c

RRB-87-23

PROGRAMS FOR THE ANALYSIS OF TIME SERIES WITH MISSING OBSERVATIONS. USER MANUAL FOR THE PROGRAMS CORRCAL AND KALMAN

by
Eivind Damsleth

National Water Research Institute
River Research Branch
Canada Centre for Inland Waters
Burlington, Ontario, L7R 4A6

June 1987
NWRI Contribution #87-~~103~~ 131

Abstract

This manual presents the Fortran programming codes for fitting time series models for irregularly spaced data which are commonly encountered in environmental work, especially water quality monitoring. Two sets of interactive programs are given and their use is illustrated on data from Turkey Lakes Watershed. One of the programs computes the autocorrelation and crosscorrelation function for time series with missing observations, while the other uses KALMAN filtering techniques for estimating the parameters in the univariate and multivariate models. Although the programs are written in Fortran 5 for a CDC-180 computer system, they are easy to adopt for other computer systems.

Résumé

Dans ce manuel, on présente les codes de programmation Fortran qui servent à ajuster les modèles en série chronologique aux données à espacement irrégulier courantes dans les travaux sur l'environnement, surtout ceux ayant trait au contrôle de la qualité de l'eau. On donne deux ensembles de programmes d'interaction ainsi que leur utilisation sur des données obtenues dans le bassin hydrographique de Turkey Lake. L'un des programmes calcule la fonction d'autocorrélation et de cross-corrélation des modèles en série chronologique avec les observations manquantes, tandis que l'autre utilise les techniques de filtrage KALMAN pour estimer les paramètres des modèles à une variable et à plusieurs variables. Les programmes sont écrits en Fortran 5 pour un ordinateur CDC-180, mais ils s'adaptent facilement à d'autres ordinateurs.

Management Perspective

Many irregularly spaced time series are collected for the purpose of monitoring the changes in the status of aquatic ecosystems. One major difficulty is encountered at the stage of performing the statistical analysis, namely most techniques available require equally spaced observations. Damsleth (1987) extended the use of equally spaced time series techniques to the case of irregular time series. This manual describes the required Fortran programs for modelling irregular time series.

Perspective-gestion

De nombreuses données en série chronologique espacées de façon irrégulière sont prélevées afin de contrôler les variations qui se produisent dans les écosystèmes aquatiques. On se heurte à une principale difficulté au moment d'effectuer l'analyse statistique, notamment le fait que la plupart des techniques actuelles nécessitent des observations espacées également. Damsleth (1987) a étendu l'utilisation des techniques de modélisation de données en série chronologique à espacement égal aux données à espacement irrégulier. Ce manuel décrit les programmes Fortran nécessaires pour modéliser les séries chronologiques irrégulières.

1. Introduction.

In sampling environmental time series, irregularly spaced data is the rule more than the exeption. Irregularly spaced data can be regarded as a special case of a time series with a (short) basic spacing between the observations, but with a lot of observations missing. The analysis of time series with missing data requires special tools, since all the standard techniques are developed for the situation where the time series is completely observed.

Two interactive Fortran programs, CORRCAL and KALMAN, have been developed to perform the analysis of such series. CORRCAL computes the autocorrelation and crosscorrelation function for time series with missing observations, while KALMAN estimates the parameters in a univariate ARIMA-model or a multi-input transfer function model, using the method of maximum likelihood. This manual serves as a users guide to these two programs.

Section 2 describes the data input, which is common to both the programs. Section 3 gives a short description of program CORRCAL and its use, while Section 4 covers program KALMAN. The major part of this note consists of a number of appendixes, which gives examples of the data files, examples of the use of the programs and of the resulting output. A listing of the source code ends the note.

The programs are written in standard Fortran 5 for a CDC-180 computer operating under NOS 2.5, but they should be fairly portable and easy to transfer to another computer.

2. Data input.

In addition to the interactive input supplied by the user during program execution, both programs require two input files. One contains a description of the data structure, while the second contains the actual data values.

2.1 The description file.

The description file describes the data structure, and has the following layout:

Line 1 : Identification heading. Up to 60 characters, left justified, which will be used as heading in the printed output.

Line 2 : Number of time series in the data file, in free format.
" 3 : Name of the first time series. Up to 8 characters, left justified.
" 4 : " second " " "
. . . . .
. . . . .
" m+2: " m'th " " "
where m is the number of series given in line 2.
" m+3: Fortran format for reading one data record, where a record contains the values for all the m time series at a given point in time.

Appendix A.1 gives an example of a description file. The name of the description file must be supplied to the program by the user.

2.2 The data file.

The data file contains the actual data values. Each record contains the values for all the time series at one point in time, and the records must be ordered sequentially in time, so that the first record covers time point no. 1, the second record time point no. 2, etc. The number of records in the file equals the length of the time series.

The records must be written to match the Fortran format given in the last line of the description file. Missing values must be given a unique code which is less than all the observed values, since all values less than or equal to this code will be regarded as missing.

Important. The present version of the programs restricts the number of series to be ten or less, and the number of observations in each series must be less than or equal to 1500, missing values included.

An example of (parts of) a data file is given in Appendix A.2. Note that in the example each record contains a sequence number and the date, in addition to the data values. This information is not used by the programs, and will be skipped when the file is read according to the format in the description file.

The programs will prompt for the name of the data file.

3. Program CORRCAL.

3.1 Purpose.

Program CORRCAL computes the autocorrelation for a time series and/or the corsscorrelation function between pair of time series, where the time series may contain missing observations.

3.2 Method.

3.2.1 Autocorrelation function.

Let $\{X_t\}$, t=1,...,n denote a time series, and define the inicator series $\{C_t\}$

$$C_t = \begin{cases} 1 & \text{if } X_t > X_{lim} \\ 0 & \text{if } X_t \leq X_{lim} \end{cases} \quad (1)$$

If X_{lim} equals the missing value code, then C_t is an indicator of whether X_t is observed or not. If X_{lim} equals zero then C_t indicates whether a "real" value is observed. If, for example, X_t is a series of daily precipitation, C_t will indicate the rainy days if $X_{lim} = 0$. The program computes the autocorrelation function twice, once with X_{lim} equal to the missing data code and once with $X_{lim} = 0$, even if the result from the two computations may some times be identical.

The lag k (k > 0) autocorrelation is computed as

$$r_k = \sum_{k+1}^{n} C_t C_{t-k}(X_t - \bar{X}_k)(X_{t-k} - \bar{X}_k) / S_k^2 \quad (2)$$

where the mean and the sum of squares are given by

$$\bar{X}_k = \left(\sum_{1}^{k} C_t C_{t+k} X_t + \sum_{k+1}^{n} C_t C_{t-k} X_t\right) / \left(\sum_{1}^{k} C_t C_{t+k} + \sum_{k+1}^{n} C_t C_{t-k}\right)$$

$$S_k^2 = \sum_{1}^{k} C_t C_{t+k}(X_t - \bar{X}_k)^2 + \sum_{k+1}^{n} C_t C_{t-k}(X_t - \bar{X}_k)^2$$

Note that the mean and variance is computed using only those values which contribute to r_k. This ensures that $|r_k| < 1$.

3.2.2 Crosscorrelation function.

Let $\{X_{1t}\}$ and $\{X_{2t}\}$ denote the two time series, and define the two indicator series $\{C_{1t}\}$ and $\{C_{2t}\}$ as in the autocorrelation case, using (1). When $k \geq 0$, the lag k crosscorrelation is computed as

$$r_k = \sum_{k+1}^{n} C_t C_{t-k}(X_{1t}-\bar{X}_{1k})(X_{2t-k}-\bar{X}_{2k}) / \sqrt{S_{1k}^2 S_{2k}^2} \quad (3)$$

where the means and sums of squares are given by

$$\bar{X}_{1k} = \sum_{k+1}^{n} C_t C_{t-k} X_{1t} \Big/ \sum_{k+1}^{n} C_t C_{t-k}$$

$$\bar{X}_{2k} = \sum_{1}^{n-k} C_{t+k} C_t X_{2t} \Big/ \sum_{1}^{n-k} C_{t+k} C_t$$

$$S_{1k}^2 = \sum_{k+1}^{n} C_t C_{t-k}(X_{1t}-\bar{X}_{1k})^2$$

$$S_{2k}^2 = \sum_{1}^{n-k} C_{t+k} C_t (X_{2t}-\bar{X}_{2k})^2$$

When $k < 0$ the same formulae applies, but with k replaced by $|k|$ and with $X_{1\cdot}$ and $X_{2\cdot}$ interchanged.

3.3 Program file.

The program is stored as symbolic code in the file CORRCAL, which contains the program itself as well as the subroutine CCORR, which is called by the program. The program requires no other routines.

3.4 Running the program.

The running of the program is best explained by the example given in Appendix B.1, with explanatory comments.

3.5 Output from the program.

CORRCAL writes its output onto a file named CORRES, which is local to the job. If the file does not exist prior to the execution, it will be created by the program. After the run this file may be inspected using an editor, and/or routed to a suitable printer. The CORRES file generated by the example run is shown in Appendix B.2.

4. Program KALMAN.

4.1 Purpose.

Program KALMAN estimates the parameters in a univariate ARIMA-model or a single or multi-input transfer function model, using the method of maximum likelihood. Standard deviations of the estimates are also given, as well as the correlations between the estimates.

4.2 Model.

Let $\{Y_t\}$ denote the output series and $\{X_{1t}\}$, $\{X_{2t}\}$,... the input series. The model can then be written

$$Y_t = \sum_{i=1}^{m} (\nu_{i0} + \nu_{i1}B + \dots + \nu_{is_i}B^{s_i})X_{it-b_i} + n_t \tag{4}$$

$$(1 - \phi_1 B - \dots - \phi_p B^p)(n_t - \mu) = (1 - \theta_1 B - \dots - \theta_q B^q)a_t \tag{5a}$$

or

$$(1 - \phi_1 B - \dots - \phi_p B^p)(1-B)^d n_t = (1 - \theta_1 B - \dots - \theta_q B^q)a_t \tag{5b}$$

where (5a) applies if the noise is assumed to be stationary, while (5b) applies in the non-stationary case.

In (4), m denotes the number of input series. If $m = 0$, then we have an ordinary ARIMA-model. s_i denotes the number of impulse response weights for the i'th input, while b_i gives the delay. n_t denotes a noise series after the "regression" (4), where the noise is not neccessarily white, but follows the ARIMA-model given in (5a) or (5b). μ is the mean of the noise in the stationary case, and d is the number of differences required to obtain stationarity in the non-stationary case. p and q represents the orders of the autoregressive and moving average part of the model, respectively. In (4) and (5), B denotes the backwards shift operator which operates on the time index, so that $B^k Y_t = Y_{t-k}$.

Note that the transfer function model (4) is written on impulse response weight form. Thus, the option to have the transferfunction parameterized as the ratio of two polynomials in B, as advocated by Box & Jenkins (1974). The reason is that the rational form approach creates a number of difficulties in the case of missing data.

Restrictions. In the present version of KALMAN, the following restrictions apply:

$$m \leq 5,\ p \leq 10,\ q \leq 10,\ \sum_{i=1}^{m} s_i + m + p + q \leq 18,\ d \leq 2.$$

4.3 Method.

For a given set of parameters, the likelihood is computed in two steps:

1. The noise series $\{n_t\}$ is calculated, using (4). The noise n_t is regarded as missing if any of the observations (output or input) involved in its computation are missing. In the univariate case $\{n_t\} = \{Y_t\}$, with the same missing values.
2. The likelihood for the ARIMA-model (5) for the noise is then computed using the method from Jones (1980) with the non-stationary extension developed by Damsleth (1987).

The negative of the likelihood thus obtained is used as the objective function to the IMSL library routine ZXMIN, which provides the optimal parameter values.

When the optimal parameter values are found, we use a finite difference technique to approximate the information matrix in the solution point. The inverse of this information matrix provides estimates of the covariance matrix, which in turn gives the standard deviations of the estimates as well as their correlation matrix. Occasionally the parameter estimates are so closely correlated that the information matrix becomes too near singular to be inverted. In this case the standard deviations are given as -1, and a warning is given.

4.4 Program file.

The program is stored as symbolic code in the file KALMAN, which contains the program and the subroutines CNOISE, DDIV, DMULT, DTRAN, FM, FM1, HCAL, INIT, DALREC, RESOUT, SETUP, STABTRA and ZINIT, which are required by the program. In addition the program requires access to the routines in the IMSL-library.

4.5 Running the program.

Before starting the program, the IMSL-library must be attached and made a library. It is further recommended to set the "stop on full screen" option to no, using the (ESC)PG=N command, where (ESC) denotes the escape character. It is also recommended to set the time limit to a fairly large value, using the SETTL command, to avoid frequent interrupts during execution.

The actual running of the program is best described via the example, with comments, given in Appendix C.1.

4.6 Output from the program.

KALMAN writes its output onto a file named ESTRES, which is local to the job. If the file does not exist prior to the execution, it will be created by the program. After the run this file may be insepcted using an editor, and/or routed to a suitable printer. The ESTRES file generated by the example run is presented in Appendix C.2.

Referenced.

Box, G.E.P. and Jenkins, G.M. (1974): Time Series Analysis, Forecasting and Control. Holden Day, San Francisco.

Damsleth, E. (1987): Estimation in non-stationary ARIMA models using the Kalman Filter. Submitted for publication.

Jones, R.H. (1980): Maximum likelihood fitting of ARMA models to time series with missing observations. Technometrics, Vol. 22 No. 3, pp 389-396.

Appendix A.1

Example of a description file.

```
TURKEY LAKE WATERSHED, STATION 1
10
O.RUNOFF
S.RUNOFF
RAIN
SNOWMELT
N.SUPPLY
SNOWPACK
PH
M2+
SO4--
HCO3
(17X,10(1X,F10.3))
```

Appendix A.2

Example of a data file.

1	81-01-01	-1.000	.074	0.000	0.000	0.000	0.000	-1.000	-1.000	-1.000	-1.000
2	81-01-02	-1.000	.121	0.000	0.000	0.000	.290	-1.000	-1.000	-1.000	-1.000
3	81-01-03	-1.000	.150	0.000	0.000	0.000	.290	-1.000	-1.000	-1.000	-1.000
4	81-01-04	-1.000	.130	0.000	0.000	0.000	.290	-1.000	-1.000	-1.000	-1.000
5	81-01-05	-1.000	.228	0.000	0.000	0.000	.880	-1.000	-1.000	-1.000	-1.000
6	81-01-06	-1.000	.218	0.000	0.000	0.000	.880	-1.000	-1.000	-1.000	-1.000
7	81-01-07	-1.000	.209	0.000	0.000	0.000	1.070	-1.000	-1.000	-1.000	-1.000
8	81-01-08	-1.000	.205	0.000	0.000	0.000	1.500	-1.000	-1.000	-1.000	-1.000
9	81-01-09	-1.000	.202	0.000	0.000	0.000	1.760	-1.000	-1.000	-1.000	-1.000
10	81-01-10	-1.000	.171	0.000	0.000	0.000	1.760	-1.000	-1.000	-1.000	-1.000
11	81-01-11	-1.000	.152	0.000	0.000	0.000	1.790	-1.000	-1.000	-1.000	-1.000
12	81-01-12	-1.000	.140	0.000	0.000	0.000	1.850	5.820	105.534	-1.000	53.934
13	81-01-13	-1.000	.133	0.000	0.000	0.000	1.850	-1.000	-1.000	-1.000	-1.000
14	81-01-14	-1.000	.128	0.000	0.000	0.000	1.990	-1.000	-1.000	-1.000	-1.000
15	81-01-15	-1.000	.137	0.000	0.000	0.000	2.030	-1.000	-1.000	-1.000	-1.000
16	81-01-16	-1.000	.124	0.000	0.000	0.000	2.070	-1.000	-1.000	-1.000	-1.000
17	81-01-17	-1.000	.170	0.000	0.000	0.000	2.070	-1.000	-1.000	-1.000	-1.000
18	81-01-18	-1.000	.145	0.000	0.000	0.000	2.070	-1.000	-1.000	-1.000	-1.000
19	81-01-19	-1.000	.128	0.000	0.000	0.000	2.070	-1.000	-1.000	-1.000	-1.000
20	81-01-20	-1.000	.117	0.000	0.000	0.000	2.070	-1.000	-1.000	-1.000	-1.000
21	81-01-21	-1.000	.111	0.000	0.000	0.000	2.070	-1.000	-1.000	-1.000	-1.000
22	81-01-22	-1.000	.107	0.000	0.000	0.000	2.070	-1.000	-1.000	-1.000	-1.000
23	81-01-23	-1.000	.090	0.000	0.000	0.000	2.070	-1.000	-1.000	-1.000	-1.000
24	81-01-24	-1.000	.079	0.000	0.000	0.000	2.070	-1.000	-1.000	-1.000	-1.000
25	81-01-25	-1.000	.072	0.000	0.000	0.000	2.070	-1.000	-1.000	-1.000	-1.000
26	81-01-26	-1.000	.068	0.000	0.000	0.000	2.070	-1.000	-1.000	-1.000	-1.000
27	81-01-27	-1.000	.066	0.000	0.000	0.000	2.250	-1.000	-1.000	-1.000	-1.000
28	81-01-28	-1.000	.064	0.000	0.000	0.000	2.250	5.870	104.034	-1.000	48.033
29	81-01-29	-1.000	.063	0.000	0.000	0.000	2.250	-1.000	-1.000	-1.000	-1.000
	⋮										
1438	84-12-08	.147	.184	0.000	0.000	0.000	1.000	-1.000	-1.000	-1.000	-1.000
1439	84-12-09	.147	.176	0.000	0.000	0.000	1.950	-1.000	-1.000	-1.000	-1.000
1440	84-12-10	.147	.171	0.000	0.000	0.000	2.340	-1.000	-1.000	-1.000	-1.000
1441	84-12-11	.139	.167	0.000	0.000	0.000	2.440	-1.000	-1.000	-1.000	-1.000
1442	84-12-12	.319	.224	0.000	0.000	0.000	2.440	5.880	92.622	60.729	27.541
1443	84-12-13	.298	.386	0.000	0.000	0.000	3.260	-1.000	-1.000	-1.000	-1.000
1444	84-12-14	.269	.340	0.000	0.000	0.000	3.280	-1.000	-1.000	-1.000	-1.000
1445	84-12-15	.260	.271	0.000	0.000	0.000	3.300	-1.000	-1.000	-1.000	-1.000
1446	84-12-16	.462	.561	0.000	0.000	0.000	3.300	-1.000	-1.000	-1.000	-1.000
1447	84-12-17	.857	1.117	0.000	0.000	0.000	4.770	-1.000	-1.000	-1.000	-1.000
1448	84-12-18	.462	.725	0.000	0.000	0.000	6.890	-1.000	-1.000	-1.000	-1.000
1449	84-12-19	.286	.504	0.000	0.000	0.000	8.070	5.690	81.388	53.646	23.279
1450	84-12-20	.223	.411	0.000	0.000	0.000	8.070	-1.000	-1.000	-1.000	-1.000
1451	84-12-21	.181	.429	0.000	0.000	0.000	8.070	-1.000	-1.000	-1.000	-1.000
1452	84-12-22	.185	.406	0.000	0.000	0.000	8.070	-1.000	-1.000	-1.000	-1.000
1453	84-12-23	.185	.353	0.000	0.000	0.000	8.980	-1.000	-1.000	-1.000	-1.000
1454	84-12-24	.176	.356	1.330	.326	1.656	8.654	-1.000	-1.000	-1.000	-1.000
1455	84-12-25	.160	.359	.810	.502	1.312	8.152	-1.000	-1.000	-1.000	-1.000
1456	84-12-26	.147	.250	0.000	0.000	0.000	8.152	-1.000	-1.000	-1.000	-1.000
1457	84-12-27	.134	.254	0.000	0.000	0.000	9.572	5.720	85.961	55.000	28.361
1458	84-12-28	.227	.295	.430	.189	.619	9.383	-1.000	-1.000	-1.000	-1.000
1459	84-12-29	.336	.284	0.000	0.000	0.000	10.403	-1.000	-1.000	-1.000	-1.000
1460	84-12-30	.294	.201	0.000	0.000	0.000	10.913	-1.000	-1.000	-1.000	-1.000
1461	84-12-31	.265	.260	0.000	0.000	0.000	11.423	-1.000	-1.000	-1.000	-1.000

/lgo	Start the CORRCAL program, which in this case has been compiled onto the file LGO.
DESCRIPTION FILE NAME: ? tdesc	TDESC is the name of the description file, which must be prepared by the user prior to execution.
DATA FILE NAME: ? turkey1	TURKEY1 is the name of the data file, which also must be prepared by the user prior to execution.
NUMBER OF DATARECORDS READ: 1461	The program responds with the number of records read from TURKEY1.
MISSING VALUE CODE: ? -1.0	All values less than or equal to -1.0 will be regarded as missing.
NO. OF CORRELATIONS: ? 15	Autocorrelations will be computed for lags 1 - 15, crosscorrelations for lags -15 - 15.
1: O.RUNOFF 2: S.RUNOFF 3: RAIN 4: SNOWMELT 5: N.SUPPLY 6: SNOWPACK 7: PH 8: M2+ 9: SO4-- 10: HCO3	The menu of time series is presented, for reference.
INDEX RANGE FOR FIRST VARIABLE (0,0 TO STOP): ? 10,10 INDEX RANGE FOR SECOND VARIABLE: ? 10,10	First series 10, HCO_3, will be crossed against itself, giving the autocorrelations.
PROCESSING VARIABLE 10 VERSUS 10	Message from the program, to show the progress.
1: O.RUNOFF 2: S.RUNOFF 3: RAIN 4: SNOWMELT 5: N.SUPPLY 6: SNOWPACK 7: PH 8: M2+ 9: SO4-- 10: HCO3	The menu is presented again.
INDEX RANGE FOR FIRST VARIABLE (0,0 TO STOP): ? 1,1 INDEX RANGE FOR SECOND VARIABLE: ? 9,10	Now series 1, O.RUNOFF, will be crossed against series 9, SO_4^{--} and then against series 10, HCO_3.
PROCESSING VARIABLE 1 VERSUS 9 PROCESSING VARIABLE 1 VERSUS 10	Progress report as the program proceeds.
1: O.RUNOFF 2: S.RUNOFF 3: RAIN 4: SNOWMELT 5: N.SUPPLY 6: SNOWPACK 7: PH 8: M2+ 9: SO4-- 10: HCO3	The menu is presented again, to start a new run.
INDEX RANGE FOR FIRST VARIABLE (0,0 TO STOP): ? 0,0 14.358 CP SECONDS EXECUTION TIME. /	However, the 0,0 response terminates the program. The printed output is on the file CORRES.

Appendix B.1

Screen communication, program CORRCAL.

Appendix B.2

Printout of the output file CORRES from program CORRCAL.

TURKEY LAKE WATERSHED, STATION 1

VARIABLE	ALL VALUES MEAN	ALL VALUES VARIANCE	ALL VALUES NO. OF OBS.	VALUES ABOVE 0 ONLY MEAN	VALUES ABOVE 0 ONLY VARIANCE	VALUES ABOVE 0 ONLY NO. OF OBS.
SERIES 1: HCO3	39.993	415.589	256	39.993	415.589	256

LAG	ALL VALUES ACF	ALL VALUES MEAN	ALL VALUES VAR.	NOBS	LAG	VALUES ABOVE 0 ONLY ACF	VALUES ABOVE 0 ONLY MEAN	VALUES ABOVE 0 ONLY VAR.	NOBS
1	.704	31.801	232.343	83	1	.704	31.801	232.343	83
2	.562	32.669	224.953	82	2	.562	32.669	224.953	82
3	.441	33.266	209.302	75	3	.441	33.266	209.302	75
4	.546	33.764	281.243	74	4	.546	33.764	281.243	74
5	.540	35.303	281.950	78	5	.540	35.303	281.950	78
6	.505	38.450	370.374	107	6	.505	38.450	370.374	107
7	.602	37.061	308.579	100	7	.602	37.061	308.579	100
8	.567	40.523	367.937	101	8	.567	40.523	367.937	101
9	.463	38.423	345.599	74	9	.463	38.423	345.599	74
10	.224	36.589	255.665	62	10	.224	36.589	255.665	62
11	.280	34.531	270.277	61	11	.280	34.531	270.277	61
12	.452	36.867	339.380	73	12	.452	36.867	339.380	73
13	.479	39.260	371.408	91	13	.479	39.260	371.408	91
14	.589	42.877	439.464	114	14	.589	42.877	439.464	114
15	.205	36.636	239.312	83	15	.205	36.636	239.312	83

TURKEY LAKE WATERSHED, STATION 1

VARIABLE	ALL VALUES MEAN	ALL VALUES VARIANCE	ALL VALUES NO. OF OBS.	VALUES ABOVE 0 ONLY MEAN	VALUES ABOVE 0 ONLY VARIANCE	VALUES ABOVE 0 ONLY NO. OF OBS.
SERIES 1: O.RUNOFF	.252	.249	1357	.252	.249	1357
SERIES 2: SO4--	62.937	68.804	251	62.937	68.804	251

ALL VALUES

LAG ON S.1	CCF	MEAN-1	VAR.-1	MEAN-2	VAR.-2	NOBS
-15	.295	.400	.339	63.050	69.403	245
-14	.285	.429	.437	63.031	69.057	243
-13	.303	.448	.465	63.031	69.057	243
-12	.246	.433	.416	62.956	69.200	245
-11	.289	.419	.373	62.942	69.492	247
-10	.245	.434	.478	62.858	68.987	246
-9	.245	.410	.340	62.935	69.487	246
-8	.192	.464	.528	62.973	69.417	245
-7	.147	.465	.499	62.959	69.347	247
-6	.122	.463	.561	62.934	69.227	248
-5	.141	.429	.467	62.905	69.304	247
-4	.120	.456	.551	62.894	69.099	249
-3	.142	.442	.542	62.922	69.024	250
-2	.101	.447	.552	62.846	68.750	248
-1	.041	.433	.455	62.851	69.021	247
0	-.026	.450	.487	62.875	68.888	248
1	-.021	.457	.541	62.864	69.258	246
2	-.027	.445	.530	62.859	68.985	247
3	-.023	.391	.395	62.785	68.837	246
4	-.048	.399	.509	62.809	68.707	247
5	-.093	.362	.463	62.785	68.837	246
6	-.137	.321	.319	62.809	68.707	247
7	-.113	.308	.330	62.809	68.707	247
8	-.123	.306	.306	62.788	68.880	246
9	-.148	.294	.264	62.788	68.880	246
10	-.136	.298	.307	62.836	68.603	248
11	-.172	.283	.288	62.836	68.603	248
12	-.161	.279	.311	62.758	67.611	247
13	-.162	.256	.231	62.763	67.879	246
14	-.217	.262	.268	62.763	67.879	246
15	-.167	.279	.377	62.763	67.879	246

VALUES ABOVE 0 ONLY

LAG ON S.1	CCF	MEAN-1	VAR.-1	MEAN-2	VAR.-2	NOBS
-15	.295	.400	.339	63.050	69.403	245
-14	.285	.429	.437	63.031	69.057	243
-13	.303	.448	.465	63.031	69.057	243
-12	.246	.433	.416	62.956	69.200	245
-11	.289	.419	.373	62.942	69.492	247
-10	.245	.434	.478	62.858	68.987	246
-9	.245	.410	.340	62.935	69.487	246
-8	.192	.464	.528	62.973	69.417	245
-7	.147	.465	.499	62.959	69.347	247
-6	.122	.463	.561	62.934	69.227	248
-5	.141	.429	.467	62.905	69.304	247
-4	.120	.456	.551	62.894	69.099	249
-3	.142	.442	.542	62.922	69.024	250
-2	.101	.447	.552	62.846	68.750	248
-1	.041	.433	.455	62.851	69.021	247
0	-.026	.450	.487	62.875	68.888	248
1	-.021	.457	.541	62.864	69.258	246
2	-.027	.445	.530	62.859	68.985	247
3	-.023	.391	.395	62.785	68.837	246
4	-.048	.399	.509	62.809	68.707	247
5	-.093	.362	.463	62.785	68.837	246
6	-.137	.321	.319	62.809	68.707	247
7	-.113	.308	.330	62.809	68.707	247
8	-.123	.306	.306	62.788	68.880	246
9	-.148	.294	.264	62.788	68.880	246
10	-.136	.298	.307	62.836	68.603	248
11	-.172	.283	.288	62.836	68.603	248
12	-.161	.279	.311	62.758	67.611	247
13	-.162	.256	.231	62.763	67.879	246
14	-.217	.262	.268	62.763	67.879	246
15	-.167	.279	.377	62.763	67.879	246

TURKEY LAKE WATERSHED, STATION 1

	ALL VALUES			VALUES ABOVE 0 ONLY		
VARIABLE	MEAN	VARIANCE	NO. OF OBS.	MEAN	VARIANCE	NO. OF OBS.
SERIES 1: O.RUNOFF	.252	.249	1357	.252	.249	1357
SERIES 2: HCO3	39.993	415.589	256	39.993	415.589	256

ALL VALUES

LAG ON S.1	CCF	MEAN-1	VAR.-1	MEAN-2	VAR.-2	NOBS
-15	-.207	.395	.336	39.588	413.202	248
-14	-.205	.424	.433	39.366	410.327	246
-13	-.234	.442	.458	39.366	410.327	246
-12	-.253	.440	.457	39.465	414.281	248
-11	-.299	.408	.371	39.671	416.165	250
-10	-.265	.423	.475	39.702	416.618	249
-9	-.301	.406	.353	39.572	406.308	249
-8	-.248	.453	.522	39.421	402.260	248
-7	-.261	.452	.496	39.526	405.214	250
-6	-.254	.446	.553	39.749	415.992	251
-5	-.308	.411	.458	39.456	396.232	250
-4	-.314	.434	.532	39.562	399.173	252
-3	-.310	.425	.505	39.851	418.705	253
-2	-.361	.435	.541	39.745	417.021	250
-1	-.437	.419	.447	39.745	417.021	250
0	-.484	.431	.475	39.698	415.904	251
1	-.478	.435	.522	39.608	413.983	249
2	-.444	.413	.437	39.708	419.366	249
3	-.433	.355	.365	39.432	403.347	248
4	-.399	.349	.399	39.386	402.247	249
5	-.349	.345	.451	39.328	402.271	247
6	-.313	.306	.302	39.121	396.368	247
7	-.291	.275	.219	39.282	401.161	248
8	-.304	.287	.283	39.235	398.126	248
9	-.345	.263	.161	39.235	398.126	248
10	-.274	.273	.248	39.641	416.735	250
11	-.249	.269	.276	39.641	416.735	250
12	-.242	.257	.255	39.700	422.442	249
13	-.190	.249	.216	39.610	422.138	248
14	-.210	.262	.270	39.610	422.138	248
15	-.163	.259	.280	39.668	421.282	249

VALUES ABOVE 0 ONLY

LAG ON S.1	CCF	MEAN-1	VAR.-1	MEAN-2	VAR.-2	NOBS
-15	-.207	.395	.336	39.588	413.202	248
-14	-.205	.424	.433	39.366	410.327	246
-13	-.234	.442	.458	39.366	410.327	246
-12	-.253	.440	.457	39.465	414.281	248
-11	-.299	.408	.371	39.671	416.165	250
-10	-.265	.423	.475	39.702	416.618	249
-9	-.301	.406	.353	39.572	406.308	249
-8	-.248	.453	.522	39.421	402.260	248
-7	-.261	.452	.496	39.526	405.214	250
-6	-.254	.446	.553	39.749	415.992	251
-5	-.308	.411	.458	39.456	396.232	250
-4	-.314	.434	.532	39.562	399.173	252
-3	-.310	.425	.505	39.851	418.705	253
-2	-.361	.435	.541	39.745	417.021	250
-1	-.437	.419	.447	39.745	417.021	250
0	-.484	.431	.475	39.698	415.904	251
1	-.478	.435	.522	39.608	413.983	249
2	-.444	.413	.437	39.708	419.366	249
3	-.433	.355	.365	39.432	403.347	248
4	-.399	.349	.399	39.386	402.247	249
5	-.349	.345	.451	39.328	402.271	247
6	-.313	.306	.302	39.121	396.368	247
7	-.291	.275	.219	39.282	401.161	248
8	-.304	.287	.283	39.235	398.126	248
9	-.345	.263	.161	39.235	398.126	248
10	-.274	.273	.248	39.641	416.735	250
11	-.249	.269	.276	39.641	416.735	250
12	-.242	.257	.255	39.700	422.442	249
13	-.190	.249	.216	39.610	422.138	248
14	-.210	.262	.270	39.610	422.138	248
15	-.163	.259	.280	39.668	421.282	249

Appendix C.1

Screen communication for program KALMAN.

```
/@pg=n
PG ACCEPTED.
/settl,1200
SETTL,1200.
/attach,imslib/un=library
/library,imslib
LIBRARY,IMSLIB.
/lgo

 DESCRIPTION FILE NAME:
? tdesc
 DATA FILE NAME:
? turkey1
 NUMBER OF DATARECORDS READ: 1461

  1: O.RUNOFF   2: S.RUNOFF   3: RAIN      4: SNOWMELT   5: N.SUPPLY
  6: SNOWPACK   7: PH         8: M2+       9: SO4--     10: HCO3

 OUTPUT SERIES:
? 10
 NUMBER OF AR-PARAMETERS, AND THEIR VALUES:
? 1,0.9
 NUMBER OF DIFFERENCES:
? 0
 CONSTANT:
? 50.0
 NUMBER OF MA-PARAMETERS, AND THEIR VALUES:
? 2,0.1,0.1
 NO. OF INPUTS, INDEXES OF INPUTS:
? 0
 MISSING CODE:
? -1.0
 FMIN    814.3679019167
 FMIN    814.3556118414
 FMIN    814.3802254517
 FMIN    814.3671939974
 FMIN    814.3686100486
 FMIN    814.3680173061
 FMIN    814.367786684
 FMIN    814.3778141324
 FMIN    814.3580262312
 FMIN    814.3678983436
 FMIN    814.3679017112
 FMIN    814.3679019502
 FMIN    814.3679047893
 FMIN    808.0153469497
 FMIN    804.1631433596
 FMIN    804.1631431273
 FMIN    804.1631426599
 FMIN    804.1631433019
 FMIN    804.1631436868
 FMIN    802.5809749473
 FMIN    818.9402948257
 FMIN    802.5809724434
```

Set no stop on full screen, to avoid disrupting the output.

Set time limit to maximum, to avoid time-out during execution.
Attach the IMSL-library file, and make it a library.

Start the KALMAN program, which in this case has been compiled onto the file LGO.

TDESC is the name of the description file, which must be prepared by the user prior to execution.

TURKEY1 is the name of the data file, which also must be prepared by the user prior to execution. The program responds with the number of records read from TURKEY1.

The menu of time series is presented, for reference.

Series no. 10, HCO_3, is chosen as output series.

The initial model is the ARMA(1,2) model

$$(1 - 0.9B)(Y_t - 50.0) = (1 - 0.1B - 0.1B^2)a_t$$

No input series, that is univariate analysis.

All values less than or equal to -1.0 will be regarded as missing.

The iterative optimization starts. Every time the object function FMIN is called, its value is printed, so that the user can follow the progress.

```
        .
        .
        .
        .
        .
        .
        .
FMIN    800.5186343714
FMIN    800.5186343714
FMIN    800.5186343722
FMIN1   800.5186343722
FMIN1   800.5282804707
FMIN1   800.5281932288
FMIN1   800.5187130955
FMIN1   800.5272226905
FMIN1   800.5187137318
FMIN1   800.5187017518
FMIN1   800.5272199946
FMIN1   800.5189171953
FMIN1   800.5187011909
FMIN1   800.5186994877
FMIN1   800.5283317522
FMIN1   800.5187819802
FMIN1   800.5187701325
FMIN1   800.5186995397
FMIN1   800.518698287
FMIN1   800.5288056294
FMIN1   800.5186943132
FMIN1   800.5187008275
FMIN1   800.5187633374
FMIN1   800.5186984577
```

The iterative process has converged to a solution. Each line labeled FMIN1 stems from a function call to calculate the variance covariance matrix of the estimates.

NOISE MODEL

	PAR.NO.	ORDER	VALUE	ST.DEV.	T
AR-PARAM.	1	1	.9687E+00	.9863E-02	98.21
MA-PARAM.	2	1	.4561E+00	.1181E+00	3.86
	3	2	.1070E+00	.1271E+00	.84
CONSTANT	4	0	.4629E+02	.4057E+01	11.41
RES.VAR.	5	0	.1209E+03	.1345E+02	8.99

A I C: .80852E+03

The final model is presented . The result is

$$(1 - 0.9687B)(Y_t - 46.29) = (1 - 0.4561B - 0.1070B^2)a_t$$

with residual variance 120.9.

The value of the Akaike Information Criterion is given.

End of the estimation of the first model.

```
 1: O.RUNOFF  2: S.RUNOFF  3: RAIN     4: SNOWMELT  5: N.SUPPLY
 6: SNOWPACK  7: PH        8: M2+      9: SO4--    10: HCO3
OUTPUT SERIES:
? 10
NUMBER OF AR-PARAMETERS, AND THEIR VALUES:
? 0
NUMBER OF DIFFERENCES:
? 1
NUMBER OF MA-PARAMETERS, AND THEIR VALUES:
? 1,0.5
NO. OF INPUTS, INDEXES OF INPUTS:
? 0
MISSING CODE:
? -1.0
FMIN    810.5359584172
FMIN    810.5311610356
FMIN    810.5407594554
FMIN    810.5359570239
FMIN    807.4419360274
FMIN    810.930466659
FMIN    807.4419360776
FMIN    807.4398915845
FMIN    807.4421199886
FMIN    807.4398915834
FMIN    807.4398905817
FMIN    807.4398916028
FMIN    807.4398905817
FMIN    807.4398905817
FMIN    807.4398905817
FMIN    807.4398905817
FMIN    807.4398905817
FMIN    807.4398905817
FMIN    807.4398905817
FMIN    807.4398905817
FMIN1   807.4398905817
FMIN1   807.4400219091
FMIN1   807.4400214652
FMIN1   807.4399542468
FMIN1   807.4399791616
FMIN1   807.4399544168

NOISE MODEL    NO. OF DIFFS.=  1
              PAR.NO.  ORDER      VALUE      ST.DEV.        T
    MA-PARAM.    1       1      .6372E+00   .4839E-01     13.17

    RES.VAR.     2       0      .1302E+03   .1419E+02      9.17
A I C:  .80944E+03
```

The menu is presented again, for reference.

The output series is still no. 10, HCO_3.

Now the initial model is the non-stationary ARIMA(0,1,1)-model

$$(1 - B)Y_t = (1 - 0.5B)a_t$$

Still univariate analysis, no inputs.

The missing value code is -1.0, as before.

Start of the iterative optimization, with output for each function call.

The optimization has converged. The calls to FMIN1 relates to the computation of the variance covariance matrix.

Final model

$$(1 - B)Y_t = (1 - 0.6372B)a_t$$

with the given residual variance and AIC value

```
 1: O.RUNOFF   2: S.RUNOFF   3: RAIN      4: SNOWMELT   5: N.SUPPLY
 6: SNOWPACK   7: PH         8: M2+       9: SO4--     10: HCO3
 OUTPUT SERIES:
? 10
 NUMBER OF AR-PARAMETERS, AND THEIR VALUES:
? 1,0.96
 NUMBER OF DIFFERENCES:
? 0
 CONSTANT:
? 48.0
 NUMBER OF MA-PARAMETERS, AND THEIR VALUES:
? 1,0.46
 NO. OF INPUTS, INDEXES OF INPUTS:
? 3,1,3,4
 NO. AND VALUES OF PAR, INPUT 1:
? 3,-0.1,-0.1,-0.1
 DELAY FOR INPUT 1:
? [illegible]
 NO. AND VALUES OF PAR, INPUT 2:
? 1,-0.1
 DELAY FOR INPUT 2:
? 0
 NO. AND VALUES OF PAR, INPUT 3:
? 2,-0.1,-0.1
 DELAY FOR INPUT 3:
? 1
 MISSING CODE:
? -1.0
 FMIN    775.0060469711
 FMIN    775.0057084044
 FMIN    775.0064115657
 FMIN    775.0043779323
 FMIN    775.0077199299
 FMIN    775.0072175521
 FMIN    775.0048889711
 FMIN    775.0061325803
 FMIN    775.0059613623
 FMIN    775.0061359983
 FMIN    775.0059579443
 FMIN    775.0060829141
 FMIN    775.0060110285
 FMIN    775.0060240674
           .
           .
```

Start of a new estimation. The menu is presented for reference.

Series no. 10, HCO_3, is still the series of interest.

This time we want to estimate a transfer function model. The noise model is given first, as the ARMA(1,1)-model

$$(1 - 0.96B)(n_t - 48.0) = (1 - 0.46B)a_t$$

We want three input series, no. 1 (O.RUNOFF), 3 (RAIN) and 4 (SNOWMELT)

The initial model for the transfer functions is

$$Y_t = (1-0.1B-0.1B^2-0.1B^3)X_{1t} + (1-0.1B)X_{2t} + (1-0.1B-0.1B^2)X_{3t-1} + n_t$$

where n_t follows the ARMA(1,1)-model above, and X_{1t}, X_{2t} and X_{3t} are the O.RUNOFF, RAIN and SNOWMELT series, respectively.

Note that we want the eventual effect of series 3 to be delayed one lag.

Values below -1.0 are still regarded as missing, both for the inputs and the output series.

The iterations start, with output for each call to the object function. In this fairly complex model a rather large number of iterations is required.

```
        .
        .
FMIN    757.1771132618
FMIN    757.1771132775
*** TERMINAL ERROR              (IER = 130) FROM IMSL ROUTINE ZXMIN
FMIN1   757.1771132622
FMIN1   757.1890061809
        .
        .
        .
FMIN1   757.1771754034
FMIN1   757.1771751969
FMI"1   757.1771752142
FM::1   757.1771753449

NOISE MODEL

                PAR.NO.  ORDER      VALUE       ST.DEV.          T
     AR-PARAM.     1       1      .9754E+00    .8333E-02      117.06
     MA-PARAM.     2       1      .6526E+00    .5613E-01       11.63
      CONSTANT     3       0      .4932E+02    .4185E+01       11.78

INPUT NO. 1    O.RUNOFF

                PAR.NO.  ORDER      VALUE       ST.DEV.          T
    TRANS.PAR.     4       0     -.5401E+01    .1414E+01       -3.82
                   5       1     -.3440E+01    .1385E+01       -2.48
                   6       2     -.2601E+01    .1462E+01       -1.78

INPUT NO. 2    RAIN

                PAR.NO.  ORDER      VALUE       ST.DEV.          T
    TRANS.PAR.     7       0      .6840E+00    .1051E+01         .65

INPUT NO. 3    SNOWMELT    DELAY =   1

                PAR.NO.  ORDER      VALUE       ST.DEV.          T
    TRANS.PAR.     8       1      .4985E+00    .3437E+01         .15
                   9       2     -.1262E+00    .3252E+01        -.04


       RES.VAR.   10       0      .1182E+03    .1261E+02        9.37
A I C:   .77518E+03

 1: O.RUNOFF   2: S.RUNOFF   3: RAIN       4: SNOWMELT   5: N.SUPPLY
 6: SNOWPACK   7: PH         8: M2+        9: SO4--     10: HCO3
OUTPUT SERIES:
? 0
 280.012 CP SECONDS EXECUTION TIME.
/
```

The iteration terminates because the optimizing routine has reached the maximum number of iterations allowed to it. However, the final result is judged to be satisfactory by the program (otherwise the optimization would have been restarted), and the program continues to compute the variance covariance matrix.

The final model is

$$Y_t = (1-5.401B-3.440B^2-2.601B^3)X_{1t} + (1+0.6840B)X_{2t}$$

$$+ (1+0.4985B-0.1262B^2)X_{3t-1} + 49.32 + (1-0.9754B)^{-1}(1-0.6526B)a_t$$

with the given residual variance and AIC.

The menu is presented again, to start a new estimation.

However, the response output series no. 0 terminates the program.

The printed output is on the file ESTRES.

Appendix C.2

Printout of the output file ESTRES from program KALMAN.

TURKEY LAKE WATERSHED, STATION 1
256 NON-MISSING OBS. OUT OF 1461

OUTPUT SERIES HCO3

NOISE MODEL

	PAR.NO.	ORDER	VALUE	ST.DEV.	T
AR-PARAM.	1	1	.9687E+00	.9863E-02	98.21
MA-PARAM.	2	1	.4561E+00	.1181E+00	3.86
	3	2	.1070E+00	.1271E+00	.84
CONSTANT	4	0	.4629E+02	.4057E+01	11.41
RES.VAR.	5	0	.1209E+03	.1345E+02	8.99

CORRELATION MATRIX OF THE ESTIMATES

	1	2	3	4	5
1	1.000				
2	.025	1.000			
3	.332	-.850	1.000		
4	-.013	.001	-.016	1.000	
5	.089	.394	-.130	-.013	1.000

A I C: .80852E+03

TURKEY LAKE WATERSHED, STATION 1
255 NON-MISSING OBS. OUT OF 1461

OUTPUT SERIES HCO3

NCISE MODEL NO. OF DIFFS.= 1

	PAR.NO.	ORDER	VALUE	ST.DEV.	T
MA-PARAM.	1	1	.6372E+00	.4839E-01	13.17
RES.VAR.	2	0	.1302E+03	.1419E+02	9.17

CORRELATION MATRIX OF THE ESTIMATES

	1	2
1	1.000	
2	.582	1.000

A I C: .80944E+03

TURKEY LAKE WATERSHED, STATION 1
248 NON-MISSING OBS. OUT OF 1461

OUTPUT SERIES HCO3

NOISE MODEL

	PAR.NO.	ORDER	VALUE	ST.DEV.	T
AR-PARAM.	1	1	.9754E+00	.8333E-02	117.06
MA-PARAM.	2	1	.6526E+00	.5613E-01	11.63
CONSTANT	3	0	.4932E+02	.4185E+01	11.78

INPUT NO. 1 D.RUNOFF

	PAR.NO.	ORDER	VALUE	ST.DEV.	T
TRANS.PAR.	4	0	-.5401E+01	.1414E+01	-3.82
	5	1	-.3440E+01	.1385E+01	-2.48
	6	2	-.2601E+01	.1462E+01	-1.78

INPUT NO. 2 RAIN

	PAR.NO.	ORDER	VALUE	ST.DEV.	T
TRANS.PAR.	7	0	.6840E+00	.1051E+01	.65

INPUT NO. 3 SNOWMELT DELAY = 1

	PAR.NO.	ORDER	VALUE	ST.DEV.	T
TRANS.PAR.	8	1	.4985E+00	.3437E+01	.15
	9	2	-.1262E+00	.3252E+01	-.04
RES.VAR.	10	0	.1182E+03	.1261E+02	9.37

CORRELATION MATRIX OF THE ESTIMATES

	1	2	3	4	5	6	7	8	9	10
1	1.000									
2	.598	1.000								
3	-.037	-.006	1.000							
4	-.039	-.153	-.060	1.000						
5	-.001	-.024	-.018	-.331	1.000					
6	-.032	-.189	-.096	.114	-.350	1.000				
7	-.006	-.019	-.061	-.037	.014	.048	1.000			
8	.039	.013	-.025	-.174	.106	.058	-.107	1.000		
9	.029	-.022	-.044	.059	-.182	.057	.058	-.587	1.000	
1C	.107	.487	.007	-.089	-.015	-.115	-.011	-.003	-.023	1.000

A I C: .77518E+03

Appendix D.1

Listing of program CORRCAL
with its associated subroutines.

```
      PROGRAM CORRCAL(INPUT,OUTPUT)
C
C     PROGRAM CORRCAL COMPUTES THE AUTOCORRELATION FUNCTION FOR A
C     TIME SERIES OR THE CROSS CORRELATION BETWEEN TWO TIME SERIES,
C     WHERE THE TIME SERIES MAY CONTAIN A SUBSTANTIAL NUMBER OF
C     MISSING OBSERVATIONS.
C
C     ******************************************************
C     *****  WRITTEN BY: EIVIND DAMSLETH                *****
C     *****  LAST REVISION: MAY 21. 1987                *****
C     ******************************************************
C
      PARAMETER (MAXLEN=1500,MAXSER=10)
      COMMON/CDATA/NOBS,Y(MAXLEN,MAXSER)
      CHARACTER VARNAM(MAXSER)*8,SNR(5)*4,DATAF*7,DESCF*7
      CHARACTER TITLE*60,FORM*60
      DATA SNR/'1   ','2   ','3   ','4   ','5   '/
C
C     ************ S T A R T   I N I T I A L I Z A T I O N **********
C
C
C     THE PROGRAM READS SOME BASIC INFORMATION FROM A DESCRIPTION FILE,
C     WHICH MUST BE PROVIDED BY THE USER. THE LAYOUT OF THE DESCRIPTION
C     FILE IS:
C
C     LINE 1: IDENTIFICATION TITLE UP TO 60 CHAR., WHICH WILL BE USED IN
C             THE OUTPUT.
C     LINE 2: NUMBER OF VARIABLES IN THE DATA FILE. FREE FORMAT.
C     LINE 3: IDENTIFICATION NAME OF UP TO 8 CHAR. FOR VARIABLE NO. 1
C     LINE 4:          "                 "             "              2
C                      .                 .             .
C                      .                 .             .
C     LINE 2+N:        "                 "             "              N,
C             WHERE N IS THE NUMBER OF VARIABLES GIVEN IN LINE 2.
C     LINE 3+N: FORTRAN FORMAT, INCLUDING THE PARANTHESIS, FOR READING
C             ONE RECORD FROM THE DATA-FILE.
C
C     PROMPT FOR THE DESCRIPTION FILE NAME
C
      WRITE(*,1) 'DESCRIPTION FILE NAME'
    1 FORMAT(/,1X,A,':',I5)
      READ(*,2) DESCF
    2 FORMAT(A)
C
C     OPEN THE DESCRIPTION FILE AND READ ITS CONTENTS.
C
      OPEN(5,FILE=DESCF)
      READ(5,2) TITLE
      READ(5,*) NVAR
      READ(5,2) (VARNAM(I),I=1,NVAR)
      READ(5,2) FORM
      CLOSE(5)
C
C     THE DATA ARE READ FROM ANOTHER FILE, THE DATAFILE. THE DATAFILE
C     IS ORGANIZED WITH ONE RECORD PER TIME UNIT, WHERE THE RECORD
C     CONTAINS THE VALUES FOR ALL THE VARIABLES. MISSING VALUES ARE GIVEN
C     BY A MISSING-VALUE CODE CHOSEN BY THE USER. THE DATAFILE WILL BE
C     READ, RECORD BY RECORD, ACCORDING TO THE FORMAT READ FROM THE
C     DESCRIPTION FILE.
C
C     PROMPT FOR THE DATAFILE NAME.
C
      WRITE(*,1) 'DATA FILE NAME'
      READ(*,2) DATAF
C
C     OPEN THE DATAFILE, AND READ THE DATA ACCORDING TO THE GIVEN FORMAT.
C
      OPEN(5,FILE=DATAF)
      NOBS=0
    7 READ(5,FORM,END=8,ERR=8) (Y(NOBS+1,J),J=1,NVAR)
      NOBS=NOBS+1
      GOTO 7
    8 CLOSE(5)
C
C     NOBS CONTAINS THE NUMBER OF DATARECORDS, I.E. THE NUMBER OF OBSERVATIONS
```

```
C       IN THE TIME SERIES, INCLUDING THE MISSING.
C
        WRITE(*,1) 'NUMBER OF DATARECORDS READ',NOBS
C
C       PROMPT FOR THE MISSING VALUE CODE. ALL VALUES <= THIS CODE WILL
C       BE REGARDED AS MISSING.
C
        WRITE(*,1) 'MISSING VALUE CODE'
        READ(*,*) XMISS
C
C       THE RESULTS FROM THE ANALYSIS WILL BE WRITTEN ONTO THE FILE "CORRES",
C       WHICH AFTERWARDS MAY BE INSPECTED BY THE USER, OR ROUTED TO A PRINTER.
C
        OPEN(6,FILE='CORRES')
C
C       PROMPT FOR THE NUMBER OF CORRELATIONS TO BE CALCULATED. IF AUTO-
C       CORRELATIONS ARE REQUESTED, THEY WILL BE COMPUTED FOR LAGS 1, 2,
C       ..., NAC, WHERE NAC IS THE VALUE GIVEN. IF CROSSCORRELATIONS ARE
C       REQUESTED THEY WILL BE CALCULATED FOR LAGS -NAC,...,0,...,NAC.
C
        WRITE(*,1) 'NO. OF CORRELATIONS'
        READ(*,*) NAC
C
C       ******** E N D   O F   I N I T I A L I Z A T I O N *********
C
5       CONTINUE
C       ******** S T A R T   O F   O N E   C O R R E L A T I O N   R U N *****
C
C       PRESENT THE "MENU"
C
        WRITE(*,6) (I,VARNAM(I),I=1,NVAR)
6       FORMAT(/,5(I3,': ',A8),/,5(I3,': ',A8))
C
C       THE PROGRAMS NEED THE INDEX (NUMBER) OF TWO VARIABLES, SAY IV1
C       AND IV2. IF IV1=IV2 THEN THE AUTOCORRELATIONS ARE CALCULATED
C       FOR SERIES IV1. IF IV1 =/= IV2 THEN THE CROSSCORRELATIONS BETWEEN
C       SERIES IV1 AND IV2 ARE CALCULATED. THE PROGRAM PROMPTS FOR THE
C       VALUES OF IV1 AND IV2. FOR CONVENIENCE, THE PROGRAM PLOTS FOR A
C       RANGE OF VALUES FOR IV1 AND IV2, AND WILL AUTOMATICALLY PRODUCE
C       AUTO- AND CROSSCORRELATIONS FOR ALL COMBINATIONS OF IV1 AND IV2
C       WITHIN THEIR RANGE. A REPLY OF 0,0 FOR THE RANGE OF IV1 TERMI-
C       NATES THE PROGRAM.
C
        WRITE(*,1) 'INDEX RANGE FOR FIRST VARIABLE (0,0 TO STOP)'
        READ(*,*) IV1LOW,IV1UP
        IF(IV1LOW.LE.0) GOTO 1000
        WRITE(*,1) 'INDEX RANGE FOR SECOND VARIABLE'
        READ(*,*) IV2LOW,IV2UP
        WRITE(*,1)
C
C       START LOOP ON THE TWO VARIABLES
C
        DO 100 IV1=IV1LOW,IV1UP
        DO 100 IV2=IV2LOW,IV2UP
C
C       PRINT THE VALUES OF IV1 AND IV2 ON THE TERMINAL, SO THE USER CAN
C       FOLLOW THE PROGRESSION
C
        WRITE(*,*) 'PROCESSING VARIABLE ',IV1,' VERSUS ',IV2
        IF(IV1.EQ.IV2) THEN
C
C       IV1=IV2 IMPLIES AUTOCORRELATION, AND THE LOWER LAG VALUE = 1.
C       OTHERWISE THE LOWER VALUE IS -NAC.
C
           MAC=1
        ELSE
           MAC=-NAC
        ENDIF
C
C       CCORRR CALCULATES THE CORRELATIONS BETWEEN THE VARIABLES IN THE
C       TWO ARRAYS GIVEN IN THE TWO FIRST PARAMETERS, WITH THE NUMBER OF
C       OBSERVATIONS GIVEN AS THE THIRD PARAMETER. THE MISSING VALUE CODE
C       IS GIVEN AS PARAMETER 4. ON RETURN, CCORR PROVIDES THE MEAN AND
C       VARIANCE OF THE FIRST SERIES IN PARAMETERS 5 AND 6 RESPECTIVELY,
C       WHILE PARAMETERS 7 AND 8 CONTAINS THE MEAN AND VARIANCE FOR THE
C       SECOND SERIES. PARAMETERS 9 AND 10 GIVES THE COVARIANCE AND CORRE-
C       LATION, RESPECTIVELY, WHILE THE 11TH AND LAST PARAMETER GIVES THE
```

```
C       NUMBER OF NON-MISSING OBSERVATIONS INVOLVED IN THE COMPUTATION.
C
C       THE FOLLOWING TWO CALLS TO CCORR GIVES THE MEAN AND VARIANCE OF THE
C       IV1-SERIES, FIRST FOR ALL NON-MISSING VALUES, SECONDLY FOR ALL
C       VARIABLES ABOVE 0 (ZERO).
C
        CALL CCORR(Y(1,IV1),Y(1,IV1),NOBS,XMISS,XM1,XV1,DU1,DU2,DU3,DU4,
     1 NX1)
        CALL CCORR(Y(1,IV1),Y(1,IV1),NOBS,.0,XM2,XV2,DU1,DU2,DU3,DU4,
     1 NX2)
        IF(IV2.NE.IV1) THEN
C
C          SINCE IV2 =/= IV1, THE FOLLOWING TWO CALLS TO CCORR ARE
C          REQUIRED TO GET THE TWO SETS OF MEANS AND VARIANCES FOR THE
C          SECOND SERIES.
C
           CALL CCORR(Y(1,IV2),Y(1,IV2),NOBS,XMISS,DU1,DU2,YM1,YV1,DU3,
     1     DU4,NY1)
           CALL CCORR(Y(1,IV2),Y(1,IV2),NOBS,.001,DU1,DU2,YM2,YV2,DU3,DU4,
     1     NY2)
        ENDIF
C
C       PRINT THE TITLE AND THE BASIC INFORMATION REGARDING THE ONE OR
C       TWO SERIES.
C
        WRITE(6,12) TITLE
 12     FORMAT('1',/,1X,A60,//,35X,'ALL VALUES',23X,'VALUES ABOVE 0 ONLY'
     1 ,/,11X,'VARIABLE',2(8X,'MEAN     VARIANCE  NO. OF OBS.'),/)
        WRITE(6,13) 1,VARNAM(IV1),XM1,XV1,NX1,XM2,XV2,NX2
 13     FORMAT(1X,'SERIES',I2,':',1X,A8,2(2F12.3,I9,4X))
        IF(IV2.NE.IV1) WRITE(6,13) 2,VARNAM(IV2),YM1,YV1,NY1,YM2,YV2,NY2
        IF(IV2.EQ.IV1) THEN
           WRITE(6,14)
        ELSE
           WRITE(6,15)
        ENDIF
 14     FORMAT(//,19X,'ALL VALUES',25X,'VALUES ABOVE 0 ONLY',/,
     1 2(' LAG      ACF     MEAN       VAR.     NOBS',5X,)/)
 15     FORMAT(//,34X,'ALL VALUES',42X,'VALUES ABOVE 0 ONLY',/,
     1 2(' LAG ON S.1    CCF    MEAN-1    VAR.-1   MEAN-2',3X,
     1 'VAR.-2',4X,'NOBS',2X,)/)
C
C       START LOOP FOR THE VARIOUS LAGS, WHERE THE RANGE OF THE LOOP
C       DEPENDS ON WETHER WE COMPUTE AUTOCORRELATIONS (IV1=IV2) OR
C       CROSSCORRELATIONS (IV1=/=IV2).
C
        DO 20 K=MAC,NAC
        IF(K.LT.0) THEN
C
C          K < 0, SO THE SECOND SERIES IS LAGGING THE FIRST
C
           CALL CCORR(Y(1-K,IV1),Y(1,IV2),NOBS+K,XMISS,XM1,XV1,YM1,YV1,
     1     COV1,CORR1,NP1)
           CALL CCORR(Y(1-K,IV1),Y(1,IV2),NOBS+K,.0,XM2,XV2,YM2,YV2,
     1     COV2,CORR2,NP2)
        ELSE
C
C          K >= 0, SO THE FIRST SERIES IS LAGGING THE SECOND.
C
           CALL CCORR(Y(1,IV1),Y(1+K,IV2),NOBS-K,XMISS,XM1,XV1,YM1,YV1,
     1     COV1,CORR1,NP1)
           CALL CCORR(Y(1,IV1),Y(1+K,IV2),NOBS-K,.0,XM2,XV2,YM2,YV2,
     1     COV2,CORR2,NP2)
        ENDIF
C
C       PRINT THE RESULTS FOR A GIVEN LAG K, WITH LAYOUT DEPENDING ON
C       WETHER IV1=IV2 OR NOT.
C
        IF(IV2.EQ.IV1) THEN
           WRITE(6,18) K,CORR1,XM1,XV1,NP1,K,CORR2,XM2,XV2,NP2
        ELSE
           WRITE(6,19) K,CORR1,XM1,XV1,YM1,YV1,NP1,K,CORR2,XM2,XV2,YM2,YV2
     1     ,NP2
        ENDIF
 18     FORMAT(1X,2(I3,3F9.3,I8,6X))
 19     FORMAT(1X,2(I8,5F9.3,I8,3X))
C
```

```
C        END OF LAG LOOP
C
20       CONTINUE
C
C        ****** E N D   O F   O N E   C O R R E L A T I O N   R U N ********
100      CONTINUE
C
C        RETURN TO LABEL 5, TO START ANOTHER CORRELATION RUN, IF REQUIRED,
C        OR TO STOP.
C
         GOTO 5
1000     STOP
         END
```

```
      SUBROUTINE CCORR(X,Y,N,XMISS,XMEAN,XVAR,YMEAN,YVAR,COV,CORR,NP)
C
C     SUBROUTINE CCORR CALCULATES THE COVARIANCE AND THE CORRELATION
C     BETWEEN THE N VALUES IN THE ARRAYS X AND Y. XMISS IS A CODE
C     FOR MISSING OBSERVATIONS, AND (X,Y) PAIRS WHERE EITHER X OR Y ARE
C     <= XMISS WILL BE EXCLUDED FROM THE COMPUTATION.
C     THE ROUTINE RETURNS THE MEANS AND VARIANCES OF THE TWO SERIES IN
C     XMEAN,XVAR AND YMEAN,YVAR RESPECTIVELY. THE MEANS AND VARIANCES ARE
C     COMPUTED ONLY FROM THE VALUES INCLUDED IN THE COMPUTATION OF THE
C     CORRELATION. THUS, A X-VALUE IS NOT INCLUDED IN THE COMPUTATION OF
C     XMEAN AND XVAR IF THE MATCHING Y-VALUE IS MISSING, EVEN IF THE
C     X-VALUE IS NON-MISSING.
C     COV AND CORR RETURNS THE VALUE OF THE COVARIANCE AND THE CORRE-
C     LATION RESPECTIVELY, AND FINALLY NP RETURNS THE NUMBER OF PAIRS
C     USED IN THE COMPUTATION, THAT IS, WHERE BOTH X AND Y WERE OBSERVED.
C
C     *****************************************************
C     *****  WRITTEN BY: EIVIND DAMSLETH                *****
C     *****  LAST REVISION: MAY 21, 1987               *****
C     *****************************************************
C
      DIMENSION X(1),Y(1)
C
C     INITIALIZE
C
      XMEAN=0.
      YMEAN=0.
      XVAR=0.
      YVAR=0.
      COV=0.
      CORR=0.
      NP=0
C
C     START LOOP THROUGH THE ARRAYS X AND Y
C
      DO 10 I=1,N
C
C     TEST IF BOTH VARIABLES ARE OBSERVED, OTHERWISE SKIP
C
      IF(X(I).LE.XMISS.OR.Y(I).LE.XMISS) GOTO 10
      NP=NP+1
C
C     THE MEANS, VARIANCES AND COVARIANCE ARE CALCULATED USING THE
C     RECURSIVE FORMULA IN ORDER TO COMPUTE THEM IN JUST ONE RUN
C     THROUGH THE DATA.
C
      XMEAN=XMEAN+(X(I)-XMEAN)/NP
      YMEAN=YMEAN+(Y(I)-YMEAN)/NP
      IF(NP.GT.1) THEN
         XVAR=(1-1./NP)*XVAR+(X(I)-XMEAN)**2/(NP-1.)
         YVAR=(1-1./NP)*YVAR+(Y(I)-YMEAN)**2/(NP-1.)
         COV=(1-1./NP)*COV+(X(I)-XMEAN)*(Y(I)-YMEAN)/(NP-1.)
      ENDIF
   10 CONTINUE
C
C     COMPUTE THE CORRELATION, PROVIDED THE VARIANCES ARE DEFINED
C
      IF(NP.GT.1) CORR=COV/SQRT(XVAR*YVAR)
      RETURN
      END
```

Appendix D.2

Listing of program KALMAN
with its associated subroutines.

```
      PROGRAM KALMAN(INPUT,OUTPUT)
      PARAMETER (MAXLEN=1500,MAXSER=10)
C
C     PROGRAM KALMAN ESTIMATES THE PARAMETERS IN A UNIVARIATE ARIMA-MODEL
C     OR IN A SIMPLE MULTI-INPUT TRANSFER FUNCTION MODEL. IN THE PRESENCE
C     OF MISSING OBSERVATIONS, THE KALMAN-FILTER APPROACH IS USED.
C
C     BASIC REFERENCE: JONES, R. H. (1980): MAXIMUM LIKELIHOOD FITTING OF
C     ARMA MODELS TO TIME SERIES WITH MISSING OBSERVATIONS. TECHNOMETRICS,
C     VOL. 22 NC. 3 389-396.
C
C     **************************************************************
C     ****************** I M P O R T A N T ************************
C     *****                                                   *****
C     ***** PROGRAM KALMAN REQUIRES ACCESS TO THE ROUTINES IN *****
C     ***** THE IMSL-LIBRARY, WHICH MUST BE ATTACHED AS A     *****
C     ***** LIBRARY TO THE JOB BEFORE EXECUTION.              *****
C     *****                                                   *****
C     ****************** I M P O R T A N T ************************
C     **************************************************************
C
C     ****************************************************
C     *****   WRITTEN BY: EIVIND DAMSLETH            *****
C     *****   LAST REVISION: MAY 21, 1987            *****
C     ****************************************************
C
      DIMENSION PAR(20),HESS(210),GRAD(20),WORK(60)
      COMMON/ARP/ID,IP,FI(10)
      COMMON/MAP/IQ,TET(10)
      COMMON/MOM/IAMY,AMY,IR2,R2,ISIG2,SIG2
      COMMON/OBS/N,NONMIS,XMISS,CODE,X(MAXLEN)
      COMMON/INPUT/NSERIE,INDOUT,INDIN(5),IB(5),NP(5)
      COMMON/CDATA/NOBS,Y(MAXLEN,MAXSER)
      EXTERNAL FM,FM1
      CHARACTER STNR*1,VARNAM(MAXSER)*8,SNR(5)*4,DATAF*7,DESCF*7
      CHARACTER TITLE*60,FORM*60
      DATA AMY,R2,SIG2,FI,TET/23*0./
      DATA SNR/'1   ','2   ','3   ','4   ','5   '/
      DATA CODE/-1.E+50/
      DATA XMISS/-1./
C
C     ************ S T A R T   I N I T I A L I Z A T I O N **********
C
C
C     THE PROGRAM READS SOME BASIC INFORMATION FROM A DESCRIPTION FILE,
C     WHICH MUST BE PROVIDED BY THE USER. THE LAYOUT OF THE DESCRIPTION
C     FILE IS:
C
C     LINE 1: IDENTIFICATION TITLE UP TO 60 CHAR., WHICH WILL BE USED IN
C             THE OUTPUT.
C     LINE 2: NUMBER OF VARIABLES IN THE DATA FILE. FREE FORMAT.
C     LINE 3: IDENTIFICATION NAME OF UP TO 8 CHAR. FOR VARIABLE NO. 1
C     LINE 4:            "             "              "             2
C                        .             .              .
C                        .             .              .
C     LINE 2+N:          "             "              "             N,
C          WHERE N IS THE NUMBER OF VARIABLES GIVEN IN LINE 2.
C     LINE 3+N: FORTRAN FORMAT, INCLUDING THE PARANTHESIS, FOR READING
C          ONE RECORD FROM THE DATA-FILE.
C
C     PROMPT FOR THE DESCRIPTION FILE NAME
C
      WRITE(*,1) 'DESCRIPTION FILE NAME'
    1 FORMAT(/,1X,A,':',I5)
      READ(*,2) DESCF
    2 FORMAT(A)
C
C     OPEN THE DESCRIPTION FILE, AND READ ITS CONTENTS.
C
      OPEN(5,FILE=DESCF)
      READ(5,2) TITLE
      READ(5,*) NVAR
      READ(5,2) (VARNAM(I),I=1,NVAR)
      READ(5,2) FORM
      CLOSE(5)
```

```
C
      THE DATA ARE READ FROM ANOTHER FILE, THE DATAFILE. THE DATAFILE
C     IS ORGANIZED WITH ONE RECORD PER TIME UNIT, WHERE THE RECORD
C     CONTAINS THE VALUES FOR ALL THE VARIABLES. MISSING VALUES ARE GIVEN
C     BY A MISSING-VALUE CODE CHOSEN BY THE USER. THE DATAFILE WILL BE
C     READ, RECORD BY RECORD, ACCORDING TO THE FORMAT READ FROM THE
C     DESCRIPTION FILE.
C
C     PROMPT FOR THE DATAFILE NAME.
C
      WRITE(*,1) 'DATA FILE NAME'
      READ(*,2) DATAF
C
C     OPEN THE DATAFILE, AND READ THE DATA ACCORDING TO THE GIVEN FORMAT.
C
      OPEN(5,FILE=DATAF)
      NOBS=0
    7 READ(5,FORM,END=8,ERR=8) (Y(NOBS+1,J),J=1,NVAR)
      NOBS=NOBS+1
      GOTO 7
    8 CLOSE(5)
C
C     NOBS CONTAINS THE NUMBER OF DATARECORDS, I.E. THE NUMBER OF OBSERVATIONS
C     IN THE TIME SERIES, INCLUDING THE MISSING.
C
      WRITE(*,1) 'NUMBER OF DATARECORDS READ',NOBS
C
C     THE RESULTS FROM THE ANALYSIS WILL BE WRITTEN ONTO THE FILE "ESTRES",
C     WHICH AFTERWARDS MAY BE INSPECTED BY THE USER, OR ROUTED TO A PRINTER.
C
      OPEN(6,FILE='ESTRES')
C
C     ******** E N D  O F  I N I T I A L I Z A T I O N *********
C
C
C
C     ******** S T A R T  O F  O N E  E S T I M A T I O N  S E Q U E N C E
C
   81 CONTINUE
C
C     ******* S T A R T  O F  M O D E L  I N P U T  S E S S I O N ********
C
C     PRESENT THE "MENU" AND PROMPT FOR THE OUTPUT SERIES NO.
C
      WRITE(*,9) (I,VARNAM(I),I=1,NVAR)
    9 FORMAT(/,5(I3,': ',A8),/,5(I3,': ',A8))
      WRITE(*,1) 'OUTPUT SERIES'
      READ(*,*) INDOUT
C
C     A REPLY OF 0 (ZERO) TERMINATES THE ESTIMATION, AND THE PROGRAM STOPS.
C     LABEL 100 LABELS THE STOP STATEMENT AT THE END OF THE PROGRAM.
C
      IF(INDOUT.LE.0) GOTO 100
C
C     PROMPT FOR THE AR-PARAMETERS. IF NO AR-PARAMETER, REPLY 0 (ZERO)
C
      WRITE(*,1) 'NUMBER OF AR-PARAMETERS, AND THEIR VALUES'
      READ(*,*) IP,(FI(I),I=1,IP)
C
C     PROMPT FOR THE NUMBER OF DIFFERENCES
C
      WRITE(*,1) 'NUMBER OF DIFFERENCES'
      READ(*,*) ID
C
C     IF NO DIFFERENCES, PROMPT FOR THE MEAN (CONSTANT IN THE TRANSFER
C     FUNCTION SITUATION)
C
      IF(ID.EQ.C) THEN
         WRITE(*,1) 'CONSTANT'
         READ(*,*) AMY
         IAMY=1
         ENDIF
C
C
C     PROMPT FOR THE MA-PARAMETERS. IF NO MA-PARAMETERS, REPLY 0 (ZERO)
      WRITE(*,1) 'NUMBER OF MA-PARAMETERS, AND THEIR VALUES'
      READ(*,*) IQ,(TET(I),I=1,IQ)
```

```
C
C     PROMPT FOR THE MEASUREMENT ERROR VARIANCE.
C
C     THIS FEATURE DOES NOT WORK PROPERLY, AND HAS BEEN EXCLUDED FROM
C     THE PROGRAM PENDING FURTHER ANALYSIS.
C
C     WRITE(*,1) 'IR2,R2'
C     READ(*,*) IR2,(R2,I=1,IR2)
      IR2=0
      IF(ID.GT.0) IAMY=0
      NPAR=IP+IQ+IR2+IAMY
C
C     NPAR IS THE TOTAL NUMBER OF UNIVARIATE PARAMETERS TO BE ESTIMATED
C
C     PROMPT FOR THE NUMBER OF INPUTS AND THEIR INDEXES. IF THERE ARE NO
C     INPUTS, THAT IS AN ORDINARY UNIVARIATE MODEL, REPLY 0 (ZERO).
C
      WRITE(*,1) 'NO. OF INPUTS, INDEXES OF INPUTS'
      READ(*,*) NINP,(INDIN(I),I=1,NINP)
C
C     START LOOP FOR INPUT OF THE INFORMATION REGARDING EACH INPUT SERIES
C
      DO 15 I=1,NINP
C
C     PROMPT FOR THE PARAMETERS
C
      WRITE(*,13) 'NO. AND VALUES OF PAR, INPUT',I
13    FORMAT(/,1X,A,I2,':')
      READ(*,*) NN,(PAR(NPAR+J),J=1,NN)
      NP(I)=NN
C
C     NPAR IS UPDATED, AND WILL FINALLY CONTAIN THE TOTAL NUMBER OF
C     PARAMETERS
C
      NPAR=NPAR+NN
C
C     PROMPT FOR THE DELAY
C
      WRITE(*,13) 'DELAY FOR INPUT',I
15    READ(*,*) IB(I)
C
C     FINALLY, PROMPT FOR THE MISSING CODE. ALL DATA VALUES BELOW THE
C     MISSING CODE WILL BE REGARDED AS MISSING.
C
      WRITE(*,1) 'MISSING CODE'
      READ(*,*) XMISS
C
C
C     ******** E N D  O F  M O D E L  I N P U T  S E S S I O N ********
C
C     ******** S T A R T  O F  E S T I M A T I O N  S E S S I O N ******
C
      NSERIE=NINP+1
C
C     FIRST, THE ESTIMATION IS DONE WITH THE VARIANCE IMPLICITLY DEFINED,
C     THAT IS NOT INCLUDED IN THE PARAMETERS IN THE OPTIMIZATION.
C     ISIG2=0 CONTROLS THIS OPTION.
C
      ISIG2=0
C
C     N=-1 IS USED AS A FLAG TO THE ROUINE CNOISE TO INDICATE THAT IT
C     IS CALLED FOR THE FIRST TIME. CNOISE RESETS N, SO THAT N =/= -1
C     IN SUBSEQUENT CALLS TO CNOISE.
C
      N=-1
C
C     SETUP PACKS THE PARAMETERS IN FI,TET ETC. INTO ONE ARRAY PAR, TO BE
C     USED IN THE OPTIMIZATION. THE PARAMETERS ARE, AT THE SAME TIME,
C     TRANSFORMED TO ENSURE STATIONARITY AND INVERTIBILITY.
C
      CALL SETUP(NUNPAR,PAR,0,0)
      IF(NPAR.EQ.0) THEN
C
C        IF THERE ARE NO PARAMETERS (BUT THE VARIANCE) IN THE MODEL, THERE
C        IS NOTHING TO OPTIMIZE. THE CALL TO SUBROUTINE FM FINDS THE
C        VARIANCE (STORED IN SIG2 IN COMMON /MOM/)
C
```

```
            CALL FM(NPAR,PAR,FMIN)
         ELSE
C
C           THE IMSL-LIBRARY ROUTINE ZXMIN FINDS THE OPTIMAL PARAMETER
C           VALUES, AND STORES THEM IN PAR. FM IS THE ROUTINE WHICH
C           CALCULATES THE LIKELIHOOD WHEN THE VARIANCE IS NOT EXPLICITLY
C           GIVEN.
C
            CALL ZXMIN(FM,NPAR,1,500,2,PAR,HESS,GRAD,FMIN,WORK,IER)
C
C        TEST IF THE ACHIEVED ACCURACY IS SATISFACTORY, IF NOT OPTIMIZE
C        ONCE MORE WITH A STRONGER CONVERGENCE CRITERION
C
            IF(WORK(3).LT.3.) CALL ZXMIN(FM,NPAR,2,500,1,PAR,HESS,GRAD,
     1      FMIN,WORK,IER)
         ENDIF
C
C
C        PACK THE OPTIMAL PARAMETER VALUES INTO PAR, WITHOUT TRANSFORMATION.
C        THE VARIANCE IS ALSO INCLUDED IN PAR.
         CALL SETUP(NUNPAR,PAR,0,1)
         NPAR=NPAR+1
         ISIG2=1
         PAR(NPAR)=SIG2
C
C        HCAL COMPUTES AN ESTIMATE OF THE HESSIAN OF THE FUNCTION FM1,
C        AS A FUNCTION OF ALL THE PARAMETERS IN PAR. FM1 COMPUTES THE
C        LIKELIHOOD AS A FUNCTION OF ALL THE PARAMETERS, INCLUDING THE
C        VARIANCE.
C
         CALL HCAL(FM1,NPAR,PAR,HESS,FMIN)
C
C        THE IMSL-LIBRARY ROUTINE LINV3P INVERTS THE HESSIAN, TO GET THE
C        VARIANCE-COVARIANCE MATRIX OF THE PARAMETER ESTIMATES.
C
         CALL LINV3P(HESS,GRAD,1,NPAR,IER)
         IIND=0
C
C        STORE THE STANDARD DEVIATIONS IN GRAD, AND CHANGE THE HESSIAN TO
C        A CORRELATION MATRIX OF THE PARAMETER ESTIMATES. IF THE HESSIAN
C        WERE TOO CLOSE TO SINGULAR TO BE INVERTED, THEN THE STANDARD
C        DEVIATIONS ARE ALL SET TO -1.
C
         DO 25 I=1,NPAR
         IF(IER.NE.0) THEN
            GRAD(I)=-1.
         ELSE
            GRAD(I)=SQRT(HESS(IIND+I))
            DO 20 J=1,I
   20       HESS(IIND+J)=HESS(IIND+J)/GRAD(I)/GRAD(J)
         ENDIF
   25    IIND=IIND+I
C
C        ******* E N D   O F   E S T I M A T I O N   S E S S I O N *****
C
C        ****** S T A R T   O F   O U T P U T   S E S S I O N *******
C
C        WRITE PAGE HEADING
C
   27    WRITE(6,30) TITLE,NONMIS,NOBS
   30    FORMAT('1',//,1X,A60,/,I8,' NON-MISSING OBS. OUT OF',I5,/)
         WRITE(6,35) 'OUTPUT SERIES',VARNAM(INDOUT)
   35    FORMAT(//,1X,A,5X,A8,/)
         IIND=IP+IQ+IAMY+IR2
C
C        WRITE THE NOISE OR UNIVARIATE MODEL PARAMETERS. RESOUT IS A BASIC
C        PRINT ROUTINE.
C
         CALL RESOUT('NOISE MODEL',' ',0,0,ID,0,0,D,D)
         CALL RESOUT(' ','AR-PARAM.',1,IP,0,0,1,FI,GRAD(1))
         CALL RESOUT(' ','MA-PARAM.',IP+1,IQ,0,0,1,TET,GRAD(IP+1))
         CALL RESOUT(' ','CONSTANT',IP+IQ+1,IAMY,0,0,0,AMY,GRAD(IP+IQ+1))
C
C        THE MEASUREMENT ERROR OPTION IS PRESENTLY NOT INCLUDED
C
C        CALL RESOUT(' ','MEAS.VAR.',IIND,IR2,0,0,0,R2,GRAD(IIND))
C
```

```
C        WRITE THE TRANSFER FUNCTION PARAMETERS FOR EACH INPUT, IF ANY.
C
         DO 50 J=1,NINP
         J1=INDIN(J)
         NPJ=NP(J)
         CALL RESOUT('INPUT NO. '//SNR(J)//VARNAM(J1),' ',0,0,0,IB(J),
        1 0,0,0)
         CALL RESOUT(' ','TRANS.PAR.',IIND+1,NPJ,0,0,IB(J),PAR(IIND+1),
        1 GRAD(IIND+1))
   50    IIND=IIND+NPJ
C
C        WRITE THE RESIDUAL VARIANCE
C
         WRITE(6,35) ' '
         WRITE(*,35) ' '
         CALL RESOUT(' ','RES.VAR.',IIND+1,1,0,0,0,SIG2,GRAD(IIND+1))
C
C        WRITE THE CORRELATION MATRIX OF THE ESTIMATES, PROVIDED THE
C        HESSIAN WAS NOT TO CLOSE TO SINGULAR.
C
         IF(IER.NE.0) THEN
            WRITE(6,35) 'HESSIAN TOO NEAR SINGULAR - NO CORRELATIONS',' '
         ELSE
            WRITE(6,35) 'CORRELATION MATRIX OF THE ESTIMATES',' '
            WRITE(6,51) (J,J=1,NPAR)
   51       FORMAT(4X,20I6)
            IIND=0
            DO 53 I=1,NPAR
            WRITE(6,52) I,(HESS(IIND+J),J=1,I)
            IIND=IIND+I
   52       FORMAT(4X,I2,20F6.3)
   53       CONTINUE
         ENDIF
C
C        FINALLY, WRITE THE VALUE OF THE AKAIKE INFORMATION CRITERION (AIC)
C
         WRITE(6,57) FMIN+2*NPAR-2
   57    FORMAT(//,' A I C:',E12.5)
C
C        ******** E N D  O F   O U T P U T   S E S S I O N ********
C
C        ******** E N D  O F  O N E  E S T I M A T I O N  R U N ****
C
C        GO TO LABEL 81 TO START ANOTHER SESSION OR TO STOP.
C
         WRITE(*,58) FMIN+2*NPAR-2
   58    FORMAT(/,' A I C:',E12.5)
         GOTO 81
  100    STOP
         END
```

```
      SUBROUTINE CNOISE(PAR)
C
C     SUBROUTINE CNOISE TRANSFERS THE DATA OF THE OUTPUT SERIES FROM THE
C     OBSERVATION MATRIX Y IN COMMON /CDATA/ TO THE ARRAY X IN COMMON/OBS/.
C     IF THE MODEL UNDER CONSIDERATION IS A TRANSFER FUNCTION MODEL, THEN
C     THE EFFECTS OF THE INPUT(S) ARE CALCULATED BY CNOISE AND SUBTRACTED
C     FROM THE OUTPUT SERIES BEFORE STORING IN X, SO THAT X, IN THIS CASE,
C     CONTAINS THE NOISE SERIES.
C
C     ***********************************************************
C     *****   WRITTEN BY: EIVIND DAMSLETH                   *****
C     *****   LAST PEVISION: MAY 21, 1987                   *****
C     ***********************************************************
C
      PARAMETER (MAXLEN=1500,MAXSER=10)
      DIMENSION PAR(1)
      COMMON/OBS/N,NONMIS,XMISS,CODE,X(MAXSER)
      COMMON/CDATA/NOBS,Y(MAXLEN,MAXSER)
      COMMON/INPUT/NSERIE,INDOUT,INDIN(5),IB(5),NP(5)
C
C     IN THE CASE OF A UNIVARIATE MODEL, THERE IS NO NEED TO TRANSFER
C     THE DATA FROM THE Y-MATRIX TO THE X-VECTOR MORE THAN ONCE. THE
C     VALUE OF N IS SET TO -1 BY THE CALLING PROGRAM PRIOR TO THE FIRST
C     CALL TO CNOISE, AND IS RESET TO NOBS AT THE EXIT OF CNOISE, SO
C     THAT N = NOBS ON SUBSEQUENT CALLS. IN THIS CASE, THE ROUTINE GOES
C     TO LABEL 100 AND RETURNS.
C
      IF(NSERIE.EQ.1.AND.N.EQ.NOBS) GOTO 100
C
C     TRANSFER THE OUTPUT SERIES FROM THE Y-MATRIX TO THE X-VECTOR. AT
C     THE SAME TIME CHANGE THE MISSING VALUE CODE FROM THE USER DEFINED
C     XMISS IN THE Y-MATRIX TO THE INTERNAL CODE IN THE X-VECTOR.
C
      NINP=NSERIE-1
      DO 90 I=1,NOBS
      X(I)=Y(I,INDOUT)
      IF(X(I).LE.XMISS) THEN
         X(I)=CODE
         GOTO 90
      ENDIF
      IPIND=0
C
C     START LOOP TO SUBTRACT THE EFFECTS OF THE POSSIBLE INPUT(S).
C     THE RESULTING NOISE VALUE IS CONSIDERED AS OBSERVED ONLY IF ALL
C     THE VALUES FOR THE INPUT(S) WHICH ARE INVOLVED IN ITS COMPUTATION
C     ARE OBSERVED. OTHERWISE THE NOISE VALUE IS REGARDED AS MISSING.
C
      DO 40 J=1,NINP
      IF(I.LT.IB(J)+NP(J).OR.I.GT.NOBS+IB(J)) THEN
         X(I)=CODE
         GOTO 90
      ENDIF
      J1=INDIN(J)
      NPJ=NP(J)
C
C     START LOOP TO COMPUTE THE TOTAL EFFECT OF THE JTH INPUT SERIES.
C
      DO 20 K=1,NPJ
      K1=I-IB(J)-K+1
      IF(Y(K1,J1).LE.XMISS) THEN
         X(I)=CODE
         GOTO 90
      ELSE
         X(I)=X(I)-PAR(IPIND+K)*Y(K1,J1)
      ENDIF
   20 CONTINUE
   40 IPIND=IPIND+NPJ
   90 CONTINUE
C
C     RESET N TO NOBS, TO SIMPLIFY SUBSEQUENT CALLS.
C
      N=NOBS
  100 RETURN
      END
```

```
      SUBROUTINE DDIV(A,N,ID,AA)
C
C     SUBROUTINE DDIV FINDS THE N FIRST COEFFICIENTS IN THE POLYNOMIAL
C     EXPANSION OF
C         (1-A(1)*B-A(2)*B**2-...-A(N)*B**N) / (1-B)**ID =
C               (1-AA(1)*B-AA(2)*B**2-...-AA(N)*B**N) + O(B**(N+1))
C
C     ***********************************************************
C     *****   WRITTEN BY: EIVIND DAMSLETH                   *****
C     *****   LAST REVISION: MAY 21. 1987                   *****
C     ***********************************************************
C
      DIMENSION A(1),AA(1)
      DO 10 I=1,N
   10 AA(I)=A(I)
      DO 30 J=1,ID
      DO 20 I=2,N
   20 AA(I)=AA(I)+AA(I-1)
   30 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE DMULT(A,N,ID,AA,NN)
C
C     SUBROUTINE DMULT FINDS THE COEFFICIENTS AA IN THE POLYNOMIAL
C     MULTIPLICATION
C        (1-A(1)*B-A(2)*B**2-...-A(N)*B**N) * (1-B)**ID =
C                   (1-AA(1)*B-AA(2)*B**2-...-AA(NN)*B**NN)
C     WHERE A,N AND ID ARE GIVEN AS INPUT.
C
C     ******************************************************
C     *****  WRITTEN BY: EIVIND DAMSLETH                *****
C     *****  LAST REVISION: MAY 21. 1987                *****
C     ******************************************************
C
      DIMENSION A(1),AA(1)
      DO 10 I=1,N
   10 AA(I)=A(I)
      NN=N
      DO 20 J=1,ID
      AOLD=-1.
      DO 15 I=1,NN
      AST=AA(I)
      AA(I)=AST-AOLD
   15 AOLD=AST
      NN=NN+1
   20 AA(NN)=-AOLD
      RETURN
      END
```

```
      SUBROUTINE DTRAN(P,C,M,K,ID)
C
C     SUBROUTINE DTRAN TRANSFORMS THE COVARIANCE MATRIX P IN THE
C     STATIONARY CASE TO A NEW COVARIANCE MATRIX WHERE THE DIFFE-
C     RENCING IS TAKEN INTO ACCOUNT. DTRAN IS RESTRICTED TO THE
C     CASES WHERE THE NUMBER OF DIFFERENCES, ID <= 2. M IS THE
C     DIMENSION OF P (MXM), K IS THE INDEX OF THE SECOND ACTUALLY
C     OBSERVED OBSERVATION, RELATIVE TO THE FIRST, SO THAT THERE
C     ARE (K-1) MISSING VALUES BETWEEN THE FIRST AND SECOND OBSERVED
C     VALUE. K IS USED ONLY WHEN ID = 2.
C
C     C IS AN AUXILLARY ARRAY, COMPUTED IN THE CALLING ROUTINE INIT.
C
C     ***********************************************************
C     *****   WRITTEN BY: EIVIND DAMSLETH                   *****
C     *****   LAST REVISION: MAY 21. 1987                   *****
C     ***********************************************************
C
      DIMENSION P(10,1),C(1),H(10,10)
C
C     IF ID = 0 NO TRANSFORMATION IS REQUIRED, SO THE ROUTINE GOES TO
C     LABEL 200, WHICH IS THE RETURN LABEL.
C
      IF(ID.EQ.0) GOTO 200
C
C     H IS A LOCAL MATRIX TO BE USED IN THE CALCULATIONS.
C
      DO 5 I=1,M
    5 H(1,I)=0.
C
C     IF ID = 1 GOTO LABEL 100 AND RESUME CALCULATIONS THERE.
C
      IF(ID.EQ.1) GOTO 100
C
C     START OF THE ID = 2 SESSION.
C
      B=1/(K-1.)
      A=1-B
      KM3=K-3
      DO 80 J=2,M
      DO 80 I=J,M
      S=(J-1)*A*P(1,1)
      DO 10 JQ=2,J
   10 S=S+(J-JQ+1)*P(1,JQ)
      Z=(I-1)*A*S
      DO 30 JP=2,I
      S=(J-1)*A*P(JP,1)
      JPM1=JP-1
      DO 20 JQ=2,JPM1
   20 S=S+(J-JQ+1)*P(JQ,JP)
      DO 25 JQ=JP,J
   25 S=S+(J-JQ+1)*P(JP,JQ)
   30 Z=Z+(I-JP+1)*S
      S=0.
      DO 40 JQ=1,KM3
   40 S=S+(1-(JQ+1)*B)*C(JQ+1)
      Z=Z+2*(J-1)*(I-1)*A*S
      DO 50 JP=2,J
      S=0.
      DO 45 JQ=1,KM3
   45 S=S+(1-(JQ+1)*B)*C(JP+JQ)
   50 Z=Z+((I-JP+1)*(J-1)+(J-JP+1)*(I-1))*S
      JP1=J+1
      DO 60 JP=JP1,I
      S=0.
      DO 55 JQ=1,KM3
   55 S=S+(1-(JQ+1)*B)*C(JP+JQ)
   60 Z=Z+(I-JP+1)*(J-1)*S
      S=(1-2*B)**2*C(1)
      DO 70 JP=2,KM3
      S1=0.
      DO 65 JQ=2,JP
   65 S1=S1+(1-JQ*B)*C(JP-JQ+2)
   70 S=S+(1-(JP+1)*B)*(2*S1+(1-(JP+1)*B)*C(1))
   80 H(J,I)=Z+(I-1)*(J-1)*S
```

```
      C
      C     END OF THE ID = 2 SESSION, JUMP TO LABEL 150.
      C
            GOTO 150
      C
      C     START OF ID = 1 SESSION
      C
  100       DO 110 J=2,M
            H(J,J)=P(J,J)+2*H(J-1,J)-H(J-1,J-1)
            JP1=J+1
            DO 110 I=JP1,M
  110       H(J,I)=P(J,I)+H(J-1,I)+H(J,I-1)-H(J-1,I-1)
      C
      C     END OF ID = 1 SESSION.
      C
      C     THE TRANSFORMED MATRIX IS NOW STORED IN H. RESTORE IT IN P.
      C
  150       DO 190 J=1,M
            DO 190 I=J,M
  190       P(J,I)=H(J,I)
  200       RETURN
            END
```

```
      SUBROUTINE FM(NPAR,PAR,FMIN)
C
C     SUBROUTINE FM COMPUTES THE VALUE OF THE LIKELIHOOD FOR A GIVEN
C     SET OF PARAMETERS, MINIMIZED WITH RESPECT TO THE RESIDUAL VARIANCE
C     SIG2. FM ASSUMES THE PARAMETERS, GIVEN IN PAR, TO BE IN THE
C     TRANSFORMED, UNRESTRICTED PARAMETER SPACE. THE MINIMIZING VALUE OF
C     SIG2 IS ALSO COMPUTED.
C
C     *******************************************************
C     *****  WRITTEN BY: EIVIND DAMSLETH                *****
C     *****  LAST REVISION: MAY 21. 1987                *****
C     *******************************************************
C
      PARAMETER (MAXLEN=1500,MAXSER=10)
      DIMENSION PAR(1),G(10),P(10,10),Z(10)
      COMMON/ARP/ID,IP,FI(1)
      COMMON/MAP/IQ,TET(1)
      COMMON/MOM/IAMY,AMY,IR2,R2,ISIG2,SIG2
      COMMON/OBS/N,NONMIS,XMISS,CODE,X(MAXSER)
C
C     THE CALL TO SETUP UNPACKS THE PARAMETER VALUES IN PAR INTO FI,
C     TET ETC. AT THE SAME TIME, THE PARAMETERS ARE TRANSFORMED FROM
C     THE UNRESTRICTED PARAMETER SPACE TO THE RESTRICTED SPACE WHICH
C     ENSURES STATIONARITY AND INVERTIBILITY.
C
      CALL SETUP(NUNPAR,PAR,1,0)
C
C     THE ROUTINE CNOISE TRANSFERS THE DATA OF THE OUTPUT SERIES FROM THE
C     OBSERVATION MATRIX Y IN COMMON /CDATA/ TO THE ARRAY X IN COMMON/OBS/.
C     IF THE MODEL UNDER CONSIDERATION IS A TRANSFER FUNCTION MODEL, THEN
C     THE EFFECTS OF THE INPUT(S) ARE CALCULATED BY CNOISE AND SUBTRACTED
C     FROM THE OUTPUT SERIES BEFORE STORING IN X, SO THAT X, IN THIS CASE,
C     CONTAINS THE NOISE SERIES.
C
      CALL CNOISE(PAR(NUNPAR+1))
C
C     THE CALL TO INIT INITIALIZES THE KALMAN RECURSION, FINDING THE
C     INITIAL VALUES FOR THE STATE VECTOR Z AND ITS COVARIANCE MATRIX P.
C
      CALL INIT(Z,P,G,KV)
      NONMIS=0
      SIG2=0.
      VS=0.
C
C     START LOOP OF KALMAN RECURSIONS THROUGH ALL THE OBSERVATIONS
C
      DO 30 I=KV,N
C
C     SUBROUTINE KALREC PERFORMS THE KALMAN RECURSIVE STEP, UPDATING Z AND P
C     AND COMPUTING THE RESIDUAL A AND ITS VARIANCE V.
      CALL KALREC(I,Z,P,A,V,G)
C
      IF(V.GT.0.) THEN
C
C        V>0 IMPLIES THAT THE ASSOCIATED OBSERVATION IS NON-MISSING
C
         SIG2=SIG2+A**2/V
         VS=VS+ALOG(V)
      ENDIF
   30 CONTINUE
C
C     SIG2 IS THE OPTIMAL ESTIMATE FOR THE RESIDUAL VARIANCE
C
      SIG2=SIG2/NONMIS
C
C     FMIN CONTAINS THE VALUE OF THE LIKELIHOOD, MINIMIZED WITH RESPECT TO
C     THE RESIDUAL VARIANCE.
C
      FMIN=(NONMIS*ALOG(SIG2)+VS+NONMIS)*.5
C
C     PRINT THE VALUE OF FMIN ON THE TERMINAL, SO THE USER CAN FOLLOW
C     THE PROGRESSION IN THE OPTIMIZATION.
C
      WRITE(*,*) 'FMIN    ',FMIN
      RETURN
```

```
76                END
```

```
      SUBROUTINE FM1(NPAR,PAR,FMIN1)
C
C     SUBROUTINE FM1 COMPUTES THE VALUE OF THE LIKELIHOOD FOR A GIVEN
C     SET OF PARAMETERS, WHEN THE RESIDUAL VARIANCE IS INCLUDED IN THE
C     PARAMETER SET. FM ASSUMES THE PARAMETERS, GIVEN IN PAR, TO BE IN THE
C     UNTRANSFORMED, RESTRICTED PARAMETER SPACE.
C
C     ******************************************************
C     *****  WRITTEN BY: EIVIND DAMSLETH                *****
C     *****  LAST REVISION: MAY 21. 1987                *****
C     ******************************************************
C
      PARAMETER (MAXLEN=1500,MAXSER=10)
      DIMENSION PAR(1),G(10),P(10,10),Z(10)
      COMMON/ARP/ID,IP,FI(1)
      COMMON/MAP/IQ,TET(1)
      COMMON/MOM/IAMY,AMY,IR2,R2,ISIG2,SIG2
      COMMON/OBS/N,NONMIS,XMISS,CODE,X(MAXSER)
C
C     THE CALL TO SETUP UNPACKS THE PARAMETER VALUES IN PAR INTO FI,
C     TET ETC..
C
      CALL SETUP(NUNPAR,PAR,1,1)
      SIG2=PAR(NPAR)
C
C     THE ROUTINE CNOISE TRANSFERS THE DATA OF THE OUTPUT SERIES FROM THE
C     OBSERVATION MATRIX Y IN COMMON /CDATA/ TO THE ARRAY X IN COMMON/OBS/.
C     IF THE MODEL UNDER CONSIDERATION IS A TRANSFER FUNCTION MODEL, THEN
C     THE EFFECTS OF THE INPUT(S) ARE CALCULATED BY CNOISE AND SUBTRACTED
C     FROM THE OUTPUT SERIES BEFORE STORING IN X, SO THAT X IN THIS CASE
C     CONTAINS THE NOISE SERIES.
C
      CALL CNOISE(PAR(NUNPAR+1))
C
C     THE CALL TO INIT INITIALIZES THE KALMAN RECURSION, FINDING STARTING
C     VALUES FOR THE STATE VECTOR Z AND ITS COVARIANCE MATRIX P.
C
      CALL INIT(Z,P,G,KV)
      NONMIS=0
      S2=0.
      VS=0.
C
C     START LOOP OF KALMAN RECURSIONS THROUGH ALL THE OBSERVATIONS
C
      DO 30 I=KV,N
C
C     SUBROUTINE KALREC PERFORMS THE KALMAN RECURSIVE STEP, UPDATING Z AND P
C     AND COMPUTING THE RESIDUAL A AND ITS VARIANCE V.
      CALL KALREC(I,Z,P,A,V,G)
C
      IF(V.GT.0.) THEN
C
C        V>0 IMPLIES THAT THE ASSOCIATED OBSERVATION IS NON-MISSING
C
         S2=S2+A**2/V
         VS=VS+ALOG(V)
      ENDIF
   30 CONTINUE
C
C     FMIN1 CONTAINS THE VALUE OF THE LIKELIHOOD.
C
      FMIN1=(NONMIS*ALOG(SIG2)+VS+S2/SIG2)*.5
C
C     PRINT THE VALUE OF FMIN1 ON THE TERMINAL, SO THE USER CAN FOLLOW
C     THE CALCULATIONS.
C
      WRITE(*,*) 'FMIN1  ',FMIN1
      RETURN
      END
```

```
      SUBROUTINE HCAL(F,N,P,H,FMIN)
C
C     SUBROUTINE HCAL CALCULATES AN ESTIMATE OF THE HESSIAN OF THE
C     N-VARIABLE FUNCTION F IN THE POINT WITH COORDINATES GIVEN IN P.
C     FMIN IS THE VALUE OF F IN P, GIVEN AS INPUT, AND H IS THE
C     RESULTING MATRIX OF DOUBLE DERIVATIVES. THE SYMMETRY OF H IS
C     UTILIZED, AND H IS GIVEN AS A VECTOR CONTAINING THE LOWER TRIANGLE
C     STORED ROW BY ROW.
C
C     ****************************************************************
C     *****   WRITTEN BY: EIVIND DAMSLETH                         *****
C     *****   LAST REVISION: MAY 21, 1987                         *****
C     ****************************************************************
C
      DIMENSION H(1),P(1),D(20)
      EXTERNAL F
      CALL F(N,P,FMIN)
      IIND=0
      DO 30 I=1,N
      PI=P(I)
      EI=AMAX1(ABS(PI)*.001,.0005)
      P(I)=PI+EI
      CALL F(N,P,D(I))
      IM1=I-1
      DO 20 J=1,IM1
      PJ=P(J)
      EJ=AMAX1(ABS(PJ)*.001,.0005)
      P(J)=PJ+EJ
      CALL F(N,P,F11)
      H(IIND+J)=(F11-D(I)-D(J)+FMIN)/(EI*EJ)
   20 P(J)=PJ
      IIND=IIND+I
      P(I)=PI-EI
      CALL F(N,P,F11)
      H(IIND)=(D(I)-2*FMIN+F11)/(EI**2)
   30 P(I)=PI
      RETURN
      END
```

```
      SUBROUTINE INIT(Z,P,G,KV)
C
C     SUBROUTINE INIT CALCULATES THE STARTING VALUES FOR THE RECURSIVE
C     KALMAN FILTER. Z IS THE INITIAL STATE, P THE ASSOCIATED COVARIANCE
C     MATRIX, G IS AN AUXILLARY VECTOR WHICH IS NEEDED IN THE LATER
C     RECURSIONS AND KV IS THE INDEX OF THE OBSERVATION WHERE THE
C     RECURSIONS SHOULD START. KV IS THE INDEX OF THE (ID+1)ST OBSERVED
C     OBSERVATION, WHERE ID IS THE NUMBER OF DIFFERENCES IN THE MODEL.
C
C     ***********************************************************
C     *****  WRITTEN BY: EIVIND DAMSLETH                    *****
C     *****  LAST REVISION: MAY 21, 1987                    *****
C     ***********************************************************
C
      DIMENSION Z(1),P(10,1),G(1),C(10),SCRATC(10),KIND(2)
      COMMON/ARP/ID,IP,FI(1)
      COMMON/MAP/IQ,TET(1)
      COMMON/MOM/IAMY,AMY,IR2,R2,ISIG2,SIG2
      IP1=IP+ID
      M=MAXO(IP1,IQ+1)
C
C     INITIALIZE THE STATE VECTOR Z
C
      CALL ZINIT(Z,M,KIND)
C
C     INITIALIZE VECTOR G, STATIONARY VERSION
C
      G(1)=1.
      DO 4 J=2,M
      JM1=J-1
      IF(JM1.LE.IQ) THEN
         G(J)=-TET(JM1)
      ELSE
         G(J)=0.
      ENDIF
      JM1=MINO(JM1,IP)
      DO 3 I=1,JM1
    3 G(J)=G(J)+FI(I)*G(J-I)
    4 CONTINUE
C
C     INITIALIZE MATRIX P, STARTING WITH THE VECTOR C
C     STATIONARY VERSION
C
      MAX=MAXO(IP+1,M)
      DO 10 I=1,MAX
      C(I)=0.
      DO 10 J=1,MAX
   10 P(I,J)=0.
      DO 50 I=0,IP
      P(I+1,I+1)=1.+1.E-12
      DO 20 J=1,IP
      IND=IABS(I-J)
   20 P(I+1,IND+1)=P(I+1,IND+1)-FI(J)
      DO 30 J=I,IQ
      IF(J.EQ.0) THEN
         C(I+1)=1.
      ELSE
         C(I+1)=C(I+1)-TET(J)*G(J-I+1)
      ENDIF
   30 CONTINUE
   50 CONTINUE
C
C     SOLVE THE SET OF LINEAR EQUATIONS P*C=C
C
      CALL LEQT1F(P,1,IP+1,10,C,0,SCRATC,IER)
      IF(IER.GT.0) THEN
         WRITE(6,51) IER
   51    FORMAT(/,' IER =',I4,' IN SUBROUTINE LEQT1F')
         WRITE(6,53) IP,(FI(I),I=1,IP)
   53    FORMAT(' AR-ORDER=',I2,' PARAMETERS:',10F8.4)
         STOP
      ENDIF
      MIN=IP+2
      KV=KIND(2)-KIND(1)+1
      MAX=M+MAXO(KV-3,0)
```

```
      DO 80 I=MIN,MAX
      DO 60 J=1,IP
   60 C(I)=C(I)+FI(J)*C(I-J)
      JMAX=IO+1
      DO 70 J=I,JMAX
   70 C(I)=C(I)-TET(J-1)*G(J-I+1)
   80 CONTINUE
      DO 110 J=1,M
      DO 100 I=1,J
      P(I,J)=C(J-I+1)
      IM1=I-1
      DO 90 K=1,IM1
   90 P(I,J)=P(I,J)-G(K)*G(K+J-I)
  100 P(J,I)=P(I,J)
  110 CONTINUE
      IF(ID.EQ.0) GOTO 200
C
C     TRANSFORM ARRAYS P, FI AND G TO INCLUDE DIFFERENCING
C
      CALL DTRAN(P,C,M,KV,ID)
      CALL DMULT(FI,IP,ID,FI,IP1)
      CALL DDIV(G,M,ID,G)
  200 CONTINUE
      KV=KIND(2)+1
      RETURN
      END
```

```
      SUBROUTINE KALREC(IND,Z,P,A,V,G)
C
C     SUBROUTINE KALREC PERFORMS THE KALMAN RECURSIVE STEP, UPDATING THE
C     STATE VECTOR Z AND ITS COVARIANCE MATRIX P, ACCORDING TO THE
C     OBSERVATION X(IND). IF X(IND) IS OBSERVED, THEN THE ASSOCIATED
C     RESIDUAL IS RETURNED IN A, WITH ITS VARIANCE IN V. IF X(IND) IS
C     MISSING THEN A=0 AND V=-1 ON RETURN.
C
C     ************************************************************
C     *****  WRITTEN BY: EIVIND DAMSLETH                     *****
C     *****  LAST REVISION: MAY 21. 1987                     *****
C     ************************************************************
C
      PARAMETER (MAXLEN=1500,MAXSER=10)
      DIMENSION Z(1),G(1),P(10,1),H(10)
      COMMON/ARP/ID,IP,FI(1)
      COMMON/MAP/IQ,TET(1)
      COMMON/MOM/IAMY,AMY,IR2,R2,ISIG2,SIG2
      COMMON/OBS/N,NONMIS,XMISS,CODE,X(MAXSER)
      IP1=IP+ID
      M=MAXO(IP1,IQ+1)
      MM1=M-1
C
C     UPDATE Z TO GET Z(T+1/T)
C
      C=Z(1)*FI(M)
      S=FI(M)
      DO 5 J=1,MM1
      S=S+FI(M-J)
      Z(J)=Z(J+1)
    5 C=C+Z(J)*FI(M-J)
      Z(M)=C+IAMY*AMY*(1-S)
C
C     UPDATE P TO GET P(T+1/T)
C
      DO 9 I=1,M
    9 H(I)=P(1,I)
      DO 10 J=1,MM1
      DO 10 I=J,MM1
   10 P(J,I)=P(J+1,I+1)
      DO 20 J=1,MM1
      P(J,M)=H(J+1)*FI(M)
      JM1=J-1
      DO 15 I=1,JM1
   15 P(J,M)=P(J,M)+P(I,J)*FI(M-I)
      DO 20 I=J,MM1
   20 P(J,M)=P(J,M)+P(J,I)*FI(M-I)
      P(M,M)=0.
      DO 30 J=1,M
   30 P(M,M)=P(M,M)+H(J)*FI(M+1-J)
      P(M,M)=P(M,M)*FI(M)
      DO 40 J=2,M
   40 P(M,M)=P(M,M)+P(J-1,M)*FI(M+1-J)
      DO 50 J=1,M
      DO 50 I=J,M
   50 P(J,I)=P(J,I)+G(J)*G(I)
C
C     CALCULATE THE INNOVATION AND ITS VARIANCE. IF X(IND) IS MISSING
C     THEN THE INNOVATION IS SET TO 0 AND THE VARIANCE TO -1.
C     IF X(IND) IS MISSING, THEN THE UPDATING OF P(T+1/T+1) AND
C     Z(T+1/T+1) IS SKIPPED.
C
      IF(X(IND).LE.CODE) THEN
         A=0.
         V=-1.
      ELSE
         NONMIS=NONMIS+1
         A=X(IND)-Z(1)
         V=P(1,1)+R2
C
C        UPDATE Z TO GET Z(T+1/T+1)
C
         C=A/V
         DO 60 J=1,M
   60    Z(J)=Z(J)+P(1,J)*C
```

```
1      C
1      C          UPDATE P TO GET P(T+1/T+1)
1      C
1                 C=1/V
1                 P(1,1)=0.
1                 DO 75 J=2,M
1                 DO 70 I=J,M
1      70         P(J,I)=P(J,I)-P(1,J)*P(1,I)*C
1      75         P(1,J)=0.
1              ENDIF
               RETURN
               END
```

```
      SUBROUTINE RESOUT(T1,T2,NR,N,ND,IB,IO,P,SP)
C
C     SUBROUTINE RESOUT IS THE BASIC PRINT-ROUTINE, CALLED FROM THE
C     MAIN PROGRAM.
C
C     ************************************************************
C     *****   WRITTEN BY: EIVIND DAMSLETH                     *****
C     *****   LAST REVISION: MAY 21. 1987                     *****
C     ************************************************************
C
      CHARACTER T1*(*),T2*(*)
      DIMENSION P(1),SP(1)
      IF(T1.NE.' ') THEN
         IF(ND.EQ.0.AND.IB.EQ.0) THEN
            WRITE(6,10) T1
            WRITE(*,10) T1
         ENDIF
         IF(ND.GT.0) THEN
            WRITE(6,10) T1,'NO. OF DIFFS.=',ND
            WRITE(*,10) T1,'NO. OF DIFFS.=',ND
         ENDIF
         IF(IB.NE.0) THEN
            WRITE(6,10) T1,'DELAY =',IB
            WRITE(*,10) T1,'DELAY =',IB
         ENDIF
         WRITE(6,15)
         WRITE(*,15)
      ENDIF
   10 FORMAT(//,1X,A,3X,A,I3)
   15 FORMAT(/,18X,'PAR.NO. ORDER',6X,'VALUE',5X,'ST.DEV.',9X,'T')
      IF(N.GT.0) THEN
      INEG=1
      IF(SP(1).LT.0.) INEG=0
         WRITE(6,20) T2,NR,IO,P(1),(SP(1),P(1)/SP(1),IN=1,INEG)
         WRITE(*,20) T2,NR,IO,P(1),(SP(1),P(1)/SP(1),IN=1,INEG)
      ENDIF
   20 FORMAT(6X,A12,I4,I7,E15.4,E11.4,F10.2)
      DO 30 I=2,N
      WRITE(6,20) ' ',NR+I-1,IO+I-1,P(I),(SP(I),P(I)/SP(I),IN=1,INEG)
   30 WRITE(*,20) ' ',NR+I-1,IO+I-1,P(I),(SP(I),P(I)/SP(I),IN=1,INEG)
      RETURN
      END
```

```
      SUBROUTINE SETUP(NPAR,PAR,IND,IO)
C
C     SUBROUTINE SETUP PACKS THE UNIVARIATE PARAMETERS INTO ONE ARRAY
C     PAR, OR UNPACKS THE VALUES IN PAR INTO FI, TET ETC.
C     SETUP MAY ALSO TRANSFORM THE PARAMETERS FROM THE USUAL PARAMETER
C     SPACE TO AN UNRESTRICTED SPACE MORE SUITABLE FOR UNCONSTRAINED
C     OPTIMIZATION, AND VICE VERSA.
C
C     ***********************************************************
C     *****  WRITTEN BY: EIVIND DAMSLETH                    *****
C     *****  LAST REVISION: MAY 21. 1987                    *****
C     ***********************************************************
C
      DIMENSION PAR(1)
      COMMON/ARP/ID,IP,FI(1)
      COMMON/MAP/IQ,TET(1)
      COMMON/MOM/IAMY,AMY,IR2,R2,ISIG2,SIG2
      IF(IND.GT.0) THEN
C
C     IND>0 IMPLIES UNPACKING FROM PAR.
C
        IF(IAMY.GT.0) AMY=PAR(IP+IQ+1)
        IF(IO.LE.0) THEN
C
C         IO<0 IMPLIES TRANSFORMATION FROM THE UNRESTRICTED SPACE
C         BACK TO THE ORDINARY SPACE.
C         STABTRA PERFORMS THIS TRANSFORMATION FOR THE AR- AND MA-
C         PARAMETERS, TO ENSURE STATIONARITY AND INVERTIBILITY.
C
          CALL STABTRA(PAR(1),FI,IP,1)
          CALL STABTRA(PAR(IP+1),TET,IQ,1)
          R2=0.
          IF(IR2.GT.0) R2=EXP(PAR(IP+IQ+IAMY+1))
        ELSE
C
C         IO>0 IMPLIES NO TRANSFORMATIONS, JUST UNPACKING.
C
          DO 15 I=1,IP
   15     FI(I)=PAR(I)
          DO 20 I=1,IQ
   20     TET(I)=PAR(IP+I)
          R2=0.
          IF(IR2.GT.0) R2=PAR(IP+IQ+IAMY+1)
        ENDIF
C
C     UNUSED AR- AND MA-PARAMETERS ARE SET TO 0 (ZERO), FOR CONVENIENCE
C
        M=MAXO(IP,IQ+1)
        K=IP+1
        DO 30 I=K,M
   30   FI(I)=0.
        K=IQ+1
        DO 40 I=K,M
   40   TET(I)=0.
      ELSE
C
C     IND<0 IMPLIES PACKING FROM FI, TET ETC. INTO PAR, WITH OR WITHOUT
C     TRANSFORMATION.
C
      IF(IAMY.GT.0) PAR(IP+IQ+1)=AMY
        IF(IO.LE.0) THEN
C
C         IO<0 IMPLIES TRANSFORMATION FROM THE ORDINARY PARAMETER
C         SPACE TO THE UNRESTRICTED SPACE.
C
          CALL STABTRA(PAR(1),FI,IP,0)
          CALL STABTRA(PAR(IP+1),TET,IQ,0)
          IF(IR2.GT.0) PAR(IP+IQ+IAMY+1)=ALOG(R2)
        ELSE
C
C         IO>0 IMPLIES NO TRANSFORMATION, JUST PACKING.
C
          DO 45 I=1,IP
   45     PAR(I)=FI(I)
          DO 46 I=1,IQ
```

```
 46          PAR(IP+I)=TET(I)
             IF(IR2.GT.O) PAR(IP+IQ+IAMY+1)=R2
           ENDIF
         ENDIF
C
C        NPAR CONTAINS THE TOTAL NUMBER OF UNIVARIATE (NOISE) PARAMETERS
C
         NPAR=IP+IQ+IAMY+IR2
         RETURN
         END
```

```
      SUBROUTINE STABTRA(U,A,N,IND)
C
C     SUBROUTINE STABTRA TRANSFORMS AN UNRESTRICTED, RANDOM VECTOR U,
C     OF LENTH N, INTO A RESTRICTED ARRAY A, SO THAT THE ELEMENTS OF
C     ARRAY A FULFILLS THE REQUIREMENTS FOR A STATIONARY OPERATOR.
C     ALTERNATIVELY, STABTRA MAY TRANSFORM THE RESTRICTED ARRAY A
C     INTO THE UNRESTRICTED ARRAY U.
C     THE TRANSFORMATION USES THE DURBIN-LEVISON ALGORITHM.
C
C     ***********************************************************
C     *****  WRITTEN BY: EIVIND DAMSLETH                    *****
C     *****  LAST REVISION: MAY 21, 1987                    *****
C     ***********************************************************
C
      DIMENSION U(1),A(1),D(10)
      IF(IND.GT.0) THEN
C
C     FROM U INTO A, WHEN IND > 0.
C
        DO 20 J=1,N
        A(J)=2./(1+EXP(-U(J)))-1.
        JM1=J-1
        DO 10 K=1,JM1,1
   10   D(K)=A(K)-A(J)*A(J-K)
        DO 15 K=1,JM1
   15   A(K)=D(K)
   20   CONTINUE
      ELSE
C
C     FROM A INTO U, WHEN IND <= 0.
C
        DO 45 J=1,N
   45   U(J)=A(J)
        NM1=N-1
        DO 50 J=NM1,1,-1
        DO 30 K=1,J
   30   D(K)=(U(K)+U(J+1)*U(J+1-K))/(1-U(J+1)**2)
        DO 40 K=1,J
   40   U(K)=D(K)
   50   CONTINUE
        DO 60 J=1,N
   60   U(J)=ALOG((1+U(J))/(1-U(J)))
      ENDIF
      RETURN
      END
```

PROGRAMS FOR THE ANALYSIS OF TIME SERIES WITH MISSING OBSERVATIONS.

User manual for the programs CORRCAL and KALMAN

Eivind Damsleth