*) 'NO CONVERGENCE AFTER ', NI, ' ITERATIONS.'
            GOTO 998
         ENDIF
         GO TO 12121
      ENDIF

      GO TO 13131

 998  STOP
      END
```

PROGRAM 25

```
C       -------------------------------------------------------
C       ...... PROGRAM TO ESTIMATE THE CORRELATION MATRIX AND
C              THE VECTOR OF VARIANCES.
C       -------------------------------------------------------

        INTEGER            NT,NY,NV
        PARAMETER          (NT=4380)
        PARAMETER          (NV=7)
        REAL               TERM (5)
        REAL               CORR (NV,NV)
        REAL               RES (NV, NT)
        REAL               VARI (NV)

  5     FORMAT (7 F10.4)
  15    FORMAT (/, ' THE CORRELATION MATRIX: ')
  25    FORMAT (' THE VARIANCE OF EACH VARIABLE: ')

        OPEN (UNIT=10,FILE='\WATER\DATA\RESI.DAT',STATUS='OLD')
        OPEN (UNIT=12,FILE='\WATER\DATA\CORR.DAT',STATUS='UNKNOWN')
        OPEN (UNIT=9,FILE='LPT1')

        DO 20, I = 1, NT
           READ (10, *) (RES (K, I), K = 1, NV)
  20    CONTINUE
        DO 40, I = 1, NV
           CORR (I, I) = 1
  40    CONTINUE
        DO 60, K = 1, NV
           DO 70, J = K+1, NV
              DO 120, II = 1, 5
                 TERM (II) = 0
 120          CONTINUE
              DO 80, I = 1, NT
                 TERM (1) = TERM (1) + RES (K, I) * RES (J, I)
                 TERM (2) = TERM (2) + RES (K, I)
                 TERM (3) = TERM (3) + RES (J, I)
                 TERM (4) = TERM (4) + RES (K, I) ** 2
                 TERM (5) = TERM (5) + RES (J, I) ** 2
  80          CONTINUE
              TERM (1) = TERM (1) / NT
              TERM (4) = SQRT ((TERM (4) / NT) - (TERM (2) / NT) ** 2)
              VARI (K) = TERM (4) ** 2
              TERM (5) = SQRT ((TERM (5) / NT) - (TERM (3) / NT) ** 2)
              TERM (2) = TERM (2) * TERM (3) / NT ** 2
              TERM (1) = TERM (1) - TERM (2)
              TERM (4) = TERM (4) * TERM (5)
              CORR (K, J) = TERM (1) / TERM (4)
  70       CONTINUE
  60    CONTINUE
        TERM (2) = 0
        TERM (4) = 0
```

```
      DO 10, I = 1, NT
         TERM (2) = TERM (2) + RES (NV, I)
         TERM (4) = TERM (4) + RES (NV, I) ** 2
 10   CONTINUE
      VARI (NV) = (TERM (4) / NT) - (TERM (2) / NT ) ** 2
      WRITE (9, 25)
      WRITE (9, 5) (VARI (K), K = 1, NV)
      WRITE (9, 15)
      DO 110, K = 1, NV
         WRITE (9, 5) (CORR (K, J), J = K, NV)
         WRITE (12, 5) (CORR (K, J), J = K, NV)
110   CONTINUE

      STOP
      END
```

PROGRAM 26

```
C       --------------------------------------------------------------
C       ...... PROGRAM TO GENERATE CLIMATE SEQUENCES USING MODEL T
C       --------------------------------------------------------------

        INTEGER         NT,NV,NV3,NV4,NV5,NY,NP,PSTATE,STATE,A,NRAU

C       ...... PSTATE = PRESENT STATE OF DAY
C       ...... STATE = PREVIOUS STATE OF DAY

        PARAMETER       (NT=365)
        PARAMETER       (NY=51)
        PARAMETER       (NV=7)
        PARAMETER       (NV3=4)
        PARAMETER       (NV4=1)
        PARAMETER       (NV5=2)
        PARAMETER       (NP=3)
        PARAMETER       (NRAU=4)

C       ...... NT = £ OBSERVATIONS PER YEAR
C       ...... NV = £ VARIABLES
C       ...... NY = £ YEARS TO BE GENERATED
C       ...... NP = £ PARAMETERS IN SEASONAL MODEL

        INTEGER         SEED (9)
        REAL            RAIN
        REAL            GAM (2,NP)
        REAL            PHI (NP,0:NT)
        REAL            RAU3 (NRAU,NV3)
        REAL            RAU4 (NV4)
        REAL            RAU5 (NRAU,NV5)
        REAL            DECOMP (NV,NV)
        REAL            RAND (1,NV)
        REAL            SIGMA3  (NRAU,NV3)
        REAL            SIGMA4  (2,NV4,0:NT)
        REAL            SIGMA5  (2,NV5,0:NT)
        REAL            MU (2,NV,0:NT)
        REAL            OBSN (NV),TEMP(NV)
        REAL            AMP (0:NP)
        REAL            PHASE (NP)
        REAL            CORR (NV,NV)
        REAL            C (NT)

        COMMON          IDUM1,IDUM2,IDUM3,IDUM4,IDUM5,IDUM6,IDUM7

   15   FORMAT (4F9.2, 2F10.2, F9.2)
   25   FORMAT (' GIVE 9 -VE Nos. TO INITIALIZE RANDOM GENERATOR',/)

        OPEN (UNIT=9,FILE='LPT1')
        OPEN (UNIT=10,FILE='\WATER\DATA\SIMU.DAT',STATUS='UNKNOWN')
        OPEN (UNIT=22,FILE='CON')
```

```
C        ...... COMPUTE THE FOURIER SERIES TERMS

         CALL COSSIN (PHI,NP,NT)
         DO 60, I = 1, NP
            PHI (I,0) = PHI (I,NT)
   60    CONTINUE

         PI=3.14159
         SMAX=135
         SMIN=110
         AVE=(SMAX+SMIN)/2
         AMPS=SMAX-SMIN

         DO 330, I = 1, NT
            C(I) = AVE+(AMPS/2)*COS((2*PI/NT)*(I+11))
  330    CONTINUE

C        ...... READING PARAMETER ESTIMATES

         CALL DATA (GAM,RAU3,RAU4,RAU5,MU,SIGMA3,SIGMA4,SIGMA5,NP,NV,AMP,
     &               PHASE,CV,PHI,CORR,NT,NRAU,NV3,NV4,NV5)

C        ...... COMPUTE THE CHOLESKY DECOMPOSITION OF THE CORRELARTION
C               MATRIX.  INPUT MATRIX HERE AS WELL.

         CALL CHOLESKY (DECOMP,CORR,NV)

C        ...... TRANSPOSE COVARIANCE MATRIX

         CALL GTRANP (DECOMP,NV)

C        ...... INPUT SEEDS TO START RANDOM NUMBER GENERATOR.  MUST BE
C               NEGATIVE NUMBER.

         PRINT 25
         DO 50, II = 1, 9
            READ (22, *) SEED (II)
   50    CONTINUE

         IDUM1 = SEED (1)
         IDUM2 = SEED (2)
         IDUM3 = SEED (3)
         IDUM4 = SEED (4)
         IDUM5 = SEED (5)
         IDUM6 = SEED (6)
         IDUM7 = SEED (7)
         IDUM8 = SEED (8)
         IDUM9 = SEED (9)

C        ...... COMPUTE PARAMETERS NEEDED FOR COMPUTATION OF RAINFALL
C               DEPTH

         CALL CALBET (BETA,CV)
         ALPH = 1 + 1 / BETA
```

```
      GAMM = GAMMA (ALPH)
      BI = 1 /BETA
      W = 0.01721421

C     ...... SET INITIAL STATE OF DAY TO BE DRY
C     ...... SET INITIAL CLIMATE VALUE TO ZERO

C     ...... STATE = 1 ==> DRY
C     ...... STATE = 2 ==> WET

      STATE = 1
      DO 10, I = 1, NV
         OBSN (I) = MU (STATE,I,0)
 10   CONTINUE
      DO 30, I = 1, NY
         DO 40, J = 1, NT

C     ...... GENERATE RAINFALL VALUE
C            ----------------------

C     ...... COMPUTE PROBABILITY THAT A WET DAY FOLLOWS A WET DAY, OR
C            THE PROBABILITY THAT A WET DAY FOLLOWS A DRY DAY.

            CALL PIEST (NP,GAM,STATE,J,PHI,PI,NT)

C     ...... GENERATE A UNIFORM RANDOM NUMBER BETWEEN 0 AND 1.

            UNIFOR = URAN8 (IDUM8)
            IF (UNIFOR .LT. PI) THEN
               PSTATE = 2
            ELSE
               PSTATE = 1
            ENDIF

C     ...... GENERATE A NORMAL RANDOM NUMBER

            CALL GAUSS (DECOMP,RAND)

C     ...... GENERATE CLIMATE SEQUENCES
C            -------------------------

            DO 80, K = 1, NV
               IF ((K.EQ.1).OR.(K.EQ.4).OR.(K.EQ.6)) THEN
                  CALL MOD3 (RAND,STATE,NV3,NV,SIGMA3,MU,RAU3,K,
     &                       J,OBSN,PSTATE,NT,NRAU)
               ELSEIF ((K.EQ.2).OR.(K.EQ.5)) THEN
                  CALL MOD5 (RAND,STATE,NV5,NV,SIGMA5,MU,RAU5,K,
     &                       J,OBSN,PSTATE,NT,NRAU)
               ELSEIF ((K.EQ.3)) THEN
                  CALL MOD4 (RAND,STATE,NV4,NV,SIGMA4,MU,RAU4,K,
     &                       J,OBSN,PSTATE,NT)
               ENDIF
 80         CONTINUE
```

```
C        ...... DETERMINE WHETHER IT RAINED AND SET RAIN VALUE

C        ...... RAIN = 0  ==> DID NOT RAIN
C        ...... RAIN = 1  ==> RAINED

               IF (PSTATE .EQ. 1) THEN
                  RAIN = 0
               ELSE
                  RAIN = 1
               ENDIF

C        ...... GENERATE RAINFALL DEPTH IF IT RAINED
C               ------------------------------------

               IF (RAIN .EQ. 1) THEN
                  CALL DEPTH3 (IDUM9,NP,RAIN,J,AMP,PHASE,GAMM,BI,W)
               ENDIF

C        ...... TRANSFORM VARIABLES TO THE ORIGINAL FORM

               TEMP(2)=(230-100*EXP(OBSN(2)))/(EXP(OBSN(2))+1)
               TEMP(1)=(410+TEMP(2)*EXP(OBSN(1)))/(EXP(OBSN(1))+1)
               TEMP(3)=(C(J)-0.01-(0.01*EXP(OBSN(3))))/(EXP(OBSN(3))+1)
               TEMP(4)=(10000/(EXP(OBSN(4))+1))-0.01
               TEMP(6)=100/(EXP(OBSN(6))+1)
               TEMP(5)=(101+TEMP(6)*EXP(OBSN(5)))/(EXP(OBSN(5))+1)

C        ...... OUTPUT GENERATED SEQUENCES

               IF (I .NE. 1) THEN
                  WRITE (10,15) RAIN, (TEMP (K), K = 1, NV)
               ENDIF

C        ...... UPDATE THE STATE OF THE PREVIOUS DAY

               IF (PSTATE .NE. STATE) THEN
                  STATE = PSTATE
               ENDIF

  40          CONTINUE
  30       CONTINUE

           STOP
           END
```

SUBROUTINES

```
C         -------------------------------------------------------
C         ...... THIS SUBROUTINE ITERATIVELY ESTIMATES THE MODEL
C                PARAMETERS BY THE NEWTON-RAPHSON METHOD FOR M3.
C         -------------------------------------------------------

          SUBROUTINE NEWT3 (ALPHA,SIGMA,RAU,NPARM,MAXITER,NT,NY,CLIMA,SEQ,
C         -----------------------------------------------------------------
     &                 COUNT,DER,DER2,PHI,EPS,NP,NV,K,A,THETA,NRAU,CONVG)
C                      ------------------------------------------------

          INTEGER        COUNT (4,NY)
          INTEGER        SEQ (4,NY,NT)
          REAL           A (NP,0:NP)
          REAL           SIGMA (NRAU,NV)
          REAL           ALPHA (2,NV,NPARM)
          REAL           PHI (NPARM,0:NT)
          REAL           DER (NP)
          REAL           DER2 (NP,NP)
          REAL           CLIMA (NY,0:NT)
          REAL           THETA (NP)
          REAL           RAU (NRAU,NV)

   15     FORMAT (' THE SUCCESSIVE THETA VALUES FOR VARIABLE: ', I4)
   25     FORMAT (' .... DID NOT CONVERGE')
   35     FORMAT (/, ' .... ', I3, ' ITERATION', /)

          OPEN (UNIT=9,FILE='LPT1')

          IC = 0
          WRITE (9,15) K

C         ...... TRANSFORMING THE SEPARATE PARAMETER ARRAYS INTO ONE
C                VECTOR

          DO 20, J = 1, NPARM
             THETA (J) = ALPHA (1,K,J)
             THETA (J+3) = ALPHA (2,K,J)
   20     CONTINUE
          DO 70, J = 1, NRAU
             THETA (J+6) = SIGMA (J,K)
             THETA (J+10) = RAU (J,K)
   70     CONTINUE

C         ...... ITERATIVE PARAMETER ESTIMATION

          DO 10, ITER = 1, MAXITER

C         ...... VECTOR OF 1ST DERIVATIVES AND MATRIX OF 2ND DERIVATIVES
C                IS COMPUTED

             CALL M3DERV (NPARM,NY,NT,ALPHA,SIGMA,RAU,CLIMA,SEQ,COUNT,
     &                    DER,DER2,PHI,NP,NV,K,NRAU)
```

```
            DO 40, KK = 1, NP
               DO 50, J = KK, NP
                  DER2 (J,KK) = DER2 (KK,J)
   50          CONTINUE
   40       CONTINUE
            PRINT 35, ITER

C        ...... NEW PARAMETER ESTIMATES ARE COMPUTED

            CALL NEWPARM (NP,DER,DER2,THETA,EPS,IC,A)

C        ...... UPDATE PARAMETER ESTIMATES

            DO 30 J = 1, NPARM
               ALPHA (1,K,J) = THETA (J)
               ALPHA (2,K,J) = THETA (J+3)
   30       CONTINUE

            DO 80, J = 1, NRAU
               SIGMA (J,K) = THETA (J+6)
               RAU (J,K) = THETA (J+10)
   80       CONTINUE

C        ...... TEST FOR CONVERGENCE

            IF (IC) 10,10,60
   10    CONTINUE
         WRITE (9,25)
         CONVG = 0

   60    RETURN
         END




         SUBROUTINE CALBET (BETA,CV)
C        ---------------------------

         REAL              NUM, DENOM

         C2 = CV ** 2
         C3 = CV ** 3
         NUM = 339.5410 + 148.4445*CV + 192.7492*C2 + 22.4401*C3
         DENOM = 1 + 257.1162*CV + 287.8362*C2 + 157.2230*C3
         BETA = NUM / DENOM

         RETURN
         END
```

```
C        ----------------------------------------------------------
C        ...... THIS SUBROUTINE SOLVES A SYSTEM OF EQUATIONS AND
C               AND EXTRACTS NEW PARAMETER ESTIMATES
C        ----------------------------------------------------------


         SUBROUTINE NEWPARM (NP,DER,DER2,THETA,EPS,IC,A)
C        ------------------------------------------------

         REAL               A (NP,0:NP)
         REAL               DER (NP)
         REAL               DER2 (NP,NP)
         REAL               THETA (NP)

   15    FORMAT (' MATRIX IS SINGULAR')
   25    FORMAT (' NEW PARAMETER ESTIMATES: ', F10.4)

         OPEN (UNIT=9,FILE='LPT1')

C        THIS SETS UP THE A MATRIX WHICH IS USED IN SOLVING THE SYSTEM
C        OF EQUATIONS

         DO 10, I = 1, NP
            A (I,0) = DER (I)
            DO 20, J = 1, NP
               A (I,J) = DER2 (I,J)
   20       CONTINUE
   10    CONTINUE

C        THIS SOLVES THE SYSTEM OF EQUATIONS
C        THE DIFFERENCE BETWEEN THE VALUE OF THETA (Q) IN THIS
C        ITERATION AND IN THE PREVIOUS ITERATION ARE STORED IN A (Q,0)

         DO 30, I1 = 1, NP
            I2 = I1
            T1 = 0
            DO 40, I3 = I1, NP
               IF (ABS (A (I3,I1)) .GT. (ABS (T1))) THEN
                  I2 = I3
                  T1 = A (I3,I1)
               ENDIF
   40       CONTINUE
            IF (T1 .EQ. 0) THEN
               WRITE (9,15)
               STOP
            ENDIF
            IF (I2 .NE. I1) THEN
               DO 50, IO = 0, NP
                  TEMP = A (I1,IO)
                  A (I1,IO) = A (I2,IO)
                  A (I2,IO) = TEMP
   50          CONTINUE
            ENDIF
            T2 = 1 / (A (I1,I1))
```

```
          NQ = NP
          DO 60, I4 = 0, NQ
             A (I1,I4) = A (I1,I4) * T2
   60     CONTINUE
          DO 70, I3 = 1, NP
             IF (I1 .NE. I3) THEN
                T2 = A (I3,I1)
                A (I3,0) = A (I3,0)-A(I1,0)* T2
                DO 80, I0 = I1, NP
                   A(I3,I0) = A(I3,I0) - A(I1,I0) * T2
   80           CONTINUE
             ENDIF
   70     CONTINUE
   30  CONTINUE

C      ...... CONVERGENCE TEST

       CRIT = 0
       DO 205, I = 1, NP
          CRIT = CRIT + ABS(A(I,0))
  205  CONTINUE
       IF (CRIT .GT. EPS) THEN
          IC = 0
       ELSE
          IC = 1
       ENDIF

C      ..... THIS EXTRACTS THE NEW PARAMETER VALUES

       DO 90, I = 1, NP
          THETA (I) = THETA (I) - A (I,0)
C         WRITE (9,25) THETA (I)
          PRINT 25, THETA (I)
   90  CONTINUE

       RETURN
       END
```

```
C       -------------------------------------------------------------------
C       ...... THIS SUBROUTINE COMPUTES THE VECTOR OF FIRST DERIVATIVES
C              AND THE MATRIX OF SECOND DERIVATIVES FOR MODEL3.
C       -------------------------------------------------------------------


      SUBROUTINE M3DERV (NPARM,NY,NT,ALPHA,SIGMA,RAU,CLIMA,SEQ,COUNT,
C     -------------------------------------------------------------------
     &                       DER,DER2,PHI,NP,NV,K,NRAU)
C                            --------------------------

      INTEGER        COUNT (4,NY)
      INTEGER        SEQ (4,NY,NT)
      REAL           CLIMA (NY,0:NT)
      REAL           MU (2,0:365)
      REAL           SIGMA (NRAU,NV)
      REAL           DER (NP)
      REAL           DER2 (NP,NP)
      REAL           ALPHA (2,NV,NPARM)
      REAL           PHI (NPARM,0:NT)
      REAL           RAU (NRAU,NV)

      DO 10, M = 1, 2
         DO 30, I = 0, NT
            MU (M,I) = 0.0
            DO 40, L = 1, NPARM
               MU (M,I) = MU (M,I) + ALPHA (M,K,L) * PHI (L,I)
 40         CONTINUE
 30      CONTINUE
 10   CONTINUE

      DO 80, I = 1, NP
         DER (I) = 0.0
         DO 90, J = 1, NP
            DER2 (I,J) = 0.0
 90      CONTINUE
 80   CONTINUE

      CALL M3DER1(NY,NT,NP,NPARM,COUNT,SEQ,CLIMA,MU,SIGMA,RAU,PHI,DER,
     &            NV,K,NRAU)
      CALL M3DER2(NY,NT,NP,NPARM,COUNT,SEQ,SIGMA,RAU,PHI,DER2,NV,K,
     &            NRAU)
      CALL M3DER3(NY,NT,NP,NPARM,COUNT,SEQ,CLIMA,MU,SIGMA,RAU,PHI,DER2,
     &            NV,K,NRAU)
      CALL M3DER4(NY,NT,NP,COUNT,SEQ,CLIMA,MU,SIGMA,RAU,DER2,NV,K,NRAU)

      RETURN
      END
```

```
C        --------------------------------------------------------------
C        ...... THIS SUBROUTINE COMPUTES THE VECTOR OF FIRST DERIVATIVES
C               FOR MODEL 3.
C        --------------------------------------------------------------


      SUBROUTINE M3DER1 (NY,NT,NP,NPARM,COUNT,SEQ,CLIMA,MU,SIGMA,
C     ---------------------------------------------------------
     &                     RAU,PHI,DER,NV,K,NRAU)
C                          ----------------------

      INTEGER       COUNT (NRAU,NY)
      INTEGER       SEQ (NRAU,NY,NT)
      INTEGER       T,P
      REAL          CLIMA (NY,0:NT)
      REAL          MIDDLE
      REAL          DER (NP)
      REAL          MU (2,0:NT)
      REAL          SIGMA (NRAU,NV)
      REAL          PHI (NPARM,0:NT)
      REAL          RAU (NRAU,NV)

      DO 850, LL = 1, NPARM
         DO 870, M = 1, 2
            IF (M .EQ. 1) THEN
               N = 2
               NN = 1
               J = 3
               KK = 4
            ELSEIF (M .EQ. 2) THEN
               N = 1
               NN = 2
               J =
               KK = 3
            ENDIF
C        ...... THE VARIABLE DER1 COMPUTES THE DERIVATIVE FOR THE MEAN
C               FUNCTION

            DER1 = 0
            DO 10, IY = 1, NY
               DO 330, T = 1, COUNT (M,IY)
                  P = SEQ (M,IY,T)
                  IF ((CLIMA(IY,P).NE.-999).AND.(CLIMA(IY,P-1)
     &                   .NE.-999)) THEN
                  MIDDLE = ((CLIMA(IY,P)-MU(M,P))/SIGMA(M,K)-RAU(M,K)*
     &                       ((CLIMA(IY,P-1)-MU(M,P-1))/SIGMA(M,K)))
                  DER1 = DER1+MIDDLE*(-PHI(LL,P)/SIGMA(M,K)+RAU(M,K)*
     &                     PHI(LL,P-1)/SIGMA(M,K))
                  ENDIF
 330           CONTINUE

               DO 350, T = 1, COUNT (J,IY)
                  P = SEQ (J,IY,T)
```

```
                    IF ((CLIMA(IY,P).NE.-999).AND.(CLIMA(IY,P-1)
     &                   .NE.-999)) THEN
                    MIDDLE = ((CLIMA(IY,P)-MU(N,P))/SIGMA(J,K)-RAU(J,K)*
     &                         ((CLIMA(IY,P-1)-MU(NN,P-1))/SIGMA(J,K)))
                    DER1 = DER1+MIDDLE*(RAU(J,K)*PHI(LL,P-1)/SIGMA
     &                      (J,K))
                    ENDIF
 350             CONTINUE

                 DO 360, T = 1, COUNT (KK,IY)
                    P = SEQ (KK,IY,T)
                    IF ((CLIMA(IY,P).NE.-999).AND.(CLIMA(IY,P-1)
     &                   .NE.-999)) THEN
                    MIDDLE=((CLIMA(IY,P)-MU(NN,P))/SIGMA(KK,K)-RAU(KK,K)
     &                      *((CLIMA(IY,P-1)-MU(N,P-1))/SIGMA(KK,K)))
                    DER1 = DER1+MIDDLE*(-PHI(LL,P)/SIGMA(KK,K))
                    ENDIF
 360             CONTINUE
  10          CONTINUE

              IF (M .EQ. 1) THEN
                 DER (LL) = -DER1
              ELSEIF (M .EQ. 2) THEN
                 DER(LL+3) = -DER1
              ENDIF

 870       CONTINUE
 850    CONTINUE

C       ...... THE DERIVATIVE WITH RESPECT TO THE AUTOCORRELATION
C              COEFFICIENT IS COMPUTED AS WELL AS THE DERIVATIVE
C              W.R.T. THE STANDARD DEVIATIONS

        DO 20, IY = 1, NY
           DO 700, T = 1, COUNT (1,IY)
              P = SEQ (1,IY,T)
              IF ((CLIMA(IY,P).NE.-999).AND.(CLIMA(IY,P-1)
     &           .NE.-999)) THEN
                PART1 = (CLIMA(IY,P)-MU(1,P))/SIGMA(1,K)
                PART2 = (CLIMA(IY,P-1)-MU(1,P-1))/SIGMA(1,K)
                MIDDLE = PART1-RAU(1,K)*PART2
                DER(11) = DER(11)+MIDDLE*PART2
                PART1 = -((CLIMA(IY,P)-MU(1,P))/SIGMA(1,K)**2)
                PART2 = ((CLIMA(IY,P-1)-MU(1,P-1))/SIGMA(1,K)**2)
                DER(7)=DER(7)-MIDDLE*(PART1+RAU(1,K)*PART2)-1/SIGMA(1,K)
              ENDIF
 700       CONTINUE

           DO 701, T = 1, COUNT (2,IY)
              P = SEQ (2,IY,T)
              IF ((CLIMA(IY,P).NE.-999).AND.(CLIMA(IY,P-1)
     &           .NE.-999)) THEN
                PART1 = (CLIMA(IY,P)-MU(2,P))/SIGMA(2,K)
                PART2 = (CLIMA(IY,P-1)-MU(2,P-1))/SIGMA(2,K)
```

```
              MIDDLE = PART1-RAU(2,K)*PART2
              DER(12) = DER(12)+MIDDLE*PART2
              PART1 = -((CLIMA(IY,P)-MU(2,P))/SIGMA(2,K)**2)
              PART2 = ((CLIMA(IY,P-1)-MU(2,P-1))/SIGMA(2,K)**2)
              DER(8)=DER(8)-MIDDLE*(PART1+RAU(2,K)*PART2)-1/SIGMA(2,K)
            ENDIF
701       CONTINUE

          DO 702, T = 1, COUNT (3,IY)
            P = SEQ (3,IY,T)
            IF ((CLIMA(IY,P).NE.-999).AND.(CLIMA(IY,P-1)
     &           .NE.-999)) THEN
              PART1 = (CLIMA(IY,P)-MU(2,P))/SIGMA(3,K)
              PART2 = (CLIMA(IY,P-1)-MU(1,P-1))/SIGMA(3,K)
              MIDDLE = PART1-RAU(3,K)*PART2
              DER(13) = DER(13)+MIDDLE*PART2
              PART1 = -((CLIMA(IY,P)-MU(2,P))/SIGMA(3,K)**2)
              PART2 = ((CLIMA(IY,P-1)-MU(1,P-1))/SIGMA(3,K)**2)
              DER(9)=DER(9)-MIDDLE*(PART1+RAU(3,K)*PART2)-1/SIGMA(3,K)
            ENDIF
702       CONTINUE

          DO 703, T = 1, COUNT (4,IY)
            P = SEQ (4,IY,T)
            IF ((CLIMA(IY,P).NE.-999).AND.(CLIMA(IY,P-1)
     &           .NE.-999)) THEN
            PART1 = (CLIMA(IY,P)-MU(1,P))/SIGMA(4,K)
            PART2 = (CLIMA(IY,P-1)-MU(2,P-1))/SIGMA(4,K)
            MIDDLE = PART1-RAU(4,K)*PART2
            DER(14) = DER(14)+MIDDLE*PART2
            PART1 = -((CLIMA(IY,P)-MU(1,P))/SIGMA(4,K)**2)
            PART2 = ((CLIMA(IY,P-1)-MU(2,P-1))/SIGMA(4,K)**2)
            DER(10)=DER(10)-MIDDLE*(PART1+RAU(4,K)*PART2)-1/SIGMA(4,K)
            ENDIF
703       CONTINUE
 20     CONTINUE

        RETURN
        END




C       ---------------------------------------
C       ...... SUBROUTINE TO SUBTRACT TWO MATRICES
C       ---------------------------------------

        SUBROUTINE SUBTR (CLAGO,TERM,NV)
C       ------------------------------

        REAL            CLAGO (NV,NV)
        REAL            TERM (NV,NV)

        DO 10, I = 1, NV
          DO 20, J = 1, NV
            TERM (I,J) = CLAGO (I,J) - TERM (I,J)
20        CONTINUE
10      CONTINUE

        RETURN
        END
```

```
C        ------------------------------------------------------------
C        ...... THIS SUBROUTINE COMPUTES THE FOLLOWING 2ND DERIVATIVES:
C               ALPHAjD-ALPHAiD, ALPHAjD-ALPHAiW, ALPHAiW-ALPHAjW FOR
C               MODEL 3.
C        ------------------------------------------------------------


         SUBROUTINE M3DER2 (NY,NT,NP,NPARM,COUNT,SEQ,SIGMA,RAU,PHI,
C        ----------------------------------------------------------
     &                          DER2,NV,K,NRAU)
C                               --------------

         INTEGER        COUNT (NRAU,NY)
         INTEGER        SEQ (NRAU,NY,NT)
         INTEGER        T,P
         REAL           MIDDLE
         REAL           DER2 (NP,NP)
         REAL           SIGMA (NRAU,NV)
         REAL           PHI (NPARM,0:NT)
         REAL           RAU (NRAU,NV)

         OPEN (UNIT=9,FILE='LPT1')

         DO 10, LL = 1, NPARM
           DO 20, LLL = 1, NPARM
             DO 30, M = 1, 2
               IF (M .EQ. 1)THEN
                 N = 2
                 NN = 1
                 J = 3
                 KK = 4
               ELSEIF (M .EQ. 2) THEN
                 N = 1
                 NN = 2
                 J = 4
                 KK = 3
               ENDIF

C        ...... THE VARIABLE DER COMPUTES THE 2ND DERIVATIVES FOR
C               ALPHAD-ALPHAD AND ALPHAW-ALPHAW WHILE DER3 COMPUTES
C               ALPHAD-ALPHAW

               DER = 0
               DER3 = 0
               DO 40, IY = 1, NY
                 DO 50, T = 1, COUNT (M,IY)
                   P = SEQ (M,IY,T)
                   PART = (-PHI(LL,P)/SIGMA(M,K))+RAU(M,K)*PHI(LL,P-1)
     &                    /SIGMA(M,K)
                   PART2 = (-PHI(LLL,P)/SIGMA(M,K))+RAU(M,K)*PHI
     &                      (LLL,P-1)/SIGMA(M,K)
                   DER = DER+PART*PART2
   50            CONTINUE
```

```
              DO 60, T = 1, COUNT (J,IY)
                P = SEQ (J,IY,T)
                PART = (RAU(J,K)*PHI(LL,P-1)/SIGMA(J,K))
                PART2 = (RAU(J,K)*PHI(LLL,P-1)/SIGMA(J,K))
                DER = DER+PART*PART2
 60           CONTINUE

              DO 70, T = 1, COUNT (KK,IY)
                P = SEQ (KK,IY,T)
                PART = (-PHI(LL,P)/SIGMA(KK,K))
                PART2 = (-PHI(LLL,P)/SIGMA(KK,K))
                DER = DER+PART*PART2
 70           CONTINUE
 40         CONTINUE

            IF (M .EQ. 1) THEN
              DER2 (LL,LLL) = -DER
            ELSEIF (M .EQ. 2) THEN
              DER2 (LL+3,LLL+3) = -DER
            ENDIF

 30       CONTINUE
          DO 80, IY = 1, NY
             DO 90, T = 1, COUNT (3,IY)
                P = SEQ (3,IY,T)
                PART = (RAU(3,K)*PHI(LL,P-1)/SIGMA(3,K))
                DER3 = DER3+PART*(-PHI(LLL,P)/SIGMA(3,K))
 90          CONTINUE

             DO 100, T = 1, COUNT (4,IY)
                P = SEQ (4,IY,T)
                PART = (-PHI(LL,P)/SIGMA(4,K))
                DER3 = DER3+PART*(RAU(4,K)*PHI(LLL,P-1)/SIGMA(4,K))
100          CONTINUE
 80       CONTINUE

          DER2 (LL,LLL+3) = -DER3
 20     CONTINUE
 10   CONTINUE

      RETURN
      END



      SUBROUTINE TMULT(MAT2,M2,N2,PROD,M3,N3,II,KK,JJ)
C     ------------------------------------------------

      REAL        MAT2(M2,N2), PROD(M3,N3)

      DO 7000 I = 1,II,1
         DO 7010 J = 1,JJ,1
            PROD(I,J) = 0.0
            DO 7020 K = 1,KK,1
               PROD(I,J) = PROD(I,J) + MAT2(K,I) * MAT2(K,J)
7020        CONTINUE
7010     CONTINUE
7000  CONTINUE
      RETURN
      END
```

```
C        -------------------------------------------------------------
C        ...... THIS SUBROUTINE COMPUTES THE 2ND DERIVATIVES FOR:
C               RAU-ALPHAjD, ALPHAjW-SIGMA FOR MODEL3.
C        -------------------------------------------------------------


      SUBROUTINE M3DER3 (NY,NT,NP,NPARM,COUNT,SEQ,CLIMA,MU,SIGMA,RAU,
C        ------------------------------------------------------------------
     &                        PHI,DER2,NV,K,NRAU)
C                             ------------------

      INTEGER         COUNT (4,NY)
      INTEGER         SEQ (4,NY,NT)
      INTEGER         T,P
      REAL            CLIMA (NY,0:NT)
      REAL            MIDDLE
      REAL            DER2 (NP,NP)
      REAL            MU (2,0:NT)
      REAL            SIGMA (NRAU,NV)
      REAL            PHI  (NPARM,0:NT)
      REAL            RAU (NRAU,NV)

      OPEN (UNIT=9,FILE='LPT1')

      DO 850, LL = 1, NPARM
         DO 870, M = 1, 2
            IF (M .EQ. 1) THEN
               N = 2
               NN = 1
               J = 3
               KK = 4
            ELSEIF (M .EQ. 2) THEN
               N = 1
               NN = 2
               J = 4
               KK = 3
            ENDIF

C        ...... THE VARIABLE DER1 COMPUTES THE 2ND DERIVATIVES FOR RAU-
C               ALPHA, WHILE DER4 COMPUTES THE 2ND DERIVATIVES FOR ALPHA-
C               SIGMA

            DER1 = 0
            DER4 = 0
            DO 10, IY = 1, NY
              DO 530, T = 1, COUNT (M,IY)
                P = SEQ (M,IY,T)
                IF ((CLIMA(IY,P).NE.-999).AND.(CLIMA(IY,P-1)
     &              .NE.-999)) THEN
                  PART1 = (CLIMA(IY,P)-MU(M,P))/SIGMA(M,K)
                  PART2 = (CLIMA(IY,P-1)-MU(M,P-1))/SIGMA(M,K)
                  MIDDLE = PART1-RAU(M,K)*PART2
                  DER1 = DER1+MIDDLE*(-PHI(LL,P-1)/SIGMA(M,K))+
     &                   (-PHI(LL,P)/SIGMA(M,K)+RAU(M,K)*PHI(LL,P-1)
```

```
     &                          /SIGMA(M,K))*PART2
                          PART1 = ((CLIMA(IY,P)-MU(M,P))/SIGMA(M,K)**2)
                          PART2 = ((CLIMA(IY,P-1)-MU(M,P-1))/SIGMA(M,K)**2)
                          PART5 = PHI(LL,P)
                          PART6 = PHI(LL,P-1)
                          DER4 = DER4+MIDDLE*(PART5/(SIGMA(M,K)**2)-RAU
     &                         (M,K)*PART6/(SIGMA(M,K)**2))+(-PART1
     &                         +RAU(M,K)*PART2)*(-PART5/SIGMA
     &                         (M,K)+RAU(M,K)*PART6/SIGMA(M,K))
                        ENDIF
 530                  CONTINUE
  10              CONTINUE
                  IF (M .EQ. 1) THEN
                     DER2(LL,11) = DER1
                     DER2(LL,7) = -DER4
                  ELSEIF (M .EQ. 2) THEN
                     DER2(LL+3,12) = DER1
                     DER2(LL+3,8) = -DER4
                  ENDIF

                  DER1 = 0
                  DER4 = 0
                  DO 20, IY = 1, NY
                     DO 550, T = 1, COUNT (J,IY)
                       P = SEQ (J,IY,T)
                       IF ((CLIMA(IY,P).NE.-999).AND.(CLIMA(IY,P-1)
     &                     .NE.-999)) THEN
                         PART1 = (CLIMA(IY,P)-MU(N,P))/SIGMA(J,K)
                         PART2 = (CLIMA(IY,P-1)-MU(NN,P-1))/SIGMA(J,K)
                         MIDDLE = PART1-RAU(J,K)*PART2
                         DER1 = DER1+MIDDLE*(-PHI(LL,P-1)/SIGMA(J,K))+
     &                          (RAU(J,K)*PHI(LL,P-1)/SIGMA(J,K))*PART2
                         PART1 = ((CLIMA(IY,P)-MU(N,P))/SIGMA(J,K)**2)
                         PART2 = ((CLIMA(IY,P-1)-MU(NN,P-1))/SIGMA(J,K)**2)
                         PART5 = PHI(LL,P)
                         PART6 = PHI(LL,P-1)
                         DER4=DER4+MIDDLE*(-RAU(J,K)*PART6/(SIGMA(J,K)**2))
     &                        +(-PART1+RAU(J,K)*PART2)*RAU(J,K)*PART6
     &                        /SIGMA(J,K)
                       ENDIF
 550                 CONTINUE
  20              CONTINUE
                  IF (M .EQ. 1) THEN
                     DER2(LL,13) = DER1
                     DER2(LL,9) = -DER4
                  ELSEIF (M .EQ. 2) THEN
                     DER2 (LL+3,14) = DER1
                     DER2(LL+3,10) = -DER4
                  ENDIF

                  DER1 = 0
                  DER4 = 0
                  DO 30, IY = 1, NY
                     DO 560, T = 1, COUNT (KK,IY)
```

```
                  P = SEQ (KK,IY,T)
                  IF (CLIMA(IY,P-1).NE.-999) THEN
                    PART2 = (CLIMA(IY,P-1)-MU(N,P-1))/SIGMA(KK,K)
                    DER1 = DER1+(-PHI(LL,P)/SIGMA(KK,K))*PART2
                  ENDIF
                  IF ((CLIMA(IY,P).NE.-999).AND.(CLIMA(IY,P-1)
     &                .NE.-999)) THEN
                    PART1 = (CLIMA(IY,P)-MU(NN,P))/SIGMA(KK,K)
                    MIDDLE = PART1-RAU(KK,K)*PART2
                    PART1 = ((CLIMA(IY,P)-MU(NN,P))/SIGMA(KK,K)**2)
                    PART2 = ((CLIMA(IY,P-1)-MU(N,P-1))/SIGMA(KK,K)**2)
                    PART5 = PHI(LL,P)
                    PART6 = PHI(LL,P-1)
                    DER4=DER4+MIDDLE*(PART5/(SIGMA(KK,K)**2))+
     &                   (-PART1+RAU(KK,K)*PART2)*(-PART5/SIGMA(KK,K))

                  ENDIF
560             CONTINUE
 30           CONTINUE
              IF (M .EQ. 1) THEN
                 DER2(LL,14) = DER1
                 DER2(LL,10) = -DER4
              ELSEIF (M .EQ. 2) THEN
                 DER2(LL+3,13) = DER1
                 DER2(LL+3,9) = -DER4
              ENDIF
870        CONTINUE
850   CONTINUE

      RETURN
      END




      SUBROUTINE COPY(MAT1,M1,N1,MAT2,M2,N2,DIM1,DIM2)
C     ------------------------------------------------

      INTEGER DIM1,DIM2
      REAL MAT1(M1,N1), MAT2(M2,N2)

      DO 10020 I = 1,M2,1
         DO 10030 J = 1,N2,1
            MAT2(I,J) = 0.0
10030    CONTINUE
10020 CONTINUE
      DO 10000 I = 1,DIM1,1
         DO 10010 J = 1,DIM2,1
            MAT2(I,J) = MAT1(I,J)
10010    CONTINUE
10000 CONTINUE
      RETURN
      END
```

```
C        ------------------------------------------------------------
C        ...... THIS SUBROUTINE COMPUTES THE 2ND DERIVATIVES FOR:
C               RAU-SIGMA, RAU-RAU, SIGMA-SIGMA FOR MODEL3
C        ------------------------------------------------------------


         SUBROUTINE M3DER4 (NY,NT,NP,COUNT,SEQ,CLIMA,MU,SIGMA,RAU,DER2,
C        -------------------------------------------------------------------
     &                            NV,K,NRAU)
C                                 ----------

         INTEGER        COUNT (NRAU,NY)
         INTEGER        SEQ (NRAU,NY,NT)
         INTEGER        T,P
         REAL           CLIMA (NY,0:NT)
         REAL           MIDDLE
         REAL           DER2 (NP,NP)
         REAL           MU (2,0:NT)
         REAL           SIGMA (NRAU,NV)
         REAL           RAU (NRAU,NV)


         OPEN (UNIT=9,FILE='LPT1')

C        ...... THE 2ND DERIVATIVE FOR RAU-RAU, SIGMA-SIGMA AND
C               RAU-SIGMA ARE COMPUTED

         DO 20, IY = 1, NY
            DO 330, T = 1, COUNT (1,IY)
               P = SEQ (1,IY,T)
               IF (CLIMA(IY,P-1).NE.-999) THEN
                PART2 = (CLIMA(IY,P-1)-MU(1,P-1))/SIGMA(1,K)
                DER2 (11,11) = DER2 (11,11)-(PART2**2)
               ENDIF
               IF ((CLIMA(IY,P).NE.-999).AND.(CLIMA(IY,P-1)
     &             .NE.-999)) THEN
                MIDDLE = ((CLIMA(IY,P)-MU(1,P))/SIGMA(1,K)-RAU(1,K)*
     &                   ((CLIMA(IY,P-1)-MU(1,P-1))/SIGMA(1,K)))
                PART3 = ((CLIMA(IY,P)-MU(1,P))/SIGMA(1,K)**2)
                PART4 = ((CLIMA(IY,P-1)-MU(1,P-1))/SIGMA(1,K)**2)
                DER2(7,7) = DER2(7,7)+MIDDLE*(2*PART3/SIGMA(1,K)-2*
     &                      RAU(1,K)*PART4/SIGMA(1,K))+((-PART3+PART4
     &                      *RAU(1,K))**2)-1/(SIGMA(1,K)**2)
                DER2(11,7) = DER2(11,7)+MIDDLE*(-PART4)+(-PART3+RAU(1,K)
     &                       *PART4)*PART2
               ENDIF
  330       CONTINUE

            DO 340, T = 1, COUNT (2,IY)
               P = SEQ (2,IY,T)
               IF (CLIMA(IY,P-1).NE.-999) THEN
                PART2 = (CLIMA(IY,P-1)-MU(2,P-1))/SIGMA(2,K)
                DER2 (12,12) = DER2(12,12)-(PART2**2)
               ENDIF
```

```
             IF ((CLIMA(IY,P).NE.-999).AND.(CLIMA(IY,P-1)
     &            .NE.-999)) THEN
              MIDDLE = ((CLIMA(IY,P)-MU(2,P))/SIGMA(2,K)-RAU(2,K)*
     &                  ((CLIMA(IY,P-1)-MU(2,P-1))/SIGMA(2,K)))
              PART3 = ((CLIMA(IY,P)-MU(2,P))/SIGMA(2,K)**2)
              PART4 = ((CLIMA(IY,P-1)-MU(2,P-1))/SIGMA(2,K)**2)
              DER2(8,8)=DER2(8,8)+MIDDLE*(2*PART3/SIGMA(2,K)-2*
     &                   RAU(2,K)*PART4/SIGMA(2,K))+((-PART3+PART4
     &                   *RAU(2,K))**2)-1/(SIGMA(2,K)**2)
              DER2(12,8)=DER2(12,8)+MIDDLE*(-PART4)+(-PART3+RAU(2,K)
     &                    *PART4)*PART2
            ENDIF
 340      CONTINUE

          DO 350, T = 1, COUNT (3,IY)
             P = SEQ (3,IY,T)
             IF (CLIMA(IY,P-1).NE.-999) THEN
              PART2 = (CLIMA(IY,P-1)-MU(1,P-1))/SIGMA(3,K)
              DER2 (13,13) = DER2 (13,13)-(PART2**2)
             ENDIF
             IF ((CLIMA(IY,P).NE.-999).AND.(CLIMA(IY,P-1)
     &            .NE.-999)) THEN
              MIDDLE = ((CLIMA(IY,P)-MU(2,P))/SIGMA(3,K)-RAU(3,K)*
     &                  ((CLIMA(IY,P-1)-MU(1,P-1))/SIGMA(3,K)))
              PART3 = ((CLIMA(IY,P)-MU(2,P))/SIGMA(3,K)**2)
              PART4 = ((CLIMA(IY,P-1)-MU(1,P-1))/SIGMA(3,K)**2)
              DER2(9,9)=DER2(9,9)+MIDDLE*(2*PART3/SIGMA(3,K)-2*
     &                   RAU(3,K)*PART4/SIGMA(3,K))+((-PART3+PART4
     &                   *RAU(3,K))**2)-1/(SIGMA(3,K)**2)
              DER2(13,9)=DER2(13,9)+MIDDLE*(-PART4)+(-PART3+RAU(3,K)
     &                    *PART4)*PART2
             ENDIF
 350      CONTINUE

          DO 360, T = 1, COUNT (4,IY)
             P = SEQ (4,IY,T)
             IF (CLIMA(IY,P-1).NE.-999) THEN
              PART2 = (CLIMA(IY,P-1)-MU(2,P-1))/SIGMA(4,K)
              DER2 (14,14) = DER2 (14,14)-(PART2**2)
             ENDIF
             IF ((CLIMA(IY,P).NE.-999).AND.(CLIMA(IY,P-1)
     &            .NE.-999)) THEN
              MIDDLE = ((CLIMA(IY,P)-MU(1,P))/SIGMA(4,K)-RAU(4,K)*
     &                  ((CLIMA(IY,P-1)-MU(2,P-1))/SIGMA(4,K)))
              PART3 = ((CLIMA(IY,P)-MU(1,P))/SIGMA(4,K)**2)
              PART4 = ((CLIMA(IY,P-1)-MU(2,P-1))/SIGMA(4,K)**2)
              DER2(10,10)=DER2(10,10)+MIDDLE*(2*PART3/SIGMA(4,K)-2*
     &                     RAU(4,K)*PART4/SIGMA(4,K))+((-PART3+PART4
     &                     *RAU(4,K))**2)-1/(SIGMA(4,K)**2)
              DER2(14,10)=DER2(14,10)+MIDDLE*(-PART4)+(-PART3+RAU(4,K)
     &                     *PART4)*PART2
             ENDIF
 360      CONTINUE
  20    CONTINUE
        DER2(7,7) = -DER2(7,7)
        DER2(8,8) = -DER2(8,8)
        DER2(9,9) = -DER2(9,9)
        DER2(10,10) = -DER2(10,10)

      RETURN
      END
```

```
C          ------------------------------------------------
C          ...... THIS SUBROUTINE ITERATIVELY ESTIMATES THE MODEL
C                 PARAMETERS BY THE NEWTON-RAPHSON METHOD (M4).
C          ------------------------------------------------

      SUBROUTINE NEWT4 (ALPHA,PSI,RAU,NPARM,MAXITER,NT,NY,CLIMA,SEQ,
C     -----------------------------------------------------------------
     &                COUNT,DER,DER2,PHI,EPS,NP,NV,K,A,THETA,NRAU,CONVG)
C                     ---------------------------------------------------

       INTEGER        COUNT (4,NY)
       INTEGER        SEQ (4,NY,NT)
       REAL           A (NP,0:NP)
       REAL           PSI (2,NV,NPARM)
       REAL           ALPHA (2,NV,NPARM)
       REAL           PHI (NPARM,0:NT)
       REAL           DER (NP)
       REAL           DER2 (NP,NP)
       REAL           CLIMA (NY,0:NT)
       REAL           THETA (NP)
       REAL           RAU (NRAU,NV)

 15    FORMAT (' THE SUCCESSIVE THETA VALUES FOR VARIABLE: ', I4)
 25    FORMAT (' .... DID NOT CONVERGE')
 35    FORMAT (/, ' .... ', I3, ' ITERATION', /)

       OPEN (UNIT=9,FILE='LPT1')

       IC = 0
       WRITE (9,15) K

C      ...... TRANSFORMING THE SEPARATE PARAMETER ARRAYS INTO ONE
C             VECTOR

       DO 20, J = 1, NPARM
          THETA (J) = ALPHA (1,K,J)
          THETA (J+3) = ALPHA (2,K,J)
          THETA (J+6) = PSI (1,K,J)
          THETA (J+9) = PSI (2,K,J)
 20    CONTINUE
       DO 70, J = 1, NRAU
          THETA (J+12) = RAU (J,K)
 70    CONTINUE

C      ...... ITERATIVE PARAMETER ESTIMATION

       DO 10, ITER = 1, MAXITER

C      ...... VECTOR OF 1ST DERIVATIVES AND MATRIX OF 2ND DERIVATIVES
C             IS COMPUTED

          CALL M4DERV (NPARM,NY,NT,ALPHA,PSI,RAU,CLIMA,SEQ,COUNT,
     &                 DER,DER2,PHI,NP,NV,K,NRAU)
```

```
            DO 40, KK = 1, NP
               DO 50, J = KK, NP
                  DER2 (J,KK) = DER2 (KK,J)
   50          CONTINUE
   40       CONTINUE
            PRINT 35, ITER

C        ...... NEW PARAMETER ESTIMATES ARE COMPUTED

            CALL NEWPARM (NP,DER,DER2,THETA,EPS,IC,A)

C        ...... UPDATE PARAMETER ESTIMATES

            DO 30 J = 1, NPARM
               ALPHA (1,K,J) = THETA (J)
               ALPHA (2,K,J) = THETA (J+3)
               PSI (1,K,J) = THETA (J+6)
               PSI (2,K,J) = THETA (J+9)
   30       CONTINUE

            DO 80, J = 1, NRAU
               RAU (J,K) = THETA (J+12)
   80       CONTINUE

C        ...... TEST FOR CONVERGENCE

            IF (IC) 10,10,60
   10    CONTINUE
         WRITE (9,25)
         CONVG = 0

   60    RETURN
         END



      SUBROUTINE PMAT(MAT,M,N,DIM1,DIM2)
C     ----------------------------------

      REAL MAT(M,N)
      INTEGER DIM1,DIM2

      OPEN (UNIT=12,FILE='\WATER\DATA\RESI.DAT',STATUS='UNKNOWN')

CC    *** THIS ROUTINE PRINTS OUT A MATRIX OF SIZE M BY N
CC    *** EACH ELEMENT IS PRINT IN A FIELD OF . CHARACTERS WITH
CC    *** TWO DECIMAL PLACES (I.E. NNN NNN.NN)

      DO 50 I = 1,DIM1,1
         WRITE (12,510) (MAT(I,J), J = 1,DIM2)
510      FORMAT(' ',7(F10.4))
50    CONTINUE
      RETURN
      END
```

```
C       ---------------------------------------------------------------
C       ...... THIS SUBROUTINE COMPUTES THE VECTOR OF FIRST DERIVATIVES
C              AND THE MATRIX OF SECOND DERIVATIVES FOR MODEL4.
C       ---------------------------------------------------------------

        SUBROUTINE M4DERV (NPARM,NY,NT,ALPHA,PSI,RAU,CLIMA,SEQ,COUNT,
C       -------------------------------------------------------------
     &                           DER,DER2,PHI,NP,NV,K,NRAU)
C                                --------------------------

        INTEGER         COUNT (4,NY)
        INTEGER         SEQ (4,NY,NT)
        REAL            CLIMA (NY,0:NT)
        REAL            MU (2,0:365)
        REAL            SIGMA (2,0:365)
        REAL            DER (NP)
        REAL            DER2 (NP,NP)
        REAL            PSI (2,NV,NPARM)
        REAL            ALPHA (2,NV,NPARM)
        REAL            PHI (NPARM,0:NT)
        REAL            RAU (NRAU,NV)

        DO 10, M = 1, 2
           DO 30, I = 0, NT
              MU (M,I) = 0.0
              SIGMA (M,I) = 0.0
              DO 40, L = 1, NPARM
                 MU (M,I) = MU (M,I) + ALPHA (M,K,L) * PHI (L,I)
                 SIGMA (M,I) = SIGMA (M,I) + PSI (M,K,L) * PHI (L,I)
40            CONTINUE
30         CONTINUE
10      CONTINUE

        DO 80, I = 1, NP
           DER (I) = 0.0
           DO 90, J = 1, NP
              DER2 (I,J) = 0.0
90         CONTINUE
80      CONTINUE

        CALL M4DER1(NY,NT,NP,NPARM,COUNT,SEQ,CLIMA,MU,SIGMA,RAU,PHI,DER,
     &              NV,K,NRAU)
        CALL M4DER2(NY,NT,NP,NPARM,COUNT,SEQ,CLIMA,MU,SIGMA,RAU,PHI,DER2,
     &              NV,K,NRAU)
        CALL M4DER3(NY,NT,NP,NPARM,COUNT,SEQ,CLIMA,MU,SIGMA,RAU,PHI,DER2,
     &              NV,K,NRAU)
        CALL M4DER4(NY,NT,NP,NPARM,COUNT,SEQ,CLIMA,MU,SIGMA,RAU,PHI,DER2,
     &              NV,K,NRAU)

        RETURN
        END
```

```
C          -----------------------------------------------------------------
C          ...... THIS SUBROUTINE COMPUTES THE VECTOR OF FIRST DERIVATIVES
C                 FOR MODEL4.
C          -----------------------------------------------------------------


      SUBROUTINE M4DER1 (NY,NT,NP,NPARM,COUNT,SEQ,CLIMA,MU,SIGMA,
C     -------------------------------------------------------------
     &                   RAU,PHI,DER,NV,K,NRAU)
C                        ---------------------

      INTEGER         COUNT (4,NY)
      INTEGER         SEQ (4,NY,NT)
      INTEGER         T,P
      REAL            CLIMA (NY,0:NT)
      REAL            MIDDLE
      REAL            DER (NP)
      REAL            MU (2,0:365)
      REAL            SIGMA (2,0:365)
      REAL            PHI (NPARM,0:NT)
      REAL            RAU (NRAU,NV)

      DO 850, LL = 1, NPARM
         DO 870, M = 1, 2
            IF (M .EQ. 1) THEN
               N = 2
               NN = 1
               J = 3
               KK = 4
            ELSEIF (M .EQ. 2) THEN
               N = 1
               NN = 2
               J = 4
               KK = 3
            ENDIF

C          ...... THE VARIABLE DER1 COMPUTES THE DERIVATIVE FOR THE MEAN
C                 FUNCTION, WHILE DER2 AND DER3 COMPUTE THE DERIVATIVE FOR
C                 THE VARIANCE FUNCTION

            DER2 = 0
            DER1 = 0
            DER3 = 0
            DO 10, IY = 1, NY
               DO 330, T = 1, COUNT (M,IY)
                  P = SEQ (M,IY,T)
                  IF ((CLIMA(IY,P).NE.-999).AND.(CLIMA(IY,P-1).NE.
     &                -999)) THEN
                   MIDDLE = ((CLIMA(IY,P)-MU(M,P))/SIGMA(M,P)-RAU(1,K)*
     &                      ((CLIMA(IY,P-1)-MU(M,P-1))/SIGMA(M,P-1)))
                   DER1 = DER1+MIDDLE*(-PHI(LL,P)/SIGMA(M,P)+RAU(1,K)*
     &                    PHI(LL,P-1)/SIGMA(M,P-1))
                   PART1 = (-((CLIMA(IY,P)-MU(M,P))/SIGMA(M,P)**2)*
     &                     PHI(LL,P))
```

```
                  PART2 = ((CLIMA(IY,P-1)-MU(M,P-1))/SIGMA(M,P-1)**2)
     &                    *PHI(LL,P-1)
                  DER3 = DER3+MIDDLE*(PART1+RAU(1,K)*PART2)
                 ENDIF
                 DER2 = DER2+PHI(LL,P)/SIGMA(M,P)
  330          CONTINUE

               DO 350, T = 1, COUNT (J,IY)
                 P = SEQ (J,IY,T)
                 IF ((CLIMA(IY,P).NE.-999).AND.(CLIMA(IY,P-1).NE.
     &                -999)) THEN
                  MIDDLE = ((CLIMA(IY,P)-MU(N,P))/SIGMA(N,P)-RAU(1,K)*
     &                      ((CLIMA(IY,P-1)-MU(NN,P-1))/SIGMA(NN,P-1)))
                  DER1 = DER1+MIDDLE*(RAU(1,K)*PHI(LL,P-1)/SIGMA
     &                   (NN,P-1))
                  PART2 = ((CLIMA(IY,P-1)-MU(NN,P-1))/SIGMA(NN,P-1)
     &                    **2)*PHI(LL,P-1)
                  DER3 = DER3+MIDDLE*(RAU(1,K)*PART2)
                 ENDIF
  350          CONTINUE

               DO 360, T = 1, COUNT (KK,IY)
                 P = SEQ (KK,IY,T)
                 IF ((CLIMA(IY,P).NE.-999).AND.(CLIMA(IY,P-1).NE.
     &                -999)) THEN
                  MIDDLE =((CLIMA(IY,P)-MU(NN,P))/SIGMA(NN,P)-RAU(1,K)
     &                     *((CLIMA(IY,P-1)-MU(N,P-1))/SIGMA(N,P-1)))
                  DER1 = DER1+MIDDLE*(-PHI(LL,P)/SIGMA(NN,P))
                  PART1 = (-((CLIMA(IY,P)-MU(NN,P))/SIGMA(NN,P)**2)*
     &                    PHI(LL,P))
                  DER3 = DER3+MIDDLE*PART1
                 ENDIF
                 DER2 = DER2+PHI(LL,P)/SIGMA(NN,P)
  360          CONTINUE
   10        CONTINUE

             IF (M .EQ. 1) THEN
                DER (LL) = -DER1
                DER (LL+6) = (-DER3-DER2)
             ELSEIF (M .EQ. 2) THEN
                DER(LL+3) = -DER1
                DER(LL+9) = (-DER3-DER2)
             ENDIF
  870      CONTINUE
  850   CONTINUE

C       ...... THE DERIVATIVE WITH RESPECT TO THE AUTOCORRELATION
C              COEFFICIENT IS COMPUTED

        DER (NP) = 0
        DO 20, IY = 1, NY
           DO 700, T = 1, COUNT (1,IY)
              P = SEQ (1,IY,T)
              IF ((CLIMA(IY,P).NE.-999).AND.(CLIMA(IY,P-1).NE.-999))
```

```
     &         THEN
              PART1 = (CLIMA(IY,P)-MU(1,P))/SIGMA(1,P)
              PART2 = (CLIMA(IY,P-1)-MU(1,P-1))/SIGMA(1,P-1)
              DER(NP) = DER(NP)+(PART1-RAU(1,K)*PART2)*PART2
             ENDIF
  700      CONTINUE

           DO 701, T = 1, COUNT (2,IY)
             P = SEQ (2,IY,T)
             IF ((CLIMA(IY,P).NE.-999).AND.(CLIMA(IY,P-1).NE.-999))
     &         THEN
              PART1 = (CLIMA(IY,P)-MU(2,P))/SIGMA(2,P)
              PART2 = (CLIMA(IY,P-1)-MU(2,P-1))/SIGMA(2,P-1)
              DER(NP) = DER(NP)+(PART1-RAU(1,K)*PART2)*PART2
             ENDIF
  701      CONTINUE

           DO 702, T = 1, COUNT (3,IY)
             P = SEQ (3,IY,T)
             IF ((CLIMA(IY,P).NE.-999).AND.(CLIMA(IY,P-1).NE.-999))
     &         THEN
              PART1 = (CLIMA(IY,P)-MU(2,P))/SIGMA(2,P)
              PART2 = (CLIMA(IY,P-1)-MU(1,P-1))/SIGMA(1,P-1)
              DER(NP) = DER(NP)+(PART1-RAU(1,K)*PART2)*PART2
             ENDIF
  702      CONTINUE

           DO 703, T = 1, COUNT (4,IY)
             P = SEQ (4,IY,T)
             IF ((CLIMA(IY,P).NE.-999).AND.(CLIMA(IY,P-1).NE.-999))
     &         THEN
              PART1 = (CLIMA(IY,P)-MU(1,P))/SIGMA(1,P)
              PART2 = (CLIMA(IY,P-1)-MU(2,P-1))/SIGMA(2,P-1)
              DER(NP) = DER(NP)+(PART1-RAU(1,K)*PART2)*PART2
             ENDIF
  703      CONTINUE
   20    CONTINUE

         RETURN
         END



C        ---------------------------------------------------------
C        ...... SUBROUTINE TO COMPUTE THE TRANSPOSE OF A MATRIX
C        ---------------------------------------------------------

         SUBROUTINE TRNSP (PHI,NP,NTT,TRSP,NPARM,NT)
C        ---------------------------------------------

         REAL            PHI  (NT,NPARM)
         REAL            TRSP (NPARM,NT)


         DO 10, I = 1, NP
           DO 20, J = 1, NTT
             TRSP (I,J) = PHI (J,I)
   20      CONTINUE
   10    CONTINUE

         RETURN
         END
```

```
C        -----------------------------------------------------------------
C        ...... THIS SUBROUTINE COMPUTES THE FOLLOWING 2ND DERIVATIVES:
C               ALPHAjD-ALPHAiD, PSIjD-PSIiD, ALPHAjD-PSIiD, ALPHAjW-
C               ALPHAiW, PSIjW-PSIiW, ALPHAjW-PSIiW FOR MODEL 4
C        -----------------------------------------------------------------


         SUBROUTINE M4DER2 (NY,NT,NP,NPARM,COUNT,SEQ,CLIMA,MU,SIGMA,
C        -------------------------------------------------------------
     &                          RAU,PHI,DER2,NV,K,NRAU)
C                               ----------------------

         INTEGER        COUNT (4,NY)
         INTEGER        SEQ (4,NY,NT)
         INTEGER        T,P
         REAL           CLIMA (NY,0:NT)
         REAL           MIDDLE
         REAL           DER2 (NP,NP)
         REAL           MU (2,0:365)
         REAL           SIGMA (2,0:365)
         REAL           PHI (NPARM,0:NT)
         REAL           RAU (NRAU,NV)

         DO 850, LL = 1, NPARM
           DO 870, LLL = 1, NPARM
             DO 880, M = 1, 2
               IF (M .EQ. 1)THEN
                 N = 2
                 NN = 1
                 J = 3
                 KK = 4
               ELSEIF (M .EQ. 2) THEN
                 N = 1
                 NN = 2
                 J = 4
                 KK = 3
               ENDIF

C        ...... THE VARIABLE DER COMPUTES THE 2ND DERIVATIVES FOR
C               ALPHA-ALPHA, DER3 THE DERIVATIVES PSI-PSI AND DER4 THE
C               DERIVATIVES ALPHA-PSI

               DER = 0
               DER3 = 0
               DER4 = 0
               DO 10, IY = 1, NY
                 DO 330, T = 1, COUNT (M,IY)
                   P = SEQ (M,IY,T)
                   PART = (-PHI(LL,P)/SIGMA(NN,P))+RAU(1,K)*PHI(LL,P-1)
     &                    /SIGMA(NN,P-1)
                   PART2 = (-PHI(LLL,P)/SIGMA(NN,P))+RAU(1,K)*PHI(LLL,
     &                      P-1)/SIGMA(NN,P-1)
                   DER = DER+PART*PART2
                   IF ((CLIMA(IY,P).NE.-999).AND.(CLIMA(IY,P-1).NE.
```

```
     &              -999)) THEN
                  MIDDLE = ((CLIMA(IY,P)-MU(NN,P))/SIGMA(NN,P)-RAU(1,K)*
     &                    ((CLIMA(IY,P-1)-MU(NN,P-1))/SIGMA(NN,P-1)))
                  PART1 = ((CLIMA(IY,P)-MU(NN,P))/SIGMA(NN,P)**2)
                  PART2 = ((CLIMA(IY,P-1)-MU(NN,P-1))/SIGMA(NN,P-1)**2)
                  PART3 = PHI(LLL,P)
                  PART4 = PHI(LLL,P-1)
                  PART5 = PHI(LL,P)
                  PART6 = PHI(LL,P-1)
                  DER3 = DER3+MIDDLE*(2*PART1/SIGMA(NN,P)*PART3*PART5
     &                -2*RAU(1,K)*PART2/SIGMA(NN,P-1)*PART4*PART6)+
     &                (-PART1*PART3+PART2*RAU(1,K)*PART4)*(-PART1*
     &                PART5+PART2*PART6*RAU(1,K))-PART3*PART5/
     &                (SIGMA(NN,P)**2)
                  DER4 = DER4+MIDDLE*(PART3*PART5/(SIGMA(NN,P)**2)-
     &                RAU(1,K)*PART4*PART6/(SIGMA(NN,P-1)**2))+(-
     &                PART1*PART3+RAU(1,K)*PART2*PART4)*(-PART5/
     &                SIGMA(NN,P)+RAU(1,K)*PART6/SIGMA(NN,P-1))
                  ENDIF
 330            CONTINUE

                DO 350, T = 1, COUNT (J,IY)
                  P = SEQ (J,IY,T)
                  PART = (RAU(1,K)*PHI(LL,P-1)/SIGMA(NN,P-1))
                  PART2 = (RAU(1,K)*PHI(LLL,P-1)/SIGMA(NN,P-1))
                  DER = DER+PART*PART2
                  IF ((CLIMA(IY,P).NE.-999).AND.(CLIMA(IY,P-1).NE.
     &              -999)) THEN
                  MIDDLE = ((CLIMA(IY,P)-MU(N,P))/SIGMA(N,P)-RAU(1,K)*
     &                    ((CLIMA(IY,P-1)-MU(NN,P-1))/SIGMA(NN,P-1)))
                  PART2 = ((CLIMA(IY,P-1)-MU(NN,P-1))/SIGMA(NN,P-1)**2)
                  PART3 = PHI(LLL,P)
                  PART4 = PHI(LLL,P-1)
                  PART5 = PHI(LL,P)
                  PART6 = PHI(LL,P-1)
                  DER3 = DER3+MIDDLE*(-2*RAU(1,K)*PART2/SIGMA(NN,P-1)*
     &                PART4*PART6)+(RAU(1,K)*PART2*PART4)*(RAU(1,K)*
     &                PART2*PART6)
                  DER4 = DER4+MIDDLE*(-RAU(1,K)*PART4*PART6/(SIGMA(NN
     &                ,P-1)**2))+RAU(1,K)*PART2*PART4*RAU(1,K)*PART6/
     &                SIGMA(NN,P-1)
                  ENDIF
 350            CONTINUE

                DO 360, T = 1, COUNT (KK,IY)
                  P = SEQ (KK,IY,T)
                  PART = (-PHI(LL,P)/SIGMA(NN,P))
                  PART2 = (-PHI(LLL,P)/SIGMA(NN,P))
                  DER = DER+PART*PART2
                  IF ((CLIMA(IY,P).NE.-999).AND.(CLIMA(IY,P-1).NE.
     &              -999)) THEN
                  MIDDLE = ((CLIMA(IY,P)-MU(NN,P))/SIGMA(NN,P)-RAU(1,K)*
     &                    ((CLIMA(IY,P-1)-MU(N,P-1))/SIGMA(N,P-1)))
                  PART1 = ((CLIMA(IY,P)-MU(NN,P))/SIGMA(NN,P) ** 2)
```

```
                 PART3 = PHI(LLL,P)
                 PART4 = PHI(LLL,P-1)
                 PART5 = PHI(LL,P)
                 PART6 = PHI(LL,P-1)
                 DER3 = DER3+MIDDLE*(2*PART1/SIGMA(NN,P))*PART3*PART5
     &                +(-PART1*PART3)*(-PART1*PART5)-PART3*PART5/
     &                (SIGMA(NN,P)**2)
                 DER4 = DER4+MIDDLE*PART3*PART5/(SIGMA(NN,P)**2)+
     &                (-PART1*PART3*(-PART5/SIGMA(NN,P)))
                 ENDIF
  360          CONTINUE
   10        CONTINUE

             IF (M .EQ. 1) THEN
               DER2 (LL,LLL) = -DER
               DER2 (LL+6,LLL+6) = -DER3
               DER2 (LL,LLL+6) = -DER4
             ELSEIF (M .EQ. 2) THEN
               DER2 (LL+3,LLL+3) = -DER
               DER2 (LL+9,LLL+9) = -DER3
               DER2 (LL+3,LLL+9) = -DER4
             ENDIF
  880      CONTINUE
  870    CONTINUE
  850  CONTINUE

       RETURN
       END




C      ---------------------------------------------------------------
C      ...... THIS FUNCTION COMPUTES THE GAMMA FUNCTION OF X GIVEN
C             BY:
C             THE DEFINITE INTEGRAL BETWEEN 0 & INFINITY OF THE
C             FUNCTION:
C                      Y ** (X-1) * EXP(-Y)
C             w.r.t. Y.
C      ---------------------------------------------------------------



       FUNCTION GAMMA (ALPH)
C      ----------------------

       A = ALPH
       G = 1
    4  IF (A .GE. 10) THEN
          GOTO 2
       ELSE
          G = G * A
          A = A + 1
          GOTO 4
       ENDIF
    2  T = (1 + (0.0833333 + 0.00347222 - 0.002681327 / A) / A) / A
       GAMMA = EXP(-1 * A + (A - 0.5) * LOG(A) + 0.918939)*T*A/G

       RETURN
       END
```

```
C        ---------------------------------------------------------
C        ...... THIS SUBROUTINE COMPUTES THE 2ND DERIVATIVES FOR:
C               RAU-ALPHAjD, RAU-PSIjD, RAU-RAU, RAU-ALPHAjW AND
C               RAU-PSIjW FOR MODEL 4
C        ---------------------------------------------------------


      SUBROUTINE M4DER3 (NY,NT,NP,NPARM,COUNT,SEQ,CLIMA,MU,SIGMA,RAU,
C     ------------------------------------------------------------------
     &                   PHI,DER2,NV,K,NRAU)
C                        ------------------

      INTEGER          COUNT (4,NY)
      INTEGER          SEQ (4,NY,NT)
      INTEGER          T,P
      REAL             CLIMA (NY,0:NT)
      REAL             MIDDLE
      REAL             DER2 (NP,NP)
      REAL             MU (2,0:365)
      REAL             SIGMA (2,0:365)
      REAL             PHI (NPARM,0:NT)
      REAL             RAU (NRAU,NV)

      DO 850, LL = 1, NPARM
         DO 870, M = 1, 2
            IF (M .EQ. 1) THEN
               N = 2
               NN = 1
               J = 3
               KK = 4
            ELSEIF (M .EQ. 2) THEN
               N = 1
               NN = 2
               J = 4
               KK = 3
            ENDIF

C        ...... THE VARIABLE DER1 COMPUTES THE 2ND DERIVATIVES FOR RAU-
C               ALPHA, WHILE DER3 COMPUTES THE 2ND DERIVATIVES FOR RAU-
C               PSI

            DER1 = 0
            DER3 = 0
            DO 10, IY = 1, NY
              DO 530, T = 1, COUNT (M,IY)
                P = SEQ (M,IY,T)
                IF ((CLIMA(IY,P).NE.-999).AND.(CLIMA(IY,P-1).NE.
     &               -999)) THEN
                PART1 = (CLIMA(IY,P)-MU(M,P))/SIGMA(M,P)
                PART2 = (CLIMA(IY,P-1)-MU(M,P-1))/SIGMA(M,P-1)
                MIDDLE = PART1-RAU(1,K)*PART2
                DER1 = DER1+MIDDLE*(-PHI(LL,P-1)/SIGMA(M,P-1))+
     &                  (-PHI(LL,P)/SIGMA(M,P)+RAU(1,K)*PHI(LL,P-1)
     &                  /SIGMA(M,P-1))*PART2
```

```
                DER3 = DER3+MIDDLE*(PART2/SIGMA(M,P-1))*(-PHI
     &                 (LL,P-1))+((PART1/SIGMA(M,P))*(-PHI(LL,P))+RAU
     &                 (1,K)*(PART2/SIGMA(M,P-1))*PHI(LL,P-1))*PART2
                ENDIF
 530          CONTINUE

              DO 550, T = 1, COUNT (J,IY)
                P = SEQ (J,IY,T)
                IF ((CLIMA(IY,P).NE.-999).AND.(CLIMA(IY,P-1).NE.
     &              -999)) THEN
                PART1 = (CLIMA(IY,P)-MU(N,P))/SIGMA(N,P)
                PART2 = (CLIMA(IY,P-1)-MU(NN,P-1))/SIGMA(NN,P-1)
                MIDDLE = PART1-RAU(1,K)*PART2
                DER1 = DER1+MIDDLE*(-PHI(LL,P-1)/SIGMA(NN,P-1))+
     &                 (RAU(1,K)*PHI(LL,P-1)/SIGMA(NN,P-1))*PART2
                DER3 = DER3+MIDDLE*(-PHI(LL,P-1)/SIGMA(NN,P-1))*
     &                 PART2+(RAU(1,K)*PHI(LL,P-1)/SIGMA(NN,P-1))*
     &                 PART2**2
                ENDIF
 550          CONTINUE

              DO 560, T = 1, COUNT (KK,IY)
                P = SEQ (KK,IY,T)
                IF (CLIMA(IY,P-1).NE.-999) THEN
                PART2 = (CLIMA(IY,P-1)-MU(N,P-1))/SIGMA(N,P-1)
                DER1 = DER1+(-PHI(LL,P)/SIGMA(NN,P))*PART2
                ENDIF
                IF ((CLIMA(IY,P).NE.-999).AND.(CLIMA(IY,P-1).NE.
     &              -999)) THEN
                PART3 = ((CLIMA(IY,P)-MU(NN,P))/SIGMA(NN,P))
                MIDDLE = (PART3-RAU(1,K)*PART2)
                DER3 = DER3+(-PHI(LL,P)/SIGMA(NN,P))*PART3*PART2
                ENDIF
 560          CONTINUE
  10        CONTINUE

            IF (M .EQ. 1) THEN
               DER2(LL,NP) = DER1
               DER2(LL+6,NP) = DER3
            ELSEIF (M .EQ. 2) THEN
               DER2 (LL+3,NP) = DER1
               DER2 (LL+9,NP) =DER3
            ENDIF
 870      CONTINUE
 850    CONTINUE

C       ...... THE 2ND DERIVATIVE RAU-RAU IS COMPUTED

        DER2 (NP,NP) = 0
        DO 20, IY = 1, NY
           DO 330, T = 1, COUNT (1,IY)
              P = SEQ (1,IY,T)
              IF (CLIMA(IY,P-1).NE.-999) THEN
               PART2 = (CLIMA(IY,P-1)-MU(1,P-1))/SIGMA(1,P-1)
```

```
            DER2(NP,NP) = DER2(NP,NP)+PART2**2
           ENDIF
 330    CONTINUE

        DO 340, T = 1, COUNT (2,IY)
           P = SEQ (2,IY,T)
           IF (CLIMA(IY,P-1).NE.-999) THEN
            PART2 = (CLIMA(IY,P-1)-MU(2,P-1))/SIGMA(2,P-1)
            DER2(NP,NP) = DER2(NP,NP)+PART2**2
           ENDIF
 340    CONTINUE

        DO 350, T = 1, COUNT (3,IY)
           P = SEQ (3,IY,T)
           IF (CLIMA(IY,P-1).NE.-999) THEN
            PART2 = (CLIMA(IY,P-1)-MU(1,P-1))/SIGMA(1,P-1)
            DER2(NP,NP) = DER2(NP,NP)+PART2**2
           ENDIF
 350    CONTINUE

        DO 360, T = 1, COUNT (4,IY)
           P = SEQ (4,IY,T)
           IF (CLIMA(IY,P-1).NE.-999) THEN
            PART2 = (CLIMA(IY,P-1)-MU(2,P-1))/SIGMA(2,P-1)
            DER2(NP,NP) = DER2(NP,NP)+PART2**2
           ENDIF
 360    CONTINUE
  20  CONTINUE
      DER2 (NP,NP) = -DER2 (NP,NP)

      RETURN
      END



C     -------------------------------------------------------------
C     ...... THIS SUBROUTINE COMPUTES PI=PROBABILITY THAT A WET DAY
C            FOLLOWS A WET DAY OR THE PROBABILITY THAT A WET DAY
C            FOLLOWS A DRY DAY.
C     -------------------------------------------------------------


      SUBROUTINE PIEST (NP,GAM,STATE,K,PHI,PI,NT)
C     -------------------------------------------

      INTEGER          STATE
      REAL             LAMBDA
      REAL             GAM (2,NP)
      REAL             PHI (NP,0:NT)
      REAL             PI

      LAMBDA = 0
      DO 10, I = 1, NP
         LAMBDA = LAMBDA + GAM (STATE,I) * PHI (I,K)
 10   CONTINUE
      PI = EXP (LAMBDA) / (1 + EXP (LAMBDA))

      RETURN
      END
```

```
C        -------------------------------------------------------
C        ...... THIS SUBROUTINE COMPUTES THE 2ND DERIVATIVES FOR:
C               ALPHAjD-ALPHAiW, ALPHAjD-PSIiW, PSIjD-PSIiW AND
C               ALPHAjW-PSIiD FOR MODEL 4.
C        -------------------------------------------------------

         SUBROUTINE M4DER4 (NY,NT,NP,NPARM,COUNT,SEQ,CLIMA,MU,SIGMA,RAU,
C        ------------------------------------------------------------------
     &                      PHI,DER2,NV,K,NRAU)
C                           -------------------

         INTEGER       COUNT (4,NY)
         INTEGER       SEQ (4,NY,NT)
         INTEGER       T,P
         REAL          CLIMA (NY,O:NT)
         REAL          DER2 (NP,NP)
         REAL          MU (2,0:365)
         REAL          SIGMA (2,0:365)
         REAL          PHI (NPARM,O:NT)
         REAL          RAU (NRAU,NV)

         DO 850, LL = 1, NPARM
            DO 870, LLL = 1, NPARM

C        ...... THE VARIABLE DER COMPUTES THE 2ND DERIVATIVE ALPHAD-
C               ALPHAW, DER3 THE DERIVATIVE ALPHAD-PSIW, DER4 THE
C               DERIVATIVE PSID-PSIW AND DER5 THE DERIVATIVE ALPHAW-
C               PSID

               DER = O
               DER3 = O
               DER4 = O
               DER5 = O
               DO 10, IY = 1, NY
                  DO 350, T = 1, COUNT (3,IY)
                     P = SEQ (3,IY,T)
                     PART = (RAU(1,K)*PHI(LL,P-1)/SIGMA(1,P-1))
                     DER = DER+PART*(-PHI(LLL,P)/SIGMA(2,P))
                     IF (CLIMA(IY,P).NE.-999) THEN
                     DER3 = DER3+PART*(-PHI(LLL,P)/SIGMA(2,P))*((CLIMA
     &                     (IY,P)-MU(2,P))/SIGMA(2,P))
                     ENDIF
                     IF ((CLIMA(IY,P).NE.-999).AND.(CLIMA(IY,P-1).NE.
     &                    -999)) THEN
                     PART2 = ((CLIMA(IY,P-1)-MU(1,P-1))/SIGMA(1,P-1)**2)
     &                     *PHI(LL,P-1)
                     PART1 = (-(CLIMA(IY,P)-MU(2,P))/SIGMA(2,P)**2)*
     &                     PHI(LLL,P)
                     DER4 = DER4+(PART1*RAU(1,K)*PART2)
                     ENDIF
                     IF (CLIMA(IY,P-1).NE.-999) THEN
                     PART = (-PHI(LL,P)/SIGMA(2,P))
                     DER5 = DER5+PART*RAU(1,K)*PHI(LLL,P-1)/SIGMA(1,P-1)*
```

```
   &                  (CLIMA(IY,P-1)-MU(1,P-1))/SIGMA(1,P-1)
                   ENDIF
350             CONTINUE

                DO 360, T = 1, COUNT (4,IY)
                   P = SEQ (4,IY,T)
                   PART = (-PHI(LL,P)/SIGMA(1,P))
                   DER = DER+PART*(RAU(1,K)*PHI(LLL,P-1)/SIGMA(2,P-1))
                   IF (CLIMA(IY,P-1).NE.-999) THEN
                   DER3 = DER3+PART*(RAU(1,K)*PHI(LLL,P-1)/SIGMA(2,P-1)
   &                      )*(CLIMA(IY,P-1)-MU(2,P-1))/SIGMA(2,P-1)
                   ENDIF
                   IF ((CLIMA(IY,P).NE.-999).AND.(CLIMA(IY,P-1).NE.
   &                   -999)) THEN
                   PART1 = (-((CLIMA(IY,P)-MU(1,P))/SIGMA(1,P)**2)*
   &                         PHI(LL,P))
                   PART2 = ((CLIMA(IY,P-1)-MU(2,P-1))/SIGMA(2,P-1)**2)*
   &                         PHI(LLL,P-1)
                   DER4 = DER4+RAU(1,K)*PART2*PART1
                   ENDIF
                   IF (CLIMA(IY,P).NE.-999) THEN
                   PART = (RAU(1,K)*PHI(LL,P-1)/SIGMA(2,P-1))
                   DER5 = DER5+PART*(-PHI(LLL,P)/SIGMA(1,P))*(CLIMA
   &                      (IY,P)-MU(1,P))/SIGMA(1,P)
                   ENDIF
360             CONTINUE
 10          CONTINUE

             DER2 (LL,LLL+3) = -DER
             DER2 (LL,LLL+9) = -DER3
             DER2 (LL+6,LLL+9) = -DER4
             DER2 (LL+3,LLL+6) = -DER5
870       CONTINUE
850    CONTINUE

       RETURN
       END



       SUBROUTINE PPMAT(MAT,M,N,DIM1,DIM2)
C      ----------------------------------

       REAL MAT(M,N)
       INTEGER DIM1,DIM2

       OPEN (UNIT=9,FILE='LPT1')

CC     *** THIS ROUTINE PRINTS OUT A MATRIX OF SIZE M BY N
CC     *** EACH ELEMENT IS PRINT IN A FIELD OF . CHARACTERS WITH
CC     *** TWO DECIMAL PLACES (I.E. NNN NNN.NN)

       WRITE (9,5020)
       DO 5000 I = 1,DIM1,1
          WRITE (9,5010) (MAT(I,J), J = 1,DIM2)
5010      FORMAT(' ',7(F15.6))
5000   CONTINUE
       WRITE (9,5020)
5020   FORMAT(/)
       RETURN
       END
```

```
C        ---------------------------------------------------------
C        ...... THIS SUBROUTINE ITERATIVELY ESTIMATES THE MODEL
C               PARAMETERS BY THE NEWTON-RAPHSON METHOD FOR M5.
C        ---------------------------------------------------------


         SUBROUTINE NEWT5 (ALPHA,PSI,RAU,NPARM,MAXITER,NT,NY,CLIMA,SEQ,
C        ----------------------------------------------------------------
     &                COUNT,DER,DER2,PHI,EPS,NP,NV,K,A,THETA,NRAU,CONVG)
C                     --------------------------------------------------

         INTEGER         COUNT (4,NY)
         INTEGER         SEQ (4,NY,NT)
         REAL            A (NP,0:NP)
         REAL            PSI (2,NV,NPARM)
         REAL            ALPHA (2,NV,NPARM)
         REAL            PHI (NPARM,0:NT)
         REAL            DER (NP)
         REAL            DER2 (NP,NP)
         REAL            CLIMA (NY,0:NT)
         REAL            THETA (NP)
         REAL            RAU (NRAU,NV)

   15    FORMAT (' THE SUCCESSIVE THETA VALUES FOR VARIABLE: ', I4)
   25    FORMAT (' .... DID NOT CONVERGE')
   35    FORMAT (/, ' .... ', I3, ' ITERATION', /)

         OPEN (UNIT=9,FILE='LPT1')

         IC = 0
         WRITE (9,15) K

C        ...... TRANSFORMING THE SEPARATE PARAMETER ARRAYS INTO ONE
C               VECTOR

         DO 20, J = 1, NPARM
            THETA (J) = ALPHA (1,K,J)
            THETA (J+3) = ALPHA (2,K,J)
            THETA (J+6) = PSI (1,K,J)
            THETA (J+9) = PSI (2,K,J)
   20    CONTINUE
         DO 70, J = 1, NRAU
            THETA (J+12) = RAU (J,K)
   70    CONTINUE

C        ...... ITERATIVE PARAMETER ESTIMATION

         DO 10, ITER = 1, MAXITER

C        ...... VECTOR OF 1ST DERIVATIVES AND MATRIX OF 2ND DERIVATIVES
C               IS COMPUTED

            CALL M5DERV (NPARM,NY,NT,ALPHA,PSI,RAU,CLIMA,SEQ,COUNT,
     &                   DER,DER2,PHI,NP,NV,K,NRAU)
```

```
         DO 40, KK = 1, NP
            DO 50, J = KK, NP
               DER2 (J,KK) = DER2 (KK,J)
 50         CONTINUE
 40      CONTINUE
         PRINT 35, ITER

C     ...... NEW PARAMETER ESTIMATES ARE COMPUTED

         CALL NEWPARM (NP,DER,DER2,THETA,EPS,IC,A)

C     ...... UPDATE PARAMETER ESTIMATES

         DO 30 J = 1, NPARM
            ALPHA (1,K,J) = THETA (J)
            ALPHA (2,K,J) = THETA (J+3)
            PSI (1,K,J) = THETA (J+6)
            PSI (2,K,J) = THETA (J+9)
 30      CONTINUE

         DO 80, J = 1, NRAU
            RAU (J,K) = THETA (J+12)
 80      CONTINUE

C     ...... TEST FOR CONVERGENCE

         IF (IC) 10,10,60
 10   CONTINUE
      WRITE (9,25)
      CONVG = 0

 60   RETURN
      END


C     -------------------------------
C     ...... FUNCTION OF ONE VARIALBE
C     -------------------------------


      FUNCTION DIM1 (X)
C     ----------------

      INTEGER          NPMAX
      PARAMETER        (NPMAX=20)

      COMMON  /ONE/ NPP,THET(NPMAX),DERI(NPMAX)
      DIMENSION        XT(NPMAX)

      OPEN (UNIT=9,FILE='LPT1')

      DO 10, J=1,NPP
         XT(J)=THET(J)+X*DERI(J)
 10   CONTINUE
      DIM1=FUNC (XT)

      RETURN
      END
```

```
C       ------------------------------------------------------------------
C       ...... THIS SUBROUTINE COMPUTES THE VECTOR OF FIRST DERIVATIVES
C              AND THE MATRIX OF SECOND DERIVATIVES FOR MODEL5.
C       ------------------------------------------------------------------


        SUBROUTINE M5DERV (NPARM,NY,NT,ALPHA,PSI,RAU,CLIMA,SEQ,COUNT,
C       ----------------------------------------------------------------
     &                     DER,DER2,PHI,NP,NV,K,NRAU)
C                          --------------------------

        INTEGER        COUNT (4,NY)
        INTEGER        SEQ (4,NY,NT)
        REAL           CLIMA (NY,0:NT)
        REAL           MU (2,0:365)
        REAL           SIGMA (2,0:365)
        REAL           DER (NP)
        REAL           DER2 (NP,NP)
        REAL           PSI (2,NV,NPARM)
        REAL           ALPHA (2,NV,NPARM)
        REAL           PHI (NPARM,0:NT)
        REAL           RAU (NRAU,NV)

        DO 10, M = 1, 2
           DO 30, I = 0, NT
              MU (M,I) = 0.0
              SIGMA (M,I) = 0.0
              DO 40, L = 1, NPARM
                 MU (M,I) = MU (M,I) + ALPHA (M,K,L) * PHI (L,I)
                 SIGMA (M,I) = SIGMA (M,I) + PSI (M,K,L) * PHI (L,I)
 40           CONTINUE
 30        CONTINUE
 10     CONTINUE

        DO 80, I = 1, NP
           DER (I) = 0.0
           DO 90, J = 1, NP
              DER2 (I,J) = 0.0
 90        CONTINUE
 80     CONTINUE

        CALL M5DER1(NY,NT,NP,NPARM,COUNT,SEQ,CLIMA,MU,SIGMA,RAU,PHI,DER,
     &              NV,K,NRAU)
        CALL M5DER2(NY,NT,NP,NPARM,COUNT,SEQ,CLIMA,MU,SIGMA,RAU,PHI,DER2,
     &              NV,K,NRAU)
        CALL M5DER3(NY,NT,NP,NPARM,COUNT,SEQ,CLIMA,MU,SIGMA,RAU,PHI,DER2,
     &              NV,K,NRAU)
        CALL M5DER4(NY,NT,NP,NPARM,COUNT,SEQ,CLIMA,MU,SIGMA,RAU,PHI,DER2,
     &              NV,K,NRAU)

        RETURN
        END
```

```
C         ---------------------------------------------------------------
C         ...... THIS SUBROUTINE COMPUTES THE VECTOR OF FIRST DERIVATIVES
C                FOR MODEL 5.
C         ---------------------------------------------------------------

          SUBROUTINE M5DER1 (NY,NT,NP,NPARM,COUNT,SEQ,CLIMA,MU,SIGMA,
C         ------------------------------------------------------------
     &                       RAU,PHI,DER,NV,K,NRAU)
C                            ---------------------

          INTEGER        COUNT (NRAU,NY)
          INTEGER        SEQ (NRAU,NY,NT)
          INTEGER        T,P
          REAL           CLIMA (NY,0:NT)
          REAL           MIDDLE
          REAL           DER (NP)
          REAL           MU (2,0:365)
          REAL           SIGMA (2,0:365)
          REAL           PHI (NPARM,0:NT)
          REAL           RAU (NRAU,NV)

          DO 850, LL = 1, NPARM
             DO 870, M = 1, 2
                IF (M .EQ. 1) THEN
                   N = 2
                   NN = 1
                   J = 3
                   KK = 4
                ELSEIF (M .EQ. 2) THEN
                   N = 1
                   NN = 2
                   J = 4
                   KK = 3
                ENDIF

C         ...... THE VARIABLE DER1 COMPUTES THE DERIVATIVE FOR THE MEAN
C                FUNCTION, WHILE DER2 AND DER3 COMPUTE THE DERIVATIVE FOR
C                THE VARIANCE FUNCTION

                DER2 = 0
                DER1 = 0
                DER3 = 0
                DO 10, IY = 1, NY
                   DO 330, T = 1, COUNT (M,IY)
                      P = SEQ (M,IY,T)
                      IF ((CLIMA(IY,P).NE.-999).AND.(CLIMA(IY,P-1).NE.
     &                   -999)) THEN
                      MIDDLE = ((CLIMA(IY,P)-MU(M,P))/SIGMA(M,P)-RAU(M,K)*
     &                         ((CLIMA(IY,P-1)-MU(M,P-1))/SIGMA(M,P-1)))
                      DER1 = DER1+MIDDLE*(-PHI(LL,P)/SIGMA(M,P)+RAU(M,K)*
     &                       PHI(LL,P-1)/SIGMA(M,P-1))
                      PART1 = (-((CLIMA(IY,P)-MU(M,P))/SIGMA(M,P)**2)*
     &                        PHI(LL,P))
```

```
                 PART2 = ((CLIMA(IY,P-1)-MU(M,P-1))/SIGMA(M,P-1)**2)
     &                  *PHI(LL,P-1)
                 DER3 = DER3+MIDDLE*(PART1+RAU(M,K)*PART2)
                 ENDIF
                 DER2 = DER2+PHI(LL,P)/SIGMA(M,P)
  330          CONTINUE

               DO 350, T = 1, COUNT (J,IY)
                 P = SEQ (J,IY,T)
                 IF ((CLIMA(IY,P).NE.-999).AND.(CLIMA(IY,P-1).NE.
     &               -999)) THEN
                 MIDDLE = ((CLIMA(IY,P)-MU(N,P))/SIGMA(N,P)-RAU(J,K)*
     &                   ((CLIMA(IY,P-1)-MU(NN,P-1))/SIGMA(NN,P-1)))
                 DER1 = DER1+MIDDLE*(RAU(J,K)*PHI(LL,P-1)/SIGMA
     &                 (NN,P-1))
                 PART2 = ((CLIMA(IY,P-1)-MU(NN,P-1))/SIGMA(NN,P-1)
     &                  **2)*PHI(LL,P-1)
                 DER3 = DER3+MIDDLE*(RAU(J,K)*PART2)
                 ENDIF
  350          CONTINUE

               DO 360, T = 1, COUNT (KK,IY)
                 P = SEQ (KK,IY,T)
                 IF ((CLIMA(IY,P).NE.-999).AND.(CLIMA(IY,P-1).NE.
     &               -999)) THEN
                 MIDDLE=((CLIMA(IY,P)-MU(NN,P))/SIGMA(NN,P)-RAU(KK,K)
     &                 *((CLIMA(IY,P-1)-MU(N,P-1))/SIGMA(N,P-1)))
                 DER1 = DER1+MIDDLE*(-PHI(LL,P)/SIGMA(NN,P))
                 PART1 = (-((CLIMA(IY,P)-MU(NN,P))/SIGMA(NN,P)**2)*
     &                  PHI(LL,P))
                 DER3 = DER3+MIDDLE*PART1
                 ENDIF
                 DER2 = DER2+PHI(LL,P)/SIGMA(NN,P)
  360          CONTINUE
   10        CONTINUE

             IF (M .EQ. 1) THEN
               DER (LL) = -DER1
               DER (LL+6) = (-DER3-DER2)
             ELSEIF (M .EQ. 2) THEN
               DER(LL+3) = -DER1
               DER(LL+9) = (-DER3-DER2)
             ENDIF

  870      CONTINUE
  850   CONTINUE

C       ...... THE DERIVATIVE WITH RESPECT TO THE AUTOCORRELATION
C              COEFFICIENT IS COMPUTED

        DO 20, IY = 1, NY
           DO 700, T = 1, COUNT (1,IY)
              P = SEQ (1,IY,T)
              IF ((CLIMA(IY,P).NE.-999).AND.(CLIMA(IY,P-1).NE.
```

```
     &              -999)) THEN
                PART1 = (CLIMA(IY,P)-MU(1,P))/SIGMA(1,P)
                PART2 = (CLIMA(IY,P-1)-MU(1,P-1))/SIGMA(1,P-1)
                DER(13) = DER(13)+(PART1-RAU(1,K)*PART2)*PART2
               ENDIF
 700        CONTINUE

            DO 701, T = 1, COUNT (2,IY)
               P = SEQ (2,IY,T)
               IF ((CLIMA(IY,P).NE.-999).AND.(CLIMA(IY,P-1).NE.
     &              -999)) THEN
                PART1 = (CLIMA(IY,P)-MU(2,P))/SIGMA(2,P)
                PART2 = (CLIMA(IY,P-1)-MU(2,P-1))/SIGMA(2,P-1)
                DER(14) = DER(14)+(PART1-RAU(2,K)*PART2)*PART2
               ENDIF
 701        CONTINUE

            DO 702, T = 1, COUNT (3,IY)
               P = SEQ (3,IY,T)
               IF ((CLIMA(IY,P).NE.-999).AND.(CLIMA(IY,P-1).NE.
     &              -999)) THEN
                PART1 = (CLIMA(IY,P)-MU(2,P))/SIGMA(2,P)
                PART2 = (CLIMA(IY,P-1)-MU(1,P-1))/SIGMA(1,P-1)
                DER(15) = DER(15)+(PART1-RAU(3,K)*PART2)*PART2
               ENDIF
 702        CONTINUE

            DO 703, T = 1, COUNT (4,IY)
               P = SEQ (4,IY,T)
               IF ((CLIMA(IY,P).NE.-999).AND.(CLIMA(IY,P-1).NE.
     &              -999)) THEN
                PART1 = (CLIMA(IY,P)-MU(1,P))/SIGMA(1,P)
                PART2 = (CLIMA(IY,P-1)-MU(2,P-1))/SIGMA(2,P-1)
                DER(NP) = DER(NP)+(PART1-RAU(4,K)*PART2)*PART2
               ENDIF
 703        CONTINUE
  20     CONTINUE

         RETURN
         END


C        --------------------------------------------------------
C        ...... SUBROUTINE TO COMPUTE THE TRANSPOSE OF A MATRIX
C        --------------------------------------------------------

         SUBROUTINE TRANSP (PHI,NPARM,NT,TRSP)
C        -------------------------------------

         REAL               PHI (NT,NPARM)
         REAL               TRSP (NPARM,NT)


         DO 10, I = 1, NPARM
            DO 20, J = 1, NT
               TRSP (I,J) = PHI (J,I)
 20         CONTINUE
 10      CONTINUE

         RETURN
         END
```

```
C         ------------------------------------------------------------------
C         ...... THIS SUBROUTINE COMPUTES THE FOLLOWING 2ND DERIVATIVES:
C                ALPHAjD-ALPHAiD, PSIjD-PSIiD, ALPHAjD-PSIiD, ALPHAjW-
C                ALPHAiW, PSIjW-PSIiW, ALPHAjW-PSIiW FOR MODEL 5.
C         ------------------------------------------------------------------


          SUBROUTINE M5DER2 (NY,NT,NP,NPARM,COUNT,SEQ,CLIMA,MU,SIGMA,
C         ---------------------------------------------------------------
     &                          RAU,PHI,DER2,NV,K,NRAU)
C                               ----------------------

          INTEGER          COUNT (4,NY)
          INTEGER          SEQ (4,NY,NT)
          INTEGER          T,P
          REAL             CLIMA (NY,O:NT)
          REAL             MIDDLE
          REAL             DER2 (NP,NP)
          REAL             MU (2,0:365)
          REAL             SIGMA (2,0:365)
          REAL             PHI (NPARM,O:NT)
          REAL             RAU (NRAU,NV)

          DO 850, LL = 1, NPARM
            DO 870, LLL = 1, NPARM
              DO 880, M = 1, 2
                IF (M .EQ. 1)THEN
                  N = 2
                  NN = 1
                  J = 3
                  KK = 4
                ELSEIF (M .EQ. 2) THEN
                  N = 1
                  NN = 2
                  J = 4
                  KK = 3
                ENDIF

C         ...... THE VARIABLE DER COMPUTES THE 2ND DERIVATIVES FOR
C                ALPHA-ALPHA, DER3 THE DERIVATIVES PSI-PSI AND DER4 THE
C                DERIVATIVES ALPHA-PSI

                DER = O
                DER3 = O
                DER4 = O
                DO 10, IY = 1, NY
                  DO 330, T = 1, COUNT (M,IY)
                    P = SEQ (M,IY,T)
                    PART = (-PHI(LL,P)/SIGMA(NN,P))+RAU(M,K)*PHI(LL,P-1)
     &                        /SIGMA(NN,P-1)
                    PART2 = (-PHI(LLL,P)/SIGMA(NN,P))+RAU(M,K)*PHI
     &                         (LLL,P-1)/SIGMA(NN,P-1)
                    DER = DER+PART*PART2
                    IF ((CLIMA(IY,P).NE.-999).AND.(CLIMA(IY,P-1).NE.-999))
```

```
     &              THEN
               MIDDLE = ((CLIMA(IY,P)-MU(NN,P))/SIGMA(NN,P)-RAU(M,K)*
     &                   ((CLIMA(IY,P-1)-MU(NN,P-1))/SIGMA(NN,P-1)))
               PART1 = ((CLIMA(IY,P)-MU(NN,P))/SIGMA(NN,P)**2)
               PART2 = ((CLIMA(IY,P-1)-MU(NN,P-1))/SIGMA(NN,P-1)**2)
               PART3 = PHI(LLL,P)
               PART4 = PHI(LLL,P-1)
               PART5 = PHI(LL,P)
               PART6 = PHI(LL,P-1)
               DER3 = DER3+MIDDLE*(2*PART1/SIGMA(NN,P)*PART3*PART5
     &                -2*RAU(M,K)*PART2/SIGMA(NN,P-1)*PART4*PART6)+
     &                (-PART1*PART3+PART2*RAU(M,K)*PART4)*(-PART1*
     &                PART5+PART2*PART6*RAU(M,K))-PART3*PART5/
     &                (SIGMA(NN,P)**2)
               DER4 = DER4+MIDDLE*(PART3*PART5/(SIGMA(NN,P)**2)-RAU
     &                (M,K)*PART4*PART6/(SIGMA(NN,P-1)**2))+(-PART1
     &                *PART3+RAU(M,K)*PART2*PART4)*(-PART5/SIGMA
     &                (NN,P)+RAU(M,K)*PART6/SIGMA(NN,P-1))
               ENDIF
330          CONTINUE

             DO 350, T = 1, COUNT (J,IY)
               P = SEQ (J,IY,T)
               PART = (RAU(J,K)*PHI(LL,P-1)/SIGMA(NN,P-1))
               PART2 = (RAU(J,K)*PHI(LLL,P-1)/SIGMA(NN,P-1))
               DER = DER+PART*PART2
               IF ((CLIMA(IY,P).NE.-999).AND.(CLIMA(IY,P-1).NE.-999))
     &              THEN
               MIDDLE = ((CLIMA(IY,P)-MU(N,P))/SIGMA(N,P)-RAU(J,K)*
     &                   ((CLIMA(IY,P-1)-MU(NN,P-1))/SIGMA(NN,P-1)))
               PART2 = ((CLIMA(IY,P-1)-MU(NN,P-1))/SIGMA(NN,P-1)**2)
               PART3 = PHI(LLL,P)
               PART4 = PHI(LLL,P-1)
               PART5 = PHI(LL,P)
               PART6 = PHI(LL,P-1)
               DER3 = DER3+MIDDLE*(-2*RAU(J,K)*PART2/SIGMA(NN,P-1)*
     &                PART4*PART6)+(RAU(J,K)*PART2*PART4)*(RAU(J,K)
     &                *PART2*PART6)
               DER4 = DER4+MIDDLE*(-RAU(J,K)*PART4*PART6/(SIGMA
     &                (NN,P-1)**2))+RAU(J,K)*PART2*PART4*RAU(J,K)*
     &                PART6/SIGMA(NN,P-1)
               ENDIF
350          CONTINUE

             DO 360, T = 1, COUNT (KK,IY)
               P = SEQ (KK,IY,T)
               PART = (-PHI(LL,P)/SIGMA(NN,P))
               PART2 = (-PHI(LLL,P)/SIGMA(NN,P))
               DER = DER+PART*PART2
               IF ((CLIMA(IY,P).NE.-999).AND.(CLIMA(IY,P-1).NE.-999))
     &              THEN
               MIDDLE = ((CLIMA(IY,P)-MU(NN,P))/SIGMA(NN,P)-RAU(KK,K)
     &                   *((CLIMA(IY,P-1)-MU(N,P-1))/SIGMA(N,P-1)))
               PART1 = ((CLIMA(IY,P)-MU(NN,P))/SIGMA(NN,P) ** 2)
```

```
                 PART3 = PHI(LLL,P)
                 PART4 = PHI(LLL,P-1)
                 PART5 = PHI(LL,P)
                 PART6 = PHI(LL,P-1)
                 DER3 = DER3+MIDDLE*(2*PART1/SIGMA(NN,P))*PART3*PART5
     &                 +(-PART1*PART3)*(-PART1*PART5)-PART3*PART5/
     &                 (SIGMA(NN,P)**2)
                 DER4 = DER4+MIDDLE*PART3*PART5/(SIGMA(NN,P)**2)+
     &                 (-PART1*PART3*(-PART5/SIGMA(NN,P)))
                 ENDIF
360            CONTINUE
 10          CONTINUE

             IF (M .EQ. 1) THEN
               DER2 (LL,LLL) = -DER
               DER2 (LL+6,LLL+6) = -DER3
               DER2 (LL,LLL+6) = -DER4
             ELSEIF (M .EQ. 2) THEN
               DER2 (LL+3,LLL+3) = -DER
               DER2 (LL+9,LLL+9) = -DER3
               DER2 (LL+3,LLL+9) = -DER4
             ENDIF

880        CONTINUE
870      CONTINUE
850    CONTINUE

       RETURN
       END




       SUBROUTINE CNTRAL(MAT,M,N,MATOR,M1,N1,DIM1,DIM2)
C      ----------------------------------------------

       REAL        MAT(M,N), MATOR(M1,N1)
       INTEGER     DIM1, DIM2
       REAL        AVE(25)

       DO 6000 J = 1,DIM2,1
          AVE(J) = 0.0
          DO 6010 I = 1,DI   ,1
             AVE(J) = AVE(J) + MATOR(I,J)
6010      CONTINUE
          AVE(J) = AVE(J) / FLOAT(DIM1)
6000   CONTINUE

CC     ***AVE(J) NOW CONTAINS THE AVERAGE OF THE ELEMENTS IN EACH
CC     ***COLUMN OF THE MATRIX                                     ***

       DO 6020 I = 1,DIM1,1
          DO 6030 J = 1,DIM2,1
             MAT(I,J) =MATOR(I,J) - AVE(J)
6030      CONTINUE
6020   CONTINUE
       RETURN
       END
```

```
C        ---------------------------------------------------------------
C        ...... THIS SUBROUTINE COMPUTES THE 2ND DERIVATIVES FOR:
C               RAU-ALPHAjD, RAU-PSIjD, RAU-RAU, RAU-ALPHAjW AND
C               RAU-PSIjW FOR MODEL5
C        ---------------------------------------------------------------

         SUBROUTINE M5DER3 (NY,NT,NP,NPARM,COUNT,SEQ,CLIMA,MU,SIGMA,RAU,
C        -----------------------------------------------------------------------
     &                        PHI,DER2,NV,K,NRAU)
C                             -------------------

         INTEGER        COUNT (4,NY)
         INTEGER        SEQ (4,NY,NT)
         INTEGER        T,P
         REAL           CLIMA (NY,0:NT)
         REAL           MIDDLE
         REAL           DER2 (NP,NP)
         REAL           MU (2,0:365)
         REAL           SIGMA (2,0:365)
         REAL           PHI (NPARM,0:NT)
         REAL           RAU (NRAU,NV)

         DO 850, LL = 1, NPARM
            DO 870, M = 1, 2
               IF (M .EQ. 1) THEN
                  N = 2
                  NN = 1
                  J = 3
                  KK = 4
               ELSEIF (M .EQ. 2) THEN
                  N = 1
                  NN = 2
                  J = 4
                  KK = 3
               ENDIF

C        ...... THE VARIABLE DER1 COMPUTES THE 2ND DERIVATIVES FOR RAU-
C               ALPHA, WHILE DER3 COMPUTES THE 2ND DERIVATIVES FOR RAU-
C               PSI

               DER1 = 0
               DER3 = 0
               DO 10, IY = 1, NY
                 DO 530, T = 1, COUNT (M,IY)
                   P = SEQ (M,IY,T)
                   IF ((CLIMA(IY,P).NE.-999).AND.(CLIMA(IY,P-1).NE.
     &                  -999)) THEN
                   PART1 = (CLIMA(IY,P)-MU(M,P))/SIGMA(M,P)
                   PART2 = (CLIMA(IY,P-1)-MU(M,P-1))/SIGMA(M,P-1)
                   MIDDLE = PART1-RAU(M,K)*PART2
                   DER1 = DER1+MIDDLE*(-PHI(LL,P-1)/SIGMA(M,P-1))+
     &                    (-PHI(LL,P)/SIGMA(M,P)+RAU(M,K)*PHI(LL,P-1)
     &                    /SIGMA(M,P-1))*PART2
```

```
                 DER3 = DER3+MIDDLE*(PART2/SIGMA(M,P-1))*(-PHI
     &                (LL,P-1))+((PART1/SIGMA(M,P))*(-PHI(LL,P))+RAU
     &                (M,K)*(PART2/SIGMA(M,P-1))*PHI(LL,P-1))*PART2

                 ENDIF
 530           CONTINUE
  10        CONTINUE
            IF (M .EQ. 1) THEN
               DER2(LL,13) = DER1
               DER2(LL+6,13) = DER3
            ELSEIF (M .EQ. 2) THEN
               DER2 (LL+3,14) = DER1
               DER2 (LL+9,14) =DER3
            ENDIF

            DER1 = 0
            DER3 = 0

            DO 20, IY = 1, NY
               DO 550, T = 1, COUNT (J,IY)
                 P = SEQ (J,IY,T)
                 IF ((CLIMA(IY,P).NE.-999).AND.(CLIMA(IY,P-1).NE.
     &              -999)) THEN
                 PART1 = (CLIMA(IY,P)-MU(N,P))/SIGMA(N,P)
                 PART2 = (CLIMA(IY,P-1)-MU(NN,P-1))/SIGMA(NN,P-1)
                 MIDDLE = PART1-RAU(J,K)*PART2
                 DER1 = DER1+MIDDLE*(-PHI(LL,P-1)/SIGMA(NN,P-1))+
     &                (RAU(J,K)*PHI(LL,P-1)/SIGMA(NN,P-1))*PART2
                 DER3 = DER3+MIDDLE*(-PHI(LL,P-1)/SIGMA(NN,P-1))*
     &                PART2+(RAU(J,K)*PHI(LL,P-1)/SIGMA(NN,P-1))*
     &                PART2**2

                 ENDIF
 550           CONTINUE
  20        CONTINUE
            IF (M .EQ. 1) THEN
               DER2(LL,15) = DER1
               DER2(LL+6,15) = DER3
            ELSEIF (M .EQ. 2) THEN
               DER2 (LL+3,16) = DER1
               DER2 (LL+9,16) =DER3
            ENDIF

            DER1 = 0
            DER3 = 0

            DO 30, IY = 1, NY
               DO 560, T = 1, COUNT (KK,IY)
                 P = SEQ (KK,IY,T)
                 IF (CLIMA(IY,P-1).NE.-999) THEN
                  PART2 = (CLIMA(IY,P-1)-MU(N,P-1))/SIGMA(N,P-1)
                  DER1 = DER1+(-PHI(LL,P)/SIGMA(NN,P))*PART2
                 ENDIF
                 IF ((CLIMA(IY,P).NE.-999).AND.(CLIMA(IY,P-1).NE.
```

```
     &                     -999)) THEN
                       PART3 = ((CLIMA(IY,P)-MU(NN,P))/SIGMA(NN,P))
                       DER3 = DER3+(-PHI(LL,P)/SIGMA(NN,P))*PART3*PART2
                      ENDIF

  560                CONTINUE
   30              CONTINUE
                   IF (M .EQ. 1) THEN
                      DER2(LL,16) = DER1
                      DER2(LL+6,16) = DER3
                   ELSEIF (M .EQ. 2) THEN
                      DER2 (LL+3,15) = DER1
                      DER2 (LL+9,15) =DER3
                   ENDIF
  870           CONTINUE
  850        CONTINUE

C            ...... THE 2ND DERIVATIVE RAU-RAU IS COMPUTED

             DO 40, IY = 1, NY
                DO 330, T = 1, COUNT (1,IY)
                   P = SEQ (1,IY,T)
                   IF (CLIMA(IY,P-1).NE.-999) THEN
                    PART2 = (CLIMA(IY,P-1)-MU(1,P-1))/SIGMA(1,P-1)
                    DER2 (13,13) = DER2 (13,13)-(PART2**2)
                   ENDIF
  330           CONTINUE

                DO 340, T = 1, COUNT (2,IY)
                   P = SEQ (2,IY,T)
                   IF (CLIMA(IY,P-1).NE.-999) THEN
                    PART2 = (CLIMA(IY,P-1)-MU(2,P-1))/SIGMA(2,P-1)
                    DER2 (14,14) = DER2(14,14)-(PART2**2)
                   ENDIF
  340           CONTINUE

                DO 350, T = 1, COUNT (3,IY)
                   P = SEQ (3,IY,T)
                   IF (CLIMA(IY,P-1).NE.-999) THEN
                    PART2 = (CLIMA(IY,P-1)-MU(1,P-1))/SIGMA(1,P-1)
                    DER2 (15,15) = DER2 (15,15)-(PART2**2)
                   ENDIF
  350           CONTINUE

                DO 360, T = 1, COUNT (4,IY)
                   P = SEQ (4,IY,T)
                   IF (CLIMA(IY,P-1).NE.-999) THEN
                    PART2 = (CLIMA(IY,P-1)-MU(2,P-1))/SIGMA(2,P-1)
                    DER2 (NP,NP) = DER2 (NP,NP)-(PART2**2)
                   ENDIF
  360           CONTINUE
   40        CONTINUE

             RETURN
             END
```

```
C        ------------------------------------------------------------
C        ...... THIS SUBROUTINE COMPUTES THE 2ND DERIVATIVES FOR:
C               ALPHAjD-ALPHAiW, ALPHAjD-PSIiW, PSIjD-PSIiW AND
C               ALPHAjW-PSIiD FOR MODEL 5
C        ------------------------------------------------------------


         SUBROUTINE M5DER4 (NY,NT,NP,NPARM,COUNT,SEQ,CLIMA,MU,SIGMA,RAU,
C        --------------------------------------------------------------
     &                        PHI,DER2,NV,K,NRAU)
C                             ------------------

         INTEGER        COUNT (4,NY)
         INTEGER        SEQ (4,NY,NT)
         INTEGER        T,P
         REAL           CLIMA (NY,0:NT)
         REAL           DER2 (NP,NP)
         REAL           MU (2,0:365)
         REAL           SIGMA (2,0:365)
         REAL           PHI (NPARM,0:NT)
         REAL           RAU (NRAU,NV)

         DO 850, LL = 1, NPARM
            DO 870, LLL = 1, NPARM

C        ...... THE VARIABLE DER COMPUTES THE 2ND DERIVATIVE ALPHAD-
C               ALPHAW, DER3 THE DERIVATIVE ALPHAD-PSIW, DER4 THE
C               DERIVATIVE PSID-PSIW AND DER5 THE DERIVATIVE ALPHAW-
C               PSID

               DER = 0
               DER3 = 0
               DER4 = 0
               DER5 = 0
               DO 10, IY = 1, NY
                  DO 350, T = 1, COUNT (3,IY)
                     P = SEQ (3,IY,T)
                     PART = (RAU(3,K)*PHI(LL,P-1)/SIGMA(1,P-1))
                     DER = DER+PART*(-PHI(LLL,P)/SIGMA(2,P))
                     IF (CLIMA(IY,P).NE.-999) THEN
                     DER3 = DER3+PART*(-PHI(LLL,P)/SIGMA(2,P))*((CLIMA
     &                      (IY,P)-MU(2,P))/SIGMA(2,P))
                     ENDIF
                     IF ((CLIMA(IY,P).NE.-999).AND.(CLIMA(IY,P-1).NE.
     &                  -999)) THEN
                     PART2 = ((CLIMA(IY,P-1)-MU(1,P-1))/SIGMA(1,P-1)**2)
     &                       *PHI(LL,P-1)
                     PART1 = (-(CLIMA(IY,P)-MU(2,P))/SIGMA(2,P)**2)*
     &                       PHI(LLL,P)
                     DER4 = DER4+(PART1*RAU(3,K)*PART2)
                     ENDIF
                     IF (CLIMA(IY,P-1).NE.-999) THEN
                     PART = (-PHI(LL,P)/SIGMA(2,P))
                     DER5 = DER5+PART*RAU(3,K)*PHI(LLL,P-1)/SIGMA(1,P-1)*
```

```
     &                     (CLIMA(IY,P-1)-MU(1,P-1))/SIGMA(1,P-1)
                     ENDIF
  350             CONTINUE

                  DO 360, T = 1, COUNT (4,IY)
                     P = SEQ (4,IY,T)
                     PART = (-PHI(LL,P)/SIGMA(1,P))
                     DER = DER+PART*(RAU(4,K)*PHI(LLL,P-1)/SIGMA(2,P-1))
                     IF (CLIMA(IY,P-1).NE.-999) THEN
                     DER3 = DER3+PART*(RAU(4,K)*PHI(LLL,P-1)/SIGMA(2,P-1
     &                     ))*(CLIMA(IY,P-1)-MU(2,P-1))/SIGMA(2,P-1)
                     ENDIF
                     IF ((CLIMA(IY,P).NE.-999).AND.(CLIMA(IY,P-1).NE.
     &                   -999)) THEN
                     PART1 = (-((CLIMA(IY,P)-MU(1,P))/SIGMA(1,P)**2)*
     &                      PHI(LL,P))
                     PART2 = ((CLIMA(IY,P-1)-MU(2,P-1))/SIGMA(2,P-1)**2)*
     &                      PHI(LLL,P-1)
                     DER4 = DER4+RAU(4,K)*PART2*PART1
                     ENDIF
                     IF (CLIMA(IY,P).NE.-999) THEN
                     PART = (RAU(4,K)*PHI(LL,P-1)/SIGMA(2,P-1))
                     DER5 = DER5+PART*(-PHI(LLL,P)/SIGMA(1,P))*(CLIMA
     &                     (IY,P)-MU(1,P))/SIGMA(1,P)
                     ENDIF
  360             CONTINUE
   10          CONTINUE

               DER2 (LL,LLL+3) = -DER
               DER2 (LL,LLL+9) = -DER3
               DER2 (LL+6,LLL+9) = -DER4
               DER2 (LL+3,LLL+6) = -DER5

  870       CONTINUE
  850    CONTINUE

         RETURN
         END




C        ----------------------------------------------------
C        ...... THIS SUBROUTINE GENERATES RAINFALL DEPTH ON DAYS
C               WHEN RAIN OCCURS
C        ----------------------------------------------------


         SUBROUTINE DEPTH3 (IDUM9,NP,RAIN,K,AMP,PHASE,GAMM,BI,W)
C        ----------------------------------------------------

         REAL           AMP (0:NP)
         REAL           PHASE (NP)
         REAL           RAIN

         AM = (AMP(0)+AMP(1)*COS(W*((K-1)-PHASE(1)))+AMP(2)*COS(2*W*
     &        ((K-1)-PHASE(2)))) / GAMM
         UNIFOR = URAN9 (IDUM9)
         RAIN = AM * (-1 * LOG(UNIFOR)) ** BI

         RETURN
         END
```

```
C        -------------------------------------------------------------
C        ...... THIS SUBROUTINE READS IN THE PARAMETER ESTIMATES OF THE
C               RAINFALL MODEL AND OF THE CLIMATE MODEL.
C        -------------------------------------------------------------


         SUBROUTINE DATA (GAM,RAU3,RAU4,RAU5,MU,SIGMA3,SIGMA4,SIGMA5,NP,
C        ----------------------------------------------------------------
     &                         NV,AMP,PHASE,CV,PHI,CORR,NT,NRAU,NV3,NV4,NV5)
C                              ---------------------------------------------

         REAL                  GAM (2,NP)
         REAL                  PSI4 (2,1,3)
         REAL                  PSI5 (2,4,3)
         REAL                  ALPHA (2,7,3)
         REAL                  SIGMA3 (NRAU,NV3)
         REAL                  SIGMA4 (2,NV4,0:NT)
         REAL                  SIGMA5 (2,NV5,0:NT)
         REAL                  MU (2,NV,0:NT)
         REAL                  PHI (NP,0:NT)
         REAL                  RAU3 (NRAU,NV3)
         REAL                  RAU4 (NV4)
         REAL                  RAU5 (NRAU,NV5)
         REAL                  AMP (0:NP)
         REAL                  PHASE (NP)
         REAL                  CORR (NV,NV)


   5     FORMAT (7F10.3)

         OPEN (UNIT=12,FILE='\WATER\DATA\EST.DAT',STATUS='OLD')

         READ (12, *) (GAM (1,J), J = 1, NP)
         READ (12, *) (GAM (2,J), J = 1, NP)
         DO 30, K = 1, NV
            DO 10, M = 1, 2
               READ (12, *) (ALPHA (M,K,J), J = 1, NP)
  10        CONTINUE
            IF ((K.EQ.4).OR.(K.EQ.6)) THEN
               IF (K.EQ.4) THEN
                  KK = 1
               ELSEIF (K.EQ.6) THEN
                  KK = 2
               ENDIF
               READ (12,*) (SIGMA3 (L,KK), L = 1, NRAU)
               READ (12,*) (RAU3 (L,KK), L = 1, NRAU)
            ELSEIF (K .EQ. 5) THEN
               KK = 1
               DO 20, M = 1, 2
                  READ (12, *) (PSI4 (M,KK,J), J = 1, NP)
  20           CONTINUE
               READ (12,*) RAU4 (KK)
            ELSEIF ((K.EQ.1).OR.(K.EQ.2).OR.(K.EQ.3).OR.(K.EQ.7)) THEN
               IF (K.EQ.1) THEN
```

```
                  KK = 1
               ELSEIF (K.EQ.2) THEN
                  KK = 2
               ELSEIF (K.EQ.3) THEN
                  KK = 3
               ELSEIF (K.EQ.7) THEN
                  KK = 4
               ENDIF
               DO 120, M = 1, 2
                  READ (12, *) (PSI5 (M,KK,J), J = 1, NP)
 120           CONTINUE
               READ (12, *) (RAU5 (I,KK), I = 1, NRAU)
            ENDIF
  30     CONTINUE
         READ (12, *) (AMP (I), I = 0, 1)
         READ (12, *) (PHASE (I), I = 1, 1)
         READ (12, *) CV

C        ...... INPUT CORRELATION MATRIX

         DO 80, I = 1, NV
            READ (12, *) (CORR (I,J), J = I, NV)
   80    CONTINUE

C        ...... COMPUTE THE MEAN AND STD.DEV. FUNCTION

         DO 40, M = 1, 2
            DO 50, I = 0, NT
               SIGMA4 (M,1,I) = 0.0
               SIGMA5 (M,1,I) = 0.0
               SIGMA5 (M,2,I) = 0.0
               SIGMA5 (M,3,I) = 0.0
               SIGMA5 (M,4,I) = 0.0
               DO 70, K = 1, NV
                  MU (M,K,I) = 0.0
                  DO 60, L = 1, NP
                     MU(M,K,I) = MU(M,K,I) + ALPHA(M,K,L) * PHI(L,I)
                     IF ((K.EQ.1).OR.(K.EQ.2).OR.(K.EQ.3).OR.(K.EQ.7))
     &                  THEN
                        IF (K.EQ.1) THEN
                           KK = 1
                        ELSEIF (K.EQ.2) THEN
                           KK = 2
                        ELSEIF (K.EQ.3) THEN
                           KK = 3
                        ELSEIF (K.EQ.7) THEN
                           KK = 4
                        ENDIF
                        SIGMA5(M,KK,I) = SIGMA5(M,KK,I)+PSI5(M,KK,L) *
     &                                   PHI(L,I)
                     ELSEIF (K.EQ.5) THEN
                        KK = 1
                        SIGMA4(M,KK,I) = SIGMA4(M,KK,I) + PSI4(M,KK,L) *
     &                                   PHI(L,I)
```

```
                  ENDIF
 60             CONTINUE
 70          CONTINUE
 50       CONTINUE
 40    CONTINUE

       RETURN
       END




C     ---------------------------------------------------------------
C     ...... THIS SUBROUTINE COMPUTES THE CHOLESKY DECOMPOSITION
C            OF A MATRIX
C     ---------------------------------------------------------------


      SUBROUTINE CHOLKY (DECOMP,CORR,NV)
C     ----------------------------------

      REAL                 CORR (NV,NV)
      REAL                 DECOMP (NV,NV)


C     ...... COMPUTE CHOLESKY DECOMPOSITION

      DO 40, I = 1, NV
         DO 50, J = 1, NV
            DECOMP (I,J) = 0
 50      CONTINUE
 40   CONTINUE
      DECOMP (1,1) = SQRT (CORR (1,1))
      DO 60, J = 2, NV
         DECOMP (J,1) = CORR (1,J) / DECOMP (1,1)
 60   CONTINUE
      DO 70, I = 2, NV-1
         TERM = 0
         DO 80, J = 1, I-1
            TERM = TERM + DECOMP (I,J) ** 2
 80      CONTINUE
         DECOMP (I,I) = SQRT (CORR (I,I) - TERM)
         DO 90, J = I+1, NV
            TERM = 0
            DO 100, K = 1, I-1
               TERM = TERM + DECOMP (I,K) * DECOMP (J,K)
 100        CONTINUE
            DECOMP (J,I) = (CORR (I,J) - TERM) / DECOMP (I,I)
 90      CONTINUE
 70   CONTINUE
      TERM = 0
      DO 110, J = 1, NV-1
         TERM = TERM + DECOMP (NV,J) ** 2
 110  CONTINUE
      DECOMP (NV,NV) = SQRT (CORR (NV,NV) - TERM)

      RETURN
      END
```

```
C       ------------------------------------------------------------
C       ...... THIS SUBROUTINE READS IN THE PARAMETER ESTIMATES OF THE
C              RAINFALL MODEL AND OF THE CLIMATE MODEL 1.
C       ------------------------------------------------------------


        SUBROUTINE DATA1 (GAM,MU,SIGMA,NP,NV,AMP,PHASE,CV,PHI,A,B,NT)
C       -----------------------------------------------------------

        REAL                GAM (2,NP)
        REAL                PSI (2,7,3)
        REAL                ALPHA (2,7,3)
        REAL                SIGMA (2,NV,0:NT)
        REAL                MU (2,NV,0:NT)
        REAL                PHI (NP,0:NT)
        REAL                AMP (0:NP)
        REAL                PHASE (NP)
        REAL                A (NV,NV)
        REAL                B (NV,NV)

    5   FORMAT (7F10.3)

        OPEN (UNIT=12,FILE='\WATER\DATA\EST1.DAT',STATUS='OLD')

        DO 10, M = 1, 2
           READ (12, *) (GAM (M,J), J = 1, NP)
   10   CONTINUE
        DO 20, K = 1, NV
           DO 30, M = 1, 2
              READ (12, *) (ALPHA (M,K,J), J = 1, NP)
              READ (12, *) (PSI (M,K,J), J = 1, NP)
   30      CONTINUE
   20   CONTINUE
        READ (12, *) (AMP (I), I = 0, 1)
        READ (12, *) (PHASE (I), I = 1, 1)
        READ (12, *) CV

C       ...... INPUT A & B MATRICES

        DO 80, I = 1, NV
           READ (12, *) (A (I,J), J = 1, NV)
   80   CONTINUE
        DO 180, I = 1, NV
           READ (12, *) (B (I,J), J = 1, NV)
  180   CONTINUE

C       ...... COMPUTE THE MEAN AND STD.DEV. FUNCTION

        DO 40, M = 1, 2
           DO 50, I = 0, NT
              DO 70, K = 1, NV
                 MU (M,K,I) = 0.0
                 SIGMA (M,K,I) = 0.0
                 DO 60, L = 1, NP
```

```
                    MU (M,K,I) = MU (M,K,I) + ALPHA (M,K,L) * PHI (L,I)
                    SIGMA (M,K,I) = SIGMA (M,K,I) + PSI (M,K,L) *
     &                              PHI (L,I)
   60             CONTINUE
   70           CONTINUE
   50         CONTINUE
   40       CONTINUE

            RETURN
            END


C           --------------------------------------------------------------
C           ...... A ROUTINE TO GENERATE PSEUDO RANDOM NUMBERS FROM A
C                  GAUSSIAN DISTRIBUTION WITH A MEAN OF ZERO AND A
C                  STANDARD DEVIATION OF UNITY AS SPECIFIED BY THE USER.
C                  THIS ROUTINE REFERENCES UNIF TO GENERATE THE UNIFORMLY
C                  DISTRIBUTED RANDOM NUMBERS.
C           --------------------------------------------------------------

            SUBROUTINE GRAND2 (NRAND,NV)
C           ----------------------------

            REAL            NRAND (1,NV)

            COMMON          IDUM1,IDUM2,IDUM3,IDUM4,IDUM5,IDUM6,IDUM7

            R1 = URAN1 (IDUM1)
            R2 = URAN1 (IDUM1)
            T = SQRT(-2*LOG(R1))
            NRAND(1,1) = T * SIN (6.283185*R2)
            R1 = URAN2 (IDUM2)
            R2 = URAN2 (IDUM2)
            T = SQRT(-2*LOG(R1))
            NRAND(1,2) = T * SIN (6.283185*R2)
            R1 = URAN3 (IDUM3)
            R2 = URAN3 (IDUM3)
            T = SQRT(-2*LOG(R1))
            NRAND(1,3) = T * SIN (6.283185*R2)
            R1 = URAN4 (IDUM4)
            R2 = URAN4 (IDUM4)
            T = SQRT(-2*LOG(R1))
            NRAND(1,4) = T * SIN (6.283185*R2)
            R1 = URAN5 (IDUM5)
            R2 = URAN5 (IDUM5)
            T = SQRT(-2*LOG(R1))
            NRAND(1,5) = T * SIN (6.283185*R2)
            R1 = URAN6 (IDUM6)
            R2 = URAN6 (IDUM6)
            T = SQRT(-2*LOG(R1))
            NRAND(1,6) = T * SIN (6.283185*R2)
            R1 = URAN7 (IDUM7)
            R2 = URAN7 (IDUM7)
            T = SQRT(-2*LOG(R1))
            NRAND(1,7) = T * SIN (6.283185*R2)

            RETURN
            END
```

```
C       ------------------------------------------------------------
C       ...... THIS SUBROUTINE SOLVES A SYSTEM OF EQUATIONS
C       ------------------------------------------------------------


        SUBROUTINE LINEAR (NPMAX,NP,DER,DER2,THETA)
C       --------------------------------------------------

        REAL                A (13,0:13)
        REAL                DER (NPMAX)
        REAL                DER2 (NPMAX,NPMAX)
        REAL                THETA (NPMAX)

   15   FORMAT (' MATRIX IS SINGULAR')
   25   FORMAT (' NEW PARAMETER ESTIMATES: ',F10.4)

        OPEN (UNIT=9,FILE='LPT1')

C       ...... THIS SETS UP THE A MATRIX WHICH IS USED IN SOLVING
C              THE SYSTEM OF EQUATIONS

        DO 10, I = 1, NP
           A (I,0) = DER (I)
           DO 20, J = 1, NP
              A (I,J) = DER2 (I,J)
   20      CONTINUE
   10   CONTINUE

C       ...... THIS SOLVES THE SYSTEM OF EQUATIONS
C              THE DIFFERENCE BETWEEN THE VALUE OF THETA(Q) IN THIS
C              ITERATION AND THE PREVIOUS ITERATION IS STORED IN A(Q,0)

        DO 30, I1 = 1, NP
           I2 = I1
           T1 = 0
           DO 40, I3 = I1, NP
              IF (ABS (A (I3,I1)) .GT. (ABS (T1))) THEN
                 I2 = I3
                 T1 = A (I3,I1)
              ENDIF
   40      CONTINUE
           IF (T1 .EQ. 0) THEN
              PRINT 15
              STOP
           ENDIF
           IF (I2 .NE. I1) THEN
              DO 50, IO = 0, NP
                 TEMP = A (I1,IO)
                 A (I1,IO) = A (I2,IO)
                 A (I2,IO) = TEMP
   50         CONTINUE
           ENDIF
           T2 = 1 / (A (I1,I1))
           NQ = NP
```

```
          DO 60, I4 = 0, NQ
             A (I1,I4) = A (I1,I4) * T2
 60       CONTINUE
          DO 70, I3 = 1, NP
             IF (I1 .NE. I3) THEN
                T2 = A (I3,I1)
                A (I3,0) = A (I3,0)-A(I1,0)* T2
                DO 80, I0 = I1, NP
                   A(I3,I0) = A(I3,I0) - A(I1,I0) * T2
 80             CONTINUE
             ENDIF
 70       CONTINUE
 30    CONTINUE

C      ..... THIS EXTRACTS THE NEW PARAMETER VALUES

       DO 90, I = 1, NP
          THETA(I) = THETA(I) - A(I,0)
          WRITE (9,25) THETA(I)
 90    CONTINUE

       RETURN
       END



C      ----------------------------------------------------------
C      ...... THIS SUBROUTINE COMPUTES TOTAL MEANS AND STD DEVS
C             TO BE USED IN THE COMPUTATION OF CROSS-CORRELATIONS
C      ----------------------------------------------------------

       SUBROUTINE AVSTD3 (CLIMA,AVEG,DEV,NY,NT,NV)
C      -----------------------------------------

       INTEGER          DENOM (7)
       REAL             CLIMA (NV,NY*NT)
       REAL             AVEG (NV)
       REAL             DEV (NV)

       DO 10, K = 1, NV
          AVEG (K) = 0.0
          DEV (K) = 0.0
          DENOM (K) = 0
 10    CONTINUE

       DO 30, K = 1, NV
          DO 20, I = 1, NY*NT
             IF (CLIMA (K,I) .GT. -900) THEN
                AVEG (K) = AVEG (K) + CLIMA (K,I)
                DEV (K) = DEV (K) + (CLIMA (K,I) ** 2)
                DENOM (K) = DENOM (K) + 1
             ENDIF
 20       CONTINUE
       DEV (K) =SQRT( (DEV(K)-((AVEG(K)**2)/DENOM(K)))/DENOM(K))
       AVEG (K) = AVEG(K)/DENOM(K)
       PRINT *, AVEG (K)
       PRINT *, DEV (K)
 30    CONTINUE

       RETURN
       END
```

```
C         -------------------------------------------------------------
C         ...... THIS SUBROUTINE COMPUTES THE CHOLESKY DECOMPOSITION
C                OF A MATRIX
C         -------------------------------------------------------------


          SUBROUTINE CHOLESKY (DECOMP,CORR,NV)
C         -------------------------------------

          REAL                 CORR (NV,NV)
          REAL                 DECOMP (NV,NV)

    5     FORMAT (7F10.3)

C         ...... FILL IN SYMMETRICAL PART OF MATRIX

          DO 20, I = 1, NV-1
             DO 30, J = I+1, NV
                CORR (J,I) = CORR (I,J)
   30        CONTINUE
   20     CONTINUE

C         ...... COMPUTE CHOLESKY DECOMPOSITION

          DO 40, I = 1, NV
             DO 50, J = 1, NV
                DECOMP (I,J) = 0
   50        CONTINUE
   40     CONTINUE
          DECOMP (1,1) = SQRT (CORR (1,1))
          DO 60, J = 2, NV
             DECOMP (J,1) = CORR (1,J) / DECOMP (1,1)
   60     CONTINUE
          DO 70, I = 2, NV-1
             TERM = 0
             DO 80, J = 1, I-1
                TERM = TERM + DECOMP (I,J) ** 2
   80        CONTINUE
             DECOMP (I,I) = SQRT (CORR (I,I) - TERM)
             DO 90, J = I+1, NV
                TERM = 0
                DO 100, K = 1, I-1
                   TERM = TERM + DECOMP (I,K) * DECOMP (J,K)
  100           CONTINUE
                DECOMP (J,I) = (CORR (I,J) - TERM) / DECOMP (I,I)
   90        CONTINUE
   70     CONTINUE
          TERM = 0
          DO 110, J = 1, NV-1
             TERM = TERM + DECOMP (NV,J) ** 2
  110     CONTINUE
          DECOMP (NV,NV) = SQRT (CORR (NV,NV) - TERM)

          RETURN
          END
```

```
C        -------------------------------------------------------------
C        ...... THIS SUBROUTINE GENERATES CLIMATE SEQUENCES ACCORDING
C               TO THE SPECIFICATIONS OF MODEL5
C        -------------------------------------------------------------


         SUBROUTINE MOD5 (RAND,STATE,NV5,NV,SIGMA5,MU,RAU5,K,J,OBSN,
C        ------------------------------------------------------------
     &                           PSTATE,NT,NRAU)
C                                ---------------


         INTEGER           PSTATE, STATE
         REAL              RAU5 (NRAU,NV5)
         REAL              RAND (1,NV)
         REAL              OBSN (NV)
         REAL              SIGMA5 (2,NV5,0:NT)
         REAL              MU (2,NV,0:NT)

    5    FORMAT (7F10.3)

         IF ((STATE .EQ. 1) .AND. (PSTATE .EQ. 1)) THEN
            JJ = 1
         ELSEIF ((STATE .EQ. 2) .AND. (PSTATE .EQ. 2)) THEN
            JJ = 2
         ELSEIF ((STATE .EQ. 1) .AND. (PSTATE .EQ. 2)) THEN
            JJ = 3
         ELSEIF ((STATE .EQ. 2) .AND. (PSTATE .EQ. 1)) THEN
            JJ = 4
         ENDIF

         IF (J-1 .EQ. 0) THEN
            L = NT
         ELSE
            L = J-1
         ENDIF

         IF (K.EQ.2) THEN
            KK = 1
         ENDIF

         OBSN (K) = SIGMA5(PSTATE,KK,J)*(RAND(1,K)+RAU5(JJ,KK)*(OBSN(K)-
     &              MU(STATE,K,L))/SIGMA5(STATE,KK,L))+MU(PSTATE,K,J)

         RETURN
         END
```

```
C         ----------------------------------------------------------------
C         ...... THIS SUBROUTINE GENERATES CLIMATE SEQUENCES ACCORDING
C                TO THE SPECIFICATIONS OF MODEL3
C         ----------------------------------------------------------------


          SUBROUTINE MOD3 (RAND,STATE,NV3,NV,SIGMA3,MU,RAU3,K,J,OBSN,
C         ---------------------------------------------------------------
     &                          PSTATE,NT,NRAU)
C                               --------------

          INTEGER           PSTATE, STATE
          REAL              RAU3 (NRAU,NV3)
          REAL              RAND (1,NV)
          REAL              OBSN (NV)
          REAL              SIGMA3 (NRAU,NV3)
          REAL              MU (2,NV,0:NT)

    5     FORMAT (7F10.3)

          IF ((STATE .EQ. 1) .AND. (PSTATE .EQ. 1)) THEN
             JJ = 1
          ELSEIF ((STATE .EQ. 2) .AND. (PSTATE .EQ. 2)) THEN
             JJ = 2
          ELSEIF ((STATE .EQ. 1) .AND. (PSTATE .EQ. 2)) THEN
             JJ = 3
          ELSEIF ((STATE .EQ. 2) .AND. (PSTATE .EQ. 1)) THEN
             JJ = 4
          ENDIF

          IF (J-1 .EQ. 0) THEN
             L = NT
          ELSE
             L = J-1
          ENDIF

          IF (K.EQ.1) THEN
             KK = 1
          ELSEIF (K.EQ.3) THEN
             KK = 2
          ELSEIF (K.EQ.4) THEN
             KK = 3
          ELSEIF (K.EQ.7) THEN
             KK = 4
          ENDIF

          OBSN (K) = SIGMA3(JJ,KK)*(RAND(1,K)+RAU3(JJ,KK)*(OBSN(K)-
     &               MU(STATE,K,L)) / SIGMA3(JJ,KK))+MU(PSTATE,K,J)

          RETURN
          END
```

```
C     ---------------------------------------------------------
C     ...... THIS SUBROUTINE COMPUTES THE AMPLITUDE & PHASE
C            REPRESENTATION
C     ---------------------------------------------------------

      SUBROUTINE AMPHA (AM,PH,THETA,NPMAX,KMAX,K,PI,NT)
C     ------------------------------------------------------

      REAL              AM(0:KMAX)
      REAL              PH(KMAX)
      REAL              THETA(NPMAX)

  45  FORMAT (/, ' AMPLITUDE: ')
  55  FORMAT (/, ' PHASE: ')
  65  FORMAT (9F8.3)

      OPEN (UNIT=9,FILE='LPT1')

      AM(0) = THETA(1)
      DO 140, I = 1, K
         TA = THETA (2*I)
         TB = THETA(2*I+1)
         AM(I) = SQRT(TA**2 + TB**2)
         IF (TA .LT. 0) THEN
            PH(I) = ATAN(TB / TA) + PI
         ELSEIF (TA .EQ. 0) THEN
            IF (TB .GE. 0) THEN
               PH(I) = 0.5 * PI
            ELSE
               PH(I) = 1.5 *PI
            ENDIF
         ELSEIF (TA .GT. 0) THEN
            IF (TB .GE. 0) THEN
               PH(I) = ATAN (TB/TA)
            ELSE
               PH(I) = ATAN(TB/TA) + 2 * PI
            ENDIF
         ENDIF
         PH(I) = PH(I) * NT / (2*PI*I)
 140  CONTINUE

      WRITE (9,45)
      WRITE (9,65) (AM(I),I=0,K)
      WRITE (9,55)
      WRITE (9,65) (PH(I),I=1,K)

      RETURN
      END
```

```
C       ------------------------------------------------------------------
C       THIS SUBROUTINE COMPUTES THE MATRIX OF SIN AND COS TERMS FOR
C       THE FOURIER TRANSFORMATION. THIS DIFFERS FROM THE SUBROUTINE
C       COSSIN IN THAT HERE THE MATRIX PHI HAS DIMENSION GIVEN BY:
C               ---> PHI (NT,NPARM)
C       ------------------------------------------------------------------


        SUBROUTINE TRIG (PHI,NPARM,NT)
C       ------------------------------

        REAL            PI
        PARAMETER       (PI = 3.14159265)
        REAL            PHI (NT,NPARM)
        REAL            THETA
        REAL            OMEGA
        INTEGER         T

        OMEGA = 2 * PI / NT
        K = (NPARM - 1) / 2
        DO 10, T = 1, NT
           PHI (T,1) = 1
   10   CONTINUE
        DO 20, J = 1, K
           J1 = 2 * J
           J2 = J1 + 1
           THETA = OMEGA * J
           A = 2 * COS (THETA)
           PHI (1,J1) = 1
           PHI (2,J1) = A / 2
           PHI (1,J2) = 0
           PHI (2,J2) = SIN (THETA)
           DO 30, T = 3, NT
              PHI (T,J1) = A * PHI (T-1,J1) - PHI (T-2,J1)
              PHI (T,J2) = A * PHI (T-1,J2) - PHI (T-2,J2)
   30      CONTINUE
   20   CONTINUE
        RETURN
        END



C       ------------------------------------------------------------------
C       ...... SUBROUTINE TO COMPUTE THE TRANSPOSE OF A MATRIX.  THE
C              RESULT IS WRITTEN INTO THE SAME MATRIX.
C       ------------------------------------------------------------------

        SUBROUTINE GTRANP (DECOMP,NV)
C       -----------------------------

        REAL                DECOMP (NV,NV)

        DO 10, I = 1, NV
           DO 20, J = I+1, NV
              TEMP1 = DECOMP (I,J)
              TEMP2 = DECOMP (J,I)
              DECOMP (I,J) = TEMP2
              DECOMP (J,I) = TEMP1
   20      CONTINUE
   10   CONTINUE

        RETURN
        END
```

```
C         ----------------------------------------
C         ...... SUBROUTINE TO BRACKET THE MINIMUM
C         ----------------------------------------


          SUBROUTINE BRACK (A,B,C,FA,FB,FC,DIM1)
C         ---------------------------------------

          PARAMETER         (GLD=1.618034)
          PARAMETER         (GLIM=100.)
          PARAMETER         (T=1.E-20)

          OPEN (UNIT=9,FILE='LPT1')

          FA=DIM1(A)
          FB=DIM1(B)
          WRITE (9,*) ' FA FB', FA, FB
          IF (FB.GT.FA) THEN
             DUM=A
             A=B
             B=DUM
             DUM=FB
             FB=FA
             FA=DUM
          ENDIF
          C=B+GLD*(B-A)
          FC=DIM1(C)
          WRITE (9,*) ' FC', FC
1         IF (FB.GE.FC) THEN
             R=(B-A)*(FB-FC)
             Q=(B-C)*(FB-FA)
             U=B-((B-C)*Q-(B-A)*R)/(2.*SIGN(MAX(ABS(Q-R),T),Q-R))
             ULIM=B+GLIM*(C-B)
             IF ((B-U)*(U-C).GT.0.) THEN
                FU=DIM1(U)
                IF (FU.LT.FC) THEN
                   A=B
                   FA=FB
                   B=U
                   FB=FU
                   GOTO 1
                ELSEIF (FU.GT.FB) THEN
                   C=U
                   FC=FU
                   GOTO 1
                ENDIF
                U=C+GLD*(C-B)
                FU=DIM1(U)
             ELSEIF ((C-U)*(U-ULIM).GT.0.) THEN
                FU=DIM1(U)
                IF (FU.LT.FC) THEN
                   B=C
                   C=U
                   U=C+GLD*(C-B)
```

```
                FB=FC
                FC=FU
                FU=DIM1(U)
             ENDIF
          ELSEIF ((U-ULIM)*(ULIM-C).GE.0.) THEN
             U=ULIM
             FU=DIM1(U)
          ELSE
             U=C+GLD*(C-B)
             FU=DIM1(U)
          ENDIF
          A=B
          B=C
          C=U
          FA=FB
          FB=FC
          FC=FU
          GOTO 1
       ENDIF

       RETURN
       END

C        ----------------------------------------
C        ...... SUBROUTINE TO MULTIPLY TWO MATRICES
C        ----------------------------------------


       SUBROUTINE MULT (FIRST,SECOND,THIRD,ROWX,COLX,ROWA,COLA)
C      ----------------------------------------------------------

       INTEGER          ROWX,COLX,ROWA,COLA,TEST
       REAL             FIRST (ROWX,COLX)
       REAL             SECOND (ROWA,COLA)
       REAL             THIRD (ROWX,COLA)

       TEST=1
       DO 40, KK = 1, ROWA
          DO 50, JJ = 1, COLA
             IF (SECOND (KK,JJ) .EQ. -999.0) THEN
                TEST=0
             ENDIF
   50     CONTINUE
   40  CONTINUE

       IF (TEST.EQ.1) THEN
          IF (COLX .NE. ROWA) THEN
             PRINT *, 'MATRICES ARE NOT COMPATIBLE'
          ELSE
             DO 10, I = 1, ROWX
                DO 20, J = 1, COLA
                   THIRD (I,J) = 0
                   DO 30, K = 1, COLX
                      THIRD(I,J) = THIRD(I,J)+FIRST(I,K)*SECOND(K,J)
   30              CONTINUE
   20           CONTINUE
   10        CONTINUE
          ENDIF
       ENDIF

       RETURN
       END
```

```
C        ------------------------------------
C        ...... FUNCTION TO FIND LOCAL MINIMUM
C        ------------------------------------


      FUNCTION BMIN (AA,BB,C,DIM1,EPS,XMIN)
C     ------------------------------------

      INTEGER           MAXITER
      PARAMETER         (MAXITER=100)
      PARAMETER         (CG=.3819660)
      PARAMETER         (T=1.0E-10)
      REAL              HALF

      A=MIN(AA,C)
      B=MAX(AA,C)
      V=BB
      W=V
      X=V
      E=0.
      FX=DIM1(X)
      FV=FX
      FW=FX

      DO 10, I=1,MAXITER
         HALF=0.5*(A+B)
         TOL=EPS*ABS(X)+T
         T2=2.*TOL
         IF (ABS(X-HALF).LE.(T2-.5*(B-A))) THEN
            GOTO 3
         ENDIF
         IF (ABS(E).GT.TOL) THEN
            R=(X-W)*(FX-FV)
            Q=(X-V)*(FX-FW)
            P=(X-V)*Q-(X-W)*R
            Q=2.*(Q-R)
            IF (Q.GT.0.) THEN
               P=-P
            ELSE
               Q=-Q
            ENDIF
            TEMP=E
            E=D
            IF ((ABS(P).GE.ABS(.5*Q*TEMP)).OR.(P.LE.Q*(A-X)).OR.
     &         (P.GE.Q*(B-X))) THEN
               GOTO 1
            ENDIF
            D=P/Q
            U=X+D
            IF ((U-A.LT.T2).OR.(B-U.LT.T2)) THEN
               D=SIGN(TOL,HALF-X)
            ENDIF
            GOTO 2
         ENDIF
```

```
1           IF (X.GE.HALF) THEN
               E=A-X
            ELSE
               E=B-X
            ENDIF
            D=CG*E
2           IF (ABS(D).GE.TOL) THEN
               U=X+D
            ELSE
               U=X+SIGN(TOL,D)
            ENDIF
            FU=DIM1(U)
            IF (FU.LE.FX) THEN
               IF (U.LT.X) THEN
                  B=X
               ELSE
                  A=X
               ENDIF
               V=W
               FV=FW
               W=X
               FW=FX
               X=U
               FX=FU
            ELSE
               IF (U.LT.X) THEN
                  A=U
               ELSE
                  B=U
               ENDIF
               IF ((FU.LE.FW).OR.(W.EQ.X)) THEN
                  V=W
                  FV=FW
                  W=U
                  FW=FU
               ELSEIF ((FU.LE.FV).OR.(V.EQ.X).OR.(V.EQ.W)) THEN
                  V=U
                  FV=FU
               ENDIF
            ENDIF
 10      CONTINUE

         PRINT *, 'DID NOT CONVERGE '
3        XMIN=X
         BMIN=FX

         RETURN
         END
```

```
C        --------------------------------------------------------------
C        ...... THIS SUBROUTINE COMPUTES THE VECTOR OF FIRST DERIVATIVES
C               FOR MODEL 3 FOR USAGE IN NUMERICAL RECIPES.
C        --------------------------------------------------------------


         SUBROUTINE DFUNC (THETA,DER)
C        --------------------------
         INTEGER         T,P,NV,NY,NT,NP,NPARM,NRAU
         PARAMETER       (NV=6,NY=7,NT=365,NP=14,NPARM=3,NRAU=4)

         COMMON          K,ICOUNT(NRAU,NY),ISEQ(NRAU,NY,NT),CLIMA(NY,0:NT),
     &                   ALPHA(2,NV,NPARM),SIGMA(NRAU,NV),PHI(NPARM,0:NT),
     &                   RAU(NRAU,NV),ISCALE(3,NV)

         DIMENSION       THETA (NP), DER (NP)

         REAL            MU (2,0:NT)
         REAL            MIDDLE

C        ...... UPDATE PARAMETER ESTIMATES

         DO 160 J = 1, NPARM
            ALPHA (1,K,J) = THETA (J)/ISCALE(3,K)
            ALPHA (2,K,J) = THETA (J+3)/ISCALE(3,K)
  160    CONTINUE

         DO 170, J = 1, NRAU
            SIGMA (J,K) = THETA (J+6)/ISCALE(2,K)
            RAU (J,K) = THETA (J+10)/ISCALE(1,K)
  170    CONTINUE

         DO 10, M = 1, 2
            DO 20, I = 0, NT
               MU (M,I) = 0.0
               DO 30, L = 1, NPARM
                  MU (M,I) = MU (M,I) + ALPHA (M,K,L) * PHI (L,I)
   30          CONTINUE
   20       CONTINUE
   10    CONTINUE

         DO 40, I = 1, NP
            DER (I) = 0.0
   40    CONTINUE

         DO 50, LL = 1, NPARM
            DO 60, M = 1, 2
               IF (M .EQ. 1) THEN
                  N = 2
                  NN = 1
                  J = 3
                  KK = 4
               ELSEIF (M .EQ. 2) THEN
                  N = 1
```

```
                NN = 2
                J = 4
                KK = 3
             ENDIF

C       ...... THE VARIABLE DER1 COMPUTES THE DERIVATIVE FOR THE MEAN
C              FUNCTION

             DER1 = 0
             DO 70, IY = 1, NY
                DO 80, T = 1, ICOUNT (M,IY)
                   P = ISEQ (M,IY,T)
                   IF ((CLIMA(IY,P).NE.-999).AND.(CLIMA(IY,P-1).NE.-999
     &                )) THEN
                   MIDDLE = ((CLIMA(IY,P)-MU(M,P))/SIGMA(M,K)-RAU(M,K)*
     &                      ((CLIMA(IY,P-1)-MU(M,P-1))/SIGMA(M,K)))
                   DER1 = DER1+MIDDLE*(-PHI(LL,P)/SIGMA(M,K)+RAU(M,K)*
     &                    PHI(LL,P-1)/SIGMA(M,K))
                   ENDIF
   80           CONTINUE

                DO 90, T = 1, ICOUNT (J,IY)
                   P = ISEQ (J,IY,T)
                   IF ((CLIMA(IY,P).NE.-999).AND.(CLIMA(IY,P-1).NE.-999
     &                )) THEN
                   MIDDLE = ((CLIMA(IY,P)-MU(N,P))/SIGMA(J,K)-RAU(J,K)*
     &                      ((CLIMA(IY,P-1)-MU(NN,P-1))/SIGMA(J,K)))
                   DER1 = DER1+MIDDLE*(RAU(J,K)*PHI(LL,P-1)/SIGMA
     &                    (J,K))
                   ENDIF
   90           CONTINUE

                DO 100, T = 1, ICOUNT (KK,IY)
                   P = ISEQ (KK,IY,T)
                   IF ((CLIMA(IY,P).NE.-999).AND.(CLIMA(IY,P-1).NE.-999
     &                )) THEN

                   MIDDLE=((CLIMA(IY,P)-MU(NN,P))/SIGMA(KK,K)-RAU(KK,K)
     &                    *((CLIMA(IY,P-1)-MU(N,P-1))/SIGMA(KK,K)))
                   DER1 = DER1+MIDDLE*(-PHI(LL,P)/SIGMA(KK,K))
                   ENDIF
  100           CONTINUE
   70        CONTINUE

             IF (M .EQ. 1) THEN
                DER (LL) = -DER1
             ELSEIF (M .EQ. 2) THEN
                DER(LL+3) = -DER1
             ENDIF

   60     CONTINUE
   50   CONTINUE

C       ...... THE DERIVATIVE WITH RESPECT TO THE AUTOCORRELATION
```

```
C               COEFFICIENT IS COMPUTED AS WELL AS THE DERIVATIVE
C               W.R.T. THE STANDARD DEVIATIONS

        DO 110, IY = 1, NY
           DO 120, T = 1, ICOUNT (1,IY)
              P = ISEQ (1,IY,T)
              IF ((CLIMA(IY,P).NE.-999).AND.(CLIMA(IY,P-1).NE.-999
     &            )) THEN
              PART1 = (CLIMA(IY,P)-MU(1,P))/SIGMA(1,K)
              PART2 = (CLIMA(IY,P-1)-MU(1,P-1))/SIGMA(1,K)
              MIDDLE = PART1-RAU(1,K)*PART2
              DER(11) = DER(11)+MIDDLE*PART2
              PART1 = -((CLIMA(IY,P)-MU(1,P))/SIGMA(1,K)**2)
              PART2 = ((CLIMA(IY,P-1)-MU(1,P-1))/SIGMA(1,K)**2)
              DER(7) = DER(7)-MIDDLE*(PART1+RAU(1,K)*PART2)-1/SIGMA(1,K)
              ENDIF
  120      CONTINUE

           DO 130, T = 1, ICOUNT (2,IY)
              P = ISEQ (2,IY,T)
              IF ((CLIMA(IY,P).NE.-999).AND.(CLIMA(IY,P-1).NE.-999
     &            )) THEN
              PART1 = (CLIMA(IY,P)-MU(2,P))/SIGMA(2,K)
              PART2 = (CLIMA(IY,P-1)-MU(2,P-1))/SIGMA(2,K)
              MIDDLE = PART1-RAU(2,K)*PART2
              DER(12) = DER(12)+MIDDLE*PART2
              PART1 = -((CLIMA(IY,P)-MU(2,P))/SIGMA(2,K)**2)
              PART2 = ((CLIMA(IY,P-1)-MU(2,P-1))/SIGMA(2,K)**2)
              DER(8) = DER(8)-MIDDLE*(PART1+RAU(2,K)*PART2)-1/SIGMA(2,K)
              ENDIF
  130      CONTINUE

           DO 140, T = 1, ICOUNT (3,IY)
              P = ISEQ (3,IY,T)
              IF ((CLIMA(IY,P).NE.-999).AND.(CLIMA(IY,P-1).NE.-999
     &            )) THEN
              PART1 = (CLIMA(IY,P)-MU(2,P))/SIGMA(3,K)
              PART2 = (CLIMA(IY,P-1)-MU(1,P-1))/SIGMA(3,K)
              MIDDLE = PART1-RAU(3,K)*PART2
              DER(13) = DER(13)+MIDDLE*PART2
              PART1 = -((CLIMA(IY,P)-MU(2,P))/SIGMA(3,K)**2)
              PART2 = ((CLIMA(IY,P-1)-MU(1,P-1))/SIGMA(3,K)**2)
              DER(9) = DER(9)-MIDDLE*(PART1+RAU(3,K)*PART2)-1/SIGMA(3,K)
              ENDIF
  140      CONTINUE

           DO 150, T = 1, ICOUNT (4,IY)
              P = ISEQ (4,IY,T)
              IF ((CLIMA(IY,P).NE.-999).AND.(CLIMA(IY,P-1).NE.-999
     &            )) THEN
              PART1 = (CLIMA(IY,P)-MU(1,P))/SIGMA(4,K)
              PART2 = (CLIMA(IY,P-1)-MU(2,P-1))/SIGMA(4,K)
              MIDDLE = PART1-RAU(4,K)*PART2
              DER(14) = DER(14)+MIDDLE*PART2
```

```
           PARTI = -((CLIMA(IY,P)-MU(1,P))/SIGMA(4,K)**2)
           PART2 = ((CLIMA(IY,P-1)-MU(2,P-1))/SIGMA(4,K)**2)
           DER(10)=DER(10)-MIDDLE*(PART1+RAU(4,K)*PART2)-1/SIGMA(4,K)
           ENDIF
 150     CONTINUE
 110   CONTINUE

       DO 200, I = 1, NP
          DER(I) = -DER(I)
 200   CONTINUE

       RETURN
       END




C      ---------------------------------------------
C      ...... SUBROUTINE TO MINIMIZE ALONG A LINE
C      ---------------------------------------------


       SUBROUTINE MINL (THETA,DER,NP,FMIN)
C      -----------------------------------

       INTEGER           NPMAX
       PARAMETER         (NPMAX=20)
       PARAMETER         (EPS=1.E-4)

       EXTERNAL DIM1
       DIMENSION         THETA(NP),DER(NP)
       COMMON   /ONE/  NPP,THET(NPMAX),DERI(NPMAX)

       OPEN (UNIT=9,FILE='LPT1')

       NPP=NP
       DO 10, J=1,NP
          THET(J)=THETA(J)
          DERI(J)=DER(J)
 10    CONTINUE

       A=0.
       B=1.
       C=2.

       CALL BRACK (A,B,C,FA,FB,FC,DIM1)
       FMIN=BMIN (A,B,C,DIM1,EPS,XMIN)
       DO 20, J=1,NP
          DER(J)=XMIN*DER(J)
          THETA(J)=THETA(J)+DER(J)
 20    CONTINUE

       RETURN
       END
```

```
C        ----------------------------------------------------------------
C        ...... THIS SUBROUTINE COMPUTES THE VECTOR OF FIRST DERIVATIVES
C               FOR MODEL 4 FOR USAGE IN NUMERICAL RECIPES.
C        ----------------------------------------------------------------


         SUBROUTINE DFUNC (THETA,DER)
C        ----------------------------
         INTEGER          T,P,NV,NY,NT,NP,NPARM,NRAU
         PARAMETER        (NV=6,NY=7,NT=365,NP=13,NPARM=3,NRAU=1)

         COMMON           K,ICOUNT(4,NY),ISEQ(4,NY,NT),CLIMA(NY,0:NT),
     &                    ALPHA(2,NV,NPARM),PSI(2,NV,NPARM),PHI(NPARM,0:NT),
     &                    RAU(NRAU,NV),ISCALE(3,NV)

         DIMENSION        THETA (NP), DER (NP)

         REAL             MU (2,0:NT)
         REAL             SIGMA (2,0:NT)
         REAL             MIDDLE

C        ...... UPDATE PARAMETER ESTIMATES

         DO 160 J = 1, NPARM
            ALPHA (1,K,J) = THETA (J)/ISCALE(3,K)
            ALPHA (2,K,J) = THETA (J+3)/ISCALE(3,K)
            PSI (1,K,J) = THETA (J+6)/ISCALE(2,K)
            PSI (2,K,J) = THETA (J+9)/ISCALE(2,K)
 160     CONTINUE

         DO 170, J = 1, NRAU
            RAU (J,K) = THETA (J+12)/ISCALE(1,K)
 170     CONTINUE

         DO 10, M = 1, 2
            DO 20, I = 0, NT
               MU (M,I) = 0.0
               SIGMA (M,I) = 0.0
               DO 30, L = 1, NPARM
                  MU (M,I) = MU (M,I) + ALPHA (M,K,L) * PHI (L,I)
                  SIGMA (M,I) = SIGMA (M,I) + PSI (M,K,L) * PHI (L,I)
  30           CONTINUE
  20        CONTINUE
  10     CONTINUE

         DO 40, I = 1, NP
            DER (I) = 0.0
  40     CONTINUE

         DO 850, LL = 1, NPARM
            DO 870, M = 1, 2
               IF (M .EQ. 1) THEN
                  N = 2
                  NN = 1
```

```
            J = 3
            KK = 4
         ELSEIF (M .EQ. 2) THEN
            N = 1
            NN = 2
            J = 4
            KK = 3
         ENDIF
C     ...... THE VARIABLE DER1 COMPUTES THE DERIVATIVE FOR THE MEAN
C            FUNCTION, WHILE DER2 AND DER3 COMPUTE THE DERIVATIVE FOR
C            THE VARIANCE FUNCTION

         DER2 = 0
         DER1 = 0
         DER3 = 0
         DO 310, IY = 1, NY
            DO 330, T = 1, ICOUNT (M,IY)
              P = ISEQ (M,IY,T)
              IF ((CLIMA(IY,P).NE.-999).AND.(CLIMA(IY,P-1).NE.
     &             -999)) THEN
               MIDDLE = ((CLIMA(IY,P)-MU(M,P))/SIGMA(M,P)-RAU(1,K)*
     &                 ((CLIMA(IY,P-1)-MU(M,P-1))/SIGMA(M,P-1)))
               DER1 = DER1+MIDDLE*(-PHI(LL,P)/SIGMA(M,P)+RAU(1,K)*
     &              PHI(LL,P-1)/SIGMA(M,P-1))
               PART1 = (-((CLIMA(IY,P)-MU(M,P))/SIGMA(M,P)**2)*
     &                PHI(LL,P))
               PART2 = ((CLIMA(IY,P-1)-MU(M,P-1))/SIGMA(M,P-1)**2)
     &                *PHI(LL,P-1)
               DER3 = DER3+MIDDLE*(PART1+RAU(1,K)*PART2)
              ENDIF
              DER2 = DER2+PHI(LL,P)/SIGMA(M,P)
 330        CONTINUE

            DO 350, T = 1, ICOUNT (J,IY)
              P = ISEQ (J,IY,T)
              IF ((CLIMA(IY,P).NE.-999).AND.(CLIMA(IY,P-1).NE.
     &             -999)) THEN
               MIDDLE = ((CLIMA(IY,P)-MU(N,P))/SIGMA(N,P)-RAU(1,K)*
     &                 ((CLIMA(IY,P-1)-MU(NN,P-1))/SIGMA(NN,P-1)))
               DER1 = DER1+MIDDLE*(RAU(1,K)*PHI(LL,P-1)/SIGMA
     &              (NN,P-1))
               PART2 = ((CLIMA(IY,P-1)-MU(NN,P-1))/SIGMA(NN,P-1)
     &                **2)*PHI(LL,P-1)
               DER3 = DER3+MIDDLE*(RAU(1,K)*PART2)
              ENDIF
 350        CONTINUE

            DO 360, T = 1, ICOUNT (KK,IY)
              P = ISEQ (KK,IY,T)
              IF ((CLIMA(IY,P).NE.-999).AND.(CLIMA(IY,P-1).NE.
     &             -999)) THEN
               MIDDLE =((CLIMA(IY,P)-MU(NN,P))/SIGMA(NN,P)-RAU(1,K)
     &                 *((CLIMA(IY,P-1)-MU(N,P-1))/SIGMA(N,P-1)))
```

```
                  DER1 = DER1+MIDDLE*(-PHI(LL,P)/SIGMA(NN,P))
                  PART1 = (-((CLIMA(IY,P)-MU(NN,P))/SIGMA(NN,P)**2)*
     &                     PHI(LL,P))
                  DER3 = DER3+MIDDLE*PART1
                 ENDIF
                 DER2 = DER2+PHI(LL,P)/SIGMA(NN,P)
  360          CONTINUE
  310        CONTINUE

             IF (M .EQ. 1) THEN
                DER (LL) = -DER1
                DER' (LL+6) = (-DER3-DER2)
             ELSEIF (M .EQ. 2) THEN
                DER(LL+3) = -DER1
                DER(LL+9) = (-DER3-DER2)
             ENDIF
  870      CONTINUE
  850   CONTINUE

C       ...... THE DERIVATIVE WITH RESPECT TO THE AUTOCORRELATION
C              COEFFICIENT IS COMPUTED

        DER (NP) = 0
        DO 420, IY = 1, NY
           DO 700, T = 1, ICOUNT (1,IY)
              P = ISEQ (1,IY,T)
              IF ((CLIMA(IY,P).NE.-999).AND.(CLIMA(IY,P-1).NE.-999))
     &           THEN
               PART1 = (CLIMA(IY,P)-MU(1,P))/SIGMA(1,P)
               PART2 = (CLIMA(IY,P-1)-MU(1,P-1))/SIGMA(1,P-1)
               DER(NP) = DER(NP)+(PART1-RAU(1,K)*PART2)*PART2
              ENDIF
  700      CONTINUE

           DO 701, T = 1, ICOUNT (2,IY)
              P = ISEQ (2,IY,T)
              IF ((CLIMA(IY,P).NE.-999).AND.(CLIMA(IY,P-1).NE.-999))
     &           THEN
               PART1 = (CLIMA(IY,P)-MU(2,P))/SIGMA(2,P)
               PART2 = (CLIMA(IY,P-1)-MU(2,P-1))/SIGMA(2,P-1)
               DER(NP) = DER(NP)+(PART1-RAU(1,K)*PART2)*PART2
              ENDIF
  701      CONTINUE

           DO 702, T = 1, ICOUNT (3,IY)
              P = ISEQ (3,IY,T)
              IF ((CLIMA(IY,P).NE.-999).AND.(CLIMA(IY,P-1).NE.-999))
     &           THEN
               PART1 = (CLIMA(IY,P)-MU(2,P))/SIGMA(2,P)
               PART2 = (CLIMA(IY,P-1)-MU(1,P-1))/SIGMA(1,P-1)
               DER(NP) = DER(NP)+(PART1-RAU(1,K)*PART2)*PART2
              ENDIF
  702      CONTINUE
```

```
          DO 703, T = 1, ICOUNT (4,IY)
             P = ISEQ (4,IY,T)
             IF ((CLIMA(IY,P).NE.-999).AND.(CLIMA(IY,P-1).NE.-999))
     &            THEN
              PART1 = (CLIMA(IY,P)-MU(1,P))/SIGMA(1,P)
              PART2 = (CLIMA(IY,P-1)-MU(2,P-1))/SIGMA(2,P-1)
              DER(NP) = DER(NP)+(PART1-RAU(1,K)*PART2)*PART2
             ENDIF
 703      CONTINUE
 420   CONTINUE

       DO 200, I = 1, NP
          DER(I) = -DER(I)
 200   CONTINUE

       RETURN
       END




C      ----------------------------------------------------------------
C      THIS SUBROUTINE COMPUTES THE MATRIX OF SIN AND COS TERMS FOR
C      THE FOURIER TRANSFORMATION
C      ----------------------------------------------------------------


       SUBROUTINE COSSIN (PHI,NPARM,NT)
C      --------------------------------

       REAL            PI
       PARAMETER       (PI = 3.14159265)
       REAL            PHI (NPARM,0:NT)
       REAL            THETA
       REAL            OMEGA
       INTEGER         T

       OMEGA = 2 * PI / NT
       K = (NPARM - 1) / 2
       DO 10, T = 1, NT
          PHI (1,T) = 1
 10    CONTINUE
       DO 20, J = 1, K
          J1 = 2 * J
          J2 = J1 + 1
          THETA = OMEGA * J
          A = 2 * COS (THETA)
          PHI (J1,1) = 1
          PHI (J1,2) = A / 2
          PHI (J2,1) = 0
          PHI (J2,2) = SIN (THETA)
          DO 30, T = 3, NT
             PHI (J1,T) = A * PHI (J1,T-1) - PHI (J1,T-2)
             PHI (J2,T) = A * PHI (J2,T-1) - PHI (J2,T-2)
 30       CONTINUE
 20    CONTINUE

       RETURN
       END
```

```
C          ---------------------------------------------------------------
C          ...... THIS SUBROUTINE COMPUTES THE VECTOR OF FIRST DERIVATIVES
C                 FOR MODEL 5 FOR USAGE IN NUMERICAL RECIPES.
C          ---------------------------------------------------------------


      SUBROUTINE DFUNC (THETA,DER)
C     ----------------------------
      INTEGER         T,P,NV,NY,NT,NP,NPARM,NRAU
      PARAMETER       (NV=6,NY=7,NT=365,NP=16,NPARM=3,NRAU=4)

      COMMON          K,ICOUNT(4,NY),ISEQ(4,NY,NT),CLIMA(NY,0:NT),
     &                ALPHA(2,NV,NPARM),PSI(2,NV,NPARM),PHI(NPARM,0:NT),
     &                RAU(NRAU,NV),ISCALE(3,NV)

      DIMENSION       THETA (NP), DER (NP)

      REAL            MU (2,0:NT)
      REAL            SIGMA (2,0:NT)
      REAL            MIDDLE

C     ...... UPDATE PARAMETER ESTIMATES

      DO 160 J = 1, NPARM
         ALPHA (1,K,J) = THETA (J)/ISCALE(3,K)
         ALPHA (2,K,J) = THETA (J+3)/ISCALE(3,K)
         PSI (1,K,J) = THETA (J+6)/ISCALE(2,K)
         PSI (2,K,J) = THETA (J+9)/ISCALE(2,K)
 160  CONTINUE

      DO 170, J = 1, NRAU
         RAU (J,K) = THETA (J+12)/ISCALE(1,K)
 170  CONTINUE

      DO 10, M = 1, 2
         DO 20, I = 0, NT
            MU (M,I) = 0.0
            SIGMA (M,I) = 0.0
            DO 30, L = 1, NPARM
               MU (M,I) = MU (M,I) + ALPHA (M,K,L) * PHI (L,I)
               SIGMA (M,I) = SIGMA (M,I) + PSI (M,K,L) * PHI (L,I)
  30        CONTINUE
  20     CONTINUE
  10  CONTINUE

      DO 40, I = 1, NP
         DER (I) = 0.0
  40  CONTINUE

      DO 850, LL = 1, NPARM
         DO 870, M = 1, 2
            IF (M .EQ. 1) THEN
               N = 2
               NN = 1
```

```
               J = 3
               KK = 4
            ELSEIF (M .EQ. 2) THEN
               N = 1
               NN = 2
               J = 4
               KK = 3
            ENDIF
C      ...... THE VARIABLE DER1 COMPUTES THE DERIVATIVE FOR THE MEAN
C             FUNCTION, WHILE DER2 AND DER3 COMPUTE THE DERIVATIVE FOR
C             THE VARIANCE FUNCTION

            DER2 = 0
            DER1 = 0
            DER3 = 0
            DO 50, IY = 1, NY
               DO 330, T = 1, ICOUNT (M,IY)
                  P = ISEQ (M,IY,T)
                  IF ((CLIMA(IY,P).NE.-999).AND.(CLIMA(IY,P-1).NE.
     &               -999)) THEN
                  MIDDLE = ((CLIMA(IY,P)-MU(M,P))/SIGMA(M,P)-RAU(M,K)*
     &                     ((CLIMA(IY,P-1)-MU(M,P-1))/SIGMA(M,P-1)))
                  DER1 = DER1+MIDDLE*(-PHI(LL,P)/SIGMA(M,P)+RAU(M,K)*
     &                   PHI(LL,P-1)/SIGMA(M,P-1))
                  PART1 = (-((CLIMA(IY,P)-MU(M,P))/SIGMA(M,P)**2)*
     &                    PHI(LL,P))
                  PART2 = ((CLIMA(IY,P-1)-MU(M,P-1))/SIGMA(M,P-1)**2)
     &                    *PHI(LL,P-1)
                  DER3 = DER3+MIDDLE*(PART1+RAU(M,K)*PART2)
                  ENDIF
                  DER2 = DER2+PHI(LL,P)/SIGMA(M,P)
  330          CONTINUE

               DO 350, T = 1, ICOUNT (J,IY)
                  P = ISEQ (J,IY,T)
                  IF ((CLIMA(IY,P).NE.-999).AND.(CLIMA(IY,P-1).NE.
     &               -999)) THEN
                  MIDDLE = ((CLIMA(IY,P)-MU(N,P))/SIGMA(N,P)-RAU(J,K)*
     &                     ((CLIMA(IY,P-1)-MU(NN,P-1))/SIGMA(NN,P-1)))
                  DER1 = DER1+MIDDLE*(RAU(J,K)*PHI(LL,P-1)/SIGMA
     &                   (NN,P-1))
                  PART2 = ((CLIMA(IY,P-1)-MU(NN,P-1))/SIGMA(NN,P-1)
     &                    **2)*PHI(LL,P-1)
                  DER3 = DER3+MIDDLE*(RAU(J,K)*PART2)
                  ENDIF
  350          CONTINUE

               DO 360, T = 1, ICOUNT (KK,IY)
                  P = ISEQ (KK,IY,T)
                  IF ((CLIMA(IY,P).NE.-999).AND.(CLIMA(IY,P-1).NE.
     &               -999)) THEN
                  MIDDLE=((CLIMA(IY,P)-MU(NN,P))/SIGMA(NN,P)-RAU(KK,K)
     &                   *((CLIMA(IY,P-1)-MU(N,P-1))/SIGMA(N,P-1)))
```

```
                     DER1 = DER1+MIDDLE*(-PHI(LL,P)/SIGMA(NN,P))
                     PART1 = (-((CLIMA(IY,P)-MU(NN,P))/SIGMA(NN,P)**2)*
     &                       PHI(LL,P))
                     DER3 = DER3+MIDDLE*PART1
                     ENDIF
                     DER2 = DER2+PHI(LL,P)/SIGMA(NN,P)
 360              CONTINUE
  50           CONTINUE

               IF (M .EQ. 1) THEN
                  DER (LL) = -DER1
                  DER (LL+6) = (-DER3-DER2)
               ELSEIF (M .EQ. 2) THEN
                  DER(LL+3) = -DER1
                  DER(LL+9) = (-DER3-DER2)
               ENDIF

 870        CONTINUE
 850     CONTINUE

C        ...... THE DERIVATIVE WITH RESPECT TO THE AUTOCORRELATION
C               COEFFICIENT IS COMPUTED

         DO 60, IY = 1, NY
            DO 700, T = 1, ICOUNT (1,IY)
               P = ISEQ (1,IY,T)
               IF ((CLIMA(IY,P).NE.-999).AND.(CLIMA(IY,P-1).NE.-999))
     &            THEN
                  PART1 = (CLIMA(IY,P)-MU(1,P))/SIGMA(1,P)
                  PART2 = (CLIMA(IY,P-1)-MU(1,P-1))/SIGMA(1,P-1)
                  DER(13) = DER(13)+(PART1-RAU(1,K)*PART2)*PART2
               ENDIF
 700        CONTINUE

            DO 701, T = 1, ICOUNT (2,IY)
               P = ISEQ (2,IY,T)
               IF ((CLIMA(IY,P).NE.-999).AND.(CLIMA(IY,P-1).NE.
     &            -999)) THEN
                  PART1 = (CLIMA(IY,P)-MU(2,P))/SIGMA(2,P)
                  PART2 = (CLIMA(IY,P-1)-MU(2,P-1))/SIGMA(2,P-1)
                  DER(14) = DER(14)+(PART1-RAU(2,K)*PART2)*PART2
               ENDIF
 701        CONTINUE

            DO 702, T = 1, ICOUNT (3,IY)
               P = ISEQ (3,IY,T)
               IF ((CLIMA(IY,P).NE.-999).AND.(CLIMA(IY,P-1).NE.
     &            -999)) THEN
                  PART1 = (CLIMA(IY,P)-MU(2,P))/SIGMA(2,P)
                  PART2 = (CLIMA(IY,P-1)-MU(1,P-1))/SIGMA(1,P-1)
                  DER(15) = DER(15)+(PART1-RAU(3,K)*PART2)*PART2
               ENDIF
 702        CONTINUE
```

```
           DO 703, T = 1, ICOUNT (4,IY)
              P = ISEQ (4,IY,T)
              IF ((CLIMA(IY,P).NE.-999).AND.(CLIMA(IY,P-1).NE.
     &           -999)) THEN
                 PART1 = (CLIMA(IY,P)-MU(1,P))/SIGMA(1,P)
                 PART2 = (CLIMA(IY,P-1)-MU(2,P-1))/SIGMA(2,P-1)
                 DER(NP) = DER(NP)+(PART1-RAU(4,K)*PART2)*PART2
              ENDIF
 703       CONTINUE
  60    CONTINUE

        DO 200, I = 1, NP
           DER(I) = -DER(I)
 200    CONTINUE

        RETURN
        END




C       ------------------------------------------------------------
C       ...... SUBROUTINE TO COMPUTE THE INVERSE OF A MATRIX WHEN
C              NOT PUTTING THE SOLUTION INTO THE OLD MATRIX
C       ------------------------------------------------------------

        SUBROUTINE INVT (CLAGO,INV,NV)
C       ------------------------------

        REAL                INV (NV,NV)
        REAL                CLAGO (NV,NV)
        REAL                RESULT (7,7)


        DO 50, K = 1, NV
           DO 60, KK = 1, NV
              INV (K,KK) = CLAGO(K,KK)
 60        CONTINUE
 50     CONTINUE

        DO 10, I = 1, NV
           DIAG = 1 / INV (I,I)
           INV (I,I) = 1
           DO 20, J = 1, NV
              INV (I,J) = INV (I,J) * DIAG
 20        CONTINUE
           DO 30, K = 1, NV
              IF (I .NE. K) THEN
                 DIAG = INV (K,I)
                 INV (K,I) = 0
                 DO 40, J = 1, NV
                    INV (K,J) = INV (K,J) - INV (I,J) * DIAG
 40              CONTINUE
              ENDIF
 30        CONTINUE
 10     CONTINUE

        RETURN
        END
```

```
C        ------------------------------------------------------------
C        ...... SUBROUTINE TO COMPUTE LOG LIKELIHOOD FUNCTION FOR
C               MODEL 3.
C        ------------------------------------------------------------


         FUNCTION FUNC (THETA)
C        ---------------------

         INTEGER         T,NV,NY,NT,NP,NPARM,NRAU
        PARAMETER       (NV=6,NY=7,NT=365,NP=14,NPARM=3,NRAU=4)

         COMMON          K,ICOUNT(NRAU,NY),ISEQ(NRAU,NY,NT),CLIMA(NY,0:NT),
     &                   ALPHA(2,NV,NPARM),SIGMA(NRAU,NV),
     &                   PHI(NPARM,0:NT),RAU(NRAU,NV),ISCALE(3,NV)

         REAL            MU (2,0:NT)
         REAL            LNLIKE,PI
         PARAMETER       (PI=3.141593)

         DIMENSION       THETA(NP)

C        ...... UPDATE PARAMETER ESTIMATES

         DO 160 J = 1, NPARM
            ALPHA (1,K,J) = THETA (J)/ISCALE(3,K)
            ALPHA (2,K,J) = THETA (J+3)/ISCALE(3,K)
 160     CONTINUE

         DO 170, J = 1, NRAU
            SIGMA (J,K) = THETA (J+6)/ISCALE(2,K)
            RAU (J,K) = THETA (J+10)/ISCALE(1,K)
 170     CONTINUE

         DO 10, M = 1, 2
            DO 20, I = 0, NT
               MU (M,I) = 0.0
               DO 30, L = 1, NPARM
                  MU (M,I) = MU (M,I) + ALPHA (M,K,L) * PHI (L,I)
  30           CONTINUE
  20        CONTINUE
  10     CONTINUE

         LNLIKE = 0
         TERM = 0

         DO 40, J = 1, 4
            IF (J .EQ. 1) THEN
               M = 1
               L = 1
            ELSEIF (J .EQ. 2) THEN
               M = 2
               L = 2
            ELSEIF (J .EQ. 3) THEN
```

```
                 M = 2
                 L = 1
              ELSEIF (J .EQ. 4) THEN
                 M = 1
                 L = 2
              ENDIF
              DO 50, I = 1, NY
                 DO 60, KK = 1, ICOUNT (J,I)
                    T = ISEQ (J,I,KK)
                    IF ((CLIMA(I,T).NE.-999).AND.(CLIMA(I,T-1).NE.-999))
     &                 THEN
                       RESID = (CLIMA(I,T)-MU(M,T))/SIGMA(J,K)-RAU(J,K)
     &                         *((CLIMA(I,T-1)-MU(L,T-1))/SIGMA(J,K))
                       LNLIKE = LNLIKE + (RESID)**2
                       TERM = TERM + LOG(SIGMA(J,K))
                    ENDIF
 60              CONTINUE
 50           CONTINUE
 40        CONTINUE

           FUNC = -(-((NY*NT)/2)*LOG(2*PI)-TERM-LNLIKE/2)

           RETURN
           END
```

```
C       -------------------------------------------------------------
C       ...... SUBROUTINE TO COMPUTE LOG LIKELIHOOD FUNCTION FOR
C              MODEL 4.
C       -------------------------------------------------------------


        FUNCTION FUNC (THETA)
C       ---------------------

        INTEGER         T,NV,NY,NT,NP,NPARM,NRAU
        PARAMETER       (NV=6,NY=7,NT=365,NP=13,NPARM=3,NRAU=1)

        COMMON          K,ICOUNT(4,NY),ISEQ(4,NY,NT),CLIMA(NY,0:NT),
     &                  ALPHA(2,NV,NPARM),PSI(2,NV,NPARM),
     &                  PHI(NPARM,0:NT),RAU(NRAU,NV),ISCALE(3,NV)

        REAL            MU (2,0:NT)
        REAL            SIGMA (2,0:NT)
        REAL            LNLIKE,PI
        PARAMETER       (PI=3.141593)

        DIMENSION       THETA(NP)

C       ...... UPDATE PARAMETER ESTIMATES

        DO 160 J = 1, NPARM
           ALPHA (1,K,J) = THETA (J)/ISCALE(3,k)
           ALPHA (2,K,J) = THETA (J+3)/ISCALE(3,K)
           PSI (1,K,J) = THETA (J+6)/ISCALE(2,K)
           PSI (2,K,J) = THETA (J+9)/ISCALE(2,K)
 160    CONTINUE

        DO 170, J = 1, NRAU
           RAU (J,K) = THETA (J+12)/ISCALE(1,K)
 170    CONTINUE

        DO 10, M = 1, 2
           DO 20, I = 0, NT
              MU (M,I) = 0.0
              SIGMA (M,I) = 0.0
              DO 30, L = 1, NPARM
                 MU (M,I) = MU (M,I) + ALPHA (M,K,L) * PHI (L,I)
                 SIGMA (M,I) = SIGMA (M,I) + PSI(M,K,L) * PHI (L,I)
  30          CONTINUE
  20       CONTINUE
  10    CONTINUE

        LNLIKE = 0
        TERM = 0

        DO 40, J = 1, 4
           IF (J .EQ. 1) THEN
              M = 1
              L = 1
```

```
          ELSEIF (J .EQ. 2) THEN
             M = 2
             L = 2
          ELSEIF (J .EQ. 3) THEN
             M = 2
             L = 1
          ELSEIF (J .EQ. 4) THEN
             M = 1
             L = 2
          ENDIF
          DO 50, I = 1, NY
             DO 60, KK = 1, ICOUNT (J,I)
                T = ISEQ (J,I,KK)
                IF ((CLIMA(I,T).NE.-999).AND.(CLIMA(I,T-1).NE.-999))
     &             THEN
                   RESID = (CLIMA(I,T)-MU(M,T))/SIGMA(M,T)-RAU(1,K)
     &                    *((CLIMA(I,T-1)-MU(L,T-1))/SIGMA(L,T-1))
                   LNLIKE = LNLIKE + (RESID)**2
                   TERM = TERM + LOG(SIGMA(M,T))
                ENDIF
   60        CONTINUE
   50     CONTINUE
   40  CONTINUE

       FUNC = -(-((NY*NT)/2)*LOG(2*PI)-TERM-LNLIKE/2)

       RETURN
       END

C      -----------------------------------------------------------
C      ...... THIS SUBROUTINE READS IN INITIAL PARAMETER ESTIMATES
C      -----------------------------------------------------------


       SUBROUTINE INT3 (ALPHA,SIGMA,RAU,NPARM,NV,NRAU,ISCALE)
C      -----------------------------------------------------

       INTEGER           ISCALE (3,NV)
       REAL              ALPHA (2,NV,NPARM)
       REAL              SIGMA (NRAU,NV)
       REAL              RAU (NRAU,NV)

   15  FORMAT (' ALPHA PARAMETERS = ')
   25  FORMAT (' SIGMA PARAMETERS = ')
   35  FORMAT (' RAU PARAMETER = ')

C      OPEN (UNIT=4,FILE='CON')
       OPEN (UNIT=4,FILE='\WATER\DATA\INIT.DAT',STATUS='OLD')

       DO 20, K = 1, NV
          READ (4,*) ISCALE (1,K),ISCALE(2,K), ISCALE(3,K)
          DO 10, M = 1, 2
             READ (4,*) (ALPHA (M,K,I), I = 1, NPARM)
   10     CONTINUE
          READ (4,*) (SIGMA (L,K), L = 1, NRAU)
          READ (4,*) (RAU (L,K), L = 1, NRAU)
   20  CONTINUE

       RETURN
       END
```

```
C         ----------------------------------------------------------
C         ...... SUBROUTINE TO COMPUTE LOG LIKELIHOOD FUNCTION FOR
C                MODEL 5.
C         ----------------------------------------------------------


          FUNCTION FUNC (THETA)
C         ---------------------

          INTEGER         T,NV,NY,NT,NP,NPARM,NRAU
          PARAMETER       (NV=6,NY=7,NT=365,NP=16,NPARM=3,NRAU=4)

          COMMON          K,ICOUNT(4,NY),ISEQ(4,NY,NT),CLIMA(NY,0:NT),
     &                    ALPHA(2,NV,NPARM),PSI(2,NV,NPARM),
     &                    PHI(NPARM,0:NT),RAU(NRAU,NV),ISCALE(3,NV)

          REAL            MU (2,0:NT)
          REAL            SIGMA (2,0:NT)
          REAL            LNLIKE,PI
          PARAMETER       (PI=3.141593)

          DIMENSION       THETA(NP)

C         ...... UPDATE PARAMETER ESTIMATES

          DO 160 J = 1, NPARM
             ALPHA (1,K,J) = THETA (J)/ISCALE(3,K)
             ALPHA (2,K,J) = THETA (J+3)/ISCALE(3,K)
             PSI (1,K,J) = THETA (J+6)/ISCALE(2,K)
             PSI (2,K,J) = THETA (J+9)/ISCALE(2,K)
  160     CONTINUE

          DO 170, J = 1, NRAU
             RAU (J,K) = THETA (J+12)/ISCALE(1,K)
  170     CONTINUE

          DO 10, M = 1, 2
             DO 20, I = 0, NT
                MU (M,I) = 0.0
                SIGMA (M,I) = 0.0
                DO 30, L = 1, NPARM
                   MU (M,I) = MU (M,I) + ALPHA (M,K,L) * PHI (L,I)
                   SIGMA (M,I) = SIGMA (M,I) + PSI(M,K,L) * PHI (L,I)
   30           CONTINUE
   20        CONTINUE
   10     CONTINUE

          LNLIKE = 0
          TERM = 0

          DO 40, J = 1, 4
             IF (J .EQ. 1) THEN
                M = 1
                L = 1
```

```
            ELSEIF (J .EQ. 2) THEN
               M = 2
               L = 2
            ELSEIF (J .EQ. 3) THEN
               M = 2
               L = 1
            ELSEIF (J .EQ. 4) THEN
               M = 1
               L = 2
            ENDIF
            DO 50, I = 1, NY
               DO 60, KK = 1, ICOUNT (J,I)
                  T = ISEQ (J,I,KK)
                  IF ((CLIMA(I,T).NE.-999).AND.(CLIMA(I,T-1).NE.-999))
     &               THEN
                     RESID = (CLIMA(I,T)-MU(M,T))/SIGMA(M,T)-RAU(J,K)
     &                       *((CLIMA(I,T-1)-MU(L,T-1))/SIGMA(L,T-1))
                     LNLIKE = LNLIKE + (RESID)**2
                     TERM = TERM + LOG(SIGMA(M,T))
                  ENDIF
   60          CONTINUE
   50       CONTINUE
   40    CONTINUE

         FUNC = -(-((NY*NT)/2)*LOG(2*PI)-TERM-LNLIKE/2)

         RETURN
         END


C        ------------------------------------------
C        ...... SUBROUTINE TO MULTIPLY TWO MATRICES
C        ------------------------------------------


         SUBROUTINE XNP (FIRST,SECOND,THIRD,ROWX,COLX,ROWA,COLA,RX,
C        ------------------------------------------------------------
     &                        CX,RA,CA)
C                             ---------

         INTEGER              ROWX,COLX,ROWA,COLA,RX,CX,RA,CA
         REAL                 FIRST (RX,CX)
         REAL                 SECOND (RA,CA)
         REAL                 THIRD (RX,CA)

         IF (COLX .NE. ROWA) THEN
            PRINT *, 'MATRICES ARE NOT COMPATIBLE'
         ELSE
            DO 10, I = 1, ROWX
               DO 20, J = 1, COLA
                  THIRD (I,J) = 0
                  DO 30, K = 1, COLX
                     IF (SECOND (K,J) .NE. -999.0) THEN
                        THIRD(I,J) = THIRD(I,J)+FIRST(I,K)*SECOND(K,J)
                     ENDIF
   30             CONTINUE
   20          CONTINUE
   10       CONTINUE
         ENDIF

         RETURN
         END
```

```
      SUBROUTINE INV(MATT,NN,MM)
C     --------------------------

      REAL MATT(NN,NN), INVER(25,25)
      REAL MATR1(25,25)

      II = 0
   20 II = II + 1

      MATR1(II,II) = 1.0 / MATT(II,II)

      DO 40 J = 1, MM
         DO 30 I = 1, MM

            IF (J .EQ. II .AND. I .EQ. II) THEN
               INVER(I,J) = (-1.0) * MATR1(II,II)

            ELSEIF (J .EQ. II .AND. I .NE. II) THEN
               INVER(I,J) = MATT(I,J) * MATR1(II,II)

            ELSEIF (I .EQ. II .AND. J .NE. II) THEN
               INVER(I,J) = MATT(I,J) * MATR1(II,II)

            ELSE
               INVER(I,J) = MATT(I,J) - ((MATT(I,II) *
     &                      MATT(II,J)) * MATR1(II,II))
            ENDIF

   30    CONTINUE
CC       PRINT*, (INVER(I,J), I = 1, MM)
   40 CONTINUE

      CALL COPY(INVER,25,25,MATT,NN,NN,MM,MM)

      IF (II .LT. MM) GO TO 20

      RETURN
      END

C     ---------------------------------------------------------------
C     ...... SUBROUTINE TO GENERATE A VECTOR ACCORDING TO THE MODEL:-
C                      S(t) = ALPHA(i)*PHI(i,t)
C     ---------------------------------------------------------------


      SUBROUTINE GMEAN (MU,PHI,NT,NPARM,ALPHA,K,NV)
C     ---------------------------------------------

      REAL             MU (2,NT)
      REAL             PHI (NT,NPARM)
      REAL             ALPHA (NV,2,NPARM)

      DO 10, M = 1, 2
         DO 20, J = 1, NT
           MU (M,J) = 0.0
           DO 30, I = 1, NPARM
              MU (M,J) = MU (M,J) + ALPHA (K,M,I) * PHI (J,I)
   30      CONTINUE
   20    CONTINUE
   10 CONTINUE
      RETURN
      END
```

```
C         -------------------------------------------------------------
C         ...... THIS SUBROUTINE GENERATES CLIMATE SEQUENCES ACCORDING
C                TO THE SPECIFICATIONS OF MODEL4
C         -------------------------------------------------------------


          SUBROUTINE MOD4 (RAND,STATE,NV4,NV,SIGMA4,MU,RAU4,K,J,OBSN,
C         ------------------------------------------------------------
     &                          PSTATE,NT)
C                               ----------

          INTEGER           PSTATE, STATE
          REAL              RAU4 (NV4)
          REAL              RAND (1,NV)
          REAL              OBSN (NV)
          REAL              SIGMA4 (2,NV4,0:NT)
          REAL              MU (2,NV,0:NT)

    5     FORMAT (7F10.3)

          IF (J-1 .EQ. 0) THEN
             L = NT
          ELSE
             L = J-1
          ENDIF

          IF (K.EQ.5) THEN
             KK = 1
          ELSEIF (K.EQ.6) THEN
             KK = 2
          ENDIF

          OBSN (K) = SIGMA4(PSTATE,KK,J)*(RAND(1,K)+RAU4(KK)*(OBSN(K)-
     &               MU(STATE,K,L))/SIGMA4(STATE,KK,L))+MU(PSTATE,K,J)

          RETURN
          END
```

```
C       ---------------------------------------------------------------
C       ...... THIS SUBROUTINE GENERATES CLIMATE SEQUENCES ACCORDING
C              TO THE SPECIFICATIONS OF MODEL3
C       ---------------------------------------------------------------


        SUBROUTINE MODEL1 (RAND,NV,SIGMA,MU,J,OBSN,PSTATE,NT,A,B,RES)
C       ----------------------------------------------------------------

        INTEGER            PSTATE
        REAL               RAND (NV,1)
        REAL               SOLN (7,1)
        REAL               RES (NV,1)
        REAL               OBSN (NV,1)
        REAL               SIGMA (2,NV,0:NT)
        REAL               MU (2,NV,0:NT)
        REAL               A (NV,NV)
        REAL               B (NV,NV)

  5     FORMAT (7F10.3)

        CALL MULT (B,RAND,SOLN,NV,NV,NV,1)
        CALL MULT (A,OBSN,RES,NV,NV,NV,1)

        DO 10, K = 1, NV
           OBSN (K,1) = RES (K,1) + SOLN (K,1)
 10     CONTINUE

        DO 20, K = 1, NV
           RES (K,1) = OBSN (K,1)*SIGMA(PSTATE,K,J)+MU(PSTATE,K,J)
 20     CONTINUE

        RETURN
        END


C       ---------------------------------------------------
C       ...... SUBROUTINE TO COMPUTE THE INVERSE OF A MATRIX
C       ---------------------------------------------------

        SUBROUTINE INVNP (NP,SOLN,NPARM)
C       --------------------------------

        REAL          SOLN (NPARM,NPARM)

        DO 10, I = 1, NP
           DIAG = 1 / SOLN (I,I)
           SOLN (I,I) = 1
           DO 20, J = 1, NP
              SOLN (I,J) = SOLN (I,J) * DIAG
 20        CONTINUE
           DO 30, K = 1, NP
              IF (I .NE. K) THEN
                 DIAG = SOLN (K,I)
                 SOLN (K,I) = 0
                 DO 40, J = 1, NP
                    SOLN (K,J) = SOLN (K,J) - SOLN (I,J) * DIAG
 40              CONTINUE
              ENDIF
 30        CONTINUE
 10     CONTINUE

        RETURN
        END
```

```
C        -----------------------------------------------------------
C        ...... THIS SUBROUTINE READS IN INITIAL PARAMETER ESTIMATES,
C               THE CONVERGENCE CRITERION AND THE MAXIMUM NUMBER OF
C               ITERATIONS TO BE PERFORMED
C        -----------------------------------------------------------

         SUBROUTINE INITIAL (EPS,MAXITER,ALPHA,PSI,RAU,NPARM,NV,NRAU)
C        -----------------------------------------------------------

         REAL              ALPHA (2,NV,NPARM)
         REAL              PSI (2,NV,NPARM)
         REAL              RAU (NRAU,NV)

    5    FORMAT (' EPS, MAXITER = ')
   15    FORMAT (' ALPHA PARAMETERS = ')
   25    FORMAT (' PSI PARAMETERS = ')
   35    FORMAT (' RAU PARAMETER = ')

C        OPEN (UNIT=4,FILE='CON')
         OPEN (UNIT=4,FILE='\WATER\DATA\INIT.DAT',STATUS='OLD')

         READ (4,*) EPS, MAXITER
         DO 20, K = 1, NV
            DO 10, M = 1, 2
               READ (4,*) (ALPHA (M,K,I), I = 1, NPARM)
               READ (4,*) (PSI (M,K,I), I = 1, NPARM)
   10       CONTINUE
            READ (4,*) (RAU (L,K), L = 1, NRAU)
   20    CONTINUE

         RETURN
         END
```

```
C        ------------------------------------------------------------
C        ...... THIS SUBROUTINE READS IN INITIAL PARAMETER ESTIMATES,
C               THE CONVERGENCE CRITERION AND THE MAXIMUM NUMBER OF
C               ITERATIONS TO BE PERFORMED
C        ------------------------------------------------------------

         SUBROUTINE INT4 (ALPHA,PSI,RAU,NPARM,NV,NRAU,ISCALE)
C        ------------------------------------------------------

         INTEGER          ISCALE (3,NV)
         REAL             ALPHA (2,NV,NPARM)
         REAL             PSI (2,NV,NPARM)
         REAL             RAU (NRAU,NV)

   15    FORMAT (' ALPHA PARAMETERS = ')
   25    FORMAT (' PSI PARAMETERS = ')
   35    FORMAT (' RAU PARAMETER = ')

C        OPEN (UNIT=4,FILE='CON')
         OPEN (UNIT=4,FILE='\WATER\DATA\INIT.DAT',STATUS='OLD')

         DO 20, K = 1, NV
            READ (4,*) ISCALE (1,K), ISCALE (2,K), ISCALE(3,K)
            DO 10, M = 1, 2
               READ (4,*) (ALPHA (M,K,I), I = 1, NPARM)
               READ (4,*) (PSI (M,K,I), I = 1, NPARM)
   10       CONTINUE
            READ (4,*) (RAU (L,K), L = 1, NRAU)
   20    CONTINUE

         RETURN
         END
```

```
C          -----------------------------------------------------------
C          . .... THIS SUBROUTINE GENERATES RANDOM NORMAL NUMBERS WITH
C                 MEAN ZERO AND STD. DEV. OF 1 AND THEN IS TRANSFORMED
C                 INTO A RANDOM NUMBER WITH STD. DEV. OF SIGMA.
C          -----------------------------------------------------------


           SUBROUTINE GAUSS (DECOMP,RAND)
C          ------------------------------

           INTEGER          NV
           PARAMETER        (NV=7)
           INTEGER          ROWX,ROWA,COLX,COLA
           REAL             DECOMP (NV,NV)
           REAL             NRAND (1,NV)
           REAL             RAND (1,NV)

    5      FORMAT (7F10.3)

C          ...... GENERATE RANDOM NORMAL (0,1) NUMBER

           CALL GRAND2 (NRAND,NV)

C          ...... GENERATE RANDOM NORMAL (0,S) NUMBER

           ROWX = 1
           ROWA = NV
           COLX = NV
           COLA = NV
           CALL MULT (NRAND,DECOMP,RAND,ROWX,COLX,ROWA,COLA)

           RETURN
           END
```

```
C         ---------------------------------------------------------------------
C         ...... SUBROUTINE TO GENERATE A VECTOR ACCORDING TO THE MODEL:-
C                         S(t) = ALPHA(i)*PHI(i,t) &
C                         S(t) = PSI(i)*PHI(i,t)
C         ---------------------------------------------------------------------

      SUBROUTINE GAVSTD (MU,PHI,NT,NPARM,ALPHA,NV,PSI,SIGMA)
C     -------------------------------------------------------

      REAL                MU (2,NV,NT)
      REAL                SIGMA (2,NV,NT)
      REAL                PHI (NT,NPARM)
      REAL                ALPHA (NV,2,NPARM)
      REAL                PSI (NV,2,NPARM)

      DO 10, M = 1, 2
         DO 20, J = 1, NT
            DO 40, K = 1, NV
               MU (M,K,J) = 0.0
               SIGMA (M,K,J) = 0.0
               DO 30, I = 1, NPARM
                  MU (M,K,J) = MU(M,K,J)+ALPHA(K,M,I)*PHI(J,I)
                  SIGMA(M,K,J) = SIGMA(M,K,J)+PSI(K,M,I)*PHI(J,I)
 30            CONTINUE
 40         CONTINUE
 20      CONTINUE
 10   CONTINUE

      RETURN
      END
```

```
C         ------------------------------------------------------------
C         ...... SUBROUTINE TO COMPUTE RESIDUAL SERIES FOR MODEL 4
C         ------------------------------------------------------------


          SUBROUTINE M4RES (RAU,ALPHA,PSI,PHI,COUNT,SEQ,CLIMA,
C         --------------------------------------------------------
     &                           NT,NY,NPARM,NV,K,NRAU,NP,CONVG)
C                                --------------------------------

          INTEGER           COUNT (4,NY)
          INTEGER           SEQ (4,NY,NT)
          INTEGER           T,CONVG
          REAL              AKAIKE,LNLIKE,PI
          PARAMETER         (PI=3.141593)
          REAL              CLIMA (NY,0:NT)
          REAL              MU (2,0:365)
          REAL              SIGMA (2,0:365)
          REAL              RESID (7,12,365)
          REAL              PSI (2,NV,NPARM)
          REAL              ALPHA (2,NV,NPARM)
          REAL              PHI (NPARM,0:NT)
          REAL              RAU (NRAU,NV)

  5       FORMAT (7F10.4)
 15       FORMAT (' AKAIKE"S CRITERION FOR VARIABLE:', I4, ' IS:', F30.4)

          OPEN (UNIT=14,FILE='\WATER\DATA\RESI4.DAT',STATUS='UNKNOWN')
          OPEN (UNIT=9,FILE='LPT1')

          IF (CONVG.EQ.0) THEN
             GOTO 250
          ENDIF

          DO 10, M = 1, 2
             DO 20, I = 0, NT
                MU (M,I) = 0.0
                SIGMA (M,I) = 0.0
                DO 30, L = 1, NPARM
                   MU (M,I) = MU (M,I) + ALPHA (M,K,L) * PHI (L,I)
                   SIGMA (M,I) = SIGMA (M,I) + PSI (M,K,L) * PHI (L,I)
 30             CONTINUE
 20          CONTINUE
 10       CONTINUE

          DO 80, I = 1, NY
             DO 90, J = 1, NT
                RESID (K,I,J) = -999.00
 90          CONTINUE
 80       CONTINUE
          LNLIKE = 0
          TERM = 0

          DO 40, J = 1, 4
```

```
      IF (J .EQ. 1) THEN
         M = 1
         L = 1
      ELSEIF (J .EQ. 2) THEN
         M = 2
         L = 2
      ELSEIF (J .EQ. 3) THEN
         M = 2
         L = 1
      ELSEIF (J .EQ. 4) THEN
         M = 1
         L = 2
      ENDIF
      DO 50, I = 1, NY
         DO 60, KK = 1, COUNT (J,I)
            T = SEQ (J,I,KK)
            IF ((CLIMA(I,T).NE.-999).AND.(CLIMA(I,T-1).NE.-999))
&               THEN
            RESID(K,I,T) = (CLIMA(I,T)-MU(M,T))/SIGMA(M,T)-RAU(1,K)
&                          *((CLIMA(I,T-1)-MU(L,T-1))/SIGMA(L,T-1))
            LNLIKE = LNLIKE + (RESID(K,I,T))**2
            ENDIF
            TERM = TERM + LOG(SIGMA(M,T))
60       CONTINUE
50    CONTINUE
40 CONTINUE

   LNLIKE = -((NY*NT)/2)*LOG(2*PI)-TERM-LNLIKE/2
   AKAIKE = -2*LNLIKE+2*NP

   WRITE (9,15) K, AKAIKE

250 IF (K .EQ. 7) THEN
      DO 100, I = 1, NY
         DO 70, T = 1, NT
            WRITE (14,5) (RESID (K,I,T), K = 1, NV)
70       CONTINUE
100   CONTINUE
   ENDIF

   RETURN
   END
```

```
C        -----------------------------------------------------------
C        ...... SUBROUTINE TO COMPUTE RESIDUAL SERIES FOR MODEL 5
C        -----------------------------------------------------------


         SUBROUTINE M5RES (RAU,ALPHA,PSI,PHI,COUNT,SEQ,CLIMA,
C        ------------------------------------------------------
     &                         NT,NY,NPARM,NV,K,NRAU,NP,CONVG)
C                              --------------------------------

         INTEGER              COUNT (4,NY)
         INTEGER              SEQ (4,NY,NT)
         INTEGER              T,CONVG
         REAL                 LNLIKE,AKAIKE,PI
         PARAMETER            (PI=3.141593)
         REAL                 CLIMA (NY,0:NT)
         REAL                 MU (2,0:365)
         REAL                 SIGMA (2,0:365)
         REAL                 RESID (7,12,365)
         REAL                 PSI (2,NV,NPARM)
         REAL                 ALPHA (2,NV,NPARM)
         REAL                 PHI (NPARM,0:NT)
         REAL                 RAU (NRAU,NV)

   5     FORMAT (7F10.4)
  15     FORMAT (' AKAIKE"S CRITERION FOR VARIABLE:', I4, ' IS:', F10.4)

C        OPEN (UNIT=14,FILE='A:RESIDU.DAT',STATUS='UNKNOWN')
         OPEN (UNIT=14,FILE='\WATER\DATA\RESI5.DAT',STATUS='UNKNOWN')
         OPEN (UNIT=9,FILE='LPT1')

         IF (CONVG.EQ.0) THEN
            GOTO 250
         ENDIF

         DO 10, M = 1, 2
            DO 20, I = 0, NT
               MU (M,I) = 0.0
               SIGMA (M,I) = 0.0
               DO 30, L = 1, NPARM
                  MU (M,I) = MU (M,I) + ALPHA (M,K,L) * PHI (L,I)
                  SIGMA (M,I) = SIGMA (M,I) + PSI (M,K,L) * PHI (L,I)
  30           CONTINUE
  20        CONTINUE
  10     CONTINUE

         DO 80, I = 1, NY
            DO 90, J = 1, NT
               RESID (K,I,J) = -999.00
  90        CONTINUE
  80     CONTINUE
         LNLIKE = 0
         TERM = 0
```

```
      DO 40, J = 1, 4
         IF (J .EQ. 1) THEN
            M = 1
            L = 1
         ELSEIF (J .EQ. 2) THEN
            M = 2
            L = 2
         ELSEIF (J .EQ. 3) THEN
            M = 2
            L = 1
         ELSEIF (J .EQ. 4) THEN
            M = 1
            L = 2
         ENDIF

         DO 50, I = 1, NY
            DO 60, KK = 1, COUNT (J,I)
               T = SEQ (J,I,KK)
               IF ((CLIMA(I,T).NE.-999).AND.(CLIMA(I,T-1).NE.-999))
     &              THEN
               RESID(K,I,T) = (CLIMA(I,T)-MU(M,T))/SIGMA(M,T)-
     &                        RAU(J,K)*((CLIMA(I,T-1)-MU(L,T-1))/
     &                        SIGMA(L,T-1))
               LNLIKE = LNLIKE + (RESID(K,I,T))**2
               ENDIF
               TERM = TERM + LOG(SIGMA(M,T))
 60         CONTINUE
 50      CONTINUE
 40   CONTINUE
      LNLIKE = -((NY*NT)/2)*LOG(2*PI)-TERM-LNLIKE/2
      AKAIKE = -2*LNLIKE+2*NP
      WRITE (9,15) K, AKAIKE


250   IF (K .EQ. 7) THEN
         DO 100, I = 1, NY
            DO 70, T = 1, NT
               WRITE (14,5) (RESID (K,I,T), K = 1, NV)
 70         CONTINUE
100      CONTINUE
       ENDIF

      RETURN
      END
```

```
C        -----------------------------------------
C        ...... SUBROUTINE TO MINIMIZE A FUNCTION
C        -----------------------------------------

      SUBROUTINE POLRIB (THETA,NP,TOL,ITER,FMIN)
C     ------------------------------------------

      INTEGER          NPMAX,MAXITER
      PARAMETER        (NPMAX=20)
      PARAMETER        (MAXITER=200)
      PARAMETER        (EPS=1.E-10)
      REAL             NUM

      DIMENSION        THETA(NP),GRAD(NPMAX),DIR(NPMAX),DER(NPMAX)

      OPEN (UNIT=9,FILE='LPT1')

      FTHETA=FUNC(THETA)
      WRITE (9,*) 'FTHETA', FTHETA
      CALL DFUNC(THETA,DER)
      DO 10, J=1,NP
         GRAD(J)=-DER(J)
         DIR(J)=GRAD(J)
         DER(J)=DIR(J)
 10   CONTINUE

      DO 20, I=1,MAXITER
         ITER=I
         CALL MINL (THETA,DER,NP,FMIN)
         WRITE (9,*) 'FMIN', FMIN
         IF (2.*ABS(FMIN-FTHETA).LE.TOL*(ABS(FMIN)+ABS(FTHETA)+EPS))
     &      RETURN
         FTHETA=FUNC(THETA)
         CALL DFUNC(THETA,DER)
         DENOM=0.
         NUM=0.
         DO 40, J=1,NP
            DENOM=DENOM+GRAD(J)**2
            NUM=NUM+(DER(J)+GRAD(J))*DER(J)
 40      CONTINUE
         IF (DENOM.EQ.0.) RETURN
         GAMMA=NUM/DENOM
         DO 50, J=1,NP
            GRAD(J)=-DER(J)
            DIR(J)=GRAD(J)+GAMMA*DIRH(J)
            DER(J)=DIR(J)
 50      CONTINUE
 20   CONTINUE

      PRINT *, 'DID NOT CONVERGE'

      RETURN
      END
```

```
C         ---------------------------------------
C         FUNCTION TO GENERATE A UNIFORM NUMBER
C         ---------------------------------------

          FUNCTION URAN1 (SEED)
C         ---------------------

          DIMENSION R(97)
          PARAMETER (M1=259200,IA1=7141,IC1=54773,RM1=3.8580247E-6)
          PARAMETER (M2=134456,IA2=8121,IC2=28411,RM2=7.4373773E-6)
          PARAMETER (M3=243000,IA3=4561,IC3=51349)

          DATA INIT /0/

          IF (SEED.LT.0.OR.INIT.EQ.0) THEN
             INIT=1
             IX1=MOD(IC1-IDUM,M1)
             IX1=MOD(IA1*IX1+IC1,M1)
             IX2=MOD(IX1,M2)
             IX1=MOD(IA1*IX1+IC1,M1)
             IX3=MOD(IX1,M3)
             DO 10 J = 1,97
                IX1=MOD(IA1*IX1+IC1,M1)
                IX2=MOD(IA2*IX2+IC2,M2)
                R(J)=(FLOAT(IX1)+FLOAT(IX2)*RM2)*RM1
 10          CONTINUE
             SEED=1
          ENDIF
          IX1=MOD(IA1*IX1+IC1,M1)
          IX2=MOD(IA2*IX2+IC2,M2)
          IX3=MOD(IA3*IX3+IC3,M3)
          J=1+(97*IX3)/M3
          IF (J.GT.97.OR.J.LT.1) PAUSE
          URAN1=R(J)
          R(J)=(FLOAT(IX1)+FLOAT(IX2)*RM2)*RM1

          RETURN
          END
```

```
C        ------------------------------------------------------------------
C        ...... THIS SUBROUTINE READS IN INITIAL PARAMETER ESTIMATES,
C               THE CONVERGENCE CRITERION AND THE MAXIMUM NUMBER OF
C               ITERATIONS TO BE PERFORMED
C        ------------------------------------------------------------------

         SUBROUTINE INTAL3 (EPS,MAXITER,ALPHA,SIGMA,RAU,NPARM,NV,NRAU)
C        ------------------------------------------------------------------

         REAL             ALPHA (2,NV,NPARM)
         REAL             SIGMA (NRAU,NV)
         REAL             RAU (NRAU,NV)

    5    FORMAT (' EPS, MAXITER = ')
   15    FORMAT (' ALPHA PARAMETERS = ')
   25    FORMAT (' SIGMA PARAMETERS = ')
   35    FORMAT (' RAU PARAMETER = ')

C        OPEN (UNIT=4,FILE='CON')
         OPEN (UNIT=4,FILE='\WATER\DATA\INIT.DAT',STATUS='OLD')

         READ (4,*) EPS, MAXITER
         DO 20, K = 1, NV
            DO 10, M = 1, 2
               READ (4,*) (ALPHA (M,K,I), I = 1, NPARM)
   10       CONTINUE
            READ (4,*) (SIGMA (L,K), L = 1, NRAU)
            READ (4,*) (RAU (L,K), L = 1, NRAU)
   20    CONTINUE

         RETURN
         END
```