LSDE

Version 1

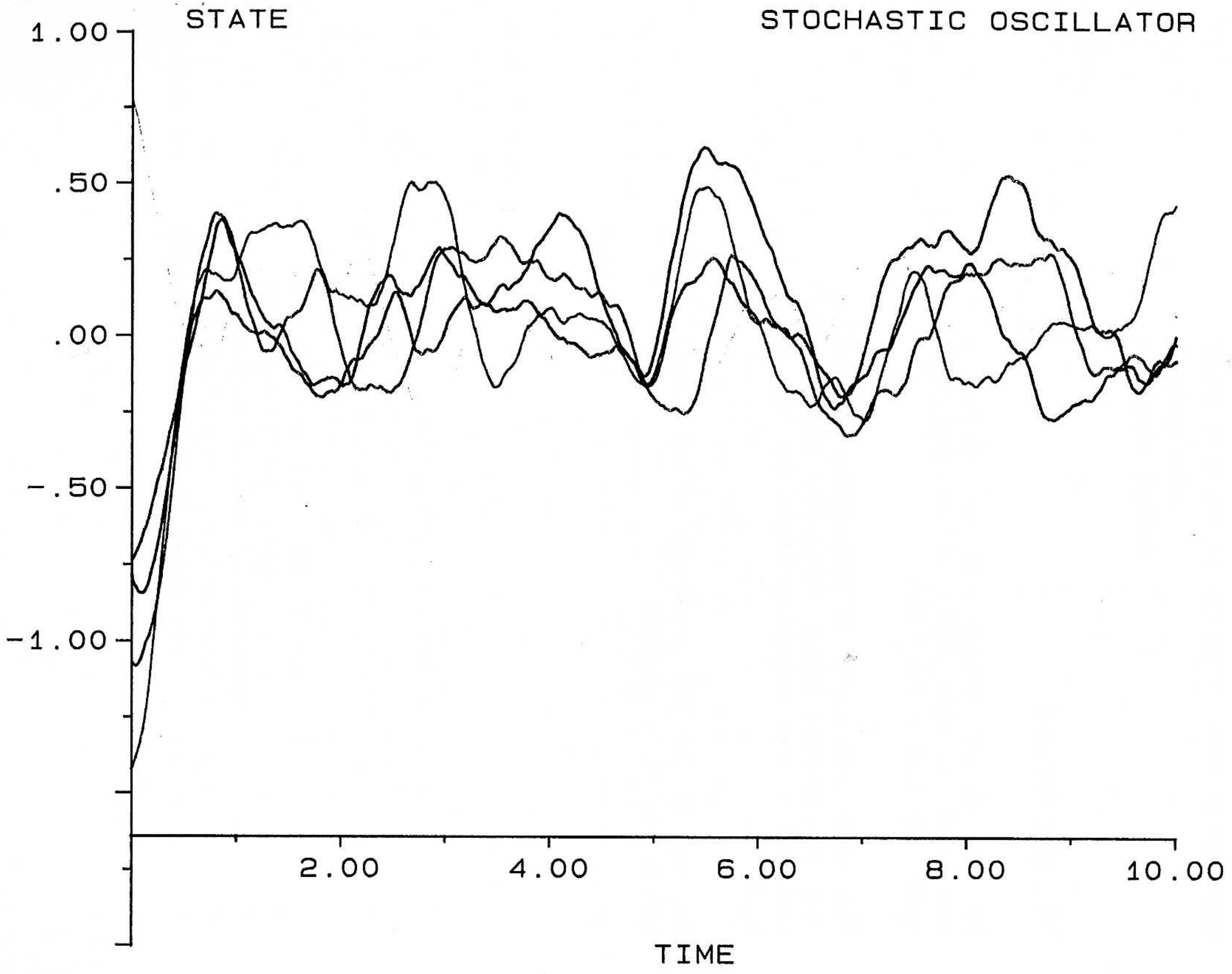
A Program Package for the

Simulation
Graphical Display
Optimal Filtering and
Maximum Likelihood Estimation of

LINEAR STOCHASTIC DIFFERENTIAL EQUATIONS

Hermann Singer

USER`S GUIDE


STATE
STOCHASTIC OSCILLATOR
1.00
.50
.00
-.50
-1.00
2.00
4.00
6.00
8.00
10.00
TIME

Copyright © 1990 - 1992 by Dr. Hermann Singer
Allmendweg 6
D-7758 Meersburg

All Rights reserved

No part of this manual may be adopted or reproduced in any form without the prior written approval of the author.

The author can accept no responsibility for any loss or damage arising from the information contained in this manual or from the use of LSDE.

The right to alter the specification of the product without warning is reserved.

CONTENTS

Chapter I
Introduction

Chapter II
Maximum Likelihood Estimation of Stochastic Differential Equations

Chapter III
Case Studies

Chapter IV
Reference

V. Appendix

Appendix I
Analysis of Discrete Time State Space Models 1

Appendix II
State Space Representations of ARMA and ARMAX models and Models containing Individual Specific Effects 3

References

Index

I. INTRODUCTION

Stochastic differential equations (SDE) are a natural choice to model the change of quantitative variables in continuous time. It is assumed that the **rate of change** of these state variables is given by a linear combination of the present state components under the influence of exogenous (control) variables and noise . For example, the change of the number of people on earth is proportional to the number already present. The proportionality constant, which is the percentual change per time unit, is normally an unknown parameter to be estimated from the data. Another example is Seligman`s theory of *learned helplessness* (cf. Singer/Hautzinger 1988): In this case the temporal variations of depression, self esteem, attributional and life event variables are explained by linear combinations of these state variables and random disturbances. The several weights form a matrix of rates to be estimated from repeated measures on study subjects.

Continuous time stochastic models have been used in many fields, e.g. physics (Einstein, 1905, Langevin, 1908, Uhlenbeck and Ornstein, 1930), econometrics (Koopmans, 1950, A.W. Phillips, 1959, Bergstrom and Wymer, 1976, Merton, 1990), sociology (Simon, 1952, Coleman, 1968, Doreian and Hummon, 1976) and psychology (Thomas and Martin, 1976, Möbus and Nagl, 1983, Singer and Hautzinger, 1988) and others.

The **advantages of continuous time models** in contrast to discrete time series and panel models are (cf. also Möbus/Nagl 1983, Singer 1990):

- changes in the variables can occur at any time instant.
- extrapolation and interpolation of observed data is possible for arbitrary time points and not only for the chosen sampling interval (e.g. annual).
- cross validation of several studies carried out with different sampling intervals (time between panel waves) is possible since the parameters correspond to the same continuous time level.

The purpose of **LSDE** (Linear Stochastic Differential Equations) is the

- simulation,
- graphical display,
- optimal filtering and
- maximum likelihood estimation

of stochastic differential equations. The focus is especially on panel data, which refers to a collection of time series data for an ensemble of N systems or persons. The program is written in the matrix language SAS/IML (Interactive Matrix Language; SAS/IML software: Usage and Reference, Version 6, 1989) to provide a high level programming environment. Furthermore, existing SAS data sets can be handled with IML and other SAS software is integrated readily.

WHY DO WE NEED STILL ANOTHER SOFTWARE PACKAGE?

Especially in the sociological literature (e.g. Tuma and Hannan (1984), Arminger (1986)) the impression prevails, that it is possible to estimate the coefficients of a SDE with the help of simultaneous equations models (e.g. LISREL).
This impression is wrong. These proposed methods attempt to estimate a nonidentified model. After establishing identification by imposing restrictions on the structural parameters of the continuous time level, these restrictions cannot be handeled on the discrete time level which is the realm of LISREL models (for details see Hamerle, Nagl, Singer, 1991).

Therefore, a maximum likelihood method aiming directly at the original structural parameters was implemented (see Singer, 1990), which can handle arbitrary restrictions on the continuous time structural matrices. Furthermore, it allows to **think directly in the language of differential equations**, i.e. in **modeling the rate of change**. This is important since thinking in terms of structural equations does not comprise dynamical systems but is essentially static.

LSDE is a collection of modules written in the matrix language SAS/IML, which can handle matrix expressions in a way similar to mathematical notation. Since the complete source code is supplied, you are able to modify or extend the library according to your needs. This option provides you with complete freedom in your specific application. So **LSDE is an open system.** Users not familiar with programming and/or SAS/IML can nevertheless use the supplied modules as described in chapters 3 and 4.

Since IML runs on every computer system where a SAS implementation is available, LSDE runs on a variety of sites. The package was developed on an IBM compatible COMPAREX mainframe under the VM/CMS operating system. A minimum of 4 megabyte of virtual memory should be available.The LSDE source code is supplied in ASCII format on 3.5 ´´ two sided double density diskettes. To run LSDE on your computer, the SAS/IML language interpreter is a necessary condition.

This user`s guide is organized as follows:

In a theoretical chapter (II) the concept of stochastic differential equations is introduced and the methods of parameter estimation are discussed. This chapter also contains the algorithms implemented in **LSDE**. Users essentially interested in applications are not required to understand chapter II completely but should aquire an understanding for the model specification. For a more detailed discussion of deterministic and stochastic differential equations and stochastic filtering theory we refer to Jazwinski (1970), V.I. Arnold (1973), L. Arnold (1974), Liptser and Shiryayev (1977, 1978), Van Kampen (1981), Möbus/Nagl (1983) and Singer (1990).

This is followed by chapter III containing case studies. We discuss several models to describe simulated and real data sets from several fields (physics, econometrics, astronomy, sociology, psychology etc.). Here we give full details of the user supplied SAS/IML code necessary to achieve the desired results. It is not necessary to aquire a full understanding of the SAS/IML language since the essential tools are described in this chapter.

Finally, in a reference section the general structure of an IML/LSDE application, the implementation of the module library and the purpose and usage of the supplied modules is explained.

In an appendix the usage of a linearized option of LSDE is described which can be used to estimate **discrete** time series and panel models with errors of measurement.

II. MAXIMUM LIKELIHOOD ESTIMATION OF STOCHASTIC DIFERENTIAL EQUATIONS

1. INTRODUCTION

Stochastic differential equations (SDE) are a natural choice to model the time evolution of dynamical systems which are subject to random influences (cf. Soong (1973), Arnold (1974), Van Kampen (1981), Tuma and Hannan (1984)). For example, in physics the dynamics of ions in superionic conductors is modeled via Langevin equations (cf. Dieterich, Fulde and Peschel (1980)), in engineering the dynamics of mechanical devices is described by differential equations under the influence of process noise and errors of measurement (cf. Gelb (1974)). Other applications are in the fields of biology (Jennrich and Bright (1976); without process noise), econometrics (Bergstrom (1976,1988)), medicine (Jones (1984)), Sociology (Möbus/Nagl (1983)) and Psychology (Singer/Hautzinger (1988)).

This chapter adresses the problem of maximum likelihood (ML) estimation of the parameters of a SDE from time discrete observations. In practical applications, this is a situation arising frequently, since the characteristic dynamics of the system can be much faster than the sampling rate Δt. Then, estimation methods based on continuous sampling (Le Breton (1976) , Liptser and Shiryayev (1977, 1978) and their discretized versions) lead to seriously biased estimators (cf. Sargan (1976), Singer (1990)). Alternatively, in the linear case a difference equation valid at the the times of measurement can be derived (´exact discrete model´ , EDM), which contains the original parameters (drift and diffusion coefficient) in a nonlinear way (cf. Bergstrom (1976, ch. 2)), Hamerle, Nagl, Singer (1991a)).

Attempts to estimate the EDM after a reparametrization with LISREL or other structural equations models (Doreian and Hummon (1976), Sinha and Lastman (1982), Tuma and Hannan (1984), Arminger (1986), Bien, Schmidt and Schürhoff, (1990)) cannot be recommended, since due to aliasing there is a serious identification problem (Phillips (1973), Hansen and Sargent (1983)) which must be resolved first by imposing restrictions on the system matrices (for details, see Singer, 1990, Hamerle, Nagl and Singer, 1991a). Furthermore, if we include control variables, a functional involving drift matrix and control variables has to be approximated. It can be shown (Hamerle, Nagl, Singer (1991b)), that only in the case of piecewise constant controls an identification problem remains, but otherwise complicated ML-equations arise, which have to be solved numerically.

Another indirect method, due to Bartlett (1946) and Phadke and Wu (1974) utilizes a discrete time ARMA representation of the sampled continuous time process. However, as noted by the autors, restrictions on the continuous time parameters can be handled more conveniently if the likelihood is expressed in these quantities.

Therefore, a direct ML estimation method aiming at the original parameters of the continuous time level is proposed which is implemented in the **software package LSDE**

(Linear Stochastic Differential Equations) written in the matrix language SAS/IML (Singer (1990)). In contrast to former work of Jones (1984), Jones and Tryon (1987), who used numerical derivatives, we take computer generated exact analytic derivatives to achieve high accuracy and efficiency.
This chapter is organized as follows: In Section 2 the continuous/discrete state space model is specified and the exact discrete model is derived. Section 3 contains the maximum likelihood equations and a scoring algorithm for the exact discrete model using analytic derivatives. In section 4 the method is tested in a simulation study using a system of three differential equations (Phillips model). Section 5 treats the general model including a measurement equation using EM , GEM and quasi-Newton algorithms. Finally, in section 6, these methods are applied to the analysis of sunspot activity.

2. Model specification : Dynamic Structural Equations

We assume that the system can be described by a p-dimensional state vector $y_n(t)$, where n indexes a collection of N replications (persons). The dynamics are given through a system of linear (Itô) stochastic differential equations (cf. Arnold (1974))

$$dy_n(t) = Ay_n(t)dt + Bx_n(t)dt + Gdw_n(t) \ ; \qquad t \in [t_0, t_T],\ n=1,\dots\dots N \qquad (1)$$

with the random initial condition $y_n(0) \sim N(\mu,\Sigma)$ i.i.d. The disturbances dw_n are given by the increments of the r-dimensional Wiener processes $w_n(t)$, $x_n(t)$ are deterministic control variables and the dimensions of the drift A, input matrix B and diffusion coefficient G are p×p, p×q and p×r, respectively.
However, only measurements at (for simplicity equally spaced) discrete time points $t_0, \dots\dots, t_T$ are available (panel data):

$$z_{ni} = Hy_{ni} + Dx_{ni} + \varepsilon_{ni} \ ; \qquad (2)$$

where $y_{ni}=y_n(t_i)$; $i=0,\dots.T$ and $\varepsilon_{ni} \sim N(0,R)$ is a discrete time white noise disturbance representing errors of measurement (R: k×k) independent of $w_n(t)$. The dimensions of factor loadings H is k×p and D (k×q) describes the influence of control variables x_{ni} on the measurements.

It should be noted that the model specification (1, 2), called **continuous/discrete state space model** or **dynamic structural equations** is very general and allows the estimation of ARMA- and ARMAX-models with errors of measurement, models including fixed and random effects (person specific parameters), dynamic factor analysis, regression analysis with autocorrelated errors, models with nonwhite process errors, models with integrated variables (flow data), spline interpolation of data etc. For examples see chapters II and III.

The system model (1) can be solved explicitly (cf. Arnold (1974), ch. 8) and written as a system of difference equations (exact discrete model) valid at the time points $t_0,.....,t_T$:

$$y_{n,i+1} = A^* y_{ni} + F_{ni} + u_{ni} \quad ; i=0.....T-1, \ n=1.....N \tag{3}$$

where $A^*=\exp(A\Delta t)$, $F_{ni}=\int_0^{\Delta t}\exp(A(\Delta t-s))Bx_n(t_i+s)ds$ and $u_{ni}\sim N(0,\Omega^*)$ i.i.d. with

$\Omega^*= \int_0^{\Delta t} \exp(As)\Omega\exp(A`s)ds$, $\Omega=GG`$.

It should be noted that, in contrast to conventional autoregressive schemes, the covariance matrix Ω^* of the disturbances u_{ni} is dependent on the drift A and the diffusion matrix Ω. If the control variables x are not observed continuously, we have to approximate the functional F_{ni} to admit an explicit calculation. For example, if $x_n(t)$ is constant between t_i and t_{i+1} (step function approximation), we have

$$F_{ni} = A^{-1}(A^*-I)Bx_{ni} := B^* x_{ni}. \tag{4}$$

where nonsingularity of A has been assumed. This approximation will be used in the sequel.
It can be shown that the above model is **not identified** without imposing restrictions on the parameter matrices A, B and G of the system model (see Phillips, 1973, Singer, 1990, Hamerle, Nagl, Singer, 1991b). Therefore, **unrestricted estimation of A^*, B^* and Ω^* and subsequent inference of the original parameter matrices A, B and Ω is not possible.**
Restrictions on the matrices A, B and Ω permitting identification lead to very complicated nonlinear restrictions on the discrete time (*)-level which cannot be handled with programs like LISREL. Therefore, as explained in the sequel, we directly maximize the likelihood function with respect to the original parameter vector Θ. If we specify a saturated model without restrictions, the nonidentifiability shows up in the singularity of the Fisher information matrix F (see (8)).
Note that other approximations of the control (exogenous) variables between measurements (for example piecewise linear variation) lead to identification without further restrictions on the continuous time matrices , but again the parameters of the discrete model are nonlinearly restricted (for details see Hamerle/Nagl/Singer 1991b).

Now the likelihood of the data z_{ni} can be calculated conveniently using the Kalman filter algorithm (see Schweppe (1965)). Furthermore it is possible to derive a recursive scheme for the score function and the Hessian (cf. Goodrich (1978), Caines (1988)). Here we follow another line using the EM algorithm in a similar fashion as in Dembo and Zeitouni (1986) (continuous/continuous case) and Watson and Engle (1983)) (discrete/discrete case).

3. Maximum Likelihood Estimation of the exact discrete model with exact analytic derivatives

We proceed in two steps: first we consider an ML-estimation procedure for the system model (3) alone, which corresponds to the choice $H=I_p$, $D=0$, $\varepsilon_{ni}=0$ and then (ch. 5) introduce the measurement model (2) treating the data z_{ni} as an incomplete data set and use the EM algorithm.

3.1. Likelihood, Score and Fisher Information

The log-likelihood function corresponding to the exact discrete model (3) is given as

$$l_{\Theta}(Y) = -\frac{NT}{2}\left[\log|\Omega^{*}| + \mathrm{tr}[\Omega^{*-1}W]\right] - \frac{N}{2}\left[\log|\Sigma| + \mathrm{tr}[\Sigma \quad S_0^{-1}\ ]\right] \tag{5}$$

where we have abbreviated $W= (1/NT)\, UU`$, $U=Y_{+}-A^{*}Y_{\sim} - F(A,B,X)$, $Y=\left[Y_0,....Y_T\right]$: $p\times N(T+1)$, $Y_{+}=\left[Y_1\ ,...Y_T\right]$: $p\times NT$, $Y_{\sim}=\left[Y_0\ ,.....Y_{T-1}\right]$: $p\times NT$ and $F(A,B,X) = \left[F_0,.....F_{T-1}\right]$: $p\times NT$. The elements of the data matrices are blocks of the form $Y_i =[y_{1i}\ ,.....y_{Ni}\ ]$: $p\times N$ etc. and the likelihood of the initial condition $y_n(0)\sim N(\mu,\Sigma)$ contains the abbreviation $S_0= (1/N) \sum(y_{n0}-\mu)(y_{n0}-\mu)`$.

The parameter matrices A, B, $\Omega=GG`$, μ and Σ are assumed to be functions of an $u\times 1$-vector Θ. This permits a convenient implementation of restrictions (including intervals, positive semidefiniteness of Σ etc.)

In order to locate the maximum of the likelihood function (5) we have to solve the maximum likelihood equations

$$s(\Theta) = \partial l/\partial\Theta = 0 \tag{6}$$

where the score function is given explicitly as

$$s(\Theta) = \left[\partial\Theta^{*}/\partial\Theta\right]\left[\partial l/\partial\Theta^{*}\right] ,$$

$$\Theta^{*}=\left[\Omega^{*},\ A^{*},\ B^{*},\ \Sigma,\ \mu\right] \text{ and}$$

$$\partial l/\partial\Theta^{*}= \begin{bmatrix} (NT/2)\ \mathrm{row}[\ \Omega^{*-1}(W-\Omega^{*})\,\Omega^{*-1}] \\ \mathrm{row}[\ \Omega^{*-1}\,UY_{\sim}`] \\ \mathrm{row}[\ \Omega^{*-1}UX_{\sim}`\ ] \\ (N/2)\ \mathrm{row}[\ \Sigma^{-1}(S_0-\Sigma)\,\Sigma^{-1}] \\ N\ \mathrm{row}[\ \Sigma^{-1}(m-\mu)] \end{bmatrix}$$

Here *row* denotes the row-wise vector operation, $\partial Y/\partial X=\partial(\mathrm{row}Y)'/\partial\ \mathrm{row}\ X$ is a partitioned McDonald-Swaminathan matrix derivative (see McDonald and Swaminathan (1973)), $X_{\sim}=\left[X_0,\ldots X_{T-1}\right]$ and $m=(1/N)\sum y_{n0}$ is the mean of the initial condition.
In order to solve the ML-equations (6) we use a scoring algorithm

$$\Theta_{k+1} = \Theta_k + F_k^{-1} s_k \tag{7}$$

where $F_k=E_\Theta[ss']\big|_{\Theta=\Theta_k}$ is the Fisher information matrix evaluated at $\Theta=\Theta_k$.
To achieve a high efficiency and accuracy it is essential to obtain analytic expressions at least for the gradient s (cf. Dennis and Schnabel (1983)). Here we will obtain analytical expressions for F as well. However, for this purpose we have to calculate the matrix derivatives $\partial A^*/\partial\Theta$, $\partial B^*/\partial\Theta$ and $\partial\Omega^*/\partial\Theta$. The first one was already obtained in Jennrich and Bright (1976). In Singer (1990, appendix 3), we give an alternative derivation using a differential equation. Methods of this kind are well known in statistical mechanics and quantum theory (Wilcox (1967)) .
The Fisher information matrix can be calculated as (cf. Rao (1973), ch. 6)

$$\begin{aligned} F = E_\Theta\left[\ s\ s'\ \right] &= -E_\Theta\left[\ \partial^2 l\ /\partial\Theta^2\ \right] \\ &= \left[\partial\Theta^*/\partial\Theta\right] F^* \ \left[\partial\Theta^*/\partial\Theta\right]' \end{aligned}$$

where $$F^* = -E_\Theta\left[\ \partial^2 l\ /\partial\Theta^{*\,2}\ \right]= \tag{8}$$

$$\begin{bmatrix} \frac{NT}{2}\left(\Omega^{*-1}\otimes\Omega^{*-1}\right) & 0 & 0 & 0 & 0 \\ 0 & \Omega^{*-1}\otimes E_\Theta\left[Y_{\sim}Y_{\sim}'\right] & \Omega^{*-1}\otimes E_\Theta\left[Y_{\sim}\right]X_{\sim}' & 0 & 0 \\ 0 & \Omega^{*-1}\otimes X_{\sim}E_\Theta\left[Y_{\sim}'\right] & \Omega^{*-1}\otimes X_{\sim}X_{\sim}' & 0 & 0 \\ 0 & 0 & 0 & \frac{N}{2}\left(\Sigma^{-1}\otimes\Sigma^{-1}\right) & 0 \\ 0 & 0 & 0 & 0 & N\Sigma^{-1} \end{bmatrix}$$

and Θ denotes the true parameter. The expectations E_Θ can be evaluated recursively, since $\mu_{ni}=E_\Theta[y_{ni}]$ and $\Sigma_i=\mathrm{Var}_\Theta\left[y_{ni}\right]$ fulfil the equations

$$\begin{aligned} \mu_{n,\,i+1} &= A^*\mu_{ni} + B^*x_{ni} \quad ; \quad \mu_{n0}=\mu=E_\Theta[y_{n0}] \\ \Sigma_{i+1} &= A^*\Sigma_i A^{*'} + \Omega^* \quad ; \quad \Sigma_0=\Sigma=\mathrm{Var}_\Theta(y_{n0}) \end{aligned} \tag{9}$$

as is seen from the exact discrete model (3).
However, from a computational point of view it is more efficient to drop the expectati-

ons and use the moment matrices $\underset{\sim}{Y}\underset{\sim}{X}'$ and $\underset{\sim}{Y}\underset{\sim}{Y}'$ instead. These can be calculated before the scoring iterations and are asymptotically equivalent (in mean square and almost surely) to the expected values (see Singer (1990)).

3.2 Implementation of the scoring algorithm

The algorithm is part of the new software package LSDE (Linear Stochastic Differential Equations, Singer (1990)) , which permits simulation, graphical display and ML-estimation of the state space model (1, 2). It is written in the matrix language SAS/IML and therefore embedded in the usual SAS-environment (usage of SAS-data sets etc.). Since the matrix A is not symmetric in general, we have to find eigenvectors and eigenvalues which are complex. Following the remarks in Jennrich and Bright (1976) we assume that all eigenvalues of A are distinct and use a QR-algorithm (see Golub and Van Loan (1983)). The scoring method can be considered as a quasi Newton algorithm (cf. Dennis and Schnabel (1983)), where F takes the role of a model Hessian. After finding the maximum $\hat{\Theta}$, $F^{-1}(\hat{\Theta})$ provides an estimate of the covariance matrix of the ML-estimator (see e.g. Mardia, Kent, Bibby (1979), ch. 4).

4. Example: The Phillips Model

In order to test the performance of the scoring algorithm a simulated data set was used. SAS/IML code necessary to execute this example is given in chapter III. We consider the modified Phillips model in the form given by Phillips (1972):

$$d\begin{bmatrix} c(t) \\ y(t) \\ k(t) \end{bmatrix} = \begin{bmatrix} -\alpha & \alpha(1-s) & 0 \\ \lambda & \lambda(\gamma v-1) & -\lambda\gamma \\ 0 & \gamma v & -\gamma \end{bmatrix} \begin{bmatrix} c(t) \\ y(t) \\ k(t) \end{bmatrix} dt + \begin{bmatrix} \alpha \\ 0 \\ 0 \end{bmatrix} Fdt + \begin{bmatrix} \Theta_6 & 0 & 0 \\ 0 & \Theta_7 & 0 \\ 0 & 0 & \Theta_8 \end{bmatrix}^{1/2} \begin{bmatrix} dw_1(t) \\ dw_2(t) \\ dw_3(t) \end{bmatrix}$$

$$dY = AYdt + BXdt + GdW$$

Here the variables c, y, k and F are to be interpreted as *real consumption*, *net national income*, *fixed capital* and *autonomous consumption*, respectively.
We simulated M=100 data sets with T=25, N=1, $\Delta t=1$ and the choice of parameters $\Theta=\{\alpha,\lambda,\gamma,v,s,\Theta_6,\Theta_7,\Theta_8\}=\{.6,4.0,.4,2.0,.25,1,1,1\}$ and F=5 in accordance with the work of Phillips. The initial condition was $Y(0)=\lim E(Y(t))=-A^{-1}BX$. As was shown by Phillips (1973), the structural matrix A is identified in a three equations model with $A_{13}=0$ from $A^*=\exp(A\Delta t)$. The control variable F is constant so that the step function approximation (4) is exact here.
Starting the iterations from the true value Θ we obtain the final estimates Θ_f after typically 10 iterations where $||\Theta_f - \Theta_{f-1}||_\infty < \varepsilon=10^{-4}$. For simulation and estimation of

the 100 samples (additionally the linearized ML-estimator (discretized continuous sampling, Le Breton (1976)) was computed) a total CPU time of 1737.53 seconds was required. The calculations were executed on a BASF 7/73 computer under VM/CMS.

Results

Table II.1 shows mean values and standard deviations of the M=100 estimates Θ^m. In accordance with the study of Phillips (1972), the estimates of λ are much better for the exact ML method as compared to the linearized model (discretized continuous samp-

	true value	linearized ML	exact ML
$\bar{\alpha}$	.6	. 4998	.5723
		(.1529)	(.1539)
$\bar{\lambda}$	4.0	2.6746	4.1283
		(.7374)	(.7631)
$\bar{\gamma}$	.4	.3786	.4020
		(.0273)	(.0202)
$\bar{v}$	2.0	1.9962	2.0008
		(.0140)	(.0115)
$\bar{s}$	.25	.2547	.2509
		(.0251)	(.0151)
$\bar{\Theta}_6$	1	.9520	.8789
		(.2889)	(.2405)
$\bar{\Theta}_7$	1	3.1514	.9088
		(1.0151)	(.8638)
$\bar{\Theta}_8$	1	1.3358	1.060
		(.4002)	(.2821)

Table II.1 : *exact and linearized ML-estimates in M=100 simulated samples (means and standard errors (in parantheses)).*

ling, Le Breton (1976)). It should be noted that in our model it is possible to obtain estimates for the diffusion matrix $\Omega=GG'$ under arbitrary restrictions (we used a diagonal form), whereas in the "minimum distance method" of Phillips (1972), $\hat{\Omega}^*$ is calculated from the residuals. Therefore, restrictions on the diffusion matrix cannot be implemented with this method. However, such restrictions usually lead to identification of A, even when it is unrestricted.

5. ML- Estimation of the Continuous/Discrete State Space Model

In this section the general case including the measurement model (2) is treated. In order to take advantage of the scoring algorithm developed in the previous section, the EM algorithm of Dempster, Laird, Rubin (1977) is used in the form given by Dembo and Zeitouni (1986).
The log likelihood of the complete data $(Z,Y)=\{z_{ni},y_{ni};\ n=1,....,N;\ i=0,.....T\}$ can be decomposed into the conditional likelihood of the measurement model and the part of the system model (eq. 5)

$$l_\psi(Z,Y) = l_\Phi(Z\,|Y) + l_\Theta(Y) \tag{10}$$

where it is assumed that the matrices $\Phi^* =[H,\ D,\ R]$ depend on the $u_2 \times 1$ parameter vector Φ, $\Theta^*=\left[\Omega^*,\ A^*,\ B^*,\ \Sigma,\ \mu\right]$ depends on the $u_1 \times 1$ parameter vector Θ and $\psi=[\Theta',\Phi']'$. Although this does not allow cross restrictions between system model and measurement model, it permits separate (and more efficient) maximization (M) of both parts Q_m and Q_s of the pseudolikelihood Q, which is obtained as conditional expectation of (10) given the measured data Z.
More exactly, we have for the likelihood ratio of the measured data

$$L_\psi(Z)/L_{\psi'}(Z) = E_{\psi'}\left[L_\psi(Z,Y)/L_{\psi'}(Z,Y)\middle|Z\right] \ . \tag{12}$$

Taking logarithms and using the Jensen inequality, we obtain

$$\begin{aligned} l_\psi(Z) - l_{\psi'}(Z) &= \log\left(E_{\psi'}\left[L_\psi(Z,Y)/L_{\psi'}(Z,Y)\middle|Z\right]\right) \\ &\geq E_{\psi'}\left[\log\left(L_\psi(Z,Y)/L_{\psi'}(Z,Y)\right)\middle|Z\right] := Q(\psi,\psi') \end{aligned} \tag{13}$$

where $Q(\psi,\psi)=0$.
Setting $\psi'=\psi_k$ as current estimate in the k-th step of an iteration, we have the inequality

$$l_{\psi_{k+1}}(Z) \geq l_{\psi_k}(Z) + Q(\psi_{k+1},\psi_k) \geq l_{\psi_k}(Z) \tag{14}$$

where $\psi_{k+1} =\operatorname{argmax}_{\psi} Q(\psi,\psi_k)$ is the maximizer of the pseudolikelihood Q.

This defines the **EM-algorithm:**

E-step:

$$Q(\psi,\psi_k)=E_{\psi_k}\left[\, l_{\psi}(Z,Y) - l_{\psi_k}(Z,Y) \,\middle|\, Z\right] \tag{15}$$

(conditional expectation of complete data likelihood using current estimate)

M-step:

$$\psi_{k+1} = \operatorname{argmax}\ Q(\psi,\psi_k) \tag{16}$$

(Maximization of estimated log likelihood)

Goto E-Step (until convergence).

Since the second term in (15) does not depend on ψ it suffices to maximize

$$E_{\psi_k}\left[\, l_{\psi}(Z,Y)|Z\right] = E_{\psi_k}\left[l_{\Phi}(Z|Y)|Z\right] + E_{\psi_k}\left[l_{\Theta}(Y)|Z\right]$$

$$:= Q_m(\Phi) + Q_s(\Theta)$$

The maximization step can be performed using the quasi Newton algorithm of section 3 if we replace (by linearity) the data matrices in the log-likelihoods and scores by their conditional expectations. Moreover, using a formula of Louis (1982)

$$s_{\psi}(Z) := \partial l_{\psi}(Z)/\partial\psi = E_{\psi}\left[s_{\psi}(Z,Y)|Z\right] , \tag{17}$$

these modified scores coincide at $\psi=\psi_k$ with the exact score function of the data Z. More explicitly, we have for the conditional likelihood of the measurement model

$$l_{\Phi}(Z|Y) = \text{const.} - \tfrac{N(T+1)}{2}\left[\log |R| + \operatorname{tr}\left(R^{-1}S\right)\right] \tag{18}$$

where $S = \frac{1}{N(T+1)} EE'$, $E = Z - HY - DX$, $Z=\left[Z_0,\ldots..Z_T\right]$ and $X=\left[X_0,\ldots..X_T\right]$. For the likelihood of the system model see (5). Due to the linearity of the likelihoods (5) and (18) in the moments

$$M_s=\left\{\frac{1}{NT}\begin{bmatrix} Y_{+}Y_{+}' & Y_{+}Y_{\sim}' & Y_{+}X_{\sim}' \\ Y_{\sim}Y_{+}' & Y_{\sim}Y_{\sim}' & Y_{\sim}X_{\sim}' \\ X_{\sim}Y_{+}' & X_{\sim}Y_{\sim}' & X_{\sim}X_{\sim}' \end{bmatrix}, \frac{1}{N}Y_0Y_0' , \frac{1}{N}Y_0 1' \right\}$$

$$\text{and} \qquad M_m= \frac{1}{N(T+1)}\begin{bmatrix} ZZ' & ZY' & ZX' \\ YZ' & YY' & YX' \\ XZ' & XY' & XX' \end{bmatrix} \tag{19}$$

we have to compute the conditional expectations

$$\hat{M}_s = E_{\Psi_k}\left[M_s|Z\right] \text{ and } \hat{M}_m = E_{\Psi_k}\left[M_m|Z\right]$$

evaluated at the current estimate ψ_k.
These conditional expectations contain expressions of the form

$$E_{\psi_k}\left[y_{n,i+1}\, y_{n\,i}^{`}|Z\right] \;,\; E_{\psi_k}\left[y_{ni} y_{n\,i}^{`}|Z\right] \text{ and } E_{\psi_k}\left[y_{n,i}|Z\right] \tag{20}$$

which can be calculated recursively using a fixed interval Kalman smoother algorithm (cf. Watson and Engle (1983), Dembo and Zeitouni (1986)). The present software implementation utilizes the form given by Rauch, Tung and Striebel (1965), cf. also Lewis (1986). The first term in (20) can be produced by using an augmented state $\eta_{ni}=[y_{n\,i}^{`}, y_{n\,,i+1}^{`}]^{`}$ and due to the formula

$$E[y_{n,i+1} y_{n\,i}^{`}|Z] = Cov(y_{n,i+1}, y_{ni}|Z) + E[y_{n,i+1}|Z]\cdot E[y_{n\,i}^{`}|Z]$$

it is sufficient to implement this augmented smoother for the conditional covariance only.

Summarizing the derivation, we finally obtain the **practicable EM-algorithm:**

E-step:

$$Q(\Psi,\Psi_k) = E_{\Psi_k}\left[l_\Phi(Z|Y)\mid Z\right] + E_{\Psi_k}\left[l_\Theta(Y)\mid Z\right] + h(\Psi_k)$$

$$:= Q_m(\Phi, \hat{M}_m) + Q_s(\Theta, \hat{M}_s) + h(\Psi_k)$$

$$\hat{M}_m = E_{\Psi_k}\left[M_m|Z\right] \;;\; \hat{M}_s = E_{\Psi_k}\left[M_s|Z\right]$$

M-step:

$$\Phi_{k+1} = \underset{\Phi}{\text{argmax}}\, Q_m(\Phi, \hat{M}_m) \;, \qquad \Theta_{k+1} = \underset{\Theta}{\text{argmax}}\, Q_s(\Theta, \hat{M}_s) \tag{21}$$

where the maximization proceeds with the quasi Newton algorithm

$$\Theta^{i+1} = \Theta^i + F_s(\Theta^i)^{-1} g_s(\Theta^i) \quad;\; \Theta^0 = \Theta_k \quad;\; i=0,....$$

$$\Phi^{i+1} = \Phi^i + F_m(\Phi^i)^{-1} g_m(\Phi^i) \;;\; \Phi^0 = \Phi_k$$

$$g_s(\Theta^i) = \partial Q_s(\Theta, \hat{M}_s)/\partial\Theta\big|_{\Theta^i}$$

$$g_m(\Phi^i) = \partial Q_m(\Phi, \hat{M}_m)/\partial\Phi\big|_{\Phi^i}$$

$$F_s = [\partial\Theta\ /\partial\Theta\ ] \begin{bmatrix} \frac{NT}{2}\ \Omega^{*-1}\otimes\Omega^{*-1} & 0 & 0 & 0 & 0 \\ 0 & \Omega^{*-1}\otimes\hat{\underset{\sim}{Y}}\underset{\sim}{Y}' & \Omega^{*-1}\otimes\underset{\sim}{Y}\underset{\sim}{X}' & 0 & 0 \\ 0 & \Omega^{*-1}\otimes\underset{\sim}{X}\underset{\sim}{Y}' & \Omega^{*-1}\otimes\underset{\sim}{X}\underset{\sim}{X}' & 0 & 0 \\ 0 & 0 & 0 & \frac{N}{2}(\Sigma^{-1}\otimes\Sigma^{-1}) & 0 \\ 0 & 0 & 0 & 0 & N\,\Sigma^{-1} \end{bmatrix} [\partial\Theta\ /\partial\Theta]$$

$$F_m = [\partial\Phi^*/\partial\Phi] \begin{bmatrix} \frac{N(T+1)}{2}(R^{-1}\otimes R^{-1}) & 0 & 0 \\ 0 & R^{-1}\otimes YY' & R^{-1}\otimes YX' \\ 0 & R^{-1}\otimes XY' & R^{-1}\otimes XX' \end{bmatrix} [\partial\Phi^*/\partial\Phi]'$$

It should be noted that the modified Fisher information matrices F_s and F_m (with conditional moments) do not coincide with the exact Fisher information of the observed data Z (see Louis (1982)). At the end of the iterations the exact observed Fisher information can be calculated using numerical derivatives of the score vector. Since the convergence of EM is very slowly in the vicinity of the maximum we used combined EM/Newton-Raphson (cf. Watson/Engle (1983)) and EM/BFGS (Broyden-Fletcher-Goldfarb-Shanno, see e.g. Dennis/Schnabel (1983)) algorithms . The EM algorithm reliably locates a vicinity of the maximum but is then replaced by a quasi-Newton-scheme based on numerical derivatives or secant updates computed from the analytic score function (17). A variant uses generalized EM (GEM), in which the M-step is canceled before reaching the maximum. These methods are implemented in the modules EM , NEWRAPH and BFGS (see chapter IV).

6. Example: The sunspot activity

The temporal fluctuations of the number of sunspots (see fig. II.1) serves to illustrate the proposed methodology (for the appropriate IML/LSDE code see ch. III.3). This time series has been discussed by many autors (e.g. Yule (1927)) , mostly using discrete time models (exceptions are Bartlett (1946, 1955), Phadke and Wu (1974) and Arato (1982)). Here a data set of annual sunspot numbers from 1749-1924 was analyzed (source: Waldmeier (1961)). We consider the average number of sunspots

$$S(t) = (1/\Delta t)\int_t^{t+\Delta t} s(u)du \qquad (22)$$

in one year as the basic variable, where only discrete measurements $S(t_i)$ are available

($\Delta t=1$ year). Alternatively, we could formulate a dynamical model for s(t), the actual number at time t (this possibility is treated in ch. III.3.4). Inspection of figure II.1 leads to the conjecture, that a kind of sine function should result as solution. In accordance with the proposal of Bartlett (1946) we used a CAR(2) continuous time autoregressive process of second order (stochastic oscillator):

$$(d^2/dt^2)\left[S(t)-D\right] + \gamma(d/dt)\left[S(t)-D\right] + \omega_0^2\left[S(t)-D\right] = g\zeta_t \qquad (23)$$

where D is the mean level of the oscillation, ω_0 the angular frequency of undamped oscillation, γ a damping constant and ζ denotes white noise. Abbreviating $S(t)-D = y_1(t)$ we can write (23) in state space form (1,2)

$$d\begin{bmatrix} y_1(t) \\ y_2(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\omega_0^2 & -\gamma \end{bmatrix}\begin{bmatrix} y_1(t) \\ y_2(t) \end{bmatrix}dt + \begin{bmatrix} 0 & 0 \\ 0 & g \end{bmatrix}\begin{bmatrix} dw_1(t) \\ dw_2(t) \end{bmatrix} \qquad (24)$$

$$S(t_i) = [\,1 \quad 0\,]\begin{bmatrix} y_1(t_i) \\ y_2(t_i) \end{bmatrix} + Dx_i \qquad ;\ i=0,......T$$

where $x_i=1$ (model I). Furthermore, we can include an error of measurement ε_i(model II). This option is supported by the fact, that the first seven sunspot cycles were obtained in retrospect from historical data (see Jagger (1977), Waldmeier (1961)). Furthermore, the Wolf number is an index depending on *...visibility conditions, the apparatus used, and the method of observation, as well as on such subjective factors as observer fatigue, the way in which the sunspots are combined into groups, and the nucleus count* (Jagger (1977)).

The deterministic initial condition $y(t_0)$ can be modeled by the choice $\mu=0$ and $\Sigma=I_p\cdot\varepsilon$, $\varepsilon\to\infty$ (uniform distribution). In the spirit of a systems approach, we do not subtract the mean level first, but estimate it simultanously with the other parameters. Phadke and Wu (1974) proposed a CARMA(2,1) process

$$(d^2/dt^2)\left[S(t)-D\right] + \gamma(d/dt)\left[S(t)-D\right] + \omega_0^2\left[S(t)-D\right] = g\zeta_t + g\,d\zeta_t/dt \qquad (25)$$

which can also be written in state space form (model III).

Results:

Table II.2 shows the ML-estimates of the three models with estimated standard errors in comparision to the results of other authors. Since the convergence of the EM algorithm is very slowly in the vicinity of the maximum we used a combined EM/BFGS-algorithm (cf. sect. 5). The algorithms converged in 30/8 iterations, respectively (CPU time 110.92 sec, 115.42 sec and 118.69 sec), to the ML-estimate ψ_f, where $||\psi_f-\psi_{f-1}||_\infty<10^{-4}$.

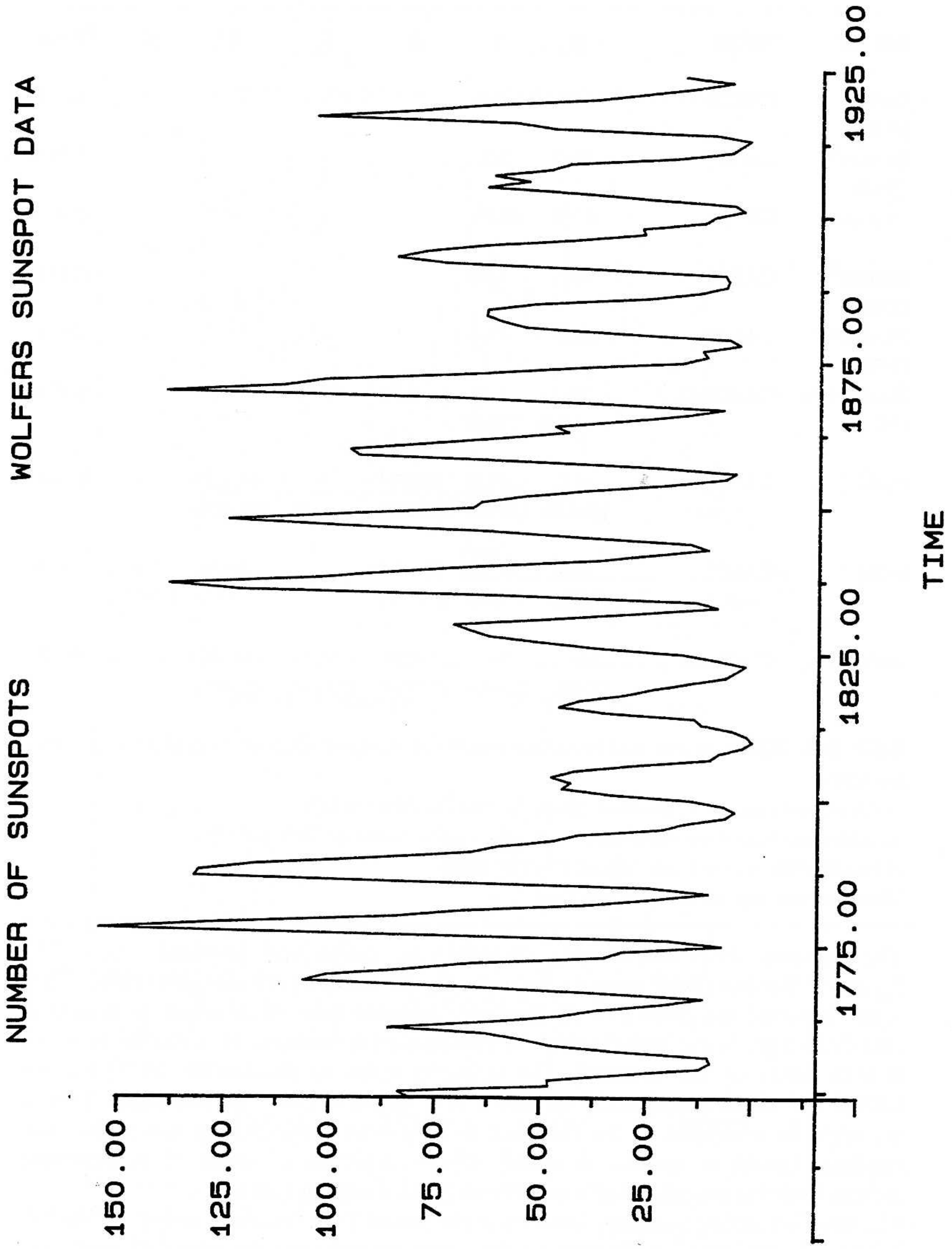


Fig. II.1 Annual number of sunspots in the years 1749 - 1924.

Author	model	ω_0	γ	g	g_1	D	R	T/years
Bartlett (1946)	CAR(2)	.6158	.3160		-		-	8.1742
Bartlett [b] (1946)	CAR(2)	.5811	.1593		-		-	8.2878
corrected	CAR(2)	.4792	.63218		-	44.75	-	10.2022
Bartlett [b] (1955)	CAR(2)	.3631	.3186		-		-	10.8118
Phadke/Wu (1974)	CAR(2)	.5028	.7853		-		-	10.6414
Phadke/Wu (1974)	CARMA(2,1)	.3593 (.0376)	.3271 (.0626)	15.70	9.9428	44.75	-	10.8956 [c]
model I	CAR(2)	.5030 (.0685)	.7931 (.1442)	30.6714 (2.5000)	-	44.1254 (4.7822)	-	10.6856
model II	CAR(2) +error	.3996 (.0463)	.3772 (.1026)	18.7239 (2.4147)	-	44.5186 (3.5720)	26.4461 (7.9072)	10.4138
model III	CARMA(2,1)	.3596 (.0459)	.3295 (.0960)	15.7189 (2.7214)	9.9383 (1.2542)	44.5781 (3.3807)	-	10.8971

Table II.2. *ML estimates and standard errors for sunspot data in comparision to other authors.*

a) Corrected figures of Bartlett given by Phadke/Wu (1974)

b) estimates based on smoothed data (graduated series of Yule (1927)) .

c) Phadke/Wu erroneously report 10.8329 years

blank entries are values not reported

The estimates differ considerably depending on author and proposed model. The figures of Bartlett (1946) seem to be incorrect, as noted by Phadke/Wu (1974). They report corrected estimates (row 3). Bartlett´s estimate (row 4) is based on smoothed data (Yule´s graduated series) using the autocorrelation function. It is similar to model II (with errors of measurement). The estimates given by Phadke/Wu (1974) for the CAR(2) and CARMA(2,1)-model are very similar to model I and II. The slight differences might be attributed to the fact that in their work a stationarity assumption was employed (usage of spectral densities). The introduction of errors of measurement reduces both the magnitude of process error g and damping constant γ.

The most interesting quantity, however, is the period $T=2\pi/\omega$, where $\omega=(\omega_0^2 - \gamma^2/4)^{1/2}$. In spite of considerable differences in the parameter estimates the period of oscillation always lies between 10 and 11 years. Fig. II.2 shows the smoothed estimates of the unob-

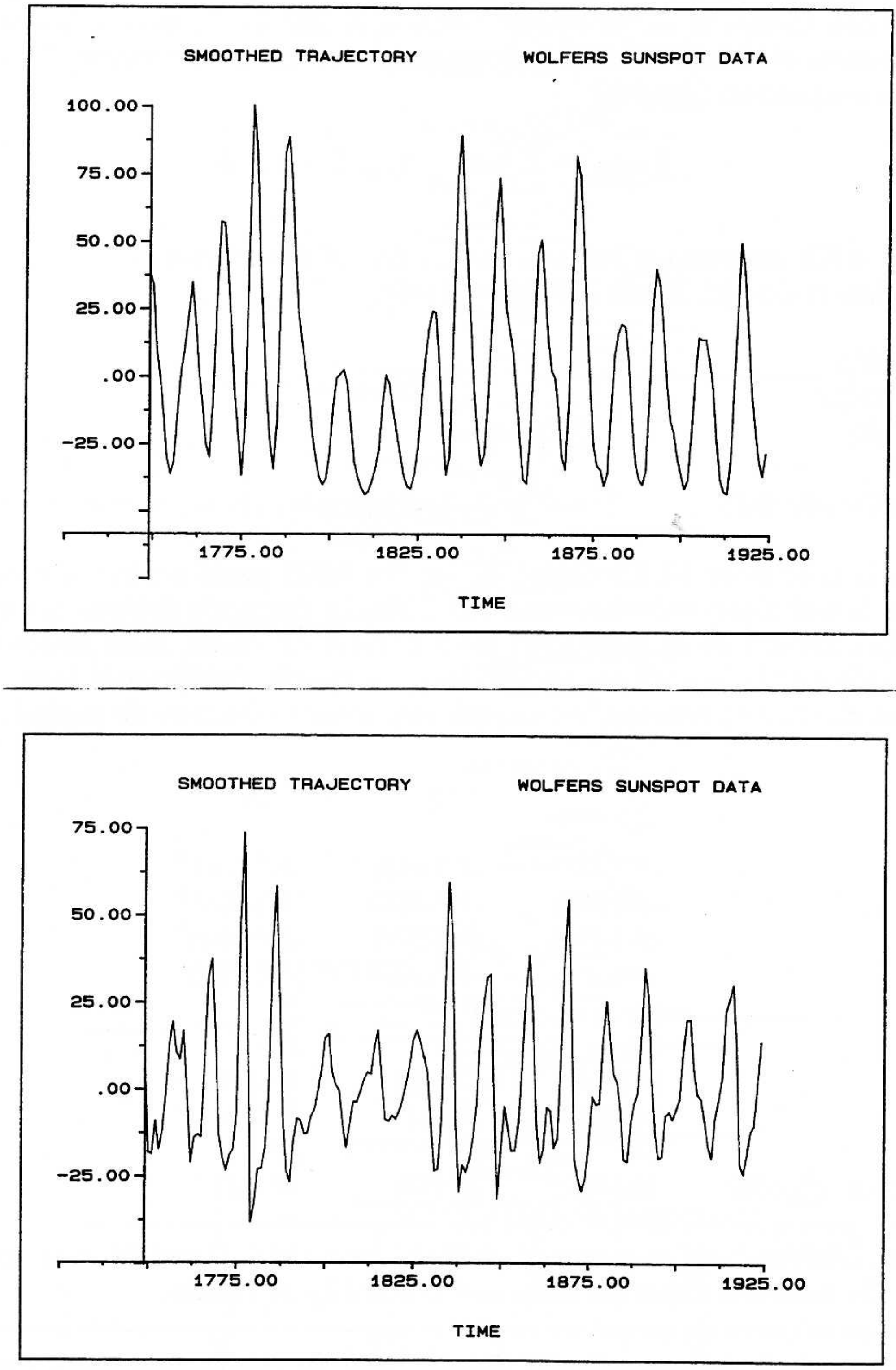


Fig. II.2. Smoothed estimates of the number of sunspots and the change in numbers (derivative)

served components $\hat{y}_1(t)=E[y_1(t)|Z]$ and $\hat{y}_2(t)=E[y_2(t)|Z]$ using the final estimates of model II. Using the concept of compensated likelihood (Azencott and Dacunha-Castelle (1986)) we now attempt to decide between model I, II and III. In this method model order u (number of parameters) and their values are simultanously estimated by maximizing the compensated likelihood

$$[\widehat{u,\psi_u}] = \underset{u,\ \psi_u \in \mathbb{R}^u}{\operatorname{argmax}}\ l(\psi_u,Z) - \delta(u,\upsilon)$$

where δ is called **compensator** and υ is the number of observations.
We used three choices (cf. Koehler/Murphree (1988)):

Akaike (AIC)	u
Schwarz (SIC)/ Rissanen (R)	$(u/2)\log\upsilon$
Azencott/ Dacunha-Castelle (AD)	$(u/2)(u+1)(1+\varepsilon)\log\log\upsilon$; (ε small, here 10^{-3})

Most criteria favor model III (CARMA(2,1)), but the AR(2)-model subject to measurement error is only slighly worse (see table II.3). However, due to the dubious quality of the data (see above) I would nevertheless favor model II. Of course, other models (e.g. higher ARMA-models or nonlinear specifications, see Priestley (1988)) would have to be tested (see ch. III. 3.4). However, the example only serves to illustrate the method.

	I	II	III
Lik	-577.8535	-571.0536	-571.0361*
AIC	-581.8535	-576.0536	-576.0361*
SIC/R	-588.1945	-583.9798	-583.9623*
AD	-594.2996*	-595.7227	-595.7053
u	4	5	5
υ	176	176	176
P(30)	50.0736	32.2761	32.6046
P`(30)	54.0523	35.8822	36.1774
df	27	26	27
critical value $\chi^2_{df}(.95)$	40.1133	38.8852	40.1133

Table II.3. *Likelihood and compensated likelihood for model I, II and III using several criteria. The maximum within one criterion is denoted by an asterisk.*

Finally, the residuals (innovations) were inspected and the portmanteau statistic given by Hosking (1980) was computed for s=30 (see Table II.3). On an $\alpha=.05$ level, the residuals of model I cannot be considered as white noise disturbances. An optical inspec-

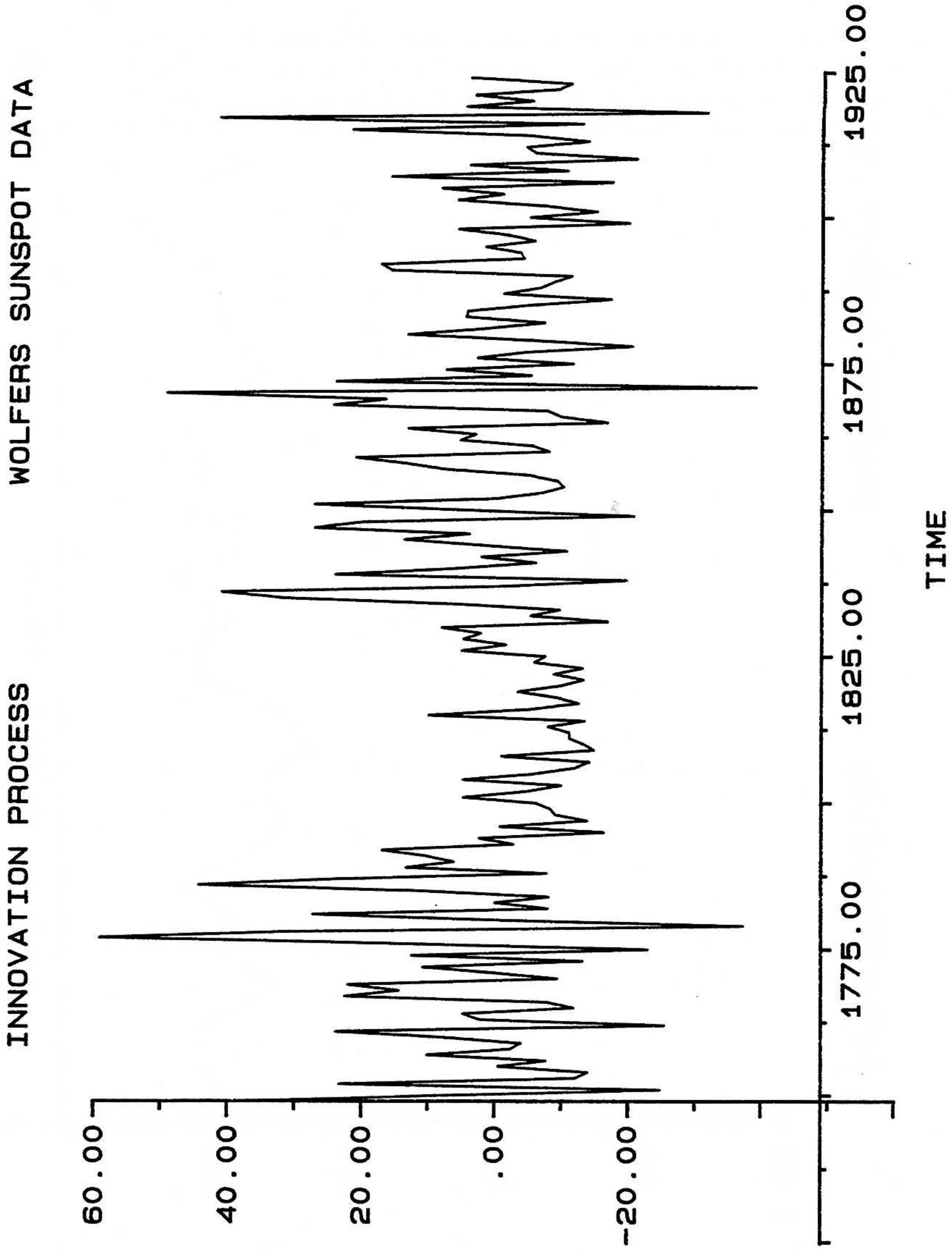


Fig. II. 3 Innovation process (residuals) for model II.

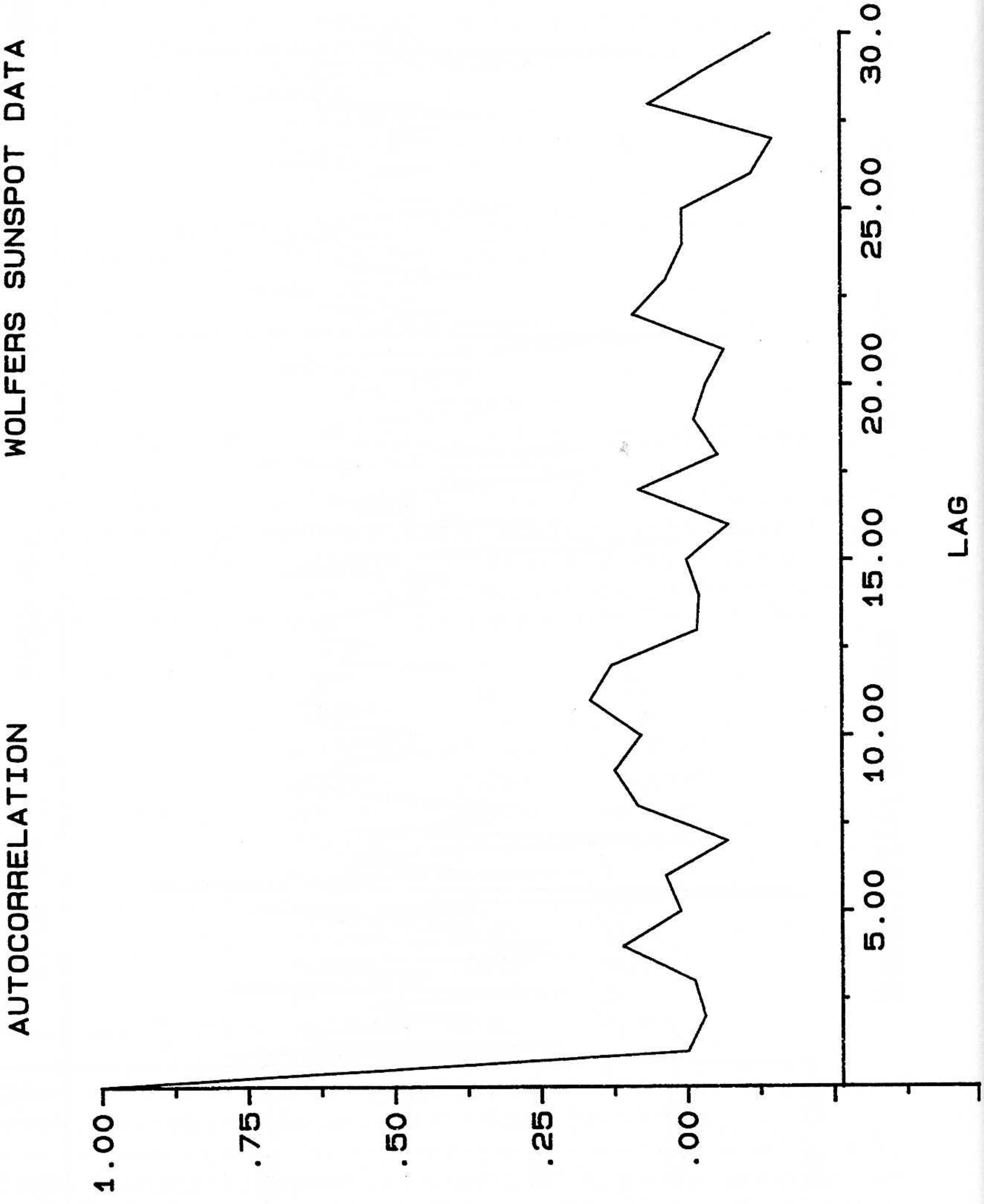


Fig. II.4 Autocorrelation of the residuals for model II.

tion of the innovations process (figure II.3) reveals, as expected, that the models do not cover the asymmetry between maxima and minima and the fluctuation in amplitudes. Finally, a plot of the autocorrelation function of the residuals (fig. II.4) corroborates the conclusion that the residuals are only correlated for a short time (white noise).

III. CASE STUDIES

0. General Remarks

In this chapter the usage of LSDE is explained with examples. A more systematic treatment and a list of supplied modules (procedures, subprograms) is given in chapter IV. As explained in chapter II, the user has to specify the state space model

$$dy_n(t) = Ay_n(t)dt + Bx_n(t)dt + Gdw_n(t) \; ; \qquad t_0 \le t \le t_T$$

$$z_{ni} = Hy_{ni} + Dx_{ni} + \varepsilon_{ni} \; ; \qquad i = 0,.......T, \; n = 1,......N$$

with the initial condition $y_n(0) \sim N(\mu,\Sigma)$ i.i.d. and a white noise error of measurement $\varepsilon_{ni} \sim N(0,R)$ i.i.d.

The structural matrices A(p×p), B(p×q), Ω = GG` (G: p×r, Ω: p×p) , μ(p×1) and Σ(p×p) must be given as functions of a parameter vector Θ and H (k×p), D (k×q) and R (k×k) must be given as functions of a parameter vector Φ. This is achieved through the user supplied modules **PAR** and **PARM** (specification of system and measurement model) and through the user supplied modules **PARD** and **PARMD** (D stands for "derivative"). After definition of these modules (model specification), a main program must be supplied which runs the several modules implementing simulation and estimation tasks. This will be explained in detail.

Knowledge of the syntax of SAS/IML (see IML user`s guide, 1989) will be helpful in understanding the examples (it is similar to GAUSS and both languages heavily borrowed their concepts from APL)

1. Case Study 1:

Continuous Time Autoregressive Process of Second Order (CAR(2)) - Ornstein-Uhlenbeck-Process

Model Specification

In the following we illustrate the maximum likelihood estimation of a sampled stochastic process with a simulated data set. The harmonic oscillator (pendulum) under the influence of random disturbances is a basic model of vibrating systems (periodic behavior). For example, the temporal variations of the annual sunspot numbers were explained by Yule (1927) and Bartlett (1946) in terms of a pendulum under the influence of a random force (AR(2)-process). Another example are the well known experiments of Kappler (1931) on the Brownian motion of a torsion pendulum. Its equation of motion

(Newton`s law: *force=mass*acceleration*) is given as (the dot $\dot{}$ is the time derivative d/dt)

$$\ddot{y}(t) + \gamma\dot{y}(t) + \omega_0^2\, y(t) = g\zeta(t) + bx(t) = F(t)$$

where γ denotes friction and ω_0^2 is the spring constant (ω_0 is the angular frequency of the undamped motion). Here $\gamma\dot{y}$ is a force due to friction (proportional to the velocity $\dot{y}$) and $\omega_0^2\, y$ is the restoring force. Note, however, that it is not necessary to interpret these terms in the language of physics. We assume that the stochastic force F(t) is composed of a white noise $\zeta(t)$ and a deterministic component x(t), which we set constant (=1). Figure III.1 shows a simulated trajectory in the interval [0,10] with parameters $\{-\omega_0,-\gamma,b,g,\mu,\Sigma\}=\{-16,-4,1,2,\begin{bmatrix}0\\0\end{bmatrix},\begin{bmatrix}1&0\\0&1\end{bmatrix}\}$ and the random initial condition $[\,y(0),\,\dot{y}(0)]` \sim N(\mu,\Sigma)$. As usual we can write the second order differential equation as a first order system

$$\dot{\begin{bmatrix}y_1\\y_2\end{bmatrix}} = \begin{bmatrix}0&1\\-\omega_0^2&-\gamma\end{bmatrix}\begin{bmatrix}y_1\\y_2\end{bmatrix} + \begin{bmatrix}0\\b\end{bmatrix}x \quad + \begin{bmatrix}0&0\\0&g\end{bmatrix}\begin{bmatrix}\zeta_1\\\zeta_2\end{bmatrix}$$

or $\dot{y} = Ay + Bx + G\zeta\,;\ y(0) \sim N(\mu,\Sigma)$

The Itô-form of this equation (see Arnold, 1974) reads

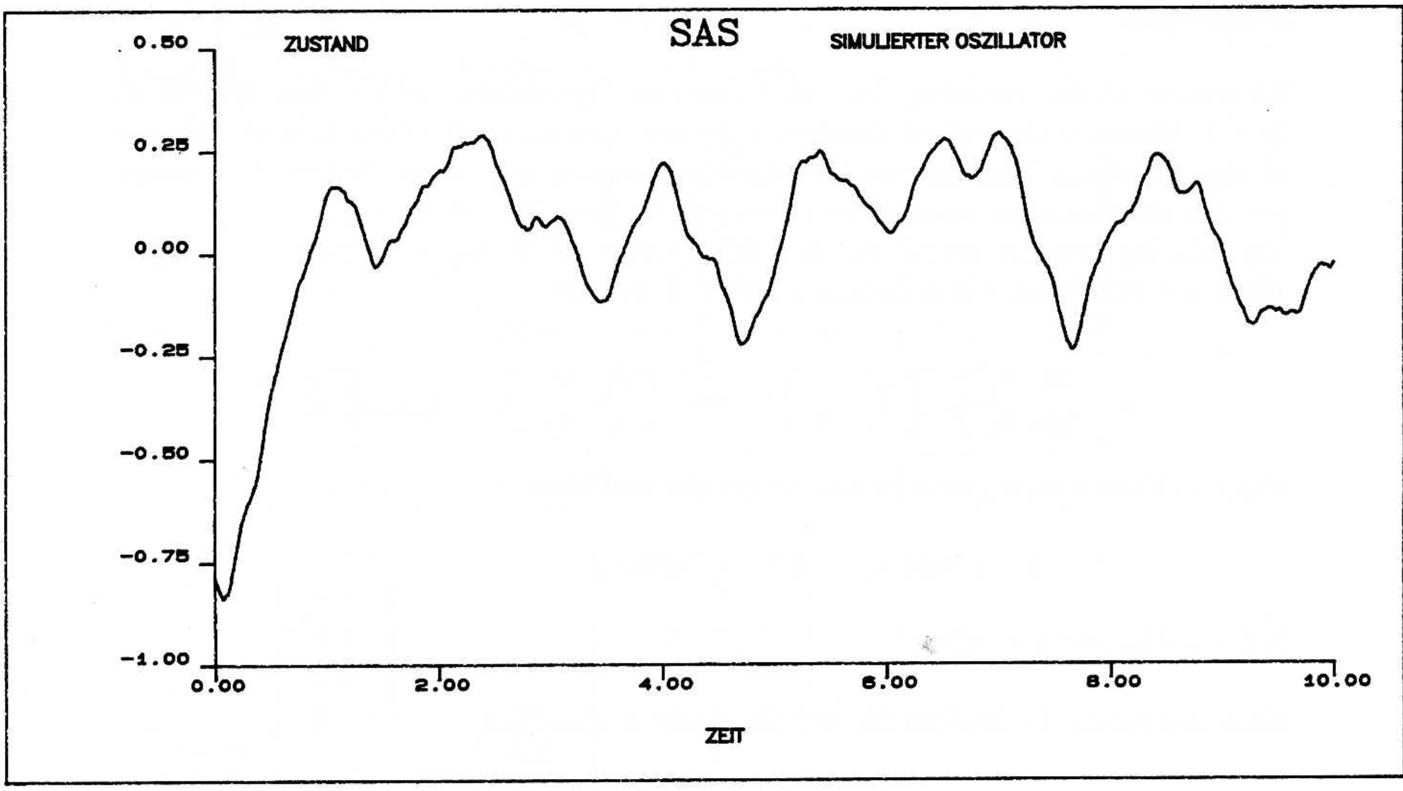
$$dy = Aydt + Bydt + Gdw.$$

The description of Brownian motion with a differentiable trajectory (i.e. with a velocity) was already given by Uhlenbeck and Ornstein (1930). The angular frequency $\omega_0=2\pi\nu_0=4$ and period $T_0= 2\pi/\omega_0=1.57$ of the undamped oscillator is changed by the friction γ yielding

$$\lambda_{1,2} = -\gamma/2 \pm \sqrt{\gamma^2/4 - \omega_0^2} \quad = -2 \pm \sqrt{-12} \quad = -2 \pm i\sqrt{12}$$

(damping and slower oscillation frequency $\omega= \sqrt{\omega_0^2 - \gamma^2/4}$)

Since the eigenvalues of A have a negative real part the process is asymptotically stationary, i.e. the expectation value $\mu(t)=E[y(t)]$ and the covariance matrix $\Sigma(t)=Var(y(t))$ approach the constant values $\mu_s=-A^{-1}Bx$ and $row(\Sigma_s)= -(A\otimes I+I\otimes A)^{-1}\ row(\Omega)$ (see Arnold 1974, ch. 8). In the present example the numerical values are $\mu_s=\begin{bmatrix}1/16\\0\end{bmatrix}$ and $\Sigma_s=\begin{bmatrix}.177&0\\0&.7071\end{bmatrix}$. These values can be inspected qualitatively in fig. III.1.



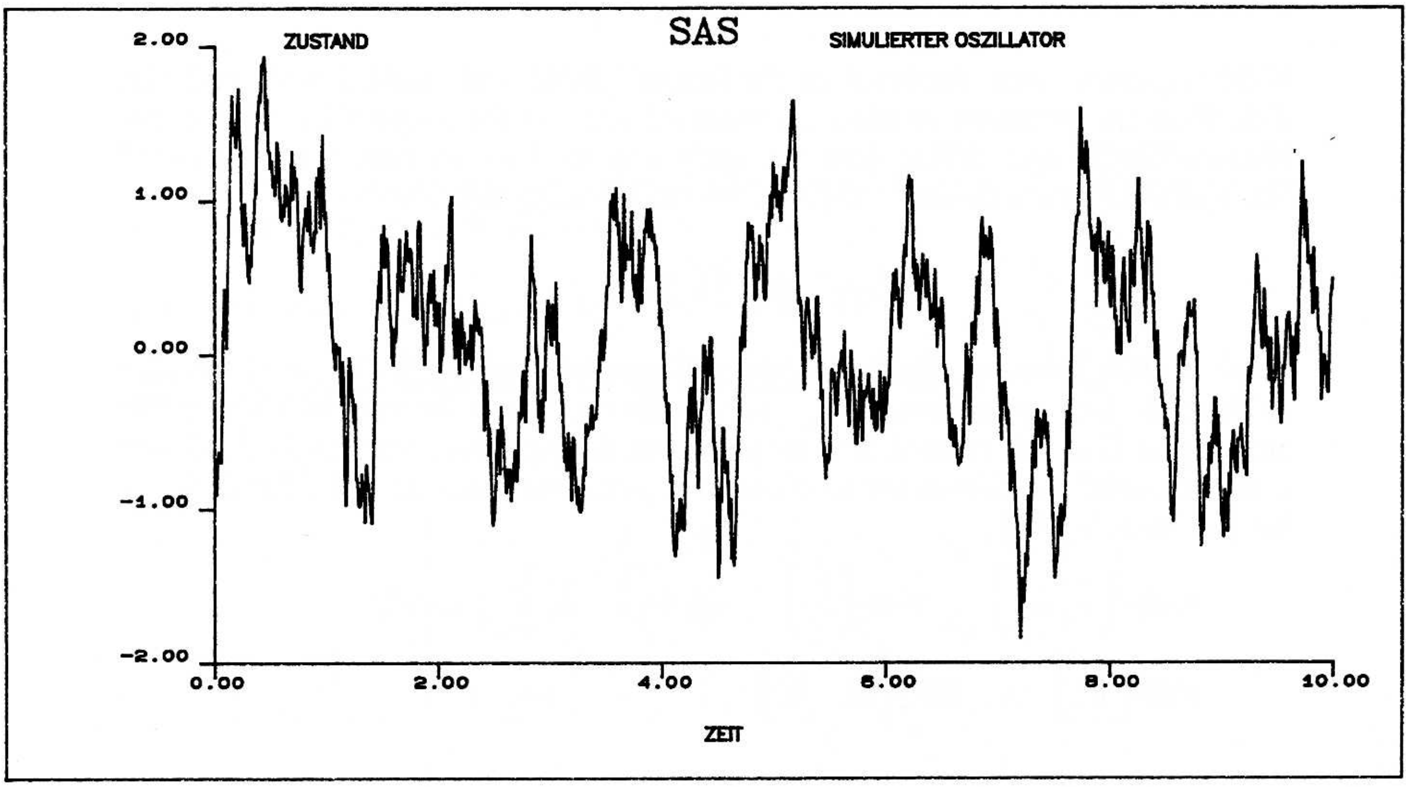


Fig. III.1 Simulated trajectory of a continuous time AR(2) process (Ornstein-Uhlenbeck-process)

Identification

We assume in the following that the trajectories are sampled with a time interval of $\Delta t = 2$. Moreover the control variables $x_n(t)$ are constant (= 1), which is a special case of step functions. Therefore an identification problem may occur, but in the present problem the parameter matrices are restricted. We have $A_{11} = 0$, $A_{12} = 1$.
The aliasing matrices are of the form (cf. Phillips 1973) $A_N = A+(2\pi i/\Delta t)P\begin{bmatrix} n & \\ & -n \end{bmatrix}P^{-1}$ where $A = P\Lambda P^{-1}$ and n is a integral number. If we write

$$P = \begin{bmatrix} \Phi_1 & \Phi_1^* \\ \Phi_2 & \Phi_2^* \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ \lambda_1 & \lambda_2 \end{bmatrix} \quad ; \; P^{-1} = \begin{bmatrix} \psi_1 & \psi_2 \\ \psi_1^* & \psi_2^* \end{bmatrix} = -(1/2i\omega)\begin{bmatrix} \lambda_2 & -1 \\ -\lambda_1 & 1 \end{bmatrix},$$

where $*$ denotes conjugate complex, we get the condition

$$0 = n\,\mathrm{Im}(\Phi_1\psi_1) \text{ und } 0 = n\,\mathrm{Im}(\Phi_1\psi_2)$$

or $0 = \gamma /4\omega$ and $0 = -n(2\omega)^{-1}$.

Since this cannot be fulfilled for $n \neq 0$ the model is identified.

Maximum likelihood estimation

N=50 trajectories were simulated in the interval [0,10] and sampled with $\Delta t=2$, i.e. T=5. Since the parameter matrices are restricted and p>1 the simple OLS-solution (see Hamerle/Nagl/Singer, 1991a) does not apply and we have to solve the complicated ML-equations by a numerical algorithm. We use a scoring algorithm

$$\Theta_{k+1} = \Theta_k + F_k^{-1} s_k$$

where $F_k=F(\Theta_k)$ denotes the Fisher-information-matrix and s_k is the score function $\partial l/\partial\Theta$ at the k-th approximation Θ_k . For details we refer to the work of Singer (1990) and chapter II of this manual. It is assumed that the parameter matrices A, B, G and μ and Σ (initial conditions) are functions of a parameter vector $\Theta \in \mathbb{R}^u$. Explicitly we set (see module **PAR**)

$$A(\Theta)=\begin{bmatrix} 0 & 1 \\ \Theta_1 & \Theta_2 \end{bmatrix} \qquad B(\Theta)=\begin{bmatrix} 0 \\ \Theta_3 \end{bmatrix} \qquad G(\Theta)=\begin{bmatrix} 0 & 0 \\ 0 & \Theta_4 \end{bmatrix} \; ; \; \Omega=GG`$$

$$\mu(\Theta)=\begin{bmatrix} \Theta_5 \\ \Theta_6 \end{bmatrix} \quad ; \quad \Sigma(\Theta)=\begin{bmatrix} \Theta_7 & \Theta_8 \\ \Theta_8 & \Theta_9 \end{bmatrix}$$

Note that Ω is parametrized via G, which guarantees the positive definiteness.
The McDonald-Swaminathan-derivatives $\partial A^*/\partial\Theta$, $\partial B^*/\partial\Theta$, $\partial\Omega^*/\partial\Theta$ appearing in the score function and the Fisher information can be calculated analytically with the help

of the chain rule $\partial A^*/\partial\Theta=(\partial A/\partial\Theta)(\partial A^*/\partial A)$ etc. (see Singer 1990). The user only has to specify (in the module **PARD**) the simple expressions

$$\partial A/\partial\Theta=\begin{bmatrix} 0&0&1&0\\ 0&0&0&1\\ 0&0&0&0\\ 0&0&0&0\\ 0&0&0&0\\ 0&0&0&0\\ 0&0&0&0\\ 0&0&0&0\\ 0&0&0&0 \end{bmatrix} \quad \partial B/\partial\Theta=\begin{bmatrix} 0&0\\ 0&0\\ 0&1\\ 0&0\\ 0&0\\ 0&0\\ 0&0\\ 0&0\\ 0&0 \end{bmatrix} \quad \partial\Omega/\partial\Theta=\begin{bmatrix} 0&0&0&0\\ 0&0&0&0\\ 0&0&0&0\\ 0&0&0&2\Theta_4\\ 0&0&0&0\\ 0&0&0&0\\ 0&0&0&0\\ 0&0&0&0\\ 0&0&0&0 \end{bmatrix}$$

$$\partial\mu/\partial\Theta=\begin{bmatrix} 0&0\\ 0&0\\ 0&0\\ 0&0\\ 1&0\\ 0&1\\ 0&0\\ 0&0\\ 0&0 \end{bmatrix} \quad \partial\Sigma/\partial\Theta=\begin{bmatrix} 0&0&0&0\\ 0&0&0&0\\ 0&0&0&0\\ 0&0&0&0\\ 0&0&0&0\\ 0&0&0&0\\ 1&0&0&0\\ 0&1&1&0\\ 0&0&0&1 \end{bmatrix}$$

The expressions above are McDonald-Swaminathan (1973) matrix derivatives which are constructed according to the following rule:

If Θ is the vector $\Theta=[\Theta_1, \Theta_2, \Theta_3, \Theta_4]$ and X is the matrix $\begin{bmatrix} x_{11} & x_{12}\\ x_{21} & x_{22} \end{bmatrix}$

then the **McDonald-Swaminathan matrix derivative** is defined as

$$\partial X/\partial\Theta = \partial\mathrm{row}`(X) / \partial\mathrm{row}(\Theta)$$

$$= \begin{bmatrix} \partial x_{11}/\partial\Theta_1 & \partial x_{12}/\partial\Theta_1 & \partial x_{21}/\partial\Theta_1 & \partial x_{22}/\Theta_1\\ \partial x_{11}/\partial\Theta_2 & \partial x_{12}/\partial\Theta_2 & \partial x_{21}/\partial\Theta_2 & \partial x_{22}/\Theta_2\\ \partial x_{11}/\partial\Theta_3 & \partial x_{12}/\partial\Theta_3 & \partial x_{21}/\partial\Theta_3 & \partial x_{22}/\Theta_3\\ \partial x_{11}/\partial\Theta_4 & \partial x_{12}/\partial\Theta_4 & \partial x_{21}/\partial\Theta_4 & \partial x_{22}/\Theta_4 \end{bmatrix}$$

The scoring algorithm with exact analytic derivatives was implemented in the software package LSDE (module NEWTON) and run on a BASF 7/73 computer under VM/CMS.

Results

In order to test the algorithm we performed a simulation study using M = 100 samples (N = 50, T = 5).
Using a starting value of Θ_0 = {-16.3, -4.3, .7, 1.7, -.3, -.3, .7, -.3, .7} the iterations converge after typically 10 iterations to the ML- estimate $\hat{\Theta} = \Theta_K$ (with a precision $\varepsilon = 10^{-4}$, i.e. $\max | \Theta^i_K - \Theta^i_{K-1} | < \varepsilon$).
The mean and standard deviation of the M=100 estimates is given in table III.1.

	$-\omega_0$	$-\gamma$	b	g	μ_1	μ_2	Σ_{11}	Σ_{12}	Σ_{22}
Θ	-16	-4	1	2	0	0	1	0	1
exact ML:									
$\bar{\Theta}$	-16.0799	-4.1346	.9930	2.0197	-.0051	-.0002	.9613	-.0045	.9958
SD	1.8557	1.1279	.1996	.2680	.1565	.1572	.1889	.1453	.1819
Linearized ML:									
$\bar{\Theta}$	-.0286	-.4990	.0030	.4988	-.0051	-.0002	.9613	-.0045	.9958
SD	.0453	.0295	.0232	.0197	.1565	.1572	.1889	.1453	.1819

Table III.1 *Mean $\bar{\Theta}$ and standard deviation SD of M=100 maximum likelihood estimates of parameter vector Θ (exact and linearized ML)*

For simulation and estimation of the M = 100 samples a cpu time of 673.88 seconds was required (additionally the linearized ML estimate (discretized continuous sampling, cf. Le Breton 1976) was computed). The cpu time of approximately 6 seconds per sample seems efficient and the estimates are close to the true values. In contrast discretized continuous sampling leads to strongly biased estimates. This is corroborated in simulation studies concerning the Phillips model (see Phillips 1973, Singer 1990, case study 2). These results are not unexpected since asympotic theory shows that good results can be obtained only if AΔt is so small that $A^* = \exp(A\Delta t)$ can be replaced by $I + A\Delta t$ (only linear terms are kept in the Taylor-expansion of exp(AΔt)). In the present case we have $A^* = \begin{bmatrix} .0210 & .0032 \\ -.0509 & .0083 \end{bmatrix}$ whereas $I + A\Delta t = \begin{bmatrix} 1 & 2 \\ -32 & -7 \end{bmatrix}$. However, in applications the true parameter is unknown. Therefore, linearization is an ad hoc approximation whose quality cannot be judged a priori.

IML-code

We now give IML code to specify and estimate the model discussed above. The main program is called **OSZILL** which contains the definition of the user supplied modules **PAR** and **PARD**. Comments are given in italics according to the syntax _*Comment;_ and modules supplied by LSDE or by the user are denoted in **bold** letters.

```
PROC IML;                                                            *begin of iml program;
RESET STORAGE=COMP.LSDE1;                                    *Loading Library LSDE from disk;
LOAD;
RUN INITLSDE ;                                          * running module INITLSDE initializes;
                                                                       * module Library LSDE;
*****************************************************************************************;
START OSZILL;                                                *begin of user supplied module;
*****************************************************************************************;

*****************************************************************************************;
START PAR ( OMEGA, A, B, SIGMA, MUE, TH);                              *model specification;
*****************************************************************************************;

A=( 0    ||  1   )//
  (TH[1]||TH[2]);                                                    *drift matrix A(Θ);

B=   0  //
   TH[3];                                                            *input matrix B(Θ);

G=(.0001||   0  )//
  (  0  ||TH[4]);                                           *diffusion coefficient G(Θ);

OMEGA=G*G`;                                                   *diffusion matrix Ω=GG`;

MUE= TH[5]//
     TH[6];                                                  *initial condition N(μ,Σ);

SIGMA=(TH[ 7]||TH[ 8])//
      (TH[ 8]||TH[ 9]);

FINISH;                                                                  *end of module PAR;
*****************************************************************************************;
START PARD (DODTH, DADTH, DBDTH, DSIGDTH, DMUEDTH, TH);                *model specification;
*****************************************************************************************;
* MC-DONALD-SWAMINATHAN-derivatives of A, B, Ω, μ and Σ;

DADTH= { 0 0 1 0,
         0 0 0 1,
         0 0 0 0,
         0 0 0 0,
         0 0 0 0,
        0 0 0 0,
        0 0 0 0,
        0 0 0 0,
        0 0 0 0 };                                    *derivative ∂A/∂Θ;

DBDTH= {  0 0,
          0 0,
          0 1,
          0 0,
          0 0,
          0 0,
          0 0,
          0 0,
          0 0 };                                      *derivative ∂B/∂Θ;


DODTH=J(3,4,0)//(0||0||0||2*TH[4])//J(5,4,0);         *derivative ∂Ω/∂Θ;

DMUEDTH={  0 0,
           0 0,
           0 0,
           0 0,
           1 0,
           0 1,
           0 0,
           0 0,
           0 0 };                                     *derivative ∂μ/∂Θ;

DSIGDTH= { 0 0 0 0,
           0 0 0 0,
           0 0 0 0,
           0 0 0 0,
           0 0 0 0,
           0 0 0 0,
           1 0 0 0,
           0 1 1 0,
           0 0 0 1 };                                 *derivative ∂Σ/∂Θ;

FINISH;                                               *end of module PARD;
****************************************************************************;
Q=1;                                        *number of exogenous variables;
P=2;                                        *number of endogenous variables;
T=5;  N=50;                                 *t_0 ,...t_T  and number of persons;

T1=T+1;                                     *number of panel waves;
DELTA=2.;                                   *sampling interval of panel;

A=  {  0   1,
     -16  -4 };                                       *true drift matrix;
B={ 0,
    1 };                                              *true input matrix;
G={ 0 0,
    0 2 };
OMEGA={ 0 0,
        0 4 };                                        *true diffusion matrix;
```

```
MUE={ 0,
      0 };                                                    *true parameters of ;
SIGMA={1 0,
       0 1};                                        *initial condition y :N(μ, Σ);

SEED=12;                                                            *random seed;
X=SHAPE(1,Q,N*T1);                                    *exogenous variables x(t)=1;

Y0=J(1,N,1) § MUE +ROOT(SIGMA)*RANNOR(SHAPE(SEED,P,N));  *random initial condition;
                                                   *using random number generator;
                                                     *RANNOR (Gauss-distribution);


RUN EXAKT(Y,A,B,G,Y0,X,P,Q,N,T,DELTA,SEED,1);                *simulation of data Y;

RUN MOMENT1(M0,M00,M11,M12,M13,M22,M23,M33,Y,X,P,Q,N,T,1);   *calculation of moment-;
                                                     *matrices for use in NEWTON;


THETA0={ -16,-4,  1,   2,   0,0,  1,0,1 } - .3;    *initial condition for iteration;


KMAX=50;                                                *maximal number of iterations;
EPS=1E-4;                                                *precision fo convergence;

RUN     NEWTON(THETA1,COV,THETA0,KMAX,EPS,
                M0,M00,M11,M12,M13,M22,M23,M33,DELTA,N,T,0,0);
                                                         *ML-estimation of Θ using;
                                                     *linearized model (nonlin=0);

RUN     NEWTON(THETA,COV,THETA0,KMAX,EPS,
                M0,M00,M11,M12,M13,M22,M23,M33,DELTA,N,T,0,1);  *ML-estimation of Θ using;
                                                          *exact model (nonlin=1);

FINISH;                                                     *end of module oszill;
RUN OSZILL;                                           *execution of module oszill;
QUIT;                                                        *end of IML program;
```

Comment:

The IML code given above provides a complete example how panel data can be simulated knowing the true parameter values A, B, G, Σ and μ. After generation of the data Y0 (random initial condition) and Y (time paths of N units at T+1 time points) it is assumed that the parameters are unknown and have to be reconstructed (estimated) from the given data. This is the situation which normally occurs in practical applications. Simulation studies of these kind serve the purpose of testing the performance of an estimation algorithm from known data (e.g. the distribution of the data and the model specification is known). Furthermore, if you have estimated parameters from an empirical data set you can get an impression how the time paths belonging to this system look like in other samples (e.g. if you repeat a study).

Two estimation prcedures have been used. The first one (NONLIN=0) utilizes discretized continuous sampling (see ch. II). The results (table III.1) clearly illustrate that in general only the exact ML-method (NONLIN=1) does lead to correct estimates. By the same token the linearization used in the trapezium method (cf. Möbus/Nagl, 1983, Singer, 1988, Singer, 1990) leads to a severe bias as well. Nevertheless the linearized model can be applied if the sampling interval is small (in relation to the parameter values) or in order to produce initial estimates for the exact iteration.

In the simple example presented above no errors of measurement are assumed. If you have indication that such errors play an important role or if you wish to use latent variables (factor scores) as dynamical variables you have to use the continuous/discrete state space model and the modules **STATESPA, EM, NEWRAPH** and **BFGS**, respectively (see case study 3).

The output produded by the above program looks like this:

```
**************************************************************
*                                                            *
*                      L  S  D  E                            *
*                      L  S  D  E                            *
*                      L  S  D  E                            *
*                      L  S  D  E                            *
*                      L  S  D  E                            *
*                      L  S  D  E  Version I                 *
*                                                            *
*                                                            *
*                      Copyright (c) 1990,1991,1992 by       *
*                                                            *
*                      Dr. HERMANN SINGER                    *
*                      Allmendweg 6                          *
*                      D-7758 Meersburg                      *
*                                                            *
*                                                            *
*                      A Program Package for the             *
*                      Simulation                            *
*                      Graphical Display                     *
*                      Optimal Filtering and                 *
*                      Maximum Likelihood Estimation of      *
*                                                            *
*                      Linear                                *
*                      Stochastic                            *
*                      Differential                          *
*                      Equations                             *
*                                                            *
**************************************************************


************************LSDE1*******************************************
***********************MODUL: EXAKT*************************************
************************************************************************

EXAKTES DISKRETES MODELL
------------------------------------------------------------------------


                P          Q          N          T         DT       SEED      PRINT

ROW1            2          1         50          5          2         12          1


A            COL1       COL2

ROW1            0          1
ROW2          -16         -4
```

```
B          COL1

ROW1          0
ROW2          1


G          COL1       COL2

ROW1          0          0
ROW2          0          2


ASTERN        COL1       COL2

ROW1   0. 0209934 0. 0031788
ROW2     -0. 05086 0. 0082783


BSTERN        COL1

ROW1   0. 0611879
ROW2   0. 0031788


V          COL1       COL2

ROW1 0. 0312312 0. 0000202
ROW2 0. 0000202 0. 4998849


Y0         COL1       COL2       COL3       COL4       COL5       COL6       COL7       COL8 ....
:         COL13      COL14      COL15      COL16      COL17      COL18      COL19      COL20
:         COL25      COL26      COL27      COL28      COL29      COL30      COL31      COL32
:         COL37      COL38      COL39      COL40      COL41      COL42      COL43      COL44
:         COL49      COL50

ROW1 -1. 100986 0. 5102031 -1. 613007  -0. 84036 0. 3063094 -0. 137757 0. 5837736 -1. 101155 .....
:    0. 5287544 0. 8516944 -0. 133082  0. 681287 1. 1035271 -2. 074337 -0. 416986 1. 9221902
:    0. 6359094 2. 0713417 -0. 553496 -1. 008133   0. 29938 -1. 338365 1. 1793219 -1. 518272
:    1. 5782633 -1. 505502 0. 2561553  0. 042185 0. 2186959 -0. 149011 -1. 128817 0. 4563077
:    -2. 305372 1. 1187851

ROW2 0. 4249926 0. 3064045 0. 0815213 0. 6593596 -0. 676497 0. 4638712 -0. 017465 -0. 867974 .....
:    -1. 418743 -1. 958731 0. 0326734 -1. 172076 0. 1572937 -0. 166277  0. 565414 1. 1940593
:    0. 4360652 -1. 047591 -0. 017632 0. 4095882 1. 0449989 0. 7487248 0. 7157709 -0. 493609
:    -0. 126856 -1. 886986 0. 0881589 -2. 055992 2. 4390881 0. 7126954 1. 6612392 2. 2704596
:    -2. 426078 -0. 588812


Y          COL1       COL2       COL3       COL4       COL5       COL6       COL7       COL8 .....
:         COL13      COL14      COL15      COL16      COL17      COL18      COL19      COL20
:         COL25      COL26      COL27      COL28      COL29      COL30      COL31      COL32
:         COL37      COL38      COL39      COL40      COL41      COL42      COL43      COL44
:         COL49      COL50      COL51      COL52      COL53      COL54      COL55      COL56
:         COL61      COL62      COL63      COL64      COL65      COL66      COL67      COL68
:         COL73      COL74      COL75      COL76      COL77      COL78      COL79      COL80
:         COL85      COL86      COL87      COL88      COL89      COL90      COL91      COL92
:         COL97      COL98      COL99     COL100     COL101     COL102     COL103     COL104
:        COL109     COL110     COL111     COL112     COL113     COL114     COL115     COL116
:        COL121     COL122     COL123     COL124     COL125     COL126     COL127     COL128
:        COL133     COL134     COL135     COL136     COL137     COL138     COL139     COL140
:        COL145     COL146     COL147     COL148     COL149     COL150     COL151     COL152
:        COL157     COL158     COL159     COL160     COL161     COL162     COL163     COL164
:        COL169     COL170     COL171     COL172     COL173     COL174     COL175     COL176
:        COL181     COL182     COL183     COL184     COL185     COL186     COL187     COL188
:        COL193     COL194     COL195     COL196     COL197     COL198     COL199     COL200
:        COL205     COL206     COL207     COL208     COL209     COL210     COL211     COL212
:        COL217     COL218     COL219     COL220     COL221     COL222     COL223     COL224
:        COL229     COL230     COL231     COL232     COL233     COL234     COL235     COL236
:       COL241    COL242    COL243    COL244    COL245    COL246    COL247    COL248
:       COL253    COL254    COL255    COL256    COL257    COL258    COL259    COL260
:       COL265    COL266    COL267    COL268    COL269    COL270    COL271    COL272
:       COL277    COL278    COL279    COL280    COL281    COL282    COL283    COL284
:       COL289    COL290    COL291    COL292    COL293    COL294    COL295    COL296

ROW1 -1.100986 0.5102031 -1.613007  -0.84036 0.3063094 -0.137757 0.5837736 -1.101155 .....
:    0.5287544 0.8516944 -0.133082  0.681287 1.1035271 -2.074337 -0.416986 1.9221902
:    0.6359094 2.0713417 -0.553496 -1.008133   0.29938 -1.338365 1.1793219 -1.518272
:    1.5782633 -1.505502 0.2561553  0.042185 0.2186959 -0.149011 -1.128817 0.4563077
:    -2.305372 1.1187851 -0.243926 0.1109998 0.0372219 -0.085549 -0.052607 0.0771608
:     -0.04872 0.1991262 0.0674431 0.0404002 -0.198509 0.0608353  -0.04278   -0.0526
:    -0.200019 -0.322247 0.2101414 -0.053752 0.0135931 0.2538245 0.3517435 0.2708867
:    -0.157511  -0.28445 -0.017414 0.0689082 0.0808218 0.0862198 0.2676685 0.0204158
:    -0.170465 0.3430358 0.2090057 -0.002725 -0.034753 0.2702058 0.2149432 0.5008493
:    -0.018726 0.0894909  0.142061 0.1803872 0.1102643   0.35851 0.0506527 -0.000407
:    0.1720143 -0.051837 -0.010171 -0.067371 -0.051732 -0.038141 -0.314836 0.0479982
:    0.0160665 -0.194046 -0.028615 0.1563287 0.0077524 -0.101216 0.3473905 0.1812392
:    0.0966794 0.1592779 -0.236369 0.0498978 0.0788471 0.1151401 0.1450518 0.1413605
:     0.137399 -0.006491 0.0273292 -0.172579 0.0595944 0.0913617 -0.152912 0.2415581
:    -0.079462 -0.044107 -0.131882 0.2708037 0.0386933 0.3803824 0.2174077 0.0733851
:    0.1886892 -0.022997 -0.048782 0.1568083 0.0066373 -0.033817 0.2847843 -0.114932
:    0.2671163 0.0652588 0.3048099 -0.240471  -0.28957    -0.094 0.0587564 -0.298295
:    -0.061271  -0.23948 0.0597063 0.0310164 -0.068017 0.1434325 -0.000139 0.2121642
:    -0.268803 -0.136107 0.4917512 -0.091679 0.0296499   0.46147 0.0939394 0.1146663
:    -0.035571 0.1796191 0.1742778 0.3124138  0.121722 -0.114123 -0.163078 0.2880078
:    0.2608644 0.5055647 0.1156067 -0.146011  0.134441 -0.138752 0.2412992 0.3725287
:    0.2892954 -0.062676 0.1441325 -0.017427 -0.061924 0.1361985 0.2213222 0.2262125
:    0.1667814 0.0067131 -0.338068 0.1991028 0.0443752 0.3135095 -0.123353 -0.056139
:    0.1692064 0.1104018 -0.030632 -0.047068 0.1028076 0.1786027 -0.051105 0.3394164
:    0.2097295 0.1355379 0.2517211 0.5178495 -0.193963 0.1835999 0.1169275 -0.234731

ROW2 0.4249926 0.3064045 0.0815213 0.6593596 -0.676497 0.4638712 -0.017465 -0.867974
:    -1.418743 -1.958731 0.0326734 -1.172076 0.1572937 -0.166277  0.565414 1.1940593
:    0.4360652 -1.047591 -0.017632 0.4095882 1.0449989 0.7487248 0.7157709 -0.493609
:    -0.126856 -1.886986 0.0881589 -2.055992 2.4390881 0.7126954 1.6612392 2.2704596
:    -2.426078 -0.588812 0.5891338 -0.091597 1.1842088 -0.136536 0.5328554 -0.298682
:    -0.383852 -0.819176 -0.389136 -0.597283 -0.104945 -0.328453 1.3583993 -0.485435
:    0.3079504 0.4989418  1.256052   1.56099 1.1071625 0.5844389 0.1292411 0.9426685
:     -0.68623 0.3629111 0.7289298 -0.450567 1.0372245 -0.956004 0.5030219 0.1681693
:    -0.065265 -0.890186 0.0972406  -0.25574 -0.590917 0.4927487  0.587238 -0.005902
:    -0.671052 1.0404846 0.1175768 -0.143582 -0.616475 0.3012437  1.151561 0.2535198
:    -0.038308  0.054376 0.5138002  -0.68987 1.2937411 -0.643377 0.9635249 -0.662235
:    -0.593739 -0.226537  -0.02469 0.2957793 -1.046079 -1.261575 -0.265756 0.4000814
:    -0.198275 0.6627909 1.9949194 -1.125654 -0.587661 -0.577556 0.4962336 -0.554475
:    -1.469166 -0.349216   0.41004 1.6172998 -1.525985 0.3018435 0.9065155   -0.3956
:    -0.151668 -0.221188 -0.225388 0.2151074 0.0179784 -0.719304 0.5091919 0.7200979
:    -0.065207 0.7326168 0.0060279 0.5149406 -0.601689 0.8043517 0.1993369  -0.51335
:    -0.435928  -0.83228 -0.561111 0.2619272 -0.151575 -0.493855 1.3830999 1.0949767
:     -1.23305 0.3842309 -0.833243 -0.328013 -0.110339 -1.073295 -0.386449 -0.232331
:    0.7933134 0.0899845 0.9525884 -0.121287 0.1362727  -0.14319 -0.957823  0.424908
:     -1.34593 -0.270528 0.0756668 -0.006935 -0.310795 0.5484622 0.5314162 -0.448536
:     0.157439 0.1298997 0.7132249 -0.375488  -1.30748 -0.677699 0.2173974 -0.124103
:    1.6231019  -0.16756 1.0977466 -1.219658 -0.638558  -0.92839 -0.297888 1.0831974
:    -0.429151 -0.460182 0.3737078 0.3418089 -0.255461 0.8336142 -0.422176 0.0805384
:    0.3083144 0.8955046 -0.167817 1.5332944 -0.606302 -0.085942 1.5972455 -0.115524
:    -0.243633 1.3703103 0.3783086  0.945855 0.4906353 -0.623279 1.3897448 -0.060079


X           COL1      COL2      COL3      COL4      COL5      COL6      COL7      COL8 .....
:          COL13     COL14     COL15     COL16     COL17     COL18     COL19     COL20
:          COL25     COL26     COL27     COL28     COL29     COL30     COL31     COL32
:          COL37     COL38     COL39     COL40     COL41     COL42     COL43     COL44
:          COL49     COL50     COL51     COL52     COL53     COL54     COL55     COL56
:          COL61     COL62     COL63     COL64     COL65     COL66     COL67     COL68
:          COL73     COL74     COL75     COL76     COL77     COL78     COL79     COL80
:          COL85     COL86     COL87     COL88     COL89     COL90     COL91     COL92
:          COL97     COL98     COL99    COL100    COL101    COL102    COL103    COL104
:         COL109    COL110    COL111    COL112    COL113    COL114    COL115    COL116
:         COL121    COL122    COL123    COL124    COL125    COL126    COL127    COL128
:        COL133   COL134   COL135   COL136   COL137   COL138   COL139   COL140
:        COL145   COL146   COL147   COL148   COL149   COL150   COL151   COL152
:        COL157   COL158   COL159   COL160   COL161   COL162   COL163   COL164
:        COL169   COL170   COL171   COL172   COL173   COL174   COL175   COL176
:        COL181   COL182   COL183   COL184   COL185   COL186   COL187   COL188
:        COL193   COL194   COL195   COL196   COL197   COL198   COL199   COL200
:        COL205   COL206   COL207   COL208   COL209   COL210   COL211   COL212
:        COL217   COL218   COL219   COL220   COL221   COL222   COL223   COL224
:        COL229   COL230   COL231   COL232   COL233   COL234   COL235   COL236
:        COL241   COL242   COL243   COL244   COL245   COL246   COL247   COL248
:        COL253   COL254   COL255   COL256   COL257   COL258   COL259   COL260
:        COL265   COL266   COL267   COL268   COL269   COL270   COL271   COL272
:        COL277   COL278   COL279   COL280   COL281   COL282   COL283   COL284
:        COL289   COL290   COL291   COL292   COL293   COL294   COL295   COL296

ROW1          1        1        1        1        1        1        1        1 .....
:             1        1        1        1        1        1        1        1
:             1        1        1        1        1        1        1        1
:             1        1        1        1        1        1        1        1
:             1        1        1        1        1        1        1        1
:             1        1        1        1        1        1        1        1
:             1        1        1        1        1        1        1        1
:             1        1        1        1        1        1        1        1
:             1        1        1        1        1        1        1        1
:             1        1        1        1        1        1        1        1
:             1        1        1        1        1        1        1        1
:             1        1        1        1        1        1        1        1
:             1        1        1        1        1        1        1        1
:             1        1        1        1        1        1        1        1
:             1        1        1        1        1        1        1        1
:             1        1        1        1        1        1        1        1
:             1        1        1        1        1        1        1        1
:             1        1        1        1        1        1        1        1
:             1        1        1        1        1        1        1        1
:             1        1        1        1        1        1        1        1
:             1        1        1        1        1        1        1        1
:             1        1        1        1        1        1        1        1
:             1        1        1        1        1        1        1        1
:             1        1        1        1        1        1        1        1
:             1        1        1        1        1        1        1        1


************************LSDE1******************************************
************************MODUL:  MOMENT1*********************************
***********************************************************************


MO         COL1

ROW1 -0.068309
ROW2 0.0577723


MOO        COL1       COL2

ROW1 1.0618557    0.02082
ROW2   0.02082 1.0414693


M          COL1       COL2       COL3       COL4       COL5

ROW1 0.0316193  -0.001755  0.0065786  -0.005116  0.0505561
ROW2 -0.001755  0.5039918  -0.016698  0.0420705  0.0520704
ROW3 0.0065786  -0.016698  0.2369732  -0.004071  0.0249859
ROW4 -0.005116  0.0420705  -0.004071  0.6007042  0.0057439
ROW5 0.0505561  0.0520704  0.0249859  0.0057439          1
```

```
***********************LSDE1********************************************
***********************MODUL: NEWTON************************************
************************************************************************

Technical Output

          KMAX        EPS       DT        N        T  OPTION    NONLIN    MAXSTEP  MINSTEP   BOUND

ROW1        50     0.0001        2       50        5       0         0       1000    1E-10    1E-6

NEWTON: STARTWERT THETA0

THETA0       COL1

ROW1        -16.3
ROW2         -1.3
ROW3          0.7
ROW4          1.7
ROW5         -0.3
ROW6         -0.3
ROW7          0.7
ROW8         -0.3
ROW9          0.7

--------------------NEWTON------------------------------

K           COL1

ROW1           1


KET                LIKNEU               LIKALT

ROW1 -16640209273.13110 -16640213975.28090


        THETA1      THETA      STEP          S

ROW1 -0.037462      -16.3 16.262538 663.24705
ROW2 -0.465492       -1.3 0.8345078 74.606483
ROW3  0.026773        0.7 -0.673227 -45.34608
ROW4 37.911247        1.7 36.211247 6264.9216
ROW5 -0.068309       -0.3 0.2316913 33.689451
ROW6 0.0577723       -0.3 0.3577723 39.993502
ROW7 1.1108705        0.7 0.4108705 64.764894
ROW8 0.1076591       -0.3 0.4076591 131.44155
ROW9 1.1661327        0.7 0.4661327 68.218782

--------------------NEWTON------------------------------

K           COL1

ROW1           2


KET                LIKNEU               LIKALT


ROW1 -16640209099.65840 -16640209273.13110


        THETA1     THETA       STEP          S

ROW1 -0.037462 -0.037462 4.458E-15 3.652E-16
ROW2 -0.465492 -0.465492 -4.11E-17 -1.54E-17
ROW3  0.026773  0.026773  -2.7E-16 -5.52E-17
ROW4 18.958901 37.911247 -18.95235 -6.593209
ROW5 -0.068309 -0.068309 6.939E-17  3.14E-15
ROW6 0.0577723 0.0577723 2.602E-18 -1.78E-16
```

```
ROW7  1.0571897  1.1108705 -0.053681   -0.81404
ROW8  0.0247664  0.1076591 -0.082893  -2.617793
ROW9  1.0381317  1.1661327 -0.128001  -2.104576

...............
...............

 --------------------NEWTON-----------------------------

K           COL1

ROW1          12


KET                  LIKNEU                 LIKALT

ROW1 -16640208314.97690 -16640208314.97690


          THETA1       THETA       STEP           S

ROW1 -0.037462 -0.037462  1.846E-18  1.036E-15
ROW2 -0.465492 -0.465492  -1.35E-17  -1.64E-14
ROW3  0.026773  0.026773   8.99E-19  1.745E-15
ROW4 0.4985223 0.4985223  -2.01E-12  -4.035E-9
ROW5 -0.068309 -0.068309          0          0
ROW6 0.0577723 0.0577723          0          0
ROW7 1.0571897 1.0571897          0          0
ROW8 0.0247664 0.0247664          0          0
ROW9 1.0381317 1.0381317          0          0

NEWTON: ENDE DER ITERATION

K           COL1

ROW1          12


KMAX        COL1

ROW1          50

ML-Sch{tzer, SCORE UND LIKELIHOOD


           THETA      SCORE

ROW1 -0.037462  9.816E-16
ROW2 -0.465492  -1.64E-14
ROW3  0.026773          0
ROW4 0.4985223    1.4E-14
ROW5 -0.068309          0
ROW6 0.0577723          0
ROW7 1.0571897          0
ROW8 0.0247664          0
ROW9 1.0381317          0


LIKELI               COL1

ROW1    -16640208314.97690

-----------WALD-TEST-----------


            CHI          U          P     ALPHA

ROW1 813.69757           9          0      0.05

NULLHYPOTHESE THETA=0 ABLEHNEN
```

```
-----------BONFERRONI-KONFIDENZINTERVALLE-----------

P{THETA AUS [THDACH-GRENZE, THDACH+GRENZE] } »= 1-ALPHA


     Z(alpha/2u)

ROW1   2.7729213


          THETA    GRENZE       STD

ROW1 -0.037462 0.1271708 0.0458617
ROW2 -0.465492 0.0797712 0.0287679
ROW3  0.026773 0.0619047 0.0223247
ROW4 0.4985223 0.0618212 0.0222946
ROW5 -0.068309 0.4032079 0.1454091
ROW6 0.0577723  0.399557 0.1440924
ROW7 1.0571897 0.5863007 0.2114379
ROW8 0.0247664 0.4109382 0.1481969
ROW9 1.0381317 0.5757315 0.2076263

Korrelation der ML-Sch{tzer

CORR  COL1  COL2  COL3  COL4  COL5  COL6  COL7  COL8  COL9

ROW1  1.00  0.01 -0.05  0.00  0.00  0.00  0.00  0.00  0.00
ROW2  0.01  1.00 -0.01  0.00  0.00  0.00  0.00  0.00  0.00
ROW3 -0.05 -0.01  1.00  0.00  0.00  0.00  0.00  0.00  0.00
ROW4  0.00  0.00  0.00  1.00  0.00  0.00  0.00  0.00  0.00
ROW5  0.00  0.00  0.00  0.00  1.00  0.02  0.00  0.00  0.00
ROW6  0.00  0.00  0.00  0.00  0.02  1.00  0.00  0.00  0.00
ROW7  0.00  0.00  0.00  0.00  0.00  0.00  1.00  0.03  0.00
ROW8  0.00  0.00  0.00  0.00  0.00  0.00  0.03  1.00  0.03
ROW9  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.03  1.00

GESCH{TZTE KOVARIANZMATRIX UND EIGENWERTE

COV        COL1      COL2      COL3      COL4      COL5      COL6      COL7      COL8      COL9

ROW1 0.0021033 0.0000148 -0.000053         0         0         0         0         0         0
ROW2 0.0000148 0.0008276 -5.122E-6         0         0         0         0         0         0
ROW3 -0.000053 -5.122E-6 0.0004984         0         0         0         0         0         0
ROW4         0         0         0  0.000497         0         0         0         0         0
ROW5         0         0         0         0 0.0211438 0.0004953         0         0         0
ROW6         0         0         0         0 0.0004953 0.0207626         0         0         0
ROW7         0         0         0         0         0         0  0.044706 0.0010473 0.0000245
ROW8         0         0         0         0         0         0 0.0010473 0.0219623 0.0010284
ROW9         0         0         0         0         0         0 0.0000245 0.0010284 0.0431087


EIGCOV      COL1

ROW1   0.0004966
ROW2    0.000497
ROW3   0.0008275
ROW4   0.0021052
ROW5   0.0204225
ROW6   0.0214839
ROW7   0.0218646
ROW8   0.0431551
ROW9   0.0447573

GESCH{TZTE FISHER-INFORMATIONS-MATRIX UND EIGENWERTE

F          COL1      COL2      COL3      COL4      COL5      COL6      COL7      COL8      COL9

ROW1 476.76026  -8.18974 50.268379         0         0         0         0         0         0
ROW2  -8.18974 1208.5412 11.555998         0         0         0         0         0         0
ROW3 50.268379 11.555998 2011.8739         0         0         0         0         0         0
ROW4         0         0         0 2011.8739         0         0         0         0         0
```

```
ROW5         0         0         0         0  47.32165 -1.128937         0         0         0
ROW6         0         0         0         0 -1.128937  48.19038         0         0         0
ROW7         0         0         0         0         0         0 22.393386 -1.068464  0.012745
ROW8         0         0         0         0         0         0 -1.068464 45.634456 -1.088078
ROW9         0         0         0         0         0         0  0.012745 -1.088078 23.223127


EIGF      COL1

ROW1 2013.6763
ROW2 2011.8739
ROW3  1208.483
ROW4 475.01597
ROW5 48.965631
ROW6 46.546398
ROW7 45.736007
ROW8 23.172254
ROW9 22.342707


KOMPENSIERTE LIKELIHOOD

U         COL1

ROW1         9


NUE       COL1

ROW1       300
  AKAIKE      SCHWARZ/RISSANEN   AZENCOTT&DACUNHA-CASTELLE


KET              LIK          AIC      SIC/RISS           AD

KOMP          0.0000       9.0000       25.6670      78.4292
KOMPLIK -16640208315 -16640208324 -16640208341 -16640208393

---------------DELTA-THEOREM----------------------


GESCH{TZTE STANDARDABWEICHUNGEN DER SYSTEM-MATRIZEN
A, B, OMEGA, SIGMA UND MUE


A         COL1      COL2

ROW1         0         1
ROW2 -0.037462 -0.465492


STDA      COL1      COL2

ROW1         0         0
ROW2 0.0458617 0.0287679


B         COL1

ROW1         0
ROW2  0.026773


STDB      COL1

ROW1         0
ROW2 0.0223247
```

```
OMEGA       COL1      COL2

ROW1        1E-8         0
ROW2           0 0.2485245


STDO       COL1      COL2

ROW1          0         0
ROW2          0 0.0222287


MUE        COL1

ROW1 -0.068309
ROW2 0.0577723


STDMUE       COL1

ROW1    0.1454091
ROW2    0.1440924


SIGMA       COL1      COL2

ROW1   1.0571897 0.0247664
ROW2   0.0247664 1.0381317


STDSIG       COL1      COL2

ROW1    0.2114379 0.1481969
ROW2    0.1481969 0.2076263

ITO-SCH{TZER

AB          COL1      COL2      COL3

ROW1 -0.488834 -0.004422 0.0250245
ROW2 -0.037462 -0.465492  0.026773


ODACH       COL1      COL2

ROW1   0.1277176 0.0079942
ROW2   0.0079942 0.5102775

***********************LSDE1*******************************************
***********************MODUL: NEWTON***********************************
***********************************************************************

Technical Output


          KMAX       EPS     DT     N     T    OPTION   NONLIN  MAXSTEP  MINSTEP  BOUND

ROW1        50    0.0001      2    50     5         0        1     1000    1E-10   1E-6

NEWTON: STARTWERT THETA0

THETA0      COL1
ROW1       -16.3
ROW2        -1.3
ROW3         0.7
ROW4         1.7
ROW5        -0.3
ROW6        -0.3
ROW7         0.7
ROW8        -0.3
ROW9         0.7
```

```
--------------------NEWTON-----------------------------

K          COL1

ROW1          1


KET                 LIKNEU                 LIKALT

ROW1      196.0499191127      104.2100809932


         THETA1      THETA       STEP           S

ROW1 -17.02468       -16.3 -0.724679 -2.472477
ROW2 -2.306611        -1.3 -1.006611 -133.6898
ROW3 0.9068924         0.7 0.2068924 2.2688975
ROW4  1.923646         1.7  0.223646 -108.3086
ROW5 -0.068309        -0.3 0.2316913 33.689451
ROW6 0.0577723        -0.3 0.3577723 39.993502
ROW7 1.1108705         0.7 0.4108705 64.764894
ROW8 0.1076591        -0.3 0.4076591  131.44155
ROW9 1.1661327         0.7 0.4661327 68.218782
--------------------NEWTON-----------------------------

K          COL1

ROW1          2


KET                 LIKNEU                 LIKALT

ROW1      221.6373629179      196.0499191127


         THETA1      THETA       STEP           S

ROW1 -17.51798 -17.02468 -0.493299 -2.225973
ROW2 -3.277828 -2.306611 -0.971217 -49.58671
ROW3 0.9218903 0.9068924 0.0149979 -0.388417
ROW4 1.9901162  1.923646 0.0664703 -89.70727
ROW5 -0.068309 -0.068309 6.939E-17  3.14E-15
ROW6 0.0577723 0.0577723 2.602E-18 -1.78E-16
ROW7 1.0571897 1.1108705 -0.053681  -0.81404
ROW8 0.0247664 0.1076591 -0.082893 -2.617793
ROW9 1.0381317 1.1661327 -0.128001 -2.104576


.................
.................


--------------------NEWTON-----------------------------

K          COL1

ROW1         11


KET                 LIKNEU                 LIKALT

ROW1      226.3492072718      226.3492072714


         THETA1      THETA       STEP           S

ROW1 -16.82291 -16.82288 -0.000035 -0.000013
ROW2 -4.131976 -4.131999 0.0000233 0.0000336
ROW3 0.8608502 0.8608487 1.4745E-6 -0.000012
```

```
ROW4 2.0228124  2.022817 -4.616E-6 -3.856E-8
ROW5 -0.068309 -0.068309         0         0
ROW6 0.0577723 0.0577723         0         0
ROW7 1.0571897 1.0571897         0         0
ROW8 0.0247664 0.0247664         0         0
ROW9 1.0381317 1.0381317         0         0

NEWTON: ENDE DER ITERATION

K          COL1

ROW1         11


KMAX       COL1

ROW1         50


ML-Sch{tzer, SCORE UND LIKELIHOOD


         THETA     SCORE

ROW1 -16.82291 3.8306E-6
ROW2 -4.131976 -9.584E-6
ROW3 0.8608502 3.4808E-6
ROW4 2.0228124 -3.228E-9
ROW5 -0.068309         0
ROW6 0.0577723         0
ROW7 1.0571897         0
ROW8 0.0247664         0
ROW9 1.0381317         0


LIKELI                COL1

ROW1       226.349207271833

-----------WALD-TEST-----------

          CHI           U         P     ALPHA

ROW1 191.00815          9         0      0.05

NULLHYPOTHESE THETA=0 ABLEHNEN

-----------BONFERRONI-KONFIDENZINTERVALLE-----------

P{THETA AUS [THDACH-GRENZE, THDACH+GRENZE] } »= 1-ALPHA

     Z(alpha/2u)

ROW1   2.7729213

         THETA    GRENZE       STD

ROW1 -16.82291 5.5282613 1.9936596
ROW2 -4.131976 2.5715146 0.9273666
ROW3 0.8608502  0.584131 0.2106554
ROW4 2.0228124 0.7091925 0.2557564
ROW5 -0.068309 0.4032079 0.1454091
ROW6 0.0577723  0.399557 0.1440924
ROW7 1.0571897 0.5863007 0.2114379
ROW8 0.0247664 0.4109382 0.1481969
ROW9 1.0381317 0.5757315 0.2076263
```

```
Korrelation der ML-Sch{tzer

CORR  COL1  COL2  COL3  COL4  COL5  COL6  COL7  COL8  COL9

ROW1  1.00  0.24 -0.47 -0.45  0.00  0.00  0.00  0.00  0.00
ROW2  0.24  1.00 -0.09 -0.94  0.00  0.00  0.00  0.00  0.00
ROW3 -0.47 -0.09  1.00  0.19  0.00  0.00  0.00  0.00  0.00
ROW4 -0.45 -0.94  0.19  1.00  0.00  0.00  0.00  0.00  0.00
ROW5  0.00  0.00  0.00  0.00  1.00  0.02  0.00  0.00  0.00
ROW6  0.00  0.00  0.00  0.00  0.02  1.00  0.00  0.00  0.00
ROW7  0.00  0.00  0.00  0.00  0.00  0.00  1.00  0.03  0.00
ROW8  0.00  0.00  0.00  0.00  0.00  0.00  0.03  1.00  0.03
ROW9  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.03  1.00

GESCH{TZTE KOVARIANZMATRIX UND EIGENWERTE

COV        COL1      COL2      COL3      COL4      COL5      COL6      COL7      COL8      COL9

ROW1 3.9746788 0.4427668 -0.197717 -0.227697         0         0         0         0         0
ROW2 0.4427668 0.8600088  -0.01787 -0.223194         0         0         0         0         0
ROW3 -0.197717  -0.01787 0.0443757 0.0103127         0         0         0         0         0
ROW4 -0.227697 -0.223194 0.0103127 0.0654114         0         0         0         0         0
ROW5         0         0         0         0 0.0211438 0.0004953         0         0         0
ROW6         0         0         0         0 0.0004953 0.0207626         0         0         0
ROW7         0         0         0         0         0         0  0.044706 0.0010473 0.0000245
ROW8         0         0         0         0         0         0 0.0010473 0.0219623 0.0010284
ROW9         0         0         0         0         0         0 0.0000245 0.0010284 0.0431087

EIGCOV    COL1

ROW1   0.0038575
ROW2   0.0204225
ROW3   0.0214839
ROW4   0.0218646
ROW5   0.0344302
ROW6   0.0431551
ROW7   0.0447573
ROW8   0.8434598
ROW9   4.0627272

GESCH{TZTE FISHER-INFORMATIONS-MATRIX UND EIGENWERTE

F          COL1      COL2      COL3      COL4      COL5      COL6      COL7      COL8      COL9

ROW1 0.5617061 1.6504182 1.4575929 7.3569789         0         0         0         0         0
ROW2 1.6504182 15.787643 -0.148465 59.638401         0         0         0         0         0
ROW3 1.4575929 -0.148465 28.969386         0         0         0         0         0         0
ROW4 7.3569789 59.638401         0 244.39299         0         0         0         0         0
ROW5         0         0         0         0  47.32165 -1.128937         0         0         0
ROW6         0         0         0         0 -1.128937  48.19038         0         0         0
ROW7         0         0         0         0         0         0 22.393386 -1.068464  0.012745
ROW8         0         0         0         0         0         0 -1.068464 45.634456 -1.088078
ROW9         0         0         0         0         0         0  0.012745 -1.088078 23.223127


EIGF       COL1

ROW1 259.23572
ROW2 48.965631
ROW3 46.546398
ROW4 45.736007
ROW5 29.044275
ROW6 23.172254
ROW7 22.342707
ROW8  1.185593
ROW9 0.2461401


KOMPENSIERTE LIKELIHOOD
```

```
U          COL1

ROW1          9


NUE        COL1

ROW1        300

 AKAIKE       SCHWARZ/RISSANEN    AZENCOTT&DACUNHA-CASTELLE


KET                LIK          AIC     SIC/RISS            AD

KOMP            0.0000       9.0000      25.6670       78.4292
KOMPLIK       226.3492     217.3492     200.6822      147.9200


---------------DELTA-THEOREM----------------------
GESCH{TZTE STANDARDABWEICHUNGEN DER SYSTEM-MATRIZEN
A, B, OMEGA, SIGMA UND MUE


A          COL1       COL2

ROW1          0          1
ROW2 -16.82291 -4.131976


STDA       COL1       COL2

ROW1          0          0
ROW2 1.9936596 0.9273666


B          COL1

ROW1          0
ROW2 0.8608502


STDB       COL1

ROW1          0
ROW2 0.2106554


OMEGA       COL1       COL2

ROW1        1E-8          0
ROW2           0 4.0917699


STDO       COL1       COL2

ROW1          0          0
ROW2          0 1.0346946


MUE        COL1

ROW1 -0.068309
ROW2 0.0577723


STDMUE      COL1

ROW1   0.1454091
ROW2   0.1440924
```

```
SIGMA       COL1      COL2

ROW1  1.0571897 0.0247664
ROW2  0.0247664 1.0381317


STDSIG       COL1      COL2

ROW1    0.2114379 0.1481969
ROW2    0.1481969 0.2076263
```

Comment:
First the data are simulated by the module EXAKT (exact discrete model) and printed. Afterwards moment matrices are computed from the data (module MOMENT1) for usage in NEWTON. This module is first used with NONLIN = 0 (discretized continuous sampling). The module prints the initial values for the scoring iteration and successive values of the score function and the parameter vector. After the iteration has converged, ML-estimator, score and likelihood function are printed. Furthermore, Wald test, Bonferroni confidence intervals, correlation matrix and covariance matrix of ML estimator are reported. Then, the Fisher information matrix, compensated likelihoods and system matrices (by virtue of the Δ-theorem) are printed. Afterwards, the same is executed for NONLIN = 1 (exact ML estimation). Comparision of the ML estimates reveals the strong bias in the linearized estimates (cf. table III.1) ■

In order to produce table III.1 a Monte Carlo simulation had to be performed. This means that the simulation and estimation procedure described above has to be repeated M times (M=100 was used). Instead of NEWTON the module NEWSIM was used to limit the output. Furthermore, the ML estimates were collected and stored on disk using the STORE command.

IML-Code:

```
PROC IML;                                        *begin of iml program;
RESET STORAGE=COMP.LSDE1;                  *Loading library LSDE from disk;
LOAD;
RUN INITLSDE ;                       * running module INITLSDE initializes;
                                                  * module library LSDE;

*************************************************************************;
START OSZSIM; *SIMULATIONSSTUDIE;
*************************************************************************;

*************************************************************************;
START PAR ( OMEGA, A, B, SIGMA, MUE, TH);
*************************************************************************;

A=( 0   ||  1  )//
  (TH[1]||TH[2]);

B=   0  //
   TH[3];
```

```
G=(  0   ||  0  )//
  (  0   ||TH[4]);

OMEGA=G*G`;

MUE= TH[5]//
     TH[6];

SIGMA=(TH[ 7]||TH[ 8]  )//
      (TH[ 8]||TH[ 9]);

*PRINT '-----------PAR-----------',G OMEGA ;
*PRINT      A,B,TH ;
FINISH;
************************************************************************;
START PARD (DODTH,DADTH,DBDTH,DSIGDTH,DMUEDTH,TH);
************************************************************************;
* MC-DONALD-SWAMINATHAN-ABLEITUNGEN ;


DADTH= { 0 0 1 0,
         0 0 0 1,
         0 0 0 0,
         0 0 0 0,
         0 0 0 0,
         0 0 0 0,
         0 0 0 0,
         0 0 0 0,
         0 0 0 0 };

DBDTH= { 0 0,
         0 0,
         0 1,
         0 0,
         0 0,
         0 0,
         0 0,
         0 0,
         0 0 };


DODTH=J(3,4,0)//(0||0||0||2*TH[4])//J(5,4,0);

DMUEDTH={ 0 0,
          0 0,
          0 0,
          0 0,
          1 0,
          0 1,
          0 0,
          0 0,
          0 0 };

DSIGDTH= { 0 0 0 0,
           0 0 0 0,
           0 0 0 0,
            0 0 0 0,
            0 0 0 0,
            0 0 0 0,
            1 0 0 0,
            0 1 1 0,
            0 0 0 1 };

*PRINT '-----------PARD-----------';
*PRINT      DODTH, DADTH, DBDTH, TH ;
FINISH;
***************************************************************************;
RESET STORAGE=SIMUL.S13579;
Q=1;
P=2;
P2=P*P;
T=5;  N=50;
T1=T+1;
DT=.01;
DELTA=2.;

A=  {  0   1,
     -16  -4 };

B={ 0,
    1 };

G={ 0 0,
    0 2 };

OMEGA={ 0 0,
        0 4 };
MUE={ 0,
      0 };
SIGMA={ 1 0,
        0 1 };
PRINT A, B, G, OMEGA, MUE, SIGMA, P  Q  DELTA;
SEED=13579;
FREE THLIN THEX;
REP=100;
DO L=1 TO REP;
X=SHAPE(1, Q, N*T1);
YO=J(1, N, 1) § MUE +ROOT(SIGMA)*RANNOR(SHAPE(SEED, P, N));  *AB N(MUE, SIGMA);
RUN EXAKT(Y, A, B, G, YO, X, P, Q, N, T, DELTA, SEED, 0);
RUN MOMENT1(MO, MOO, M11, M12, M13, M22, M23, M33, Y, X, P, Q, N, T, 0);
THETAO={ -16, -4,  1,   2,   0, 0,  1, 0, 1 } - .3;
KMAX=50;
EPS=1E-4;
RUN   NEWSIM(THETA1, COV, THETAO, KMAX, EPS,
                 MO, MOO, M11, M12, M13, M22, M23, M33, DELTA, N, T, 0, 0);
THLIN=THLIN//THETA1`;
RUN   NEWSIM(THETA, COV, THETAO, KMAX, EPS,
               MO, MOO, M11, M12, M13, M22, M23, M33, DELTA, N, T, 0, 1);
THEX=THEX//THETA`;
END;
STORE THLIN THEX;
PRINT THLIN , THEX;
```

```
THLQ=THLIN[:, ];THEXQ=THEX[:, ];
REP1=1/(REP-1);
THLS=REP1*(THLIN`*THLIN -REP*THLQ`*THLQ);
THEXS=REP1*(THEX`*THEX - REP*THEXQ`*THEXQ);
STORE THLQ THLS THEXQ THEXS;

PRINT 'MITTELWERTE UND STANDARDFEHLER DER ML-SCH{TZER' THLQ,THLS,THEXQ,THEXS;

FINISH;
RUN OSZSIM;
QUIT;
```

Comment:

The ML estimates, means and standard deviations for the exact and linearized model are stored on disk in the storage catalog SIMUL . S13579. Afterwards, the sample was analyzed using the SAS procedure UNIVARIATE (see SAS/STATISTICS) which calculates several statistics and plots histograms of the sample distribution. To achieve this, write a simple IML program which creates a SAS data set which can be used by UNIVARIATE:

```
PROC IML;
RESET STORAGE=SIMUL.S13579;
SHOW STORAGE;
LOAD ;
NAMES=STORAGE();
PRINT NAMES;
PRINT THLIN, THEX/THLQ,THLS,THEXQ,THEXS;
CREATE DATEN1 FROM THLIN;
append from thlin;
CREATE DATEN2 FROM THEX;
append from thex;
QUIT;
PROC UNIVARIATE DATA=DATEN1 PLOT FREQ NORMAL;
PROC UNIVARIATE DATA=DATEN2 PLOT FREQ NORMAL;
```

2. Case Study 2: The Phillips Model

In chapter II.4 an econometric model, the so called Phillips model was described, which is given as a three equations model. The entries of the diffusion matrix A are restricted (they are functions of a parameter vector). Furthermore it is assumed that the diffusion matrix is fixed (containing the true values).

```
PROC IML;                                              *begin of iml program;
RESET STORAGE=COMP.LSDE1;                   *Loading library LSDE from disk;
LOAD;
RUN INITLSDE ;                            * running module INITLSDE initializes;
                                                         * module library LSDE;

****************************************************************************;
START PHILLIPS;
****************************************************************************;
*FIXED DIFFUSION MATRIX DIAG(1,1,1);
****************************************************************************;
START PAR ( OMEGA,A,B,SIGMA,MUE,TH);
****************************************************************************;

A=(  -TH[1]   || TH[1]*(1-TH[5])        ||       0       ) //
  (   TH[2]   || TH[2]*(TH[3]*TH[4]-1) || -TH[2]*TH[3] ) //
  (     0     || TH[3]*TH[4]            ||     -TH[3]    ) ;

B= (  TH[1] )//
   (    0   )//
   (    0   );

OMEGA=I(3);
MUE=  0   //
      0   //
      0   ;
SIGMA=1E4*I(3);

*PRINT '-----------PAR-----------',G OMEGA ;
*PRINT       A,B,TH ;
FINISH;
****************************************************************************;
START PARD (DODTH,DADTH,DBDTH,DSIGDTH,DMUEDTH,TH);
****************************************************************************;
* MC-DONALD-SWAMINATHAN-ABLEITUNGEN ;


DADTH= ( -1 || 1-TH[5] || 0 || 0 ||       0        ||   0   || 0 ||   0   || 0 )//
       (  0 ||    0     || 0 || 1 ||TH[3]*TH[4]-1||-TH[3]|| 0 ||   0   || 0 )//
       (  0 ||    0     || 0 || 0 ||TH[2]*TH[4]   ||-TH[2]|| 0 ||TH[4]||-1 )//
       (  0 ||    0     || 0 || 0 ||TH[2]*TH[3]   ||   0   || 0 ||TH[3]||0  )//
       (  0 ||-TH[1]    || 0 || 0 ||       0        ||   0   || 0 ||   0   || 0 );

DBDTH={1 0 0,
       0 0 0,
       0 0 0,
       0 0 0,
       0 0 0 };

DODTH=J(5,9,0);

DMUEDTH=J(5,3,0);
DSIGDTH=J(5,9,0);

*PRINT '-----------PARD-----------';
*PRINT      DODTH,DADTH,DBDTH,TH ;
FINISH;
************************************************************************;

*Data set given by Phillips (1972);
Y={ 20.001465 20.001294 40.002760,
    20.724659 21.873653 40.319084,
    19.500591 20.517932 41.772445,
    17.740573 16.459774 40.902389,
    16.797718 12.794065 36.626464,
    16.304851 14.501026 32.963211,
    13.996612 13.712356 32.426635,
    15.690959 11.987287 32.672866,
    13.092237  9.756517 28.897232,
    14.220060 12.906684 27.340717,
    15.704090 16.940200 30.263877,
    17.718055 17.428257 30.470222,
    19.487106 23.287891 34.520500,
    21.442127 25.636001 37.044532,
    24.951919 30.484500 45.467407,
    26.961414 32.732826 48.174156,
    27.625804 34.344078 56.707367,
    29.802745 33.231643 62.022628,
    30.370506 29.752571 65.508667,
    27.389129 21.525676 60.045471,
    24.136940 18.777145 52.820404,
    21.597133 17.251140 47.556259,
    20.041355 14.616481 42.308754,
    18.967193 16.165267 35.948036,
    18.595157 19.830234 32.583084,
    20.269153 23.141529 37.360000 };
Y=Y`;
Q=1;
P=3;
P2=P*P;
T=25;N=1;
T1=T+1;
DT=.01;
DELTA=1.;
G=I(3);
TH={ .6, 4.0, .4, 2.0, .25 };    *true parameter vector;
RUN PAR ( OMEGA,A,B,SIGMA,MUE,TH);
PRINT A,B,G,OMEGA,MUE,SIGMA,P Q DELTA;
SEED=9753137;
X=SHAPE(5,Q,N*T1);
Y0=-SOLVE(A,B)*X[1,1];        *STATIONARY INITIAL CONDITION;
*OMIT THE NEXT STATEMENT IF YOU WANT TO ANALYZE PHILLIP`S DATA;
```

```
RUN EXAKT(Y, A, B, G, Y0, X, P, Q, N, T, DELTA, SEED, 1);         *SIMULATION OF DATA SET;
RUN MOMENT1(M0, M00, M11, M12, M13, M22, M23, M33, Y, X, P, Q, N, T, 1);

THETA0=TH;        *STARTING ITERATION AT TRUE VALUE;
KMAX=5;
EPS=1E-4;
RUN    NEWTON(THETA1, COV, THETA0, KMAX, EPS,
               M0, M00, M11, M12, M13, M22, M23, M33, DELTA, N, T, 0, 0);   *LINEARIZED MODEL;

RUN    NEWTON(THETA1, COV, THETA0, KMAX, EPS,
               M0, M00, M11, M12, M13, M22, M23, M33, DELTA, N, T, 0, 1);   *NONLINEAR MODEL;

FINISH;
RUN PHILLIPS;
QUIT;
```

Comment:

The module PHILLIPS analyzes a simulated data set (alternatively you can use Phillips` data) with the linearized model (LeBreton 1976) or with the correct nonlinear specification. Since the model is identified (A is restricted, see Phillips, 1973) the Fisher information is p.d. and you can use OPTION=0 in NEWTON.

The following code differs somewhat from the one just discussed since it is assumed that the diffusion matrix Ω contains free parameters (in the diagonal).

```
PROC IML;                                            *begin of iml program;
RESET STORAGE=COMP.LSDE1;                      *Loading library LSDE from disk;
LOAD;
RUN INITLSDE ;                           * running module INITLSDE initializes;
                                                        * module library LSDE;

*****************************************************************************;
START  PHILLIP1;
*****************************************************************************;
*DIAGONALE DIFFUSIONSMATRIX;
*****************************************************************************;
START PAR ( OMEGA, A, B, SIGMA, MUE, TH);
*****************************************************************************;

A=(  -TH[1]    || TH[1]*(1-TH[5])        ||          0        ) //
  (   TH[2]    || TH[2]*(TH[3]*TH[4]-1)  || -TH[2]*TH[3]  ) //
  (      0     || TH[3]*TH[4]            ||       -TH[3]    ) ;
B= (  TH[1] )//
   (     0   )//
   (     0   );

OMEGA=( TH[6] ||    0    ||  0     )//
      (   0   || TH[7]  ||  0     )//
      (   0   ||    0    || TH[8] );
```

```
MUE=  0   //
      0   //
      0   ;
SIGMA=1E4*I(3);

*PRINT '-----------PAR-----------',G OMEGA ;
*PRINT       A,B,TH ;
FINISH;
***********************************************************************;
START PARD (DODTH,DADTH,DBDTH,DSIGDTH,DMUEDTH,TH);
***********************************************************************;
* MC-DONALD-SWAMINATHAN-ABLEITUNGEN ;


DADTH= ( -1 || 1-TH[5] || 0 || 0 ||      0      ||  0   || 0 ||  0  || 0 )//
       (  0 ||    0    || 0 || 1 ||TH[3]*TH[4]-1||-TH[3]|| 0 ||  0  || 0 )//
       (  0 ||    0    || 0 || 0 ||TH[2]*TH[4]  ||-TH[2]|| 0 ||TH[4]||-1 )//
       (  0 ||    0    || 0 || 0 ||TH[2]*TH[3]  ||  0   || 0 ||TH[3]||0  )//
       (  0 ||-TH[1]   || 0 || 0 ||      0      ||  0   || 0 ||  0  || 0 );
DADTH=DADTH//J(3,9,0);
DBDTH={1 0 0,
       0 0 0,
       0 0 0,
       0 0 0,
       0 0 0 };
DBDTH=DBDTH//J(3,3,0);
DODTH=J(5,9,0);
HILTH={ 1 0 0 0 0 0 0 0 0,
        0 0 0 0 1 0 0 0 0,
        0 0 0 0 0 0 0 0 1 };
DODTH=DODTH//HILTH;
DMUEDTH=J(8,3,0);
DSIGDTH=J(8,9,0);

*PRINT '-----------PARD-----------';
*PRINT       DODTH,DADTH,DBDTH,TH ;
FINISH;
***********************************************************************;
Q=1;
P=3;
P2=P*P;
T=25;N=1;
T1=T+1;
DT=.01;
DELTA=1.;
G=I(3);
TH={ .6,  4.0,  .4,  2.0,  .25,  1,  1,   1};
RUN PAR ( OMEGA,A,B,SIGMA,MUE,TH);
PRINT A,B,G,OMEGA,MUE,SIGMA,P Q DELTA;;
SEED=13579;
X=SHAPE(5,Q,N*T1);
YO=-SOLVE(A,B)*X[1,1];  *STATION{RE ANFANGSBEDINGUNG;
RESET STORAGE=PHILLIP.S13579;
FREE THLIN THEX;
REP=100;
```

```
DO L=1 TO REP;

RUN EXAKT(Y, A, B, G, Y0, X, P, Q, N, T, DELTA, SEED, 0);
RUN MOMENT1(M0, M00, M11, M12, M13, M22, M23, M33, Y, X, P, Q, N, T, 0);

KMAX=50;
EPS=1E-4;
RUN    NEWSIM(THETA1, COV, TH, KMAX, EPS,
              M0, M00, M11, M12, M13, M22, M23, M33, DELTA, N, T, 0, 0);

RUN    NEWSIM(THETA, COV, TH, KMAX, EPS,
             M0, M00, M11, M12, M13, M22, M23, M33, DELTA, N, T, 0, 1);
THLIN=THLIN//THETA1`;
THEX=THEX//THETA`;
END;

STORE THLIN THEX;
PRINT THLIN ,THEX;
THLQ=THLIN[:, ];      *ROW MEAN OF LINEARIZED ML ESTIMATORS;
THEXQ=THEX[:, ];      *ROW MEAN OF EXACT ML ESTIMATORS;
REP1=1/(REP-1);
THLS=REP1*(THLIN`*THLIN -REP*THLQ`*THLQ);
THEXS=REP1*(THEX`*THEX - REP*THEXQ`*THEXQ);
STORE THLQ THLS THEXQ THEXS;
PRINT 'MITTELWERTE UND STANDARDFEHLER DER ML-SCH{TZER' THLQ, THLS, THEXQ, THEXS;
FINISH;
RUN PHILLIP1;
```

Comment:

The estimates THETA1 and THETA are collected (in THLIN and THEX) and stored on disk for later usage (analysis using PROC UNIVARIATE (SAS/STATISTICS)). Table II.1 was produced using the output of PHILLIP1.

3. Case study 3: The Sunspot Activity

Chapter II.6 contains a detailed discussion of the analysis of the annual sunspot numbers. Since only the first component of the state vector can be measured (an AR(2)-process was assumed), we need the general state space model including a measurement model (modules PARM and PARMD are required).

3.1 CAR(2) model including errors of measurement

IML Code:

```
PROC IML;                                               *begin of iml program;
RESET STORAGE=COMP.LSDE1 ;                      *Loading library LSDE from disk;
LOAD;
RUN INITLSDE ;                        * running module INITLSDE initializes;
                                                    * module library LSDE;

************************************************************************;
START SUNSPOT;     *including errors of measurement - model II (ch.II.6);
************************************************************************;

************************************************************************;
START PAR ( OMEGA, A, B, SIGMA, MUE, TH);
************************************************************************;

A=( 0    ||  1  )//
  (TH[1]||TH[2]);

B=   0  //
     0  ;

G=(  0   ||  0  )//
  (  0   ||TH[3]);
OMEGA=G*G`;
MUE= 0//
     0;
SIGMA=1E+4*I(2);
*PRINT '-----------PAR-----------',G OMEGA ;
*PRINT       A, B, TH ;
FINISH;
************************************************************************;
START PARD (DODTH, DADTH, DBDTH, DSIGDTH, DMUEDTH, TH);
************************************************************************;
* MC-DONALD-SWAMINATHAN-ABLEITUNGEN ;

DADTH= { 0 0 1 0,
         0 0 0 1,
         0 0 0 0 };
```

```
DBDTH= J(3,2,0);

DODTH=J(2,4,0)//
      (0||0||0||2*TH[3]) ;

DMUEDTH=J(3,2,0);

DSIGDTH= J(3,4,0);

*PRINT '-----------PARD-----------';
*PRINT      DODTH,DADTH,DBDTH,TH ;
FINISH;
**************************************************************************;
START PARM(R,H,D,PHI);
**************************************************************************;
*PARAMETER-MATRIZEN DES ME--MODELLS;
H={ 1 0 };
D=PHI[1];
R=PHI[2];
FINISH;
**************************************************************************;
START PARMD(DR,DH,DD,PHI);
**************************************************************************;
*ABLEITUNGEN DER PARAMETER-MATRIZEN DES ME--MODELLS;
DH={ 0 0,
     0 0 };
DD={ 1,
     0 };
DR={ 0,
     1 };

FINISH;
**************************************************************************;
*DATA FROM ANDERSON (1971);
Z={809 834 477 478 307 122 96 102 324 476 540 629 859 612 451 364 209
   114 378 698 1061 1008 816 665 348 306 70 198 925 1544 1259 848 681
   385 228 102 241 829 1320 1309 1181 899 666 600 469 410 213 160 64
   41 68 145 340 450 431 475 422 281 101 81 25 00 14 50 122 139 354 458
   411 304 239 157 66 40 18 85 166 363 497 625 670 710 478 275 85 132
   569 1215 1383 1032 858 632 368 242 107 150 401 615 985 1243 959 665
   645 542 390 206 67 43 228 548 938 957 772 591 440 470 305 163 73 373
   739 1391 1112 1017 663 447 171 113 123 34 60 323 543 597 637 635 522
   254 131 68 63 71 356 730 849 780 640 418 262 267 121 95 27 50 244
   420 635 538 620 485 439 186 57 36 14 96 474 571 1039 806 636 376 261
   142 58 167};
Z=Z/10;
T0=1749;
XY=((T0:T0+NCOL(Z)-1)//Z )`;
CALL PGRAF(XY,'*','ZEIT','ZAHL DER SONNENFLECKEN','SONNENFLECKEN');
*SCATTER PLOT ON PRINTER, SEE IML USER'S GUIDE;
ZQUER=Z[:];   *MEAN LEVEL;
Z0=Z-ZQUER;
PRINT Z,ZQUER,Z0;
FREE Z0 ZQUER XY;
```

```
K=1; Q=1; P=2;
P2=P*P;  ZP=2*P;
N=1; T=NCOL(Z)-1; T1=T+1;
DT=1.;   *ANNUAL MEASUREMENTS;
X=J(Q, N*T1, 1);
SEED=12;
THETAO={ -1 , -1,   2 };   *INITIAL CONDITION FOR ITERATION;
PHIO= { 46,  1 };

KMAX=1; IMAX=30;   *M STEP IS STOPPED AFTER 1 ITERATION (GEM) - MAXIMALLY 30 EM ITERATIONS;
EPS=1E-4;

RUN EM(THETA, PHI, F, Z, X, K, P, Q, N, T, DT, THETAO, PHIO, IMAX, KMAX, EPS, 1);

theta0=theta; phi0=phi;

KMAX=10;  *MAXIMUM OF 10 FURTHER ITERATIONS (BFGS OR NEWTON-RAPHSON);

*RUN NEWRAPH(THETA, PHI, COV, THETAO, PHIO, KMAX, EPS,
                         Z, X, K, P, Q, N, T, DT, 1, 1)   ;

Run  BFGS(THETA, PHI, COV, THETAO, PHIO, F,
           KMAX, EPS, Z, X, K, P, Q, N, T, DT, 1, 1);

pq=4; *Da  CAR(2)+eps=ARMA(2, 2);
Run  DIAGNOSE(inno, innot, gamma, rho, 30, pq,
              Z, X, K, P, Q, N, T, DT, THETA, PHI, 1);        *DIAGNOSTIC CHECKS;
RUN  GLATT1(YG, PG, YF, PF, LIK,
           Z, X, K, P, Q, N, T, DT, THETA, PHI, 1);        *SMOOTHED TRAJECTORIES OF LATENT VARIABLES;
YGS=YG`; PGS=PG`;
PRINT 'GEGL{TTETE TRAJEKTORIE UND GL{TTFEHLER'  YGS   PGS ;
RESET STORAGE=RFREE.SUNAND;
STORE THETA PHI YG PG Z inno rho;     *STORE MATRICES IN CATALOG RFREE.SUNAND;
FINISH; *SUNSPOT;
RUN SUNSPOT;

QUIT;
```

Comment:

After the model specification (CAR(2) model + errors of measurement) the data are stored in the matrix Z. A scatter plot on the printer is produced using the IML routine PGRAF. Then, the EM algorithm is used to obtain the ML estimates. Since experience shows that the convergence in the vicinity of the maximum is very slow a few iterations using BFGS (or NEWRAPH) are added. After convergence, diagnostic checks are performed (the residuals must be **white noise** if the model is correctly specified). Since the CAR(2) model with errors of measurement corresponds to a discrete time AR(p=2,q=2) model (see Phadke/Wu 1974), the parameter pq=p+q was set equal to 4 and used in the diagnostic procedure DIAGNOSE. Then, the ML estimates, smoothed estimates etc. are stored on disk for later use (for example, plots of the measured data, smoothed latent variables and smoothing errors were produced afterwards using ZEICHNE, see ch.II. , fig. II.2).

3.2. CAR(2) MODEL WITHOUT ERRORS OF MEASUREMENT

The following code specifies an AR(2) model without errors of measurement (i.e. R=0), which corresponds to **model I** in chapter II.6 .

```
PROC IML;                                          *begin of iml program;
RESET STORAGE=COMP.LSDE1 ;               *Loading library LSDE from disk;
LOAD;
RUN INITLSDE ;                     * running module INITLSDE initializes;
                                                   * module library LSDE;

************************************************************************;
START SUNSPOT1;                 *MODEL I - WITHOUT ERRORS OF MEASUREMENT;
************************************************************************;

************************************************************************;
START PAR ( OMEGA,A,B,SIGMA,MUE,TH);
************************************************************************;

A=( 0   ||  1  )//
  (TH[1]||TH[2]);

B=   0  //
     0  ;

G=(  0  ||  0  )//
  (  0  ||TH[3]);
OMEGA=G*G`;
MUE= 0//
     0;
SIGMA=1E+4*I(2);
*PRINT '------------PAR------------',G OMEGA ;
*PRINT       A,B,TH ;
FINISH;
************************************************************************;
START PARD (DODTH,DADTH,DBDTH,DSIGDTH,DMUEDTH,TH);
************************************************************************;
* MC-DONALD-SWAMINATHAN-ABLEITUNGEN ;

DADTH= { 0 0 1 0,
         0 0 0 1,

         0 0 0 0 };

DBDTH= J(3,2,0);

DODTH=J(2,4,0)//
      (0||0||0||2*TH[3]) ;
```

```
DMUEDTH=J(3,2,0);

DSIGDTH= J(3,4,0);

*PRINT '-----------PARD-----------';
*PRINT       DODTH,DADTH,DBDTH,TH ;
FINISH;
***************************************************************************;
START PARM(R,H,D,PHI);
***************************************************************************;
*PARAMETER-MATRIZEN DES ME--MODELLS;
H={ 1 0 };
D=PHI[1];
R=1E-4;            *R IS SET TO A SMALL NUMBER;
FINISH;
***************************************************************************;
START PARMD(DR,DH,DD,PHI);
***************************************************************************;
*ABLEITUNGEN DER PARAMETER-MATRIZEN DES ME--MODELLS;
DH={ 0 0 };
DD={ 1 };
DR={ 0 };

FINISH;
***************************************************************************;


*DATA FROM ANDERSON (1971);
Z={809 834 477 478 307 122 96 102 324 476 540 629 859 612 451 364 209
   114 378 698 1061 1008 816 665 348 306 70 198 925 1544 1259 848 681
   385 228 102 241 829 1320 1309 1181 899 666 600 469 410 213 160 64
   41 68 145 340 450 431 475 422 281 101 81 25 00 14 50 122 139 354 458
   411 304 239 157 66 40 18 85 166 363 497 625 670 710 478 275 85 132
   569 1215 1383 1032 858 632 368 242 107 150 401 615 985 1243 959 665
   645 542 390 206 67 43 228 548 938 957 772 591 440 470 305 163 73 373
   739 1391 1112 1017 663 447 171 113 123 34 60 323 543 597 637 635 522
   254 131 68 63 71 356 730 849 780 640 418 262 267 121 95 27 50 244
   420 635 538 620 485 439 186 57 36 14 96 474 571 1039 806 636 376 261
   142 58 167};
Z=Z/10;
T0=1749;
XY=((T0:T0+NCOL(Z)-1)//Z )`;
CALL PGRAF(XY,'*','ZEIT','ZAHL DER SONNENFLECKEN','SONNENFLECKEN');
ZQUER=Z[:];
Z0=Z-ZQUER;
PRINT Z,ZQUER,Z0;
FREE Z0 ZQUER XY;


K=1;Q=1;P=2;

N=1;T=NCOL(Z)-1;T1=T+1;
DT=1.;
X=J(Q,N*T1,1);
THETA0={ -1 ,-1,  2 };
PHI0= { 46 };
```

```
KMAX=1; IMAX=30;
EPS=1E-4;

RUN EM(THETA, PHI, F, Z, X, K, P, Q, N, T, DT, THETA0, PHI0, IMAX, KMAX, EPS, 1);

theta0=theta; phi0=phi;     *INITIAL CONDITION FOR BFGS;
KMAX=20;

*RUN NEWRAPH(THETA, PHI, COV, THETA0, PHI0, KMAX, EPS,
                     Z, X, K, P, Q, N, T, DT, 1, 1)  ;

Run BFGS(THETA, PHI, COV, THETA0, PHI0, F,
          KMAX, EPS, Z, X, K, P, Q, N, T, DT, 0, 1);

pq=3; *Da CAR(2)=ARMA(2, 1);

Run DIAGNOSE(inno, innot, gamma, rho, 30, pq,
              Z, X, K, P, Q, N, T, DT, THETA, PHI, 1);
RUN GLATT1(YG, PG, YF, PF, LIK,
            Z, X, K, P, Q, N, T, DT, THETA, PHI, 1);
YGS=YG`; PGS=PG`;
PRINT 'GEGL{TTETE TRAJEKTORIE UND GL{TTFEHLER' YGS  PGS ;
RESET STORAGE=RFIX.SUNAND;
STORE THETA PHI YG PG Z inno rho;
FINISH; *SUNSPOT1;
RUN SUNSPOT1;
```

Comment:

The error of measurement covariance matrix R was set equal to a small, fixed value. This is necessary since nonsingularity of R was assumed in the expressions for the likelihood and the score function. The only free parameter of the measurement model is the mean level D=Φ[1]. Since the CAR(2) model without errors of measurement corresponds to a discrete time AR(p=2,q=1) model (see Phadke/Wu 1974), the parameter pq=p+q was set equal to 3 and used in the diagnostic procedure DIAGNOSE.

3.3. CARMA(2,1) MODEL WITHOUT ERRORS OF MEASUREMENT

Phadke/Wu (1974) proposed another model specification which takes into account a moving average term, i.e. the disturbance contains a derivative of the white noise ζ(t):

$$d^2/dt^2\left[S(t)-D\right] + \gamma\, d/dt\left[S(t)-D\right] + \omega_0^2\left[S(t)-D\right] = g\zeta(t) + g_1\dot{\zeta}(t)$$

This model, called CARMA(2,1) can be written in state space form as (see appendix 2)

$$d/dt\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} 0 & -\omega_0^2 \\ 1 & -\gamma \end{bmatrix}\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} + \begin{bmatrix} g \\ g_1 \end{bmatrix}\zeta$$

$$z(t) = \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}$$

The CARMA(2,1) form can be recovered by eliminating the quantity $y_1(t)$ from the above equations. Thus the parameter matrices of LSDE are:

$$A = \begin{bmatrix} 0 & -\omega_0^2 \\ 1 & -\gamma \end{bmatrix}, \qquad B = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \qquad G = \begin{bmatrix} g \\ g_1 \end{bmatrix}$$

$$H = \begin{bmatrix} 0 & 1 \end{bmatrix}, \; D = D, \qquad R = 0$$

and the parameter vectors are: $\Theta = \{-\omega_0^2, -\gamma, g, g_1\}$ and $\Phi = \{D\}$.

IML-Code:

```
PROC IML;                                            *begin of iml program;
RESET STORAGE=COMP.LSDE1;                    *Loading library LSDE from disk;
LOAD;
RUN INITLSDE ;                          * running module INITLSDE initializes;
                                                       * module library LSDE;
******************************************************************************;
START ARMASUN; * CARMA(2,1)-Model - Phadke & Wu (1974) =MODEL III ;
******************************************************************************;

******************************************************************************;
START PAR ( OMEGA,A,B,SIGMA,MUE,TH);
******************************************************************************;

A=( 0    ||th[1])//
  ( 1    ||TH[2]);

B=   0  //
     0  ;

G= th[3]//
   th[4];
OMEGA=G*G`;
MUE= 0//
     0;
SIGMA=1E+4*I(2);
*PRINT '-----------PAR-----------',G OMEGA ;
*PRINT       A,B,TH ;
FINISH;
```

```
***************************************************************************;
START PARD (DODTH, DADTH, DBDTH, DSIGDTH, DMUEDTH, TH);
***************************************************************************;
* MC-DONALD-SWAMINATHAN-ABLEITUNGEN ;

DADTH= { 0 1 0 0,
         0 0 0 1,
         0 0 0 0,
         0 0 0 0 };

DBDTH= J(4,2,0);

DODTH=(           J(2,4,0)              )//
      (2*th[3]||th[4]||th[4]||    0     )//
      (     0  ||th[3]||th[3]||2*th[4]);
DMUEDTH=J(4,2,0);

DSIGDTH= J(4,4,0);

*PRINT '-----------PARD-----------';
*PRINT       DODTH, DADTH, DBDTH, TH ;
FINISH;
***************************************************************************;
START PARM(R,H,D,PHI);
***************************************************************************;
*PARAMETER-MATRIZEN DES ME--MODELLS;
H={ 0 1 };
D=PHI[1];
R=1E-4;
FINISH;
***************************************************************************;
START PARMD(DR,DH,DD,PHI);
***************************************************************************;
*ABLEITUNGEN DER PARAMETER-MATRIZEN DES ME--MODELLS;
DH={ 0 0 };
DD={ 1 };
DR={ 0 };

FINISH;
***************************************************************************;

*DATA FROM ANDERSON (1971);
Z={809 834 477 478 307 122 96 102 324 476 540 629 859 612 451 364 209
   114 378 698 1061 1008 816 665 348 306 70 198 925 1544 1259 848 681
   385 228 102 241 829 1320 1309 1181 899 666 600 469 410 213 160 64
   41 68 145 340 450 431 475 422 281 101 81 25 00 14 50 122 139 354 458
   411 304 239 157 66 40 18 85 166 363 497 625 670 710 478 275 85 132
   569 1215 1383 1032 858 632 368 242 107 150 401 615 985 1243 959 665
   645 542 390 206 67 43 228 548 938 957 772 591 440 470 305 163 73 373
   739 1391 1112 1017 663 447 171 113 123 34 60 323 543 597 637 635 522
   254 131 68 63 71 356 730 849 780 640 418 262 267 121 95 27 50 244
   420 635 538 620 485 439 186 57 36 14 96 474 571 1039 806 636 376 261
   142 58 167};
Z=Z/10;
T0=1749;
XY=((T0: T0+NCOL(Z)-1)//Z )`;
```

```
CALL PGRAF(XY, '*', 'ZEIT', 'ZAHL DER SONNENFLECKEN', 'SONNENFLECKEN');
ZQUER=Z[: ];
ZO=Z-ZQUER;
PRINT Z, ZQUER, ZO;
FREE ZO ZQUER XY;


K=1; Q=1; P=2;
N=1; T=NCOL(Z)-1; T1=T+1;
DT=1.;
X=J(Q, N*T1, 1);
THETAO={ -1 ,-1,  2 , 2};
PHIO= { 46 };

KMAX=1; IMAX=30;
EPS=1E-4;

RUN EM(THETA, PHI, F, Z, X, K, P, Q, N, T, DT, THETAO, PHIO, IMAX, KMAX, EPS, 1);

theta0=theta; phi0=phi;

KMAX=20;

*RUN NEWRAPH(THETA, PHI, COV, THETA, PHI, KMAX, EPS,
                   Z, X, K, P, Q, N, T, DT, 1, 1)  ;

Run BFGS(THETA, PHI, COV, THETAO, PHIO, F,
           KMAX, EPS, Z, X, K, P, Q, N, T, DT, 0, 1);

Run DIAGNOSE(inno, innot, gamma, rho, 30, 3,
             Z, X, K, P, Q, N, T, DT, THETA, PHI, 1);
RUN GLATT1(YG, PG, YF, PF, LIK,
           Z, X, K, P, Q, N, T, DT, THETA, PHI, 1);
YGS=YG`; PGS=PG`;
PRINT 'GEGL{TTETE TRAJEKTORIE UND GL{TTFEHLER' YGS  PGS ;
RESET STORAGE=plot.armasun;
STORE THETA PHI YG PG Z inno rho;
FINISH; *ARMASUN;
RUN ARMASUN;
```

Comment:

For results see table II. 2 and II.3. The model corresponds to a discrete time ARMA(2,1) process (see Phadke/Wu 1974), so the parameter pq was set equal to 3.

3.4. CARIMA(2,1,0) MODEL WITH ERRORS OF MEASUREMENT

As noted in chapter II.6, the number of sunspots S(t) analyzed in the models I,II and III is an annual average,

$$S(t) = (1/\Delta t) \int_{t}^{t+\Delta t} s(u)du \approx s(t) \approx s(t+\Delta t)$$

i.e. it is an integrated quantity. On the other hand, it may be argued that the number of sunspots s(t) at any time point is the basic quantity for which a dynamical model must be formulated. This means:

$$\text{(S)} \qquad d^2/dt^2[s(t)\text{-}D] + \gamma\, d/dt[s(t)\text{-}D] + \omega_0^2[s(t)\text{-}D] = g\zeta(t)$$

$$\text{(M)} \qquad S(t) = (1/\Delta t) \int_{t}^{t+\Delta t} s(u)du$$

The measurement equation (M) does not fit to the scheme provided by LSDE. However, it can be written in the required form by introduction of an integration variable (area under the curve s(t)):

$$I(t) = \int_{t_0}^{t} s(u)\, du$$

We have $S(t)=(1/\Delta t)\left[I(t+\Delta t)\text{-}I(t)\right]$ and $d/dt\, I(t) = s(t)$ with $I(t_0)=0$. This means that s(t) is the time derivative of I(t) and $I(t_i)$, i=1,......T+1, can be calculated from $S(t_i)$, i=0....T, which are the measured data , just by summing the data:

$$I(t_i)= \Delta t \sum_{k=0}^{i-1} S(t_k) \qquad , i=1,.....T+1.$$

We then have the state space form $(\Delta t=1,\ t \in [t_1,\ t_{T+1}]\ )$.

$$d/dt \begin{bmatrix} J \\ y \\ \dot{y} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & -\omega_0^2 & -\gamma \end{bmatrix} \begin{bmatrix} J \\ y \\ \dot{y} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} \zeta$$

$$I(t_i) = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} J \\ y \\ \dot{y} \end{bmatrix} + D\,(t_i\text{-}t_0) + \varepsilon_i \qquad ; \quad i=1\ ,.......T+1$$

where we used the variable $y = s - D$ (deviation from mean value) and the integrals

$$I(t) = \int_{t_0}^{t} s(u)du$$

$$= \int y\,(u)du + \int D du$$

$$:= J(t) + D(t-t_0)$$

The data $I(t_i)$ to be analyzed are the cumulative sums of the original annual averages, i.e.

$$Z = \quad [I(t_1),.....I(t_{T+1})] = \Delta t[S(t_0),\ S(t_0)+S(t_1),........,\ S(t_0)+....+S(t_T)]$$

IML-code:

```
PROC IML;                                          *begin of iml program;
RESET STORAGE=COMP.LSDE1;                     *Loading library LSDE from disk;
LOAD;
RUN INITLSDE ;                            * running module INITLSDE initializes;
                                                      * module library LSDE;

*************************************************************************;
START SUNSPOT3;  *WITH INTEGRATED DATA (FLOWS) AND ERRORS OF MEASUREMENTS;
*************************************************************************;

*************************************************************************;
START PAR ( OMEGA, A, B, SIGMA, MUE, TH);
*************************************************************************;
EPS=1E-6;                          * A SHOULD NOT BE SINGULAR;
A=(EPS||  1    ||  0  )//
  ( 0 ||  0    ||  1  )//
  ( 0 || TH[1]||TH[2]);

B=   0  //
     0  //
     0  ;
T3=TH[3]; T3=T3*T3;
OMEGA=(EPS || 0   ||   0  )//
      (  0 ||EPS ||   0  )//
      (  0 || 0   ||  T3 );
MUE= 0//
     0//
     0;
SIGMA=DIAG({1E+4 1E+4 1E+4});
*PRINT '-----------PAR-----------',G OMEGA ;
*PRINT       A, B, TH ;
FINISH;
*************************************************************************;
START PARD (DODTH, DADTH, DBDTH, DSIGDTH, DMUEDTH, TH);
*************************************************************************;
* MC-DONALD-SWAMINATHAN-ABLEITUNGEN ;

DADTH= { 0 0 0 0 0 0 0 1 0,
         0 0 0 0 0 0 0 0 1,
         0 0 0 0 0 0 0 0 0};

DBDTH= J(3,3,0);

DODTH=J(2,9,0)//
      ({ 0 0 0 0 0 0 0 0}||2*TH[3]) ;
```

```
DMUEDTH=J(3,3,0);

DSIGDTH= J(3,9,0);

*PRINT '-----------PARD-----------';
*PRINT       DODTH,DADTH,DBDTH,TH ;
FINISH;
************************************************************************;
START PARM(R,H,D,PHI);
************************************************************************;
*PARAMETER-MATRIZEN DES ME--MODELLS;
H={ 1 0 0 };
D=PHI[1];
R=PHI[2];
FINISH;
************************************************************************;
START PARMD(DR,DH,DD,PHI);
************************************************************************;
*ABLEITUNGEN DER PARAMETER-MATRIZEN DES ME--MODELLS;
DH={ 0 0 0,
     0 0 0};
DD={ 1,
     0 };
DR={ 0,
     1 };

FINISH;
************************************************************************;

*DATA FROM ANDERSON (1971);
Z={809 834 477 478 307 122 96 102 324 476 540 629 859 612 451 364 209
   114 378 698 1061 1008 816 665 348 306 70 198 925 1544 1259 848 681
   385 228 102 241 829 1320 1309 1181 899 666 600 469 410 213 160 64
   41 68 145 340 450 431 475 422 281 101 81 25 00 14 50 122 139 354 458
   411 304 239 157 66 40 18 85 166 363 497 625 670 710 478 275 85 132
   569 1215 1383 1032 858 632 368 242 107 150 401 615 985 1243 959 665
   645 542 390 206 67 43 228 548 938 957 772 591 440 470 305 163 73 373
   739 1391 1112 1017 663 447 171 113 123 34 60 323 543 597 637 635 522
   254 131 68 63 71 356 730 849 780 640 418 262 267 121 95 27 50 244
   420 635 538 620 485 439 186 57 36 14 96 474 571 1039 806 636 376 261
   142 58 167};
Z=Z/10;
TO=1749;
XY=((TO:TO+NCOL(Z)-1)//Z )`;
CALL PGRAF(XY,'*','ZEIT','ZAHL DER SONNENFLECKEN','SONNENFLECKEN');
ZQUER=Z[:];
ZO=Z-ZQUER;
PRINT Z,ZQUER,ZO;
FREE ZO ZQUER XY;
*INTEGRATED DATA;
T1=NCOL(Z);
Z=CUSUM(Z);        *CUMULATED SUM , SEE IML USER`S GUIDE;
PRINT Z;           * Z are the cumulated data I(t1),......I(tt+1);

K=1;Q=1;P=3;
```

```
N=1; T=T1-1;
DT=1.;
X=1:T1;      * this gives the term D(t-t0) in the measurement model;
THETA0={ -1 ,-1,  2 };
PHI0= { 46,  1 };
EPS=1E-4;

RUN EM(THETA, PHI, F0, Z, X, K, P, Q, N, T, DT, THETA0, PHI0, IMAX, KMAX, EPS, 1);

THETA0=THETA; PHI0=PHI;
KMAX=30;

*RUN NEWRAPH(THETA, PHI, COV, PSI0, KMAX, EPS,
                     Z, X, K, P, Q, N, T, DT, 1, 1)  ;

RUN BFGS(THETA, PHI, COV, THETA0, PHI0, F0,
          KMAX, EPS, Z, X, K, P, Q, N, T, DT, 0, 1);

pq=6; *Da CARI(2, 1)+eps=CARMA(3, 3);

Run DIAGNOSE(inno, innot, gamma, rho, 30, pq,
              Z, X, K, P, Q, N, T, DT, THETA, PHI, 1);
RUN GLATT1(YG, PG, YF, PF, LIK,
             Z, X, K, P, Q, N, T, DT, THETA, PHI, 1);
YGS=YG`; PGS=PG`;
PRINT 'GEGL{TTETE TRAJEKTORIE UND GL{TTFEHLER' YGS  PGS ;
RESET STORAGE=PLOT.SUN3;
STORE THETA PHI YG PG Z inno rho;
FINISH; *SUNSPOT3;
RUN SUNSPOT3;
```

Comment:

The cumulated data were computed using the IML routine CUSUM. Since only one trajectory (time series) is measured we cannot estimate the parameters μ, Σ of the initial condition. They were set μ=0 and $\Sigma=\infty$ (Σ=diag(10^4)), in order to model a uniform initial distribution. The ε-factors used in A and G serve the purpose of avoiding singularity of the matrices (this is assumed in the implemented formulae). The time points to be analyzed are t_1,.......t_{T+1}, since the first datum is available at the end of the year t_0, which begins at t_0 and ends at t_1. Note that we have to distinguish between the continuous time axis and the counting of the years.

The results obtained are (table II.2 continued):

	model	ω_0^2	Υ	g	D	R	T/years
model IV	CARIMA(2,1,0)	-.4326 (.0526)	-.4722 (.1177)	22.2308 (2.7041)	44.9193 (3.9040)	7.6041 (1.9587)	10.2351

Table III.1: *ML estimates and standard errors for sunspot data analyzed as integrated data (flow data) - CARIMA(2,1,0) (Model IV).*

The compensated likelihoods and test statistics reported by DIAGNOSE are:

		succession of model **
Lik	-572.1942	3
AIC	-577.1942	3
SIC/R	-585.1204	3
AD	-596.8633	4
u	5	
ν	176	
P(30)	29.9692	
P`(30)	33.3379	
df	24*	
critical value $\chi^2_{df}(.95)$	36.4151	

Table III.2: *Likelihood and compensated likelihood for model IV (integrated data - CARIMA(2,1,0)).*
**) The model corresponds to a discrete time ARMA(3,3): pq = 6.*
***) Succession of compensated likelihood in comparision to models I-III (table II.3)*

The hypothesis that the residuals are white noise must be kept on an α=5% level. The analysis of compensated likelihood shows that the succession of models according to the maximum compensated likelihood (MCL) is III, II, IVand I, but the Azencott/Dacunha-Castelle criterion gives the order I, III, II, IV. This may be attributed to the fact that the number of parameters is only u=4 in model I and the criterion depends quadratically on u (for comparisions of several criteria see, e.g., de Gooijer et al., 1985).

Case Study 4: Seligman´s Theory of Learned Helplessness

Seligman´s theory of **learned helplessness** is one of the most prominent cognitive theories of depression (cf. Seligman, 1975, Hautzinger, 1981). It claims that certain attributional styles (cognitions), especially under the influence of negative life events, will promote the development of depressive symptoms. The term "helplessness" means in this context, that

"When highly desired outcomes are believed improbable or highly aversive outcomes are believed probable, and the individual expects that no response in his repertoire will change their likelihood, (helplessness) depression results."
(Abramson et al. , 1978, p.68)

In order to test the theory empirically, Seligmans group developed an "Attributional Style Questionaire (ASQ)", which contains social and job situations (with positive and negative outcome), which have to be evaluated according to causal attributions (i.e. are the reasons for the outcome internal/external (to the person), stable/variable and global/specific). Usually, 6 scores are computed:

$$\{\text{success, failure}\} \otimes \{(\text{internal} \longleftrightarrow \text{external}), (\text{stable} \longleftrightarrow \text{variable}), (\text{global} \longleftrightarrow \text{specific})\}$$

and the correlations of these variables with depression scales (e.g. Beck´s depression inventory, BDI) are computed. In fact a negative correlation between the internal/stable/global poles of failures and depression was reported (vice versa for success). These are only cross-sectional results, however, and it may be the case, that the attributions are caused by the depression or are symptoms of it (symptom-hypothesis). There were attempts to improve the test by using longitudinal data (panel data), but the methods of analysis were dubious (e.g. the "cross lag panel correlation method" of Kenny). Others used multivariate regression analysis using LISREL (see Singer/Hautzinger, 1988, for details and references). In Singer (1986) it was proposed to use stochastic differential equations in order to test Seligman´s hypothesis. Here the several variables were interpreted as a time dependent vector y(t) and their change was modeled according to the differential equation

$$dy_n(t) = Ay_n(t)dt + B + Gdw_n(t)$$

where $n = 1,.....,N$ is a person index. The entries A_{ij} of the p×p-matrix A can be interpreted as causal actions of variable y_{nj} on the change of y_{ni}. Furthermore, B is a constant (p×1), which controls the stationary mean ($\mu_s = -A^{-1}B$) and G is the diffusion coefficient which represents the strength of random disturbances. The parameters in A, B and G were estimated with data collected in a **two wave panel** ($N = 206$, $\Delta t = 1$ month). In the following analysis the 6 variables y(t) =[self-esteem, Bf, depression, success, failure, life events] were analyzed (success and failure are sum scores over three attributional scales, respectively).

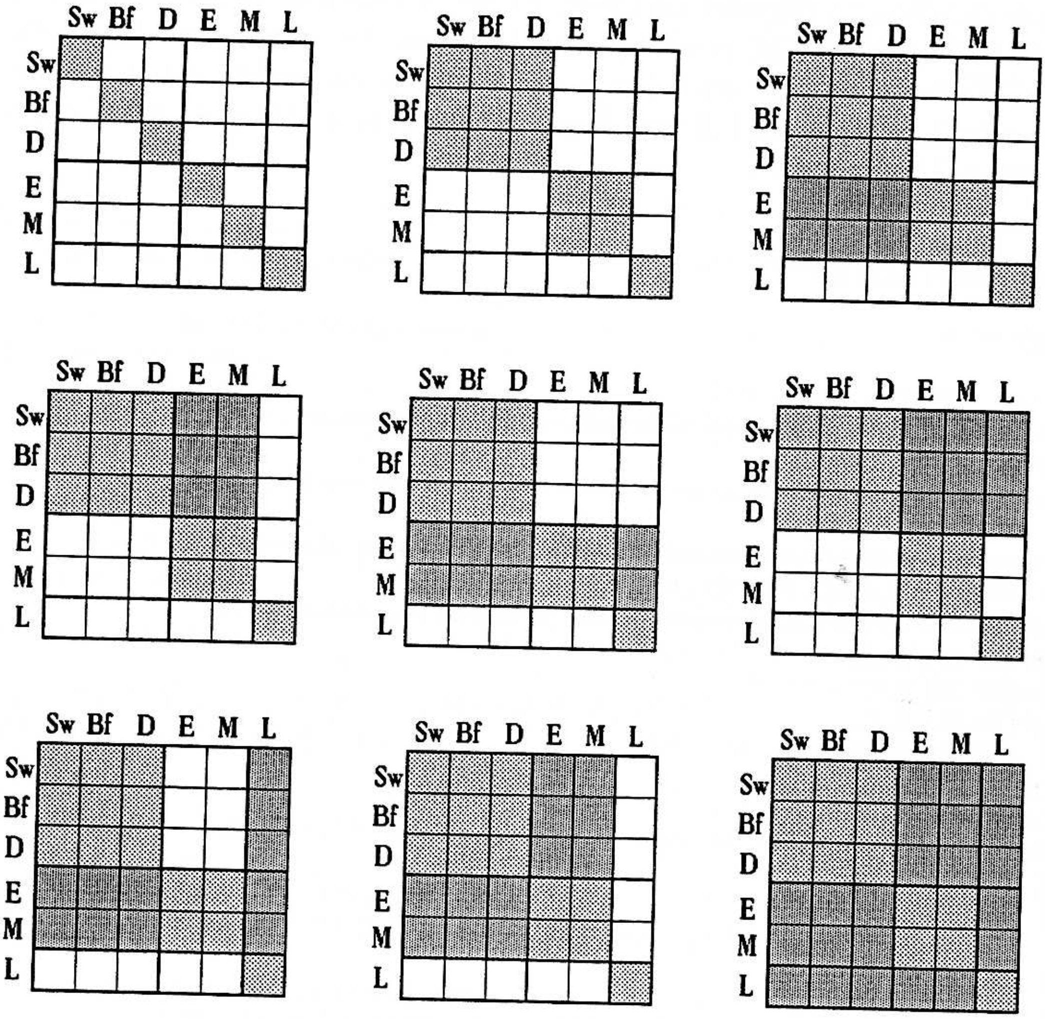


Fig. III.2 *Learned Helplessness. Several competing hypotheses in graphical representation. The squares represent elements $A_{ij}$of the structural matrix A. Elements with free parameters are shaded. The hypotheses are (from the left above): diagonal, block-diagonal, symptom, helplessness, symptom/stress I, diathesis/stress, symptom/stress II, interaction and full model. The elements of A are rate constants of the* ***change*** *of the variable y_i under the influence of variable y_j.*

In imposing null restrictions on elements of A several hypotheses can be tested (see fig. III.2). For example, the helplessness hypothesis corresponds to a causal action of success (E = Erfolg) and failure (M = Mißerfolg) on the variables self-esteem (Sw = Selbstwert), Bf. (Bf = Befindlichkeit) and depression (D = Depression). Since the matrix A contains 6×6 = 36 elements theoretically there are $2^{36} = 6.87 \times 10^{10}$ different hypotheses. This clearly shows that only several carefully selected hypotheses can be tested. These are listed in fig. III.2 and represent several independence, learned helplessness, diathesis-stress and symptom models. Since the fully parametrized model is not identified, a diagonal restriction for the influence of errors was used (G = diagonal). This means, that every equation is driven by a seperate white noise disturbance. Furthermore, the vector B was not restricted and estimation is conditional on the initial values (μ and Σ were fixed). The following IML-code specifies the symptom-stress II-hypothesis:

```
PROC IML;                                                             *begin of iml program;
RESET STORAGE=COMP.LSDE1 ;                                      *Loading library LSDE from disk;
LOAD;
RUN INITLSDE ;                                             * running module INITLSDE initializes;
                                                                        * module library LSDE;

************************************************************************;
START hilflos;
************************************************************************;

************************************************************************;
START PAR ( OMEGA,A,B,SIGMA,MUE,TH) global(sigmaols,mueols);                  *global variables;
************************************************************************;
DD=shape(th[13:21],3,3);
AA=shape(th[22:25],2,2);
LL=shape(th[26:26],1,1);
DA=shape(    0     ,3,2);                                                * no action of A on D;
DL=shape(th[35:37],3,1);
AD=shape(th[27:32],2,3);
AL=shape(th[33:34],2,1);
LD=shape(    0     ,1,3);                                                * no action of D on L;
LA=shape(    0     ,1,2);                                                * no action of A on L;

A=( DD ||  DA  || DL )//
  ( AD ||  AA  || AL )//
  ( LD ||  LA  || LL );
B=shape(th[7:12],6,1);

OMEGA=diag(th[1:6]##2);                                              *diagonal diffusion matrix ;

mue=mueols;
sigma=sigmaols;
*print a,b,omega;
FINISH;
************************************************************************;
START PARD (DODTH,DADTH,DBDTH,DSIGDTH,DMUEDTH,TH);
************************************************************************;
run PARD1(DODTH,DADTH,DBDTH,DSIGDTH,DMUEDTH,TH);                           *numerical derivatives;
finish;
************************************************************************;
 START PARM(R,H,D,PHI);
************************************************************************;
R=diag(j(1,6,1e-4));
D=j(6,1,0);
H=I(6);
finish;
************************************************************************;
```

```
use sassy.hilflos;                                                        * use SAS data set;
read all  var ('v299':'v318') into y;      *variables 299-318 are read into data matrix y;
run casedel(yout,y,.);                             *casewise deletion of missing variables;
n=nrow(yout);                                                           *number of cases;
Q=1;
P=10;
P2=P*P;
T=1;
T1=T+1;
run datstruk(y,yout,n,t1||p,3,{ 2 1});     *restructuring data matrix yout (N×t1*p) ;
                                                                        to y (p×t1*n);
free yout;                                                              *free matrix yout;
*print y,yout;
y=y[{1 2 9 3 4 5 6 7 8 10}, ];                        *interchange variables (permutation);
       *the new variables are:  1=SW, 2=Bf, 3=D, 4=IE, 5=IM, 6=SE, 7=SM, 8=GE, 9=GM, 10=LE;

erfolg=y[{4 6 8}, ][+, ];                                    *compute sum scores (success);
misser=y[{5 7 9}, ][+ ,];                                    *compute sum scores (failure);
y=y[1:3, ]//erfolg//misser//y[10, ];     *laminate parts of data matrix y with success and
                                                                          failure score;
p=6;                                                                     *6 new variables;
X=SHAPE(1,Q,N*T1);                             *exogenous dummy variable, because B = B*X;
DELTA=1.;                                                 *time difference of panel waves;
RUN MOMENT1(MO,MOO,M11,M12,M13,M22,M23,M33,Y,X,P,Q,N,T,1);
MUEOLS=MO; SIGMAOLS=MOO-MO*MO`;                     *mean and variance of initial condition;
KMAX=30;
EPS=1E-4;
theta0=.01#rannor(j(37,1,97531));                    *initial condition of scoring iteration;

*RUN    NEWTON(THETA1,COV,THETAO,KMAX,EPS,
              MO,MOO,M11,M12,M13,M22,M23,M33,DELTA,N,T,0,0);

RUN    NEWTON(THETA,COV,THETAO,KMAX,EPS,
              MO,MOO,M11,M12,M13,M22,M23,M33,DELTA,N,T,0,1); *Maximum Likelihood estimation
                                                                          of parameters;
reset storage=hilflos.par;                                            *set storage catalog;

store theta;                                                   *store ML estimator on disk;
*Interpolation (optimal smoothing) in intervall [0,10];
miss=.;                                                          *missing data are dots;
dt=.1;                                                   *discretization interval dt=0.1;
T=100;                              *the interval [0,10] is divided in 10/.1 = 100 intervals;
T1=T+1;
N=5;                             *the optimal estimates (smoothed latent data) are calculated
                                                                        for 5 persons;
X=SHAPE(1,Q,N*T1);
z=j(6,N*T1,.);                               *data matrix z is initialized with missing values;
z[ ,tree({ 1 11 },1:N,T1,N)]=y[ ,tree(1:2,1:N,2,206)];
                                                    *measured data are inserted into matrix z;
free y;
phi=0;                                                                          * dummy;
run MGLATT1(YG,PG,YF,PF,LIK,Z,X,K,P,Q,N,T,DT,THETA,PHI,1,.);
*optimal filtering and smoothing of latent variables in the interval [0,10] = [0,100×.1];
```

```
print z,yg,pg,yf,pf;
store yg pg yf pf p n t dt;                *store smoothed and filtered estimates on disk;

FINISH;
RUN hilflos;
```

The output produced by the above program is listed now:

```
............
...........

 NEWTON:ENDE DER ITERATION

 K           COL1

 ROW1          29


 KMAX        COL1

 ROW1          30

 ML-Sch{tzer, SCORE UND LIKELIHOOD


              THETA     SCORE

 ROW1   -4.798053 -2.352E-7
 ROW2   -8.930918 -4.573E-7
 ROW3   -12.97139 -1.739E-6
 ROW4   -13.12681 3.5046E-8
 ROW5   -12.95958  2.189E-6
 ROW6   -2.740212 -3.846E-7
 ROW7   -11.84569 1.396E-11
 ROW8     23.4375 2.944E-11
 ROW9   -78.56919 -2.01E-11
 ROW10  30.947444 2.849E-11
 ROW11  -56.08759 -4.42E-12
 ROW12  17.588088     -4E-12
 ROW13  -0.335278 8.5784E-7
 ROW14  0.3891408 -1.852E-6
 ROW15  -0.000833 -1.289E-6
 ROW16  0.4106704 -1.263E-6
 ROW17  -1.447265 -1.903E-6
 ROW18  0.2211672 -0.000017
 ROW19  0.8971813 -3.846E-6
 ROW20  2.1786159 2.2695E-6
 ROW21  -2.599951 -5.503E-6
 ROW22  -0.395306 7.3074E-6
 ROW23  0.2957672 -3.056E-6
 ROW24  0.3978355 0.0000152
 ROW25  -0.429478 6.2438E-6
 ROW26  -0.625216 5.4359E-7
 ROW27  -0.443333 1.6665E-7
 ROW28  -0.254468 -3.487E-6
 ROW29  -0.101163 -0.000024
 ROW30   0.349569 -4.104E-7
 ROW31  1.0896349 3.1684E-7
 ROW32  -0.639513 -0.000017
 ROW33  0.2723174 6.6301E-7
 ROW34  1.0913616 6.5918E-7
 ROW35  0.1860875 -1.053E-6
 ROW36  0.3741355 3.8829E-7
 ROW37  2.4832609 5.0078E-6


 LIKELI                 COL1

 ROW1    -5674.232600539530

 -----------WALD-TEST-----------
```

```
            CHI           U          P      ALPHA

ROW1   1611.572          37          0       0.05

NULLHYPOTHESE THETA=0 ABLEHNEN

-----------BONFERRONI-KONFIDENZINTERVALLE-----------

P{THETA AUS [THDACH-GRENZE,THDACH+GRENZE] } »= 1-ALPHA


     Z(alpha/2u)

ROW1   3.2048452


           THETA    GRENZE         STD

ROW1   -4.798053 1.0008311 0.3122869
ROW2   -8.930918 3.0259731  0.944187
ROW3   -12.97139 5.3474739 1.6685592
ROW4   -13.12681 2.7668422  0.863331
ROW5   -12.95958 2.8175745 0.8791609
ROW6   -2.740212 0.5766127  0.179919
ROW7   -11.84569  19.69212 6.1444841
ROW8     23.4375 43.772822 13.658327
ROW9   -78.56919 68.968816 21.520171
ROW10 30.947444 61.286401 19.123045
ROW11 -56.08759 61.230486 19.105599
ROW12 17.588088 10.101842  3.152053
ROW13 -0.335278 0.3384742 0.1056133
ROW14 0.3891408 0.4901249 0.1529325
ROW15 -0.000833 0.4346683 0.1356285
ROW16 0.4106704 0.6795947 0.2120523
ROW17 -1.447265 1.3229115 0.4127848
ROW18 0.2211672 0.9689206 0.3023299
ROW19 0.8971813 1.3251377 0.4134795
ROW20 2.1786159 1.8470526 0.5763313
ROW21 -2.599951 1.9821776  0.618494
ROW22 -0.395306 0.3435936 0.1072107
ROW23 0.2957672  0.296127 0.0923998
ROW24 0.3978355 0.3153883 0.0984099
ROW25 -0.429478 0.3190641 0.0995568
ROW26 -0.625216 0.3491802 0.1089538
ROW27 -0.443333  0.820563 0.2560383
ROW28 -0.254468 1.1854095 0.3698804
ROW29 -0.101163 1.0000318 0.3120375
ROW30  0.349569 0.8357467  0.260776
ROW31 1.0896349 1.2531926 0.3910306
ROW32 -0.639513 1.0621632 0.3314242
ROW33 0.2723174 1.6332793 0.5096281
ROW34 1.0913616 1.6840411 0.5254672
ROW35 0.1860875 0.6177095 0.1927424
ROW36 0.3741355 1.2335186 0.3848918
ROW37 2.4832609 2.3075366  0.720015

................


KOMPENSIERTE LIKELIHOOD

U          COL1

ROW1         37


NUE        COL1

ROW1        412

 AKAIKE        SCHWARZ/RISSANEN    AZENCOTT&DACUNHA-CASTELLE
```

```
KET               LIK          AIC     SIC/RISS           AD

KOMP           0.0000      37.0000     111.3889    1263.3279
KOMPLIK    -5674.2326   -5711.2326   -5785.6215   -6937.5605


---------------DELTA-THEOREM----------------------


GESCH{TZTE STANDARDABWEICHUNGEN DER SYSTEM-MATRIZEN
A, B, OMEGA, SIGMA UND MUE


A         COL1      COL2      COL3      COL4      COL5      COL6

ROW1 -0.335278 0.3891408 -0.000833         0         0 0.1860875
ROW2 0.4106704 -1.447265 0.2211672         0         0 0.3741355
ROW3 0.8971813 2.1786159 -2.599951         0         0 2.4832609
ROW4 -0.443333 -0.254468 -0.101163 -0.395306 0.2957672 0.2723174
ROW5  0.349569 1.0896349 -0.639513 0.3978355 -0.429478 1.0913616
ROW6         0         0         0         0         0 -0.625216


STDA      COL1      COL2      COL3      COL4      COL5      COL6

ROW1 0.1056133 0.1529325 0.1356285         0         0 0.1927424
ROW2 0.2120523 0.4127848 0.3023299         0         0 0.3848918
ROW3 0.4134795 0.5763313  0.618494         0         0  0.720015
ROW4 0.2560383 0.3698804 0.3120375 0.1072107 0.0923998 0.5096281
ROW5  0.260776 0.3910306 0.3314242 0.0984099 0.0995568 0.5254672
ROW6         0         0         0         0         0 0.1089538


B         COL1

ROW1 -11.84569
ROW2   23.4375
ROW3 -78.56919
ROW4 30.947444
ROW5 -56.08759
ROW6 17.588088


STDB      COL1

ROW1 6.1444842
ROW2 13.658323
ROW3 21.520173
ROW4  19.12304
ROW5   19.1056
ROW6 3.1520522


OMEGA      COL1      COL2      COL3      COL4      COL5      COL6

ROW1   23.02131         0         0         0         0         0
ROW2          0 79.761304         0         0         0         0
ROW3          0         0 168.25687         0         0         0
ROW4          0         0         0  172.3132         0         0
ROW5          0         0         0         0 167.95082         0
ROW6          0         0         0         0         0 7.5087623


STDO      COL1      COL2      COL3      COL4      COL5      COL6

ROW1  2.996738         0         0         0         0         0
ROW2         0 16.864914         0         0         0         0
ROW3         0         0 43.287053         0         0         0
ROW4         0         0         0 22.665569         0         0
ROW5         0         0         0         0 22.787118         0
ROW6         0         0         0         0         0 0.9860327
```

```
MUE          COL1

ROW1 21.349515
ROW2 35.495146
ROW3 34.791262
ROW4 111.50971
ROW5 98.742718
ROW6 29.165049

STDMUE        COL1

ROW1             0
ROW2             0
ROW3             0
ROW4             0
ROW5             0
ROW6             0

SIGMA       COL1         COL2        COL3         COL4         COL5        COL6

ROW1  33.460364   16.97257 30.903054  -21.99368 2.9394382 2.2869733
ROW2   16.97257  34.395608 27.448016  -13.99995 -1.493967  1.379442
ROW3  30.903054  27.448016 76.223419  -19.57322 5.8443539 5.2820247
ROW4  -21.99368  -13.99995 -19.57322  167.90039 73.150556 -2.826845
ROW5  2.9394382  -1.493967 5.8443539  73.150556 184.55031  0.688095
ROW6  2.2869733   1.379442 5.2820247  -2.826845  0.688095   6.11839

STDSIG        COL1        COL2        COL3        COL4        COL5        COL6

ROW1             0           0           0           0           0           0
ROW2             0           0           0           0           0           0
ROW3             0           0           0           0           0           0
ROW4             0           0           0           0           0           0
ROW5             0           0           0           0           0           0
ROW6             0           0           0           0           0           0
............
```

Comment:

The ML-estimates of the 37 free parameters in Θ are obtained after 29 iterations of the scoring algorithm implemented in NEWTON. The accuracy was $||s(\Theta_{29})||_{\infty} < 10^{-4}$ and $||\Theta_{29} - \Theta_{28}|| < 10^{-4}$. In comparision to other models, the symptom/stress II-hypothesis exhibits the highest compensated likelihood according to the AIC-criterion (cf. Singer, 1992, chapter 5.4). The optimal estimates of the latent variables $y_n(t)$ for $n = 1,.......,5$ and $t \in [0,10]$, $dt = 0.1$ were computed and stored on disk in the storage catalog HILFLOS.PAR.

The following program shows how the estimated trajectories can be plotted using the module ZEICHNE, which utilizes the graphics facilities in IML:

```
%INCLUDE GOINIT;
PROC IML;                                                  *begin of iml program;
RESET STORAGE=COMP.LSDE1 ;     *Loading module "zeichne" from module library LSDE;
LOAD module=(zeichne);
```

```
*************************************************************************;
START hilflos;
*************************************************************************;
RESET STORAGE=hilflos.par;
LOAD;
p=6; N=5; T=100;
ZEIT=do(0, 10, .1);

 RUN ZEICHNE(YF, ZEIT, p, N, T, 'TIME',
           'Filtered trajectory',
           'Learned Helplessness', 0);

 RUN ZEICHNE(YG, ZEIT, p, N, T, 'TIME',
           'Smoothed trajectory',
           'Learned Helplessness', 0);


FINISH;
*************************************************************************;

run hilflos;
QUIT;
```

Comment:
Only the module ZEICHNE is loaded from the module library in order to save machine storage (graphics need much storage). It is always recommended to store data in a storage catalog and use the graphics facilities in a separate run. The results of this small program are given in fig. III. 3. The filtered and smoothed estimates are shown for the five persons and for each of the 6 variables. Each plot contains the trajectories of the n = 5 persons. The measurements taken at t = 0 and t = 1 lead to a discontinuous change of the optimal extrapolation due to new information. After t = 1 (2. panel wave) no measurements are available any more and the trajectories tend to the asymptotic stationary values $y(t \to \infty) = -A^{-1}B$.
In contrast, the smoothed trajectories $E[y(t)|y(0), y(1)]$ always use the full information available from the past and the future, so that no discontinuous change occurs. However, since after the time t = 1 no information is available, the filtered and smoothed trajectories agree in the interval [1,10]. Furthermore, there is a complicated dependence between the movement of all 6 curves, since they interact due to the entries in the structural matrix A.
This example shows how the continuous time parameters of a stochastic differential equation can be estimated from discrete data (panel data) and how the unobserved scores between the measurements can be reconstructed using Kalman filter and smoothing techniques.

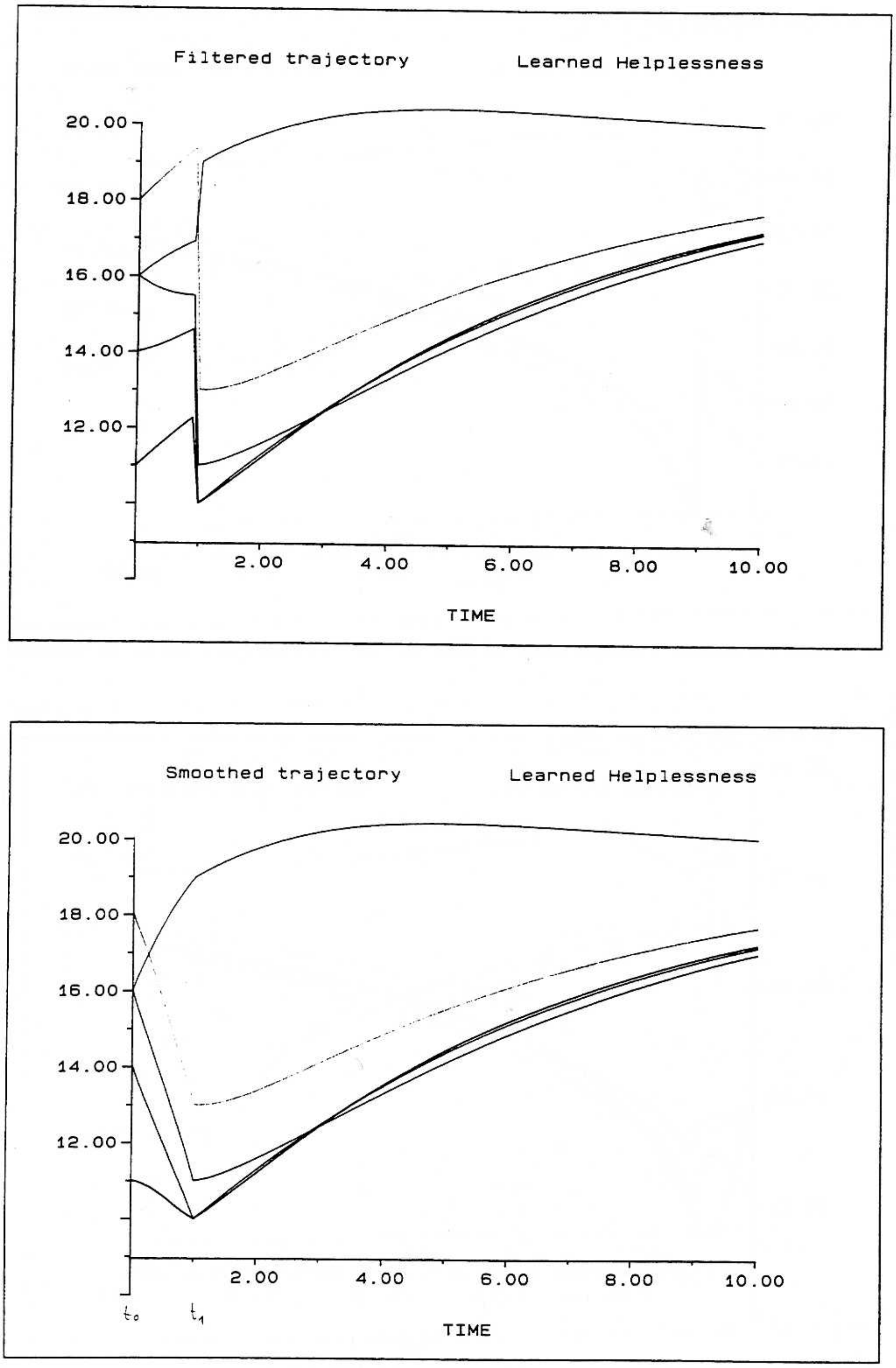


Fig. III.3 a *Learned helplessness. Filtered and smoothed trajectories of self esteem in the interval {0,10} for 5 persons. Measurements are taken at t = 0 and t = 1.*

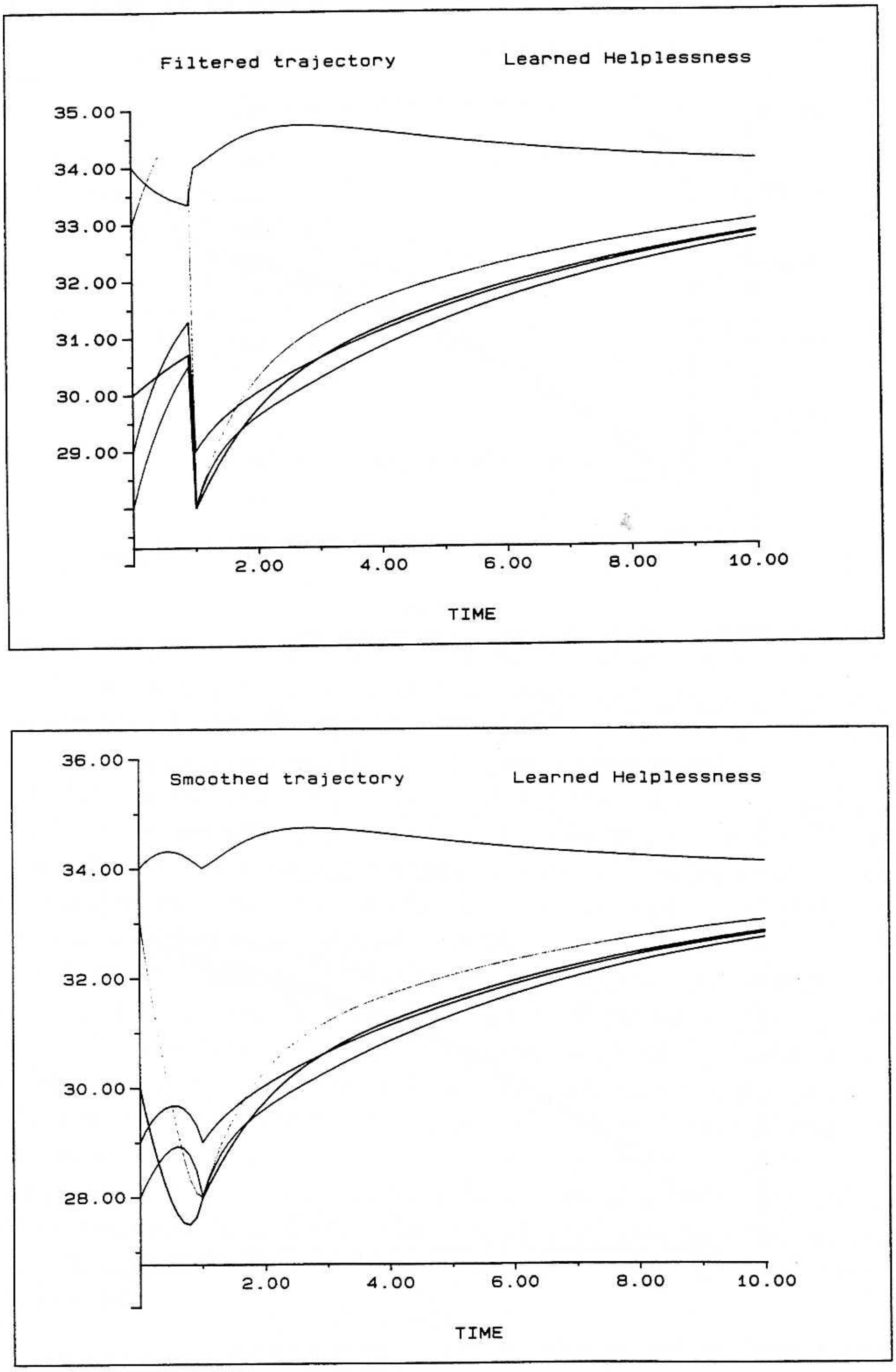
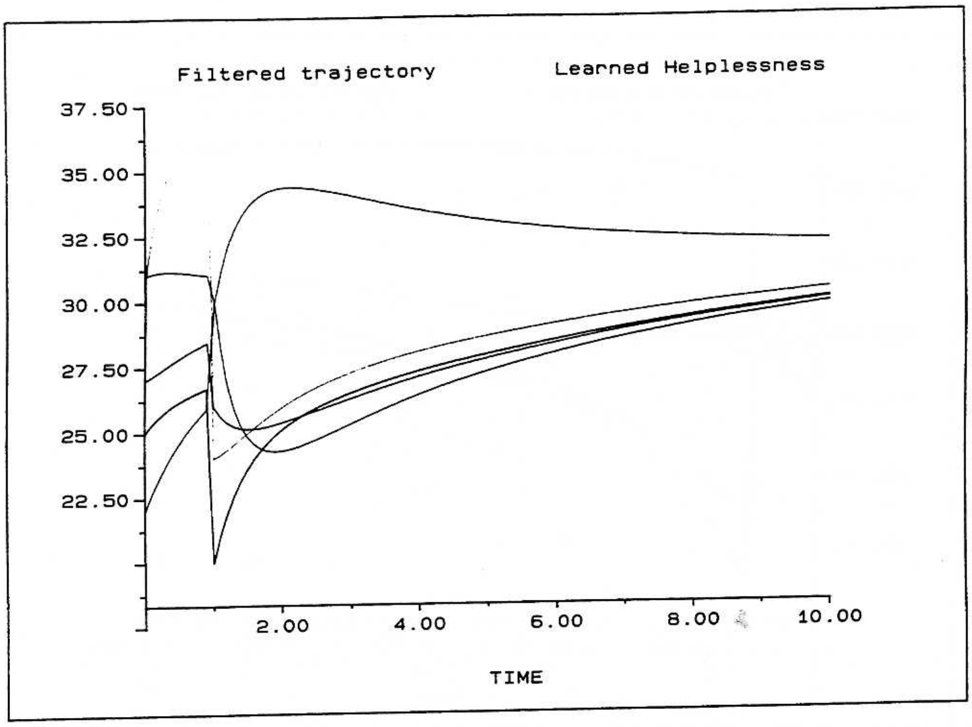


Fig. III.3 b *Learned helplessness. Filtered and smoothed trajectories of "Befindlichkeit" in the interval {0,10} for 5 persons. Measurements are taken at t = 0 and t = 1.*



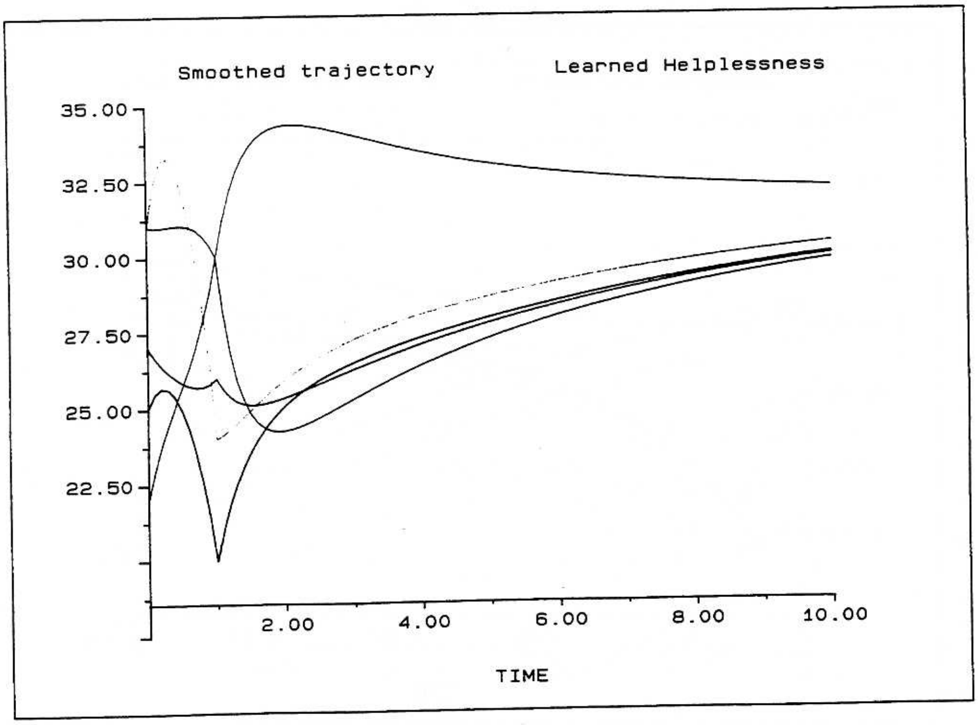
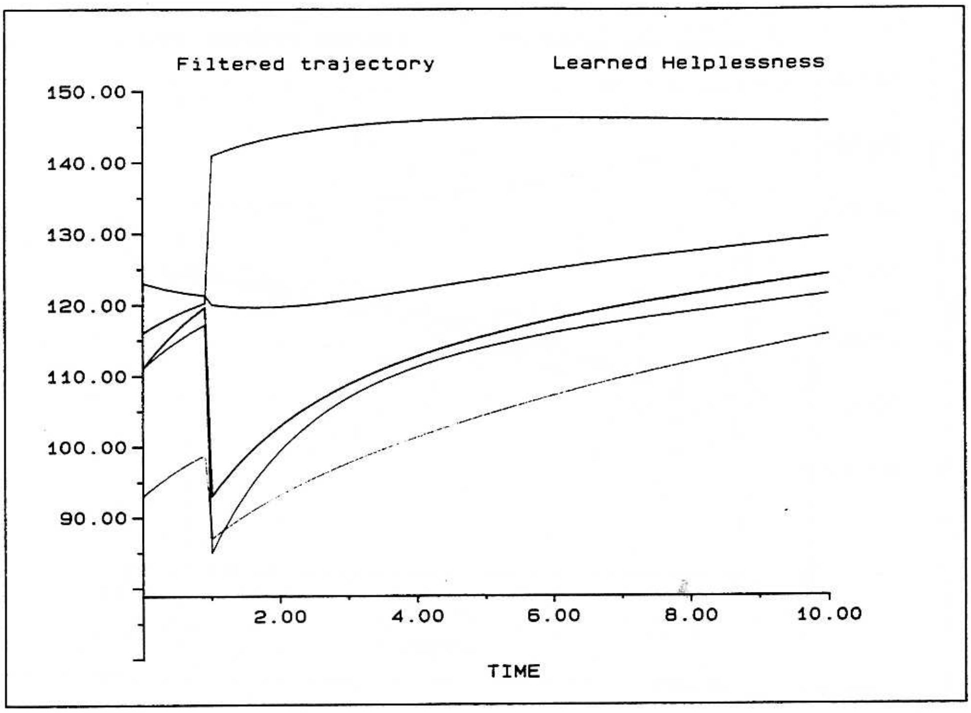


Fig. III.3 c *Learned helplessness. Filtered and smoothed trajectories of depression in the interval {0,10} for 5 persons. Measurements are taken at t = 0 and t = 1.*



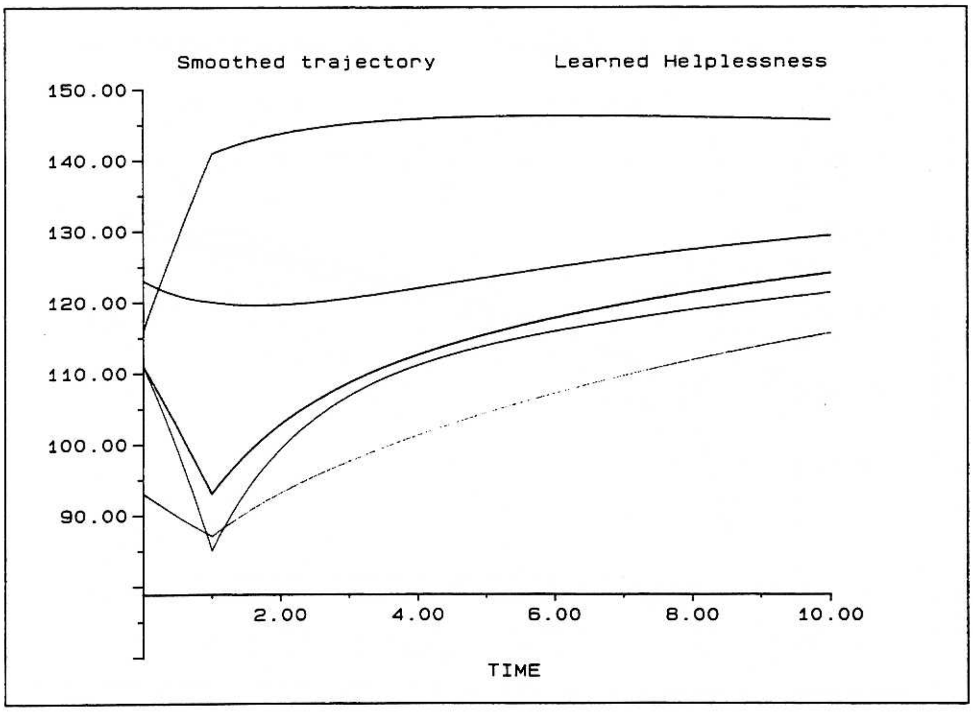
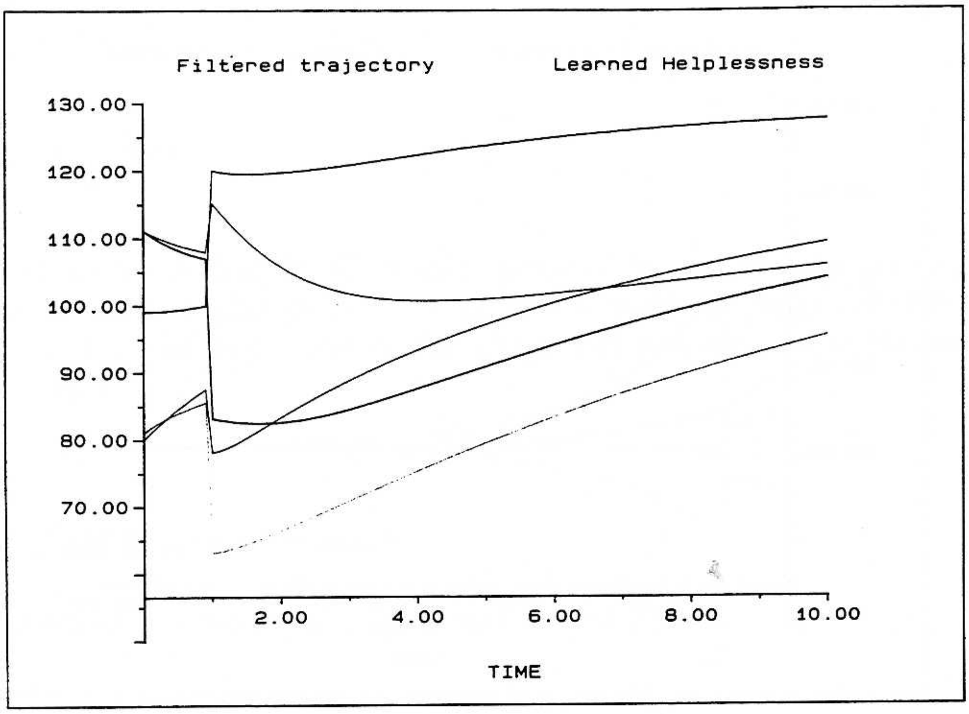


Fig. III.3 d *Learned helplessness. Filtered and smoothed trajectories of success in the interval {0,10} for 5 persons. Measurements are taken at t = 0 and t = 1.*



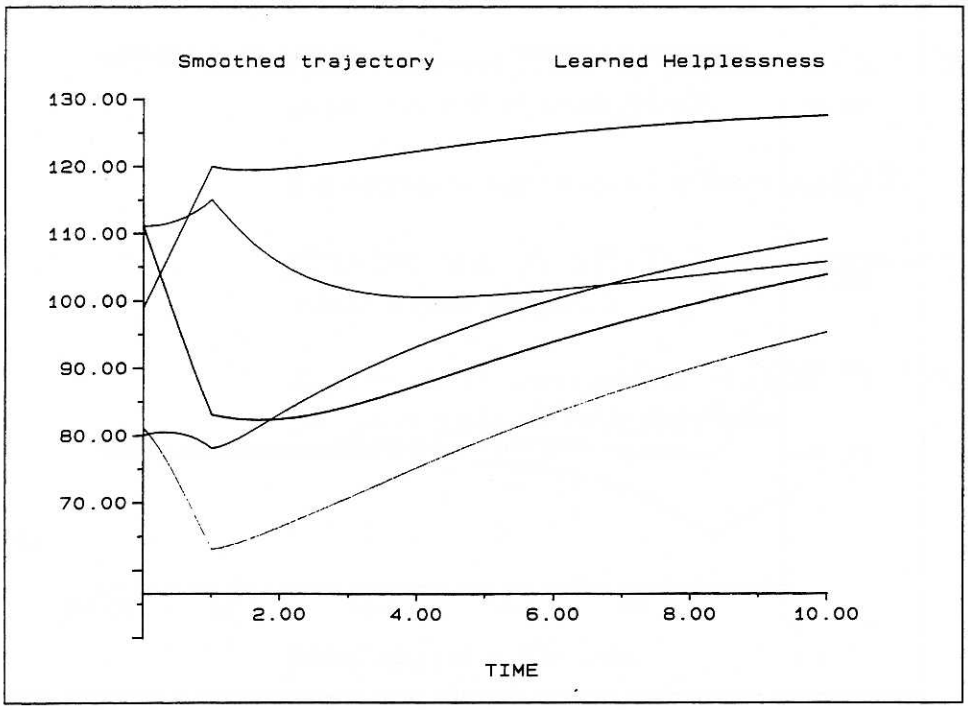
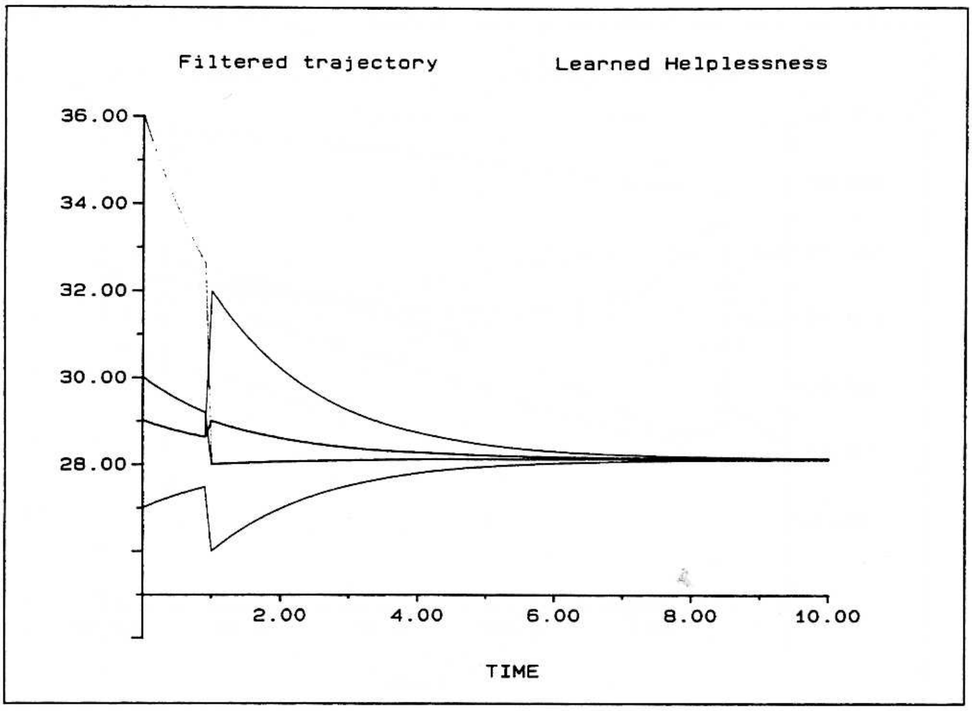


Fig. III.3 e *Learned helplessness. Filtered and smoothed trajectories of failure in the interval {0,10} for 5 persons. Measurements are taken at t = 0 and t = 1.*



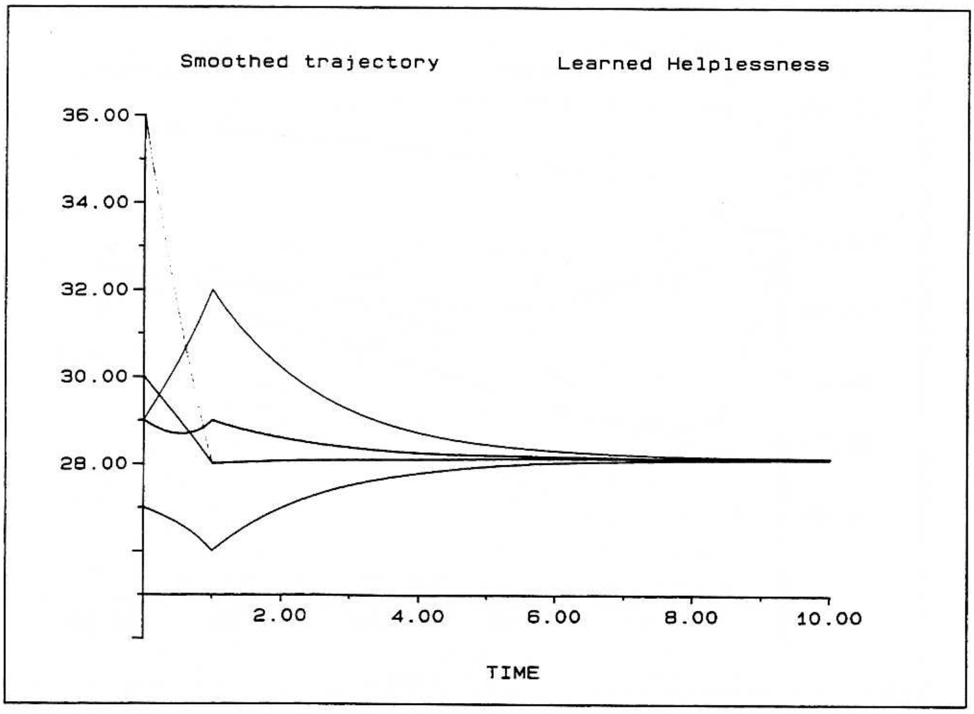


Fig. III.3 f *Learned helplessness. Filtered and smoothed trajectories of life events in the interval {0,10} for 5 persons. Measurements are taken at t = 0 and t = 1.*

IV. REFERENCE

1. Introduction

Chapter IV serves to describe all modules provided by the program package LSDE. It is *not* indended to give an introduction to the matrix language SAS/IML and its possibilities. Here we must refer to the Usage and Reference guide Version 6 (SAS Institute , 1989) .

1.1 List of LSDE supplied modules

The modules can be seperated roughly into several groups:

Model specification

PAR	System matrices A, B, Ω, Σ and μ as functions of parameter vector Θ (**user supplied**).
PARD	Derivatives of system matrices (**user supplied**).
PARM	Measurement matrices H, D, R as functions of parameter vector Φ (**user supplied**).
PARMD	Derivatives of measurement matrices as functions of parameter vector Φ (**user supplied**).

Simulation

EXAKT	Simulation of panel data.
STATESPA	Simulation of panel data with measurement model.
ZEITREIH	Simulation of discrete time panel data (see appendix).

Maximum Likelihood Estimation

NEWTON	ML parameter estimation of stochastic differential equations (scoring algorithm).
NEWSIM	for simulation studies (without print commands).
MOMENT1	Calculation of moment matrices from raw data for usage in NEWTON.
EM	EM algorithm for state space model (equ. II.1, II.2).
EM1	EM algorithm for state space model (fixed measurement model).
NEWRAPH	Newton-Raphson-algorithm for state space model.
NEWRAPH1	Newton-Raphson-algorithm for state space model (fixed measurement model).
BFGS	Broyden-Fletcher-Goldfarb-Shanno-algorithm for state space model.
BFGS1	Broyden-Fletcher-Goldfarb-Shanno-algorithm for state space model (fixed measurement model)

Kalman Filtering and Smoothing (Estimation of latent variables)

FILTER	Kalman-Filter-Algorithm (forward filter). Calculates the likelihood functon as well.
GLATT1	Kalman filter and Rauch-Tung-Striebel smoother. Optimal estimates of latent variables given measured (manifest) data.
MGLATT1	Kalman-filter and Rauch-Tung-Striebel smoother. Optimal estimates of latent variables with missing data.
NORKORR	Normal correlation. Conditional mean estimates of latent variables given data.

Graphics

ZEICHNE Plot of data matrices over time (time paths).

PHASE Plot of data matrices in phase (state) space.

Mixed

INITLSDE Initialisation of module library LSDE.

EIGEN1 Eigenvalues and eigenvectors of asymmetric matrix (possibly complex)

EXP01 Matrix exponential function

TREE Tree structure. Multi indexing for IML arrays.

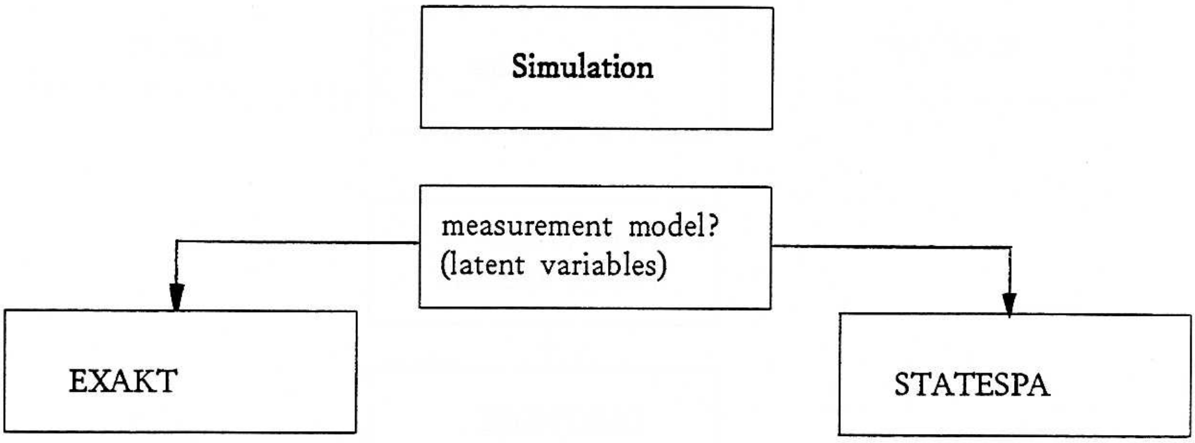
DATSTRUK Restructuring of data matrix.

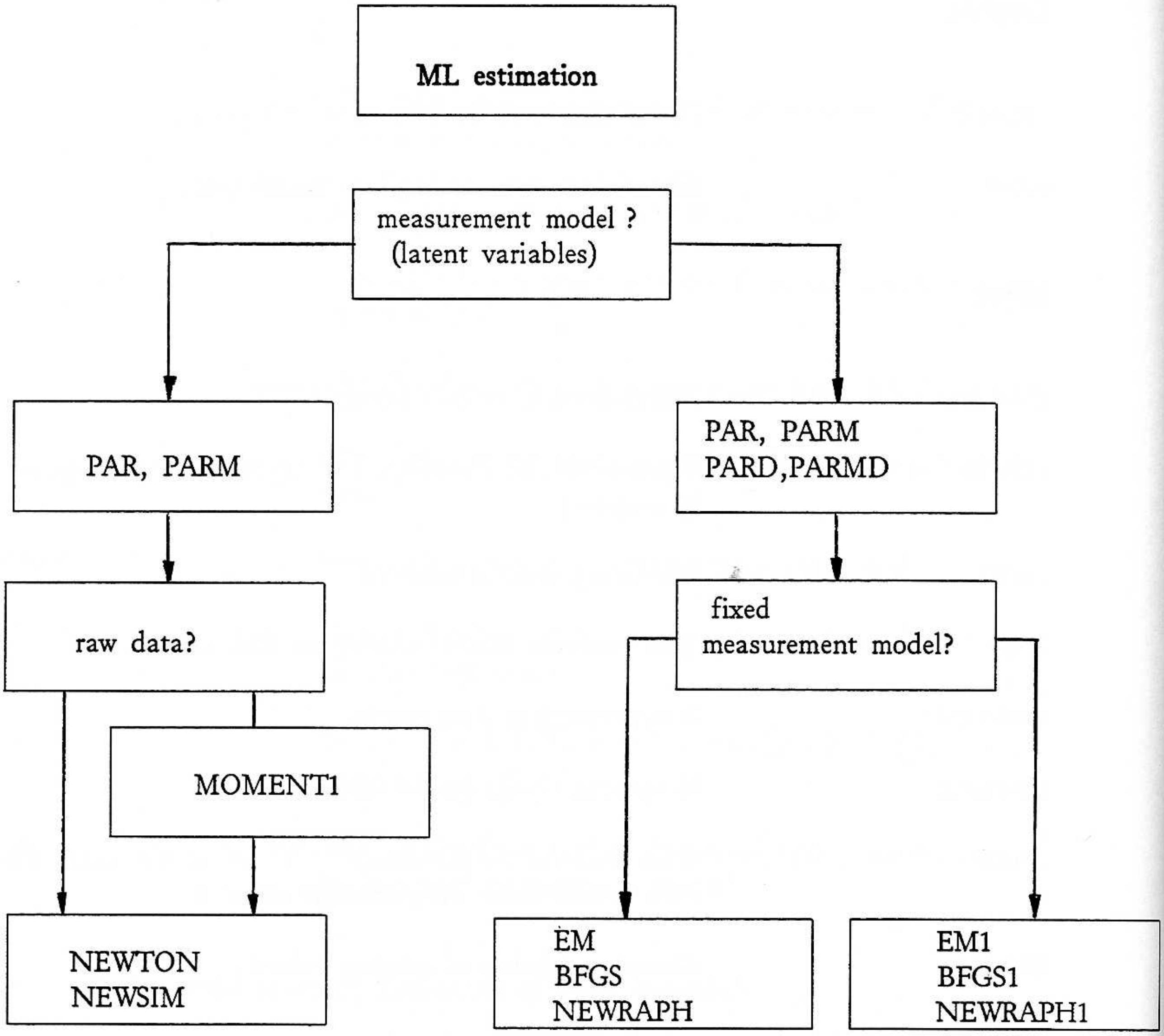
DIAGNOSE Diagnostic checks (model identification).

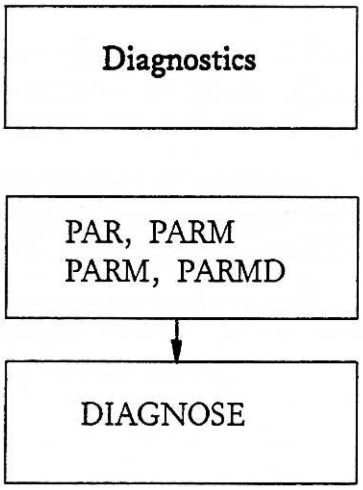
STERN Calculation of the matrices A^*, B^*, Ω^* of the exact discrete model from the parameter vector Θ.

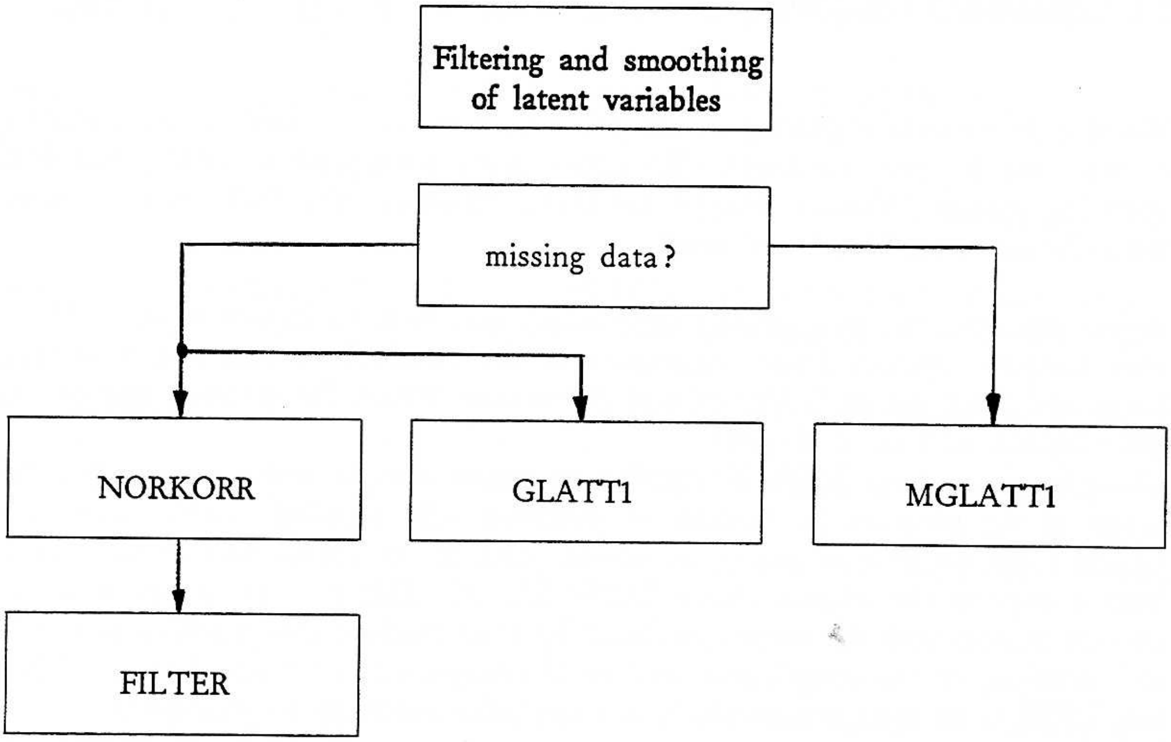
CASEDEL Casewise deletion of missing values

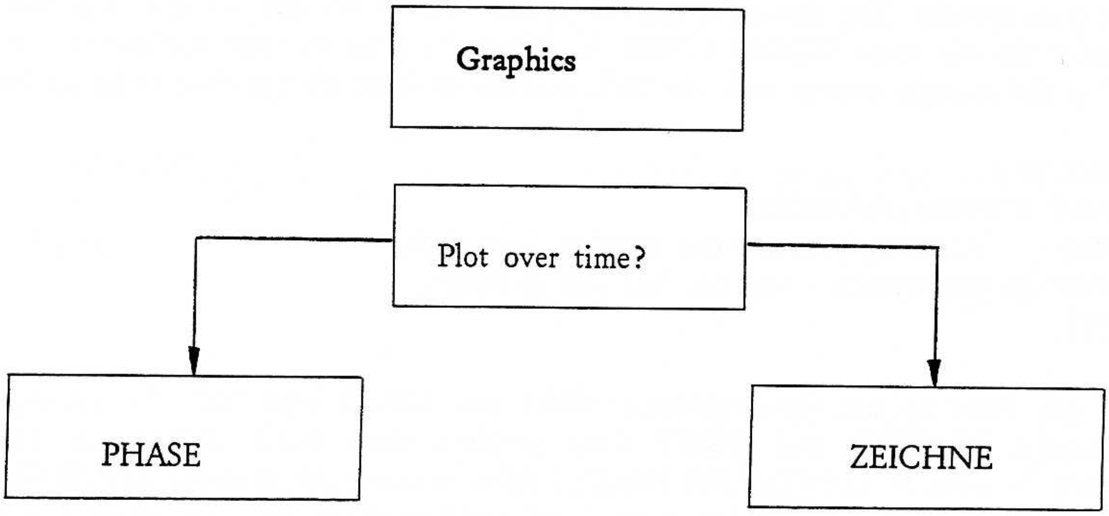
Flow diagrams of principal applications




ML estimation
measurement model ?
(latent variables)
PAR, PARM
PAR, PARM
PARD,PARMD
raw data?
fixed
measurement model?
MOMENT1
NEWTON
NEWSIM
EM
BFGS
NEWRAPH
EM1
BFGS1
NEWRAPH1


Diagnostics
PAR, PARM
PARM, PARMD
DIAGNOSE


Filtering and smoothing of latent variables
missing data?
NORKORR
GLATT1
MGLATT1
FILTER


Graphics
Plot over time?
PHASE
ZEICHNE

1.2. Implementation of LSDE

Warning: file names appearing in the following text may depend on the operating system used by your computer. The names given correspond to IBM`s VM/CMS operating system (filename filetype filemode). However, the IML code given in SWISS letters is machine independent.

Before you can start programming applications you have to implemement LSDE on your computer system. Basic requirement is the SAS/IML version 6 software (see Usage and Reference guide Version 6 (SAS Institute, 1989)). For graphics applications SAS/GRAPH software is required.
The program package **LSDE is supplied as source code** in order to give you full access to the program. In contrast to packages with compiled machine code (e.g. LISREL) you are able to change or extend parts of the source (you should always keep a copy of the original source LSDE1 SAS A). This may be necessary if you are not content with the output produced by some modules (too much or too little, too technical or too complicated etc.) or if unexpected errors are detected. Therefore **LSDE is an open system** which you can tailor according to your needs.
IML is an interpretive language executor but you can and **should pre-compile** the source code. This is done by running the file LSDE1 SAS A with the SAS/IML interpreter (Consult your SAS software consultant how the SAS system is invoked at your computer). The precompiled code is then stored on disk as a storage catalog under the file name OLSDE1 COMP A. It can be used in your application by loading the module storage with the IML statements (note the reversed order in line 2)

```
PROC IML;
RESET STORAGE=COMP.LSDE1;
LOAD;      *Loading precompiled modules from disk;
*FURTHER STATEMENTS - SEE CH. III FOR EXAMPLES;
QUIT;
```

If you want to use the graphics facilities you should load only the appropriate modules ZEICHNE and PHASE since graphics need much storage, i.e. replace LOAD; by LOAD MODULE=(ZEICHNE PHASE);. After running the modules ZEICHNE and /or PHASE a graphics catalog is produced and stored on disk. To obtain a picture on a graphics terminal or a hard copy the device drivers of SAS/GRAPH are required.

2. General Structure of an LSDE application

The general structure of an IML/LSDE program is described as follows:

```
PROC IML;                                                   *start of proc IML;

RESET STORAGE=COMP.LSDE1;                  *loading module library from disk storage;
                                      * (name on disk is dependent on operatingsystem);
LOAD;
RUN INITLSDE;                                 *initialisation of module library LSDE;

START MAIN;                                           *start of main program MAIN;

     START PAR(.....);                         * model specification (system model);
     ....
     FINISH;

     START PARD(.....);                            *derivatives of system matrices;
     ....
     FINISH;

     START PARM(..... );                  *model specification (measurement model);
     ....
     FINISH;

     START PARMD(..... );                     *derivatives of measurement matrices;
     ....
     FINISH;


.....
.....                                               *execution of modules supplied by;
......                                                *LSDE and other IML statements;

FINISH;                                                         *end of main program;

RUN MAIN;                                                *running main program MAIN;
QUIT;                                                                *end of proc IML;
```

Since IML is a programming language, the user is completely free in mixing language supplied control structures, procedures, data set operations etc. with LSDE supplied modules and user written modules.

For example, after estimation of the drift matrix A you can calculate its eigenvalues

and eigenvectors in order to investigate the stability of the system. Then you may decide to store the estimates on disk for later use (e.g. you may simulate data from the estimates and plot them with graphics facilities supplied by LSDE or IML). Other examples are provided by the case studies in chapter III.
Clearly there is a tradeoff between the flexibility to program arbitrary applications and simplicity of usage. Of course it is much simpler to use traditional procedures (e.g. GLM or CORR etc.), but you have to be content with the predefined options. Nevertheless you can use other SAS software products in combination with IML or/and LSDE (e.g. the procedure UNIVARIATE was used by the author after a simulation study, which produced a SAS data set of ML estimators; see case study 1).

3. Description of LSDE supplied modules in alphabetical order

BFGS
BFGS1

CASEDEL

DATSTRUK
DIAGNOSE

EIGEN1
EM
EM1
EXAKT
EXPO1

FILTER

GLATT1

INITLSDE

MOMENT1
MGLATT1

NEWTON
NEWRAPH
NEWRAPH1
NEWSIM
NORKORR

PAR
PARD
PARM
PARMD
PHASE

STATESPA
STERN

TREE

ZEICHNE
ZEITREIH

BFGS

Purpose: Maximum likelihood estimation of the continuous/discrete state space model using the Broyden-Fletcher-Goldfarb-Shanno-algorithm. BFGS uses an exact analytic expression for the score function and the BFGS secant update as model Hessian.

Call: `RUN BFGS(THETA,PHI,COV,`

```
                THETA0,PHI0,F0,KMAX,EPS,Z,X,K,P,Q,N,T,DT,OPTION,NONLIN);
```

Output Parameters:

THETA: $u_1 \times 1$. Final estimate of Θ (ML estimate if convergence was achieved) after KMAX iterations.
PHI: $u_2 \times 1$: Final estimate of Φ (ML estimate if convergence was achieved) after KMAX iterations
COV: u×u. $(u=u_1+u_2)$. Estimated covariance matrix of the ML-estimator $\Psi=[\Theta`,\Phi`]`$. Computed as GINV(F), where F is the estimated Fisher information matrix.

Input Parameters:

THETA0: $u_1 \times 1$. Starting value Θ_0 for BFGS iteration.
PHI0: $u_2 \times 1$. Starting value Φ_0 for iteration.
F0: u×u. Starting value for BFGS update. F0 is a symmetric, positive definite matrix. You may take the identity matrix I(u) or an approximation to the Fisher information matrix or the negative Hessian if p.d.
KMAX: Maximal number of iterations. If you start investigating a model you are recommended to use a small number (e.g. 10) to keep CPU time small. If KMAX was too small for achieving convergence (maximum of likelihood) you can continue with a bigger number using the final estimates THETA and PHI as new starting values.

EPS: Convergence criterion. If max (abs (Score)) < ε and max(abs (Step)) < ε the iteration is stopped.
Z: K×(T+1)*N. Data matrix of measured data. Z must have the block structure $Z=[Z_0,Z_1,..........Z_T]$, where Z_j: k×N are submatrices containing measurements of N units at time point t_j. If the data matrix is of another form you can restructure it using the module DATSTRUK.
X: Q×(T+1)*N. Data matrix of exogenous variables with block structure $X=[X_0,.....X_T]$, where X_i: q×N.
K: Dimension of measured variables $z_n(t)$: k×1
P: Dimension of state vector $y_n(t)$: p×1
Q: Dimension of exogeneous variables $x_n(t)$: q×1
N: Number of units in panel
T: T+1 = number of panel waves $(t_0,.....t_T)$
DT: Time interval between measurements (sampling interval)
OPTION: If=1, the positive definiteness of the model Hessian is checked, and secured if this is not fulfilled.
NONLIN: If=1, the exact nonlinear model specification (exact discrete model) is used. NONLIN=0 (linearized model) should only be used if the time intervals dt(=Δt) are small compared to the dynamics or if you are analyzing a discrete time model (see appendix A)

Explanation of output:
First BFGS prints technical output and the (user supplied) starting values THETA0 and PHI0. Then for each step kk of the iteration the following quantities are printed:

- eigenvalues of BFGS update, minimal value, condition number and shift parameter (only if OPTION=1 and in the first step). If the update is not p.s.d. the old one is used.
- if step halving is required the new and old likelihoods are printed
- new and old likelihood , score, quasi-Newton step, new and old parameter vector PSI=THETA//PHI and PSI1.
- BFGS update.

After convergence or if kk=KMAX the iteration is stopped and the following quantities are printed:

- successive values of the likelihood function (LIKI)
- successive values of parameter vector $\psi=[\Theta`,\Phi`]`$ (PSII)
- exact and numerical score (for control purposes)
- ML estimator, score and likelihood at the end of iteration.
- Wald test: Test of hypothesis H_0: $\psi=0$ against H_1: $\psi\neq0$.
- Bonferroni confidence intervals:

$$P\{\ \psi \in [\hat{\psi} \pm z_{\alpha/2u} * STD]\} \geq 1-\alpha,$$

where STD is the vector of estimated standard deviations of $\hat{\psi}$ (obtained as $STD=\sqrt{\text{vecdiag}(F)}$, F=estimated Fisher information matrix). $z_{\alpha/2u}$ is the $\alpha/2u$ quantile of the N(0,1)-Gauß-distribution and u is the number of parameters.

- correlation matrix of ML estimator (CORR).
- asymptotic covariance matrix of ML estimator ($COV=F^{-1}$).
- eigenvalues of COV.
- estimated ("observed") Fisher information matrix F (negative Hessian $F \sim -\partial^2 l/\partial\psi^2$).
- eigenvalues of F (check of positive definiteness).
- compensated likelihood (see ch. II. 6).
- Delta-theorem: approximate standard errors of parameter matrices A, B, Ω, μ, Σ and H, D, R from estimated covariance matrix COV $\sim$ $cov(\hat{\psi})$ (see Rao, 1973). In addition $A(\hat{\Theta})$, $B(\hat{\Theta})$ etc. , are printed.
- Estimated moment matrices (conditional moments, see ch. II.5).

BFGS1

Purpose: Maximum likelihood estimation of the continuous/discrete state space model using the Broyden-Fletcher-Goldfarb-Shanno-algorithm. BFGS1 uses an exact analytic expression for the score function and the BFGS secant update as model Hessian. In contrast to BFGS all parameters of the measurement model are fixed (Φ is empty). This case occurs, e.g. , if only one component of the state vector can be observed (pseudo measurement model) or if factor loadings are known in advance.

Call:
```
RUN BFGS(THETA,COV,
                THETA0,F0,KMAX,EPS,Z,X,K,P,Q,N,T,DT,OPTION,NONLIN);
```

Output Parameters:

THETA: $u_1 \times 1$. Final estimate of Θ (ML estimate if convergence was achieved) after KMAX iterations.
COV: $u_1 x u_1$. Estimated covariance matrix of the ML-estimator Θ. Computed as GINV(F), where F is the estimated Fisher information matrix.

Input Parameters:

THETA0: $u_1 \times 1$. Starting value Θ_0 for BFGS iteration.
F0: $u_1 \times u_1$. Starting value for BFGS update. F0 is a positiv definite matrix. You may take the identity matrix $I(u_1)$ or an approximation to the Fisher information matrix or the negative Hessian if p.d.
KMAX: Maximal number of iterations. If you start investigating a model you are recommended to use a small number (e.g. 10) to keep CPU time small. If KMAX was too small for achieving convergence (maximum of likelihood) you can continue with a bigger number using the final estimate THETA as new starting value.

EPS: Convergence criterion. If max (abs (Score)) $< \varepsilon$ and max(abs (Step)) $< \varepsilon$ the iteration is stopped.
Z: $P \times (T+1)N$. Data matrix of measured data.
X: $Q \times (T+1)N$. Data matrix of exogenous variables.

K: Dimension of measured variables $z_n(t)$: k×1
P: Dimension of state vector $y_n(t)$: p×1
Q: Dimension of exogeneous variables $x_n(t)$: q×1
N: Number of units in panel
T: T+1 = number of panel waves $(t_0, ...t_T)$
DT: Time interval between measurements (sampling interval)
OPTION: If=1, the positive definiteness of the model Hessian is checked and secured, if not fulfilled.
NONLIN: If=1, the exact nonlinear model specification (exact discrete model) is used. NONLIN=0 (linearized model) should only be used if the time intervals dt(=Δt) are small compared to the dynamics or if you are analyzing a discrete time model (see appendix A).

Explanation of output:
see **BFGS**

CASEDEL

Purpose: Deletion of cases with missing data. The N cases are ordered rowwise.

Call: `RUN CASEDEL(YOUT,Y,MISS);`

Output Parameters:

YOUT: N`×M. New data matrix

Input Parameters:

Y: N×M. Old data matrix containing missing values.
MISS: Code of missing values (Number or .)

DATSTRUK

Purpose: Restructuring of data matrices when the block (tree) structure is not appropriate. For example, the raw data matrix may have the structure 〈N,p*(T+1)〉, i.e. N rows and each row contains T+1 measurements for p variables. However, LSDE requires the block structure 〈p,(T+1)N〉 , i.e. the p-column vectors y_{tn} of measurements are temporally ordered.

Call: `RUN DATSTRUK (XNEU,XALT,ZEILE,SPALTE,ZPERM,SPERM);`

Output Parameters:

XNEU: restructured matrix

Input Parameters:

XALT: Data matrix to be restructured.
ZEILE: vector of row indices = $\{i_1,i_2,....i_g\}$; g=number of row indices.
SPALTE: vector of column indices = $\{i_{g+1},i_{g+2},...i_h\}$; h=total number of indices.
ZPERM: permutation vector of new row indices. The indices with number ZPERM[k] become new row indices i_k.
SPERM: permutation vector of new column indices. The indices with number SPERM[k] become new column indices i_{g+k}

Example:

```
N=3;T=4;P=2;
                                                    * {N,TP}-TREE STRUCTURE;
XALT={ 1  2  3  4  5  6  7  8 ,
       9 10 11 12 13 14 15 16 ,
      17 18 19 20 21 22 23 24 };

ZEILE={3};SPALTE={4 2};
ZPERM={3};                    *THE THIRD INDEX (P) BECOMES FIRST ROW INDEX  ;
SPERM={2 1};               * THE SECOND INDEX (T) BECOMES FIRST COLUMN INDEX
                         AND THE FIRST INDEX (N) BECOMES SECOND COLUMN INDEX;
                                  *THE NEW MATRIX HAS TREE STRUCTURE {P,TN};
RUN DATSTRUK (XNEU,XALT,ZEILE,SPALTE,ZPERM,SPERM);
PRINT XNEU;
*XNEU={1  9 17 3 11 19 5 13 21 7 15 23,
       2 10 18 4 12 20 6 14 22 8 16 24};
* in APL notation we have XNEU<— 3 2 1 ⍉ XALT , i.e. {P,TN}  = 3 2 1 {N,TP};
```

DIAGNOSE

Purpose: Diagnostic checks. After fitting a certain model it should be checked whether the specification was appropriate. If it is correct the residuals (difference of data and fitted values) should form a white noise sequence. The innovations (residuals) are computed using the Kalman filter given the ML estimates and the autocorrelation is calculated. Furthermore, the innovations are plotted over time. The module computes the multivariate portmanteau statistic (Hosking,1980), which tests the residuals for autocorrelation. The analysis is performed seperately for every unit n, n=1,....,N, in order to check the homogeneity of the sample with respect to dynamical properties. However, the number of time points should not be too small since the distribution of Hosking`s statistic is only known asymptotically.

If T is small, but N large (the panel case), we alternatively consider the N random vectors $\nu_n=\{\nu_{jni}$; j=1,.....k, i=0,......T$\}$: k(T+1) × 1 of normalized innovations, which are Gauß- distributed according to $N(0,I_{k(T+1)})$ if the parameters are known. We test the hypothesis H_0: $\mu=0$, $\Sigma=I_{k(T+1)}$ using an approximate likelihood ratio test.

Call: `RUN DIAGNOSE(INNO,GAMMA,RHO,MAXLAG,PQ,Z,X,K,P,Q,N,T,DT,`
`NSELECT,THETA,PHI,NONLIN);`

Output Parameters:

INNO: k×(T+1)N matrix of innovations ν (block structure $[\nu_0,.....\nu_T]$).
GAMMA: k^2×(T+1)N* matrix of residual autocovariances (N* = ncol(NSELECT) = number of seperately analyzed units).
RHO: k ×(T+1)N matrix of residual autocorrelations (N* = ncol(NSELECT)).

Input Parameters:

MAXLAG: Maximal lag to be used in calculating the portmanteau statistic and in the autocorrelations of the residuals.
PQ: number of autoregressive and moving average parameters.
Z: k×(T+1)N data matrix.
X: q×(T+1)N matrix of exogenous variables.
K: dimension of data vector $z_n(t)$
P: dimension of state vector $y_n(t)$
Q: dimension of exogenous variables $x_n(t)$
N: number of units.
T: T+1=number of panel waves.
DT: sampling interval
NSELECT: vector of numbers of units to be analyzed seperately.
THETA: estimated parameter vector Θ.
PHI: estimated parameter vector Φ.
NONLIN: =1 if the exact discrete model is used.

EIGEN1

Purpose: Calculation of eigenvectors and eigenvalues of a real nonsymmetric matrix. A QR-algorithm is utilized. If the matrix is symmetric you should use the IML routines EIGEN and EIGVAL (see IML reference guide)

Call: `RUN EIGEN1(EV,LAMBDA,A,EPS);`

Output Parameters:

EV: Matrix containing the eigenvectors of A as columns. The submatrix EV[1:p,] contains the real part and EV[p+1 : 2*p,] the imaginary part.
LAMBDA: Matrix containing eigenvalues. LAMBDA[1,] =real part, LAMBDA[2,]= imaginary part.

Input Parameters:

A: p×p-matrix to be analyzed.
EPS: precision parameter. A value of $\varepsilon=10^{-14}$ is recommended.

EM

Purpose: Maximum likelihood estimation of the continuous/discrete state space model using an EM algorithm. In the M-step the pseudo likelihoods of the system and measurement model are maximized seperately using a modified scoring algorithm. If you cancel the maximization before reaching the maximum a generalized EM (GEM) algorithm is obtained (see ch. II. 5). EM reliably locates a maximum of the likelihood, but near it it may converge very slowly. In such a case EM should be followed by a few iteration steps using BFGS or NEWRAPH.

Call: `RUN EM (THETA, PHI, F, Z, X, K, P, Q, N, T, DT, THETA0, PHI0, IMAX, KMAX, EPS, NONLIN);`

Output Parameters:

THETA: $u_1\times 1$. Final estimate of Θ (ML estimate if convergence was achieved) after IMAX iterations.
PHI: $u_2\times 1$: Final estimate of Φ (ML estimate if convergence was achieved) after IMAX iterations
F: u×u ($u=u_1+u_2$). Estimated Fisher information at final estimate. Can be used as starting value F0 in BFGS.

Input Parameters:

Z: K×(T+1)*N. Data matrix of measured data. Z must have the block structure $Z=[Z_0,Z_1,.........Z_T]$, where Z_j: k×N are submatrices containing measurements of N units at time point t_j. If the data matrix is of another form you can restructure it using the module DATSTRUK.
X: Q×(T+1)*N. Data matrix of exogenous variables with block structure $X=[X_0,.......X_T]$, where X_i: q×N.
K: Dimension of measured variables $z_n(t)$: k×1
P: Dimension of state vector $y_n(t)$: p×1
Q: Dimension of exogenous variables $x_n(t)$: q×1
N: Number of units in panel
T: T+1 = number of panel waves $(t_0,.....t_T)$
DT: Time interval between measurements (sampling interval)
THETA0: $u_1\times 1$. Starting value Θ_0.
PHI0: $u_2\times 1$. Starting value Φ_0.

IMAX: Maximum number of EM iterations.
KMAX: Maximum number of iterations in M-step. KMAX=1 corresponds to a modified scoring algorithm (see ch. II.5).
EPS: stopping criterion: if $\max(\mathrm{abs}(\Theta_{i+1}-\Theta_i)) < \varepsilon$ and $\max(\mathrm{abs}(\Phi_{i+1}-\Phi_i)) < \varepsilon$, the EM iteration is stopped.
NONLIN: If=1, the exact nonlinear model specification (exact discrete model) is used. NONLIN=0 (linearized model) should only be used if the time intervals dt(=Δt) are small compared to the dynamics or if you are analyzing a discrete time model (see appendix A)

Explanation of output:

First EM prints the NONLIN parameter and the (user supplied) starting values THETA0 and PHI0.
After convergence or if i=IMAX the iteration is stopped and the following quantities are printed:

- successive values of the likelihood function (LIK) and of the pseudolikelihood increments QQSYS and QQMESS. The increments of LIK are greater or equal to the sum of QQSYS and QQMESS, i.e. LIK[i+1] - LIK[i] $\geq$ QQSYS[i] + QQMESS[i].
- successive values of parameter vector Ψ`=THETAPHI=[$\Theta$`,Φ`]
- exact and numerical score (for control purposes)
- ML estimator, score and likelihood at the end of iteration.
- Wald test: Test of hypothesis H_0: $\psi=0$ against H_1: $\psi\neq 0$.
- Bonferroni confidence intervals:

$$P\{ \psi \in [\hat{\psi} \pm z_{\alpha/2u} * STD] \} \geq 1-\alpha,$$

where STD is the vector of estimated standard deviations of $\hat{\psi}$ (obtained as $STD = \sqrt{\mathrm{vecdiag}(F)}$, F=estimated Fisher information matrix). $z_{\alpha/2u}$ is the $\alpha/2u$ quantile of the N(0,1)-Gauß-distribution.

- correlation matrix of ML estimator (CORR).
- asymptotic covariance matrix of ML estimator (COV=F^{-1}).
- eigenvalues of COV.
- estimated ("observed") Fisher information matrix F (negative Hessian $F \sim -\partial^2 l/\partial\psi^2$).
- eigenvalues of F (check of positive definiteness).
- compensated likelihood (see ch. II. 6).
- Score and Fisher information of pseudo likelihoods QQSYS and QQMESS (Quantities SS, SF, SM, FM)
- Delta-theorem: approximate standard errors of parameter matrices A, B, Ω, μ, Σ and H, D, R from estimated covariance matrix COV $\sim$ cov($\hat{\psi}$) (see Rao, 1973). In addition A($\hat{\Theta}$), B($\hat{\Theta}$) etc. , are printed.
- Estimated moment matrices (conditional moments, see ch. II.5).

EM1

Purpose: Maximum likelihood estimation of the continuous/discrete state space model using an EM algorithm. In contrast to module EM the measurement model is fixed (Φ is empty). In the M-step the pseudo-likelihood of the system is maximized using a modified scoring algorithm. If you cancel the maximization before reaching the maximum a generalized EM (GEM) algorithm is obtained (see ch. II. 5). EM1 reliably locates a maximum of the likelihood, but near it it may converge very slowly. In such a case EM1 should be followed by a few iteration steps using BFGS1 or NEWRAPH1.

Call: `RUN EM1 (THETA,F,Z,X,K,P,Q,N,T,DT,THETA0,IMAX,KMAX,EPS,NONLIN);`

Output Parameters:

THETA: $u_1 \times 1$. Final estimate of Θ (ML estimate if convergence was achieved) after IMAX iterations.
F: $u_1 \times u_1$. Estimated Fisher information at final estimate. Can be used as starting value F0 in BFGS.

Input Parameters:

Z: K×(T+1)*N. Data matrix of measured data. Z must have the block structure $Z=[Z_0,Z_1,..........Z_T]$, where Z_j: k×N are submatrices containing measurements of N units at time point t_j. If the data matrix is of another form you can restructure it using the module DATSTRUK.
X: Q×(T+1)*N. Data matrix of exogenous variables with block structure $X=[X_0,......X_T]$, where X_i: q×N.
K: Dimension of measured variables $z_n(t)$: k×1
P: Dimension of state vector $y_n(t)$: p×1
Q: Dimension of exogenous variables $x_n(t)$: q×1
N: Number of units in panel
T: T+1 = number of panel waves $(t_0,.....t_T)$
DT: Time interval between measurements (sampling interval)
THETA0: $u_1 \times 1$. Starting value Θ_0.
IMAX: Maximum number of EM iterations.
KMAX: Maximum number of iterations in M-step. KMAX=1 corresponds to a modified scoring algorithm (see ch. II.5).
EPS: stopping criterion: if $\max(\mathrm{abs}(\Theta_{i+1}-\Theta_i)) < \varepsilon$, the EM iteration is stopped.
NONLIN: If=1, the exact nonlinear model specification (exact discrete model) is used. NONLIN=0 (linearized model) should only be used if the time intervals dt(=Δt) are small compared to the dynamics or if you are analyzing a discrete time model (see appendix A)

Explanation of output:

First EM1 prints the NONLIN parameter and the (user supplied) starting values THETA0.
After convergence or if i=IMAX the iteration is stopped and the following quantities are printed:

- successive values of the likelihood function (LIK) and of the pseudolikelihood increments QQSYS. The increments of LIK are greater or equal to QQSYS , i.e. LIK[i+1] - LIK[i] ≥ QQSYS[i].
- successive values of parameter vector THETA=Θ`.
- exact and numerical score (for control purposes)
- ML estimator, score and likelihood at the end of iteration.
- Wald test: Test of hypothesis H_0: $\psi=0$ against H_1: $\psi \neq 0$.
- Bonferroni confidence intervals: $P\{ \Theta \in [\hat{\Theta} \pm z_{\alpha/2u} * \mathrm{STD}] \} \geq 1-\alpha$, where STD is the vector of estimated standard deviations of Θ (obtained as $\mathrm{STD} = \sqrt{\mathrm{vecdiag(F)}}$, F=estimated Fisher information matrix). $z_{\alpha/2u}$ is the $\alpha/2u$ quantile of the N(0,1)-Gauß-distribution and u is the number of parameters.
- correlation matrix of ML estimator (CORR).
- asymptotic covariance matrix of ML estimator (COV=F^{-1}).
- eigenvalues of COV.
- estimated ("observed") Fisher information matrix F (negative Hessian $F \sim -\partial^2 l/\partial\psi^2$).
- eigenvalues of F (check of positive definiteness).
- compensated likelihood (see ch. II. 6).
- Score and Fisher information of pseudo likelihood QQSYS (Quantities SS, SF).
- Delta-theorem: approximate standard errors of parameter matrices A, B, Ω, μ, Σ and H, D, R from estimated covariance matrix COV ~ $\mathrm{cov}(\hat{\psi})$ (see Rao, 1973). In addition $A(\hat{\Theta})$, $B(\hat{\Theta})$ etc. , are printed.
- Estimated moment matrices (conditional moments, see ch. II.5).

EXAKT

Purpose: Simulation of the "exact discrete model" (see equ. 3, ch. II.2).

Call: `RUN EXAKT(Y,A,B,G,Y0,X,P,Q,N,T,DT,SEED,PRINT);`

Output parameters:

Y: p×(T+1)N data matrix of simulated values $Y=[Y_0,.........Y_T]$.

Input parameters:

A: p×p drift matrix.
B: p×q input matrix.
G: p×r diffusion coefficient (Ω=GG`).
Y0: p×N random inititial condition.

X: $q \times (T+1)N$ matrix of exogenous variables of block structure $X=[X_0,......X_T]$.
P: dimension of state variable $y_n(t)$.
Q: dimension of exogenous variable $x_n(t)$.
N: Number of units (persons).
T: Number of panel waves $(t_0,......t_T)$
DT: distance of panel waves.
SEED: random seed for random generator RANNOR. An odd integer of more than 5 places is recommended.
PRINT: if=1, the matrices Y0, Y and X are printed.

EXPO1

Purpose: Calculation of matrix exponential function $\exp(A)=\sum_{j=0}^{\infty} A^j/j!$.

Computed using the spectral decomposition $\exp(A)=P\exp(\Lambda)P^{-1}$.

Call: `RUN EXPO1(EXPA,A);`

Output Parameter:

EXPA: $p \times p$ matrix = exp(A).

Input Parameter:

A: $p \times p$ matrix.

FILTER

Purpose: Kalman filter algorithm. Calculates recursively the optimal estimates of the unobserved state vector y_{nt} (latent variables) given the measured data. Furthermore, the filtering error and the likelihood function are computed.

Call: `CALL FILTER(YF,PF,LIK,Y0,P0,Z,X,ASTERN,BSTERN,OSTERN,H,D,R,N,T);`

Output Parameters:

YF: $p \times (T+1)N$- data matrix of estimates: $YF=[\hat{Y}_0,.......\hat{Y}_T]$, where $\hat{Y}_i=E[Y_i|Z_0Z_1....Z_i]$ is the conditional mean of the latent variables Y_i given data up to time point i.
PF: $p \times (T+1)p$-matrix of filter error: $PF=[P_0,.......P_T]$, where $P_i=COV[Y_i,Y_i|Z_0,...Z_i]$. Since it is the same for all n=1.....N, it is calculated only for one trajectory.
LIK: Likelihood function calculated from prediction error decomposition.

Input Parameters:

Y0: p×N. Initial value for Kalman filter. Usually chosen as $\hat{Y}_0=E[Y_0|Z_0]$ (see NORKORR).
P0: p×p. Initial value for filter error: usually chosen as $P_0=COV(Y_0,Y_0|Z_0)$ (see NORKORR).
Z: k×(T+1)N data matrix of measured data.
X: q×(T+1)N data matrix of exogenous data.
ASTERN: p×p. $=A^*=\exp(A\Delta t)$. Regression parameter of exact discrete model (EDM). See STERN.
BSTERN: p×q. $=B^*=A^{-1}(A^*-I)B$. Input matrix of EDM. See STERN.
OSTERN: p×p $=\Omega^*=\int \exp(As)\Omega\exp(A`s)ds$. See STERN.
H: k×p matrix of factor loadings.
D: k×q input matrix (measurement model)
R: k×k covariance matrix of measurement errors.
N: number of units
T: number of time points $t_0,......t_T$

GLATT1

Purpose: Optimal filtering and interpolation (smoothing) of the latent variables (state vector) $y_n(t)$, $t \in \{ t_0,......t_T \}$ given observations $z_n(t)$. The present implementation uses a fixed interval Rauch-Tung-Striebel smoother (see ch. II). It is supposed that the parameter vectors Θ and Φ are known or have been estimated. The optimally smoothed estimates are conditional expectations of the latent variables given the measured data.

Call: `RUN GLATT1(YG,PG,YF,PF,LIK,Z,X,K,P,Q,N,T,DT,THETA,PHI,NONLIN);`

Output Parameters:

YG: p×(T+1)N matrix of smoothed estimates, i.e. $YG=[\hat{Y}_0,.....\hat{Y}_T]$, where $\hat{Y}_i=E[Y_i|Z_0.....Z_T]$.
PG: p×(T+1)p matrix of smoothing error covariances, i.e. $PG=[P_0,......P_T]$, where $P_i=COV[Y_i,Y_i|Z_0.....Z_T]$.
YF: p×(T+1)N matrix of filtered estimates, i.e. $YF=[\hat{Y}_0,.....\hat{Y}_T]$, where $\hat{Y}_i=E[Y_i|Z_0.....Z_i]$.
PF: p×(T+1)p matrix of filtering error covariances, i.e. $PF=[P_0,......P_T]$, where $P_i=COV[Y_i,Y_i|Z_0.....Z_i]$.
LIK: Likelihood of data Z.

Input Parameters:

Z: k×(T+1)N matrix of measured data.
X: q×(T+1)N matrix of exogenous (control) variables.

K: dimension of measured variables (z_{nt}: k×1)
P: dimension of latent variables (y : p×1)
Q: dimension of exogenous variables (x_{nt}: q×1)
N: number of units
T: time points $t_0, \ldots\ldots t_T$
DT: sampling interval
THETA: known or estimated parameter vector Θ .
PHI: known or estimated parameter vector Φ .
NONLIN: = 1 if exact discrete model is used (= 0, if linearized or discrete time models are used)

INITLSDE

Purpose: Sets several global parameters.

Call: `RUN INITLSDE;`

MGLATT1

Purpose: Optimal filtering and interpolation (smoothing) of the latent variables (state vector) $y_n(t)$, $t \in \{ t_0, \ldots\ldots t_T \}$ given observations $z_n(t)$. In contrast to GLATT1 missing values are allowed. These components are reconstructed by the smoother using information from the data and the system parameters. The present implementation uses a fixed interval Rauch-Tung-Striebel smoother (see ch. II). It is supposed that the parameter vectors Θ and Φ are known or have been estimated. The optimally smoothed estimates are conditional expectations of the latent variables given the measured data.

Call: `RUN MGLATT1(YG,PG,YF,PF,LIK,Z,X,K,P,Q,N,T,DT,THETA,PHI,NONLIN,MISS);`

Output Parameters:

YG: p×(T+1)N matrix of smoothed estimates, i.e. $YG=[\hat{Y}_0, \ldots\ldots \hat{Y}_T]$, where $\hat{Y}_i=E[Y_i|Z_0 \ldots\ldots Z_T]$.
PG: (p(p+1)/2)×(T+1)N matrix of smoothing error covariances, i.e. $PG=[P_0, \ldots\ldots P_T]$, where $P_i=COV[Y_i,Y_i|Z_0 \ldots\ldots Z_T]$. Only the upper triangles of the matrices P_i: p×p are stored in the columns of PG.
YF: p×(T+1)N matrix of filtered estimates, i.e. $YF=[\hat{Y}_0, \ldots\ldots \hat{Y}_T]$, where $\hat{Y}_i=E[Y_i|Z_0 \ldots\ldots Z_i]$.
PF: (p(p+1)/2)×(T+1)N matrix of filtering error covariances, i.e. $PF=[P_0, \ldots\ldots P_T]$, where $P_i=COV[Y_i,Y_i|Z_0 \ldots\ldots Z_i]$. Only the upper triangle of P_i is stored.
LIK: Likelihood of data Z.

Input Parameters:

Z: k×(T+1)N matrix of measured data.
X: q×(T+1)N matrix of exogenous (control) variables.
K: dimension of measured variables (z_{nt}: k×1)
P: dimension of latent variables (y : p×1)
Q: dimension of exogenous variables (x_{nt}: q×1)
N: number of units
T: time points $t_0,......t_T$
DT: sampling interval
THETA: known or estimated parameter vector Θ .
PHI: known or estimated parameter vector Φ .
NONLIN: = 1 if exact discrete model is used (= 0, if linearized or discrete time models are used)
MISS: Code of missing values (number or .)

MOMENT1

Purpose: Calculation of sample moment matrices for use in NEWTON.

Call: `RUN MOMENT1(M0,M00,M11,M12,M13,M22,M23,M33,Y,X,P,Q,N,T,PRINT);`

Output Parameters:

(set $Y_+=[Y_1.......Y_T]$, $Y_\sim=[Y_0.........Y_{T-1}]$ and $X_\sim=[X_0........X_{T-1}]$).
M0: p×1 M0 =(1/N) Y_0*1_N , where 1_N is N×1 vector of ones.
M00: p×p. M00=(1/NT)Y_0Y_0\`.
M11: p×p. M11 = (1/NT) Y_+Y_+\` M12 = (1/NT) $Y_+Y_\sim$\` M13 = (1/NT) $Y_+X_\sim$\`
M12: p×p. M22 = (1/NT) $Y_\sim Y_\sim$\` M23 = (1/NT) $Y_\sim X_\sim$\`
M13: p×q. M33 = (1/NT) $X_\sim X_\sim$\`
M22: p×p.
M23: p×q.
M33: q×q.

Input Parameters:

Y: p×(T+1)N matrix of measured data.
X: q×(T+1)N matrix of exogenous variables.
P: dimension of state vector $y_n(t)$.
Q: dimension of exogenous varables $x_n(t)$
N: number of units.
T: Number of panel waves $t_0,......t_T$
PRINT: If=1 the moment matrices are printed.

NEWTON

Purpose: Maximum likelihood estimation of stochastic differential equations with panel data. The data are assumed to be measured without errors of measurement. The module is recommended if no latent variables are present. It uses a scoring algorithm with an analytic score function and an analytic Fisher information matrix.

Call:
```
RUN NEWTON(THETA,COV,
           THETA0, KMAX,EPS, M0,M00,M11,M12,M13,M22,M23,M33,
           DT,N,T,OPTION,NONLIN);
```

Output Parameters:

THETA: ux1 vector. Contains the ML estimator $\hat{\Theta}$ if the scoring iteration has converged.
COV: u×u asymptotic covariance matrix of the ML estimator. Inverse of Fisher information matrix F.

Input Parameters:

THETA0: Initial value of scoring iteration.
KMAX: maximal number of iterations.
EPS: the algorithm terminates if max(abs(score)) < ε and max(abs(step)) < ε .
M0, M00,..........M33: Sample moment matrices (see MOMENT1).
DT: sampling interval of panel (=Δt).
N: number of units (persons).
T: time points t_0,......t_T
OPTION: if the identification status of the model is unclear, the Fisher information may be singular. If OPTION=1 the eigenvalues of F are calculated and the matrix is shifted if necessary. It is **strongly recommended to ensure identification by analytic calculation** (use of restrictions) and to set OPTION=0 (see ch. III.2, case study 1).
NONLIN: If=1, the exact nonlinear model specification (exact discrete model) is used. NONLIN=0 (linearized model) should only be used if the time intervals dt(=Δt) are small compared to the dynamics or if you are analyzing a discrete time model (see appendix A)

NEWRAPH

Purpose: Maximum likelihood estimation of the continuous/discrete state space model using a Newton-Raphson algorithm. NEWRAPH uses an exact analytic expression for the score function and numerical second derivatives from the analytic score as model Hessian.

Call:
```
RUN NEWRAPH(THETA,PHI,COV,
            THETA0,PHI0,KMAX,EPS,Z,X,K,P,Q,N,T,DT,OPTION,NONLIN);
```

Output Parameters:

THETA: $u_1 \times 1$. Final estimate of Θ (ML estimate if convergence was achieved) after KMAX iterations.
PHI: $u_2 \times 1$: Final estimate of Φ (ML estimate if convergence was achieved) after KMAX iterations
COV: $u \times u$. ($u=u_1+u_2$). Estimated covariance matrix of the ML-estimator $[\Theta,\Phi]$. Computed as GINV(F), where F is the estimated Fisher information matrix.

Input Parameters:

THETA0: $u_1 \times 1$. Starting value Θ_0 for Newton-Raphson iteration.
PHI0: $u_2 \times 1$. Starting value Φ_0 for iteration.
KMAX: Maximal number of iterations. If you start investigating a model you are recommended to use a small number (e.g. 10) to keep CPU time small. If KMAX was too small for achieving convergence (maximum of likelihood) you can continue with a bigger number using the final estimates THETA and PHI as new starting values.
EPS: Convergence criterion. If max (abs (Score)) $< \varepsilon$ and max(abs (Step)) $< \varepsilon$ the iteration is stopped.
Z: $K \times (T+1)*N$. Data matrix of measured data. Z must have the block structure $Z=[Z_0,Z_1,..........Z_T]$, where Z_j: $k \times N$ are submatrices containing measurements of N units at time point t_j. If the data matrix is of another form you can restructure it using the module DATSTRUK.
X: $Q \times (T+1)*N$. Data matrix of exogenous variables with block structure $X=[X_0,.....X_T]$, where X_i: $q \times N$.
K: Dimension of measured variables $z_n(t)$: $k \times 1$
P: Dimension of state vector $y_n(t)$: $p \times 1$
Q: Dimension of exogenous variables $x_n(t)$: $q \times 1$
N: Number of units in panel
T: T+1 = number of panel waves $(t_0,.....t_T)$
DT: Time interval between measurements (sampling interval)
OPTION: If=1, the positive definiteness of the model Hessian is checked, and secured if this is not fulfilled.
NONLIN: If=1, the exact nonlinear model specification (exact discrete model) is used. NONLIN=0 (linearized model) should only be used if the time intervals $dt(=\Delta t)$ are small compared to the dynamics or if you are analyzing a discrete time model (see appendix A)

Explanation of output:

First NEWRAPH prints the NONLIN parameter and the (user supplied) starting values THETA0 and PHI0. Then for each step kk of the iteration the following quantities are printed:

- eigenvalues of model Hessian, minimal value, condition number and shift parameter (only if OPTION=1).

- if step halving is required the new and old likelihoods are printed
- new and old likelihood , score, quasi-Newton step, new and old parameter vector PSI=THETA//PHI and PSI1 .

After convergence or if kk=KMAX the iteration is stopped and the following quantities are printed:

- successive values of the likelihood function (LIKI)
- successive values of parameter vector $\psi=[\Theta',\Phi']'$ (PSII)
- exact and numerical score (for control purposes)
- ML estimator, score and likelihood at the end of iteration.
- Wald test: Test of hypothesis H_0: $\psi=0$ against H_1: $\psi\neq0$.
- Bonferroni confidence intervals: $P\{\ \psi \in [\hat{\psi} \pm z_{\alpha/2u}*STD]\} \geq 1-\alpha$, where STD is the vector of estimated standard deviations of $\hat{\psi}$ (obtained as $STD = \sqrt{\text{vecdiag}(F)}$, F=estimated Fisher information matrix). $z_{\alpha/2u}$ is the $\alpha/2u$ quantile of the N(0,1) - Gauß-distribution and u is the number of parameters.
- correlation matrix of ML estimator (CORR).
- asymptotic covariance matrix of ML estimator ($COV=F^{-1}$).
- eigenvalues of COV.
- estimated ("observed") Fisher information matrix F (negative Hessian $F \sim -\partial^2 l/\partial\psi^2$).
- eigenvalues of F (check of positive definiteness).
- compensated likelihood (see ch. II. 6).
- Delta-theorem: approximate standard errors of parameter matrices A, B, Ω, μ, Σ and H, D, R from estimated covariance matrix COV $\sim$ cov($\hat{\psi}$) (see Rao, 1973). In addition A($\hat{\Theta}$), B($\hat{\Theta}$) etc. , are printed.
- Estimated moment matrices (conditional moments, see ch. II.5).

NEWRAPH1

Purpose: Maximum likelihood estimation of the continuous/discrete state space model using a Newton-Raphson algorithm. *The measurement model is fixed (Φ is empty).* NEWRAPH1 uses an exact analytic expression for the score function and numerical second derivatives from the analytic score as model Hessian.

Call:
```
RUN NEWRAPH1(THETA,COV,
             THETA0,KMAX,EPS,Z,X,K,P,Q,N,T,DT,OPTION,NONLIN);
```

Output Parameters: see NEWRAPH.
Input Parameters: see NEWRAPH.
Explanation of output: see NEWRAPH.

NEWSIM

Purpose: same as NEWTON, except that no printed output is produced. NEWSIM can be used for simulation studies, when many replications would lead to a formidable output.

Call:
```
RUN NEWSIM(THETA, COV,
                THETAO, KMAX, EPS, MO, MOO, M11, M12, M13, M22, M23, M33,
                DT, N, T, OPTION, NONLIN);
```

Output Parameters: see NEWTON
Input Parameters: see NEWTON

NORKORR

Purpose: Normal correlation (see Liptser and Shiryayev, 1978, ch. 13.1). Calculation of conditional expectations and covariances for Gaussian random vectors. These expectations are the estimated factor scores $\hat{y}_i$ of the measurement model z_i=Hy_i+Dx_i+ε_i; ε_i~N(0, R)i.i.d.; i=1.......N. It is used to calculate the initial condition for the Kalman filter FILTER.

Call: `RUN NORKORR(YZ, PZ, Z, X, MUE, SIGMA, H, D, R);`

Output Parameters:

YZ: p×N matrix of conditional expectations $E[Y_n|Z_n]$, n=1.......N.
PZ: p×p matrix of conditional covariances $COV(Y_n,Y_n|Z_n)$.

Input Parameters:

Z: k×N data matrix of N independent Gaussian random vectors with expectation $E[z_i]$=$H\mu_i$ + Dx_i. and identical covariance matrix $COV(z_i,z_i)$=HΣH`+R.
X: q×N data matrix of exogenous (deterministic) variables.
MUE: p×N matrix of expectation values $[\mu_1,......\mu_N]$
SIGMA: p×p covariance matrix Σ=$Cov(y_i,y_i)$, i=1.....N.
H: k×p matrix of factor loadings.
D: k×q matrix of input variables.
R: k×k covariance matrix of measurement errors.

PAR

Purpose: Specification of system model. ***User supplied.*** According to ch. IV.2. you are required to define a main program containing the modules PAR, PARD (and PARM, PARMD if a measurement model is used). Examples for these modules are given in chapter III and below.

Call: `RUN PAR(OMEGA,A,B,SIGMA,MUE,THETA);`

Output Parameters:

OMEGA: p×p diffusion matrix.
A: p×p drift matrix.
B: p×q input matrix.
SIGMA: p×p covariance matrix of initial condition.
MUE: p×1. Expectation of initial condition (μ=E[$y_n(t_0)$], n=1......N).

Input Parameters:

THETA: u_1×1 parameter vector.

Example:

```
START PAR(OMEGA,A,B,SIGMA,MUE,THETA);

A=( THETA[1] || THETA[2] )//
  (    2     ||    0     );                      * A[2,1] AND A[2,2] ARE CONSTANT;

B=EXP(THETA[2]) //0;                                       *B[1,1] IS POSITIVE;

OMEGA=diag( { 2 3 } );                                *DIAGONAL DIFFUSION MATRIX;

SIGMA=(THETA[3] || THETA[4])//
      (THETA[4] || THETA[5]);                                *SIGMA IS SYMMETRIC;

MUE={0,
     0};

FINISH;                                                             *END OF PAR;
```

PARD

Purpose: Matrix derivatives of A, B, Ω, μ, Σ with respect to Θ (see McDonald and Swaminathan (1973) and ch. III. 2). ***User supplied.***

Call: `RUN PARD (DOMEGA, DA, DB, DSIGMA, DMUE, THETA);`

Output Parameters:

DOMEGA: u×p^2 matrix of derivatives $\partial\Omega/\partial\Theta$
DA: u×p^2 matrix of derivatives $\partial A/\partial\Theta$
DB: u×pq matrix of derivatives $\partial B/\partial\Theta$
DSIGMA: u×p^2 matrix of derivatives $\partial\Sigma/\partial\Theta$
DMUE: u×p matrix of derivatives $\partial\mu/\partial\Theta$

Input Parameters:

THETA: u×1 parameter vector.

Example:

```
START PARD (DOMEGA, DA, DB, DSIGMA, DMUE, THETA);

DA={1 0 0 0 ,
    0 1 0 0 ,
    0 0 0 0 ,
    0 0 0 0 ,
    0 0 0 0 };

DB= (       0      ||    0 )//
    (EXP(THETA[2]))||    0 )//
    (       0      ||    0 )//
    (       0      ||    0 )//
    (       0      ||    0 );

DOMEGA= J(5, 4, 0);                                    *5X4 MATRIX OF ZEROES;

DSIGMA={1 0 0 0,
        0 1 0 0,
        0 0 0 0,
        0 0 0 0,
        0 0 0 0 };

DMUE=J(5, 2, 0);

FINISH;
```

PARM

Purpose: Specification of measurement model. ***User supplied.*** H, D, R as functions of the parameter vector Φ.

Call: `RUN PARM (R,H,D,PHI);`

Output Parameters:

R: k×k covariance matrix of measurement errors.
H: k×q matrix of factor loadings.
D: k×q matrix of input coefficients.

Input Parameters:

PHI: u_2×1 parameter vector.

Example:

```
START PARM (R,H,D,PHI);

H={1,0};
D=PHI[1];
R=PHI[2];

FINISH;
```

PARMD

Purpose: Derivatives of parameter matrices . ***User supplied.***

Call: `RUN PARM(DR,DH,DD,PHI);`

Output Parameters:

DR: $u_2 \times k^2$ matrix of derivatives $\partial R/\partial \Phi$
DH: u_2×kp matrix of derivatives $\partial H/\partial \Phi$
DD: u_2×kq matrix of derivatives $\partial D/\partial \Phi$

Input Parameters:

PHI: u_2×1 parameter vector.

Example:

```
START PARMD(DR,DH,DD,PHI);

DH=J(2,2,0);
DD={1,0};
DR={0,1};

FINISH;
```

PHASE

Purpose: Plot of data matrices in phase (state) space. If the data matrix DATEN is of block structure <p,(T+1)N>, i.e. DATEN=[Y_0,.......Y_T], p(p+1)/2 plots are produced each of which contains N curves (trajectories) for every unit n=1,.....N. Default colours are black for the axes and red for the curves. The points y_{tn} and $y_{t+1,n}$ are connected with straight lines (polygon). Can only be used if SAS/GRAPH software is available on your computer system.

Call: `RUN PHASE (DATEN,P,N,T,XTEXT,YTEXT,TITEL);`

Input Parameters:

DATEN: p×(T+1)N data matrix.
P: dimension of plotted data (number of variables).
N: number of units.
T: number of panel waves (t_0,......t_T).
XTEXT: literal containing text for x-axis (e.g. "variable 1");
YTEXT: literal containing text for y-axis (e.g. "variable 2");
TITEL: literal containing text for heading (e.g. "plot of var1 vs. var2");

STATESPA

Purpose: Simulation of continuous/discrete state space model (see ch II.2)

$$\begin{aligned} y_{t+1,n} &= A^* y_{tn} + B^* x_{tn} + u_{tn} \\ z_{tn} &= H y_{tn} + D x_{tn} + \varepsilon_{tn} \end{aligned} \quad .$$

Call: `RUN STATESPA (Z,Y,H,D,R,A,B,G,Y0,X,K,P,Q,N,T,DT,SEED,PRINT);`

Output Parameters:

Z: k×(T+1)N matrix of observed data

Y: p×(T+1)N matrix of unobserved (latent) variables.

Input Parameters:

H: k×p matrix of factor loadings.
D: k×q matrix of input coefficients.
R: k×k covariance matrix of measurement errors.
A: p×p drift matrix.
B: p×q input matrix.
G: p×r matrix of diffusion coefficients.
Y0: p×N matrix of initial conditions. Usually chosen as $y_{0n} \sim N(\mu, \Sigma)$ i.i.d.
X: q×(T+1)N matrix of control variables.
K: dimension of z_{tn}
P: dimension of y_{tn}
Q: dimension of x_{tn}
N: number of units (persons).
T: $t_0, \ldots\ldots t_T$
DT: $=\Delta t$. Difference of panel waves.
SEED: random seed for random generator RANNOR.
PRINT: little output if PRINT≠0.

STERN

Purpose: Calculation of the parameters of the exact discrete model. These are required in FILTER as inputs.

Call: `RUN STERN (OSTERN,ASTERN,BSTERN,A1,S,S1,LAMBDA,ELT,OMEGA,A,B,DT);`

Output Parameters:

OSTERN: p×p matrix = $\Omega^* = \int \exp(As)\Omega\exp(A`s)ds$.
ASTERN: p×p matrix = $A^* = \exp(A\Delta t)$.
BSTERN: p×q matrix = $A^{-1}(A^* - I)B$
A1: p×p matrix = A^{-1}
S: 2p×p Matrix of eigenvectors. S[1:p,]=real part , S[p+1:2p,]=imaginary part.
S1: inverse of S.
LAMBDA: 2×p matrix of eigenvalues. Λ[1,] =real parts, Λ[2,]= imaginary parts.
ELT:

Input Parameters:

OMEGA: p×p diffusion matrix.
A: p×p drift matrix.
B: p×q input matrix.

TREE

Purpose: Multi indexing for IML arrays. Since IML only allows two-dimensional arrays, higher dimensional arrays (for example data matrices $Y=y_{ptn}$) need a special indexing. TREE calculates an index vector from two given index vectors. For example, if nn=1....3 and tt=1.....5, and you want to select elements nn={1 3} and tt={1 2 5}, TREE calculates the appropriate index vector tn for selection: Y[,tn], i.e. all rows and tn columns of Y are chosen (see IML users guide for indexing with vectors).

Call: `TREE (TT,NN,T,N);`

Output Parameters:

TREE is a function module.

Input Parameters:

TT: index vector of first level.
NN: index vector of second level.
T: maximum value of first level.
N: maximum value of second level.

Example:
The tree structure is:

T:	1	2	3	4	5
N:	1 2 3	1 2 3	1 2 3	1 2 3	1 2 3
Selected values:	11 13	21 23			51 53

IML code:

```
T=5; N=3;
Y=SHAPE(1:30,2,15);          * 2X15 MATRIX;
YSELECT=Y[ ,TREE({1 2 5},{1 3},T,N)];       *2X6 MATRIX;
*TIME POINTS {1 2 5} AND PERSONS {1 3} ARE SELECTED;
```

ZEICHNE

Purpose: Plot of data matrices over time. If the data matrix DATEN is of block structure <p,(T+1)N>, i.e. DATEN=$[Y_0,.......Y_T]$, p plots are produced each of which contains N curves (trajectories) for every unit n=1,.....N. Default colours are black for the axes and red for the curves. The points y_{tn} and $y_{t+1,n}$ are connected with straight lines (polygon). Can only be used if SAS/GRAPH software is available on your computer system.

Call: `RUN ZEICHNE(DATEN,X,P,N,T,XTEXT,YTEXT,TITEL,OPTION);`

Output Parameters:

None.

Input Parameters:

DATEN: p×(T+1)N data matrix.
X: 1×(T+1). Coordinates of x-axis (e.g. time points or other variables).
P: dimension of plotted data (number of variables).
N: number of units.
T: number of panel waves $(t_0,......t_T)$.
XTEXT: literal containing text for x-axis (e.g. "time");
YTEXT: literal containing text for y-axis (e.g. "variable 2");
TITEL: literal containing text for heading (e.g. "plot of variable 2 over time");
OPTION: IF OPTION=1 all rows DATEN[i,], i=1,.....p are simultaneously scaled, if OPTION=0 they are separately scaled. This means that in the first case the y-axes of all p plots are identical.

ZEITREIH

Purpose: Simulation of **discrete** time state space model (see appendix I)

$$\begin{aligned} y_{n,t+1} &= \alpha\, y_{nt} + \beta\, x_{nt} + \zeta_{nt} \qquad ; \text{ where } \zeta_{nt} \sim N(0,\Gamma\Gamma'). \\ z_{nt} &= Hy_{nt} + Dx_{nt} + \varepsilon_{nt} \end{aligned}$$

Call: `RUN ZEITREIH(Z,Y,H,D,R,ALPHA,BETA,GAMMA,Y0,X,K,P,Q,N,T,DT,SEED,PRINT);`

Output Parameters:

Z: k×(T+1)N matrix of observed data
Y: p×(T+1)N matrix of unobserved (latent) variables.

Input Parameters:

H: k×p matrix of factor loadings.
D: k×q matrix of input coefficients.
R: k×k covariance matrix of measurement errors.
ALPHA: p×p matrix of autoregressive parameters.

BETA: p×q input matrix.
GAMMA: p×r matrix. $\Gamma\Gamma$` is the covariance matrix of the disturbances ζ_{nt}.
Y0: p×N matrix of initial conditions. Usually chosen as y_{0n}~$N(\mu,\Sigma)$ i.i.d.
X: q×(T+1)N matrix of control variables.
K: dimension of z_{nt}
P: dimension of y_{nt}
Q: dimension of x_{nt}
N: number of units (persons).
T: $t_0,......t_T$
DT: =Δt. Difference of panel waves.
SEED: random seed for random generator RANNOR.
PRINT: little output if PRINT=0.

V. APPENDIX

APPENDIX I

Analysis of Discrete Time State Space Models

As a by-product, LSDE can analyse discrete time series and panel models including errors of measurement. The **discrete/discrete state space model** is given as:

$$y_{n,i+1} = \alpha\, y_{ni} + \beta\, x_{ni} + \zeta_{ni} \;; \qquad i=0,.....,T$$

$$z_{ni} = H\, y_{ni} + D\, x_{ni} + \varepsilon_{ni} \;; \qquad i=0,.......T+1$$

where n=1,....,N and the structural parameter matrices are $\alpha(\Theta)$: p×p, $\beta(\Theta)$: p×q, $H(\Phi)$: k×p, $D(\Phi)$: k×q . ζ and ε are independent discrete time white noise disturbances distributed as $N(0,\Gamma\Gamma`)$ and $N(0,R)$, respectively, where $\Gamma(\Theta)$: p×r and $R(\Phi)$: k×k.
It can be simulated using the module ZEITREIH (see chapter IV).
The exact discrete model is:

$$y_{n,i+1} = A^{*} y_{ni} + B^{*} x_{ni} + u_{ni} \;; \qquad i=0,.....,T$$

$$z_{ni} = H\, y_{ni} + D\, x_{ni} + \varepsilon_{ni} \;; \qquad i=0,.......T+1$$

In contrast to the exact discrete model the matrix α does not have the special exponential structure $A^{*} = \exp(A\Delta t)$. However, if we use the linearized model (NONLIN=0), we can find a simple correspondence between α and A^{*}, β and B^{*}, Ω^{*} and $\Gamma\Gamma`$:

$$\alpha = I + A\Delta t$$
$$\beta = B\Delta t$$
$$\Gamma\Gamma` = \Omega\Delta t = GG`\Delta t$$

From this correspondence, you can specify and estimate the time discrete model as follows:

1. always use NONLIN=0 (in the modules NEWTON, BFGS etc.)
2. set $\Delta t=1$
3. Specify α , β and Γ as functions of a parameter vector Θ (implement the restrictions you require). Then we have

$$A = \alpha - I$$
$$B = \beta$$
$$G = \Gamma,\ \Omega = GG` = \Gamma\Gamma`$$

The module PAR is given as

```
START PAR(OMEGA,A,B,SIGMA,MUE,THETA);

ALPHA= <FUNCTION OF THETA> ;
P=NCOL(ALPHA);
A=ALPHA-I(P);
B= <FUNCTION OF THETA>;
G= <FUNCTION OF THETA>;
OMEGA=G*G`;
MUE= <FUNCTION OF THETA> ;
SIGMA=<FUNCTION OF THETA> ;

FINISH;
```

The derivatives $\partial A/\partial\Theta$, $\partial B/\partial\Theta$ etc. which are required in PARD are the same as $\partial\alpha/\partial\Theta$, $\partial\beta/\partial\Theta$ etc. The modules PARM and PARMD corresponding to the measurement model are the same as usual.

Problems with the estimation of discrete time panels using LISREL

Discrete time panel models can be estimated with LISREL, too (see, e.g. Arminger, 1990). There are, however, **serious restrictions** of this approach:
1. The parameter matrices cannot be nonlinearly restricted (LSDE also allows inequality restrictions, intervals etc.)
2. The expectation of the latent variables must vanish in LISREL ($E(y_{ni}) = 0$). This is not always appropriate, since the expectation of the initial condition $E[y_{n0}] = \mu$ can be nonvanishing and the deterministic exogenous variables x_{nj} are not zero.

Then, the means $\mu_{ni} = \alpha^i\mu + \sum_{j=0}^{i-1} \alpha^{i-j-1}\beta x_{nj}$
are nonvanishing and are specifically restricted (they obey the Liapunov equation). These restrictions cannot be implemented in LISREL , even if you are using intercepts.
3. LSDE utilizes the special temporal structure of the model whereas LISREL is a model designed for cross sectional data. Therefore, if the number of time points is large and you are analyzing multivariate models, the LISREL matrices become very large (e.g., if p=10 and T=10, LISREL`s BETA is 100×100, whereas α is only 10×10). Furthermore, the time series case cannot be treated in LISREL, since the data are temporally dependent and cannot be considered as a cross-section (independence!) .
4. Rewriting the panel model in terms of LISREL produces a mental deviation from the original specification. Although this is only a psychological remark, in my experience this leads to a static way of thinking disregarding the special features of dynamic models.

Appendix II

State Space Representations of ARMA and ARMAX Models and Models Containing Individual Specific Effects

1. ARMA-Models

A continuous time ARMA model (autoregressive-moving-average model) is given as the differential equation (suppressing index n)

$$\Theta(d/dt)\, y(t) = \Phi(d/dt)\zeta(t)$$

Here Θ and Φ are differential operators of the form:

$$\Theta = \sum_{j=0}^{P} a_j (d/dt)^j \quad ; \qquad a_P = I$$

$$\Phi = \sum_{j=0}^{Q} g_j (d/dt)^j \quad ; \qquad Q \leq P\text{-}1$$

The numbers P and Q are the orders of the ARMA (P,Q) model. If g_1=g_2=.....=g_{P-1}=0 an AR(P) model is obtained. In other words, we have the P-th order differential equation:

$$y^{(P)} + a_{P-1} y^{(P-1)} + \ldots\ldots\ldots + a_0 y \;=\; g_0\zeta + g_1\zeta` + \ldots\ldots\ldots\ldots + g_{P-1}\zeta^{(P-1)}$$

This equation can be written in state space form as follows:

$$d/dt \begin{bmatrix} y_1 \\ y_2 \\ . \\ . \\ . \\ . \\ y_P \end{bmatrix} = \begin{bmatrix} 0 \;\ldots & & & & & -a_0 \\ I & & & & & -a_1 \\ 0 & I & & & & . \\ 0 & . & I & & & . \\ 0 & \ldots & & I & & . \\ 0 & \ldots & & & I & . \\ 0 & \ldots\ldots & & & I & -a_{P-1} \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ . \\ . \\ . \\ . \\ y_P \end{bmatrix} + \begin{bmatrix} g_0 \\ . \\ . \\ . \\ . \\ . \\ g_{P-1} \end{bmatrix} \zeta$$

$$z = [0, \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots, I] \begin{bmatrix} y_1 \\ y_2 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ y_P \end{bmatrix}$$

In short we have

$$\begin{aligned} d/dt\, Y &= AY + G\zeta \\ z &= Hy \end{aligned}$$

and this is the form required by LSDE.

In the stationary case y(t) is a stochastic process with spectral representation

$$y(t) = \int_{-\infty}^{+\infty} e^{i\lambda t} f(\lambda) \Psi(d\lambda)$$

where $f(\lambda) = \Theta^{-1}(i\lambda)\ \Phi(i\lambda)$ is a rational spectral characteristic and $\Psi(d\lambda)$ is the orthogonal spectral measure with

$$\begin{aligned} &E[\Psi(d\lambda)] = 0 \\ &E[\Psi(d\lambda)\Psi^*(d\lambda')] = (d\lambda/2\pi)\ \delta_{\lambda\lambda'} \end{aligned}$$

(see, e.g. Liptser and Shiryayev, 1978, vol. II, ch. 15).

2. ARMAX-Models

These models include exogenous variables in the form:

$$\Theta(d/dt)\, y(t) = \Phi(d/dt)\, \zeta(t) + \chi(d/dt)\, x(t)$$

Θ and Φ are given as above and $\chi = \sum c_j (d/dt)^j$.

Using the decomposition $y(t) = \tilde{y}(t) + u(t)$ we obtain:

$$\Theta\tilde{y} + \Theta u = \Phi\zeta + \chi x$$

If we seperate the deterministic and stochastic parts this equation is implied by the two equations:

$$\Theta u = \chi x \quad \text{and} \quad \Theta \tilde{y} = \Phi \zeta$$

As shown in 1. these equations can be written in state space form as:

$$\begin{cases} d/dt\, U = AU + Cx, & C=[c_0^{\backprime},\dots\dots,c_{P-1}^{\backprime}]^{\backprime} \\ u = u_P = [0,\dots\dots,I]U, & U=[u_1^{\backprime},\dots\dots u_P^{\backprime}\,]^{\backprime} \end{cases}$$

and

$$\begin{cases} d/dt\, \tilde{Y} = A\,\tilde{Y} + G\,\zeta \\ \tilde{y} = \tilde{y}_P = [0,\dots\dots,I]\tilde{Y} \end{cases}$$

where A and G are defined above.
From this we obtain the state space form:

$$d/dt \begin{bmatrix} \tilde{Y} \\ U \end{bmatrix} = \begin{bmatrix} A & 0 \\ 0 & A \end{bmatrix} \begin{bmatrix} \tilde{Y} \\ U \end{bmatrix} + \begin{bmatrix} 0 \\ C \end{bmatrix} x + \begin{bmatrix} G \\ 0 \end{bmatrix} \zeta$$

$$y = \big[0,\dots\dots,I,0,\dots\dots,I\,\big] \begin{bmatrix} \tilde{Y} \\ U \end{bmatrix}$$

Again, this is of the form required by LSDE.

3. Models with individual specific effects

We consider the model

$$d/dt\; y_n(t) = Ay_n(t) + Bx_n(t) + \pi_n + G\zeta_n(t)$$

where π_n are time independent random effects (see Hamerle, Nagl, Singer (1991b)). They are i.i.d. as $N(0,V_\pi)$. We can write:

$$d/dt \begin{bmatrix} y_n(t) \\ \pi_n(t) \end{bmatrix} = \begin{bmatrix} A & I \\ 0 & 0 \end{bmatrix} \begin{bmatrix} y_n(t) \\ \pi_n(t) \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} x + \begin{bmatrix} G & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \zeta \\ 0 \end{bmatrix}$$

V. Appendix

$$z_{ni} = \begin{bmatrix} I , 0 \end{bmatrix} \begin{bmatrix} y_{ni} \\ \\ \pi_{ni} \end{bmatrix}$$

For the initial condition we set $\mu = \begin{bmatrix} \mu_1 \\ 0 \end{bmatrix}$ $\Sigma = \begin{bmatrix} \Sigma_{11} & 0 \\ 0 & V_\pi \end{bmatrix}$.

This is the form required by LSDE.

References

Anderson, T.W. (1971), The Statistical Analysis of Time Series, Wiley, New York

Arato , M. (1982), Linear Stochastic Systems with Constant Coefficients, Springer, Berlin, Heidelberg, New York

Arminger, G. (1986), Linear Stochastic Differential Equation Models for Panel Data with Unobserved Variables, in: Tuma, N.B. (ed.), Sociological Methodology, Washington, 187-212

Arminger, G., Müller, F. (1990), Lineare Modelle zur Analyse von Paneldaten, Westdeutscher Verlag

Arnold, L. (1973), Stochastische Differentialgleichungen, Oldenbourg, München, Wien

Arnold, L. (1974), Stochastic Differential Equations, Wiley, New York.

Arnold, VI. (1973), Ordinary Differential Equations, MIT Press, Cambridge (Mass.), London

Azencott, R., Dacunha-Castelle, D. (1986), Series of Irregular Observations, Springer, New York

Bartlett, M.S. (1946), On the theoretical specification and sampling properties of autocorrelated time-series, Journal of the Royal Statistical Society (Supplement) , 7, 27-41

Bartlett, M.S. (1955, 1978), An Introduction to Stochastic Processes, Cambridge University Press, Cambridge

Bergstrom, A.R. (Hrsg., 1976), Statistical Inference in Continuous Time Models, North Holland, Amsterdam

Bergstrom, A.R. (1988), The History of Continuous-Time Econometric Models, Econometric Theory, 4, 365-383

Bergstrom, A.R., Wymer, C.R. (1976), A Model of Disequilibrium Neoclassical Growth and its Application to the United Kingdom, in: Bergstrom (1976).

Bien, W., Schmidt, P., Schürhoff, R. (1990), Strukturgleichungsmodelle für Paneldaten und die Schätzung von Differentialgleichungsmodellen, preprint, Universität Gießen.

Blossfeld, H.P., Hamerle, A., Mayer, K.U. (1986), Ereignisanalyse, Campus, Frankfurt.

Caines, P.E. (1988), Linear Stochastic Systems, Wiley, New York

Coleman, J.S. (1968), The Mathematical Study of Change, in: Blalock, H.M., Blalock, A.B. (eds.) Methodology in Social Research, Mc Graw Hill, New York, 428-478.

Dembo, A., Zeitouni, O. (1986), Parameter Estimation of Partially Observed Continuous Time Stochastic Processes Via the EM Algorithm, Stochastic Processes and their Applications, 23, 91-113

Dempster, A.P., Laird, N.M., Rubin, D.B. (1977), Maximum Likelihood from Incomplete Data via the EM Algorithm, J. Royal Statistical Society B, 39, 1-38

Dennis, J.E. Jr, Schnabel, R.B. (1983), Numerical Methods for Unconstrained Optimization and Nonlinear Equations, Prentice Hall, Englewood Cliffs

Dieterich, W., Fulde, P., Peschel, I. (1980), Theoretical models for superionic conductors, Advances in Physics, 29, 3, 527-605

Doreian, P., Hummon, N.P. (1976), Modelling Social Processes, Elsevier, New York, Oxford, Amsterdam

Einstein, A. (1905), Über die von der molekularkinetischen Theorie der Wärme geforderte Bewegung von in ruhenden Flüssigkeiten suspendierten Teilchen, Annalen der Physik, 4, 17, 549-560

Gelb, A. (ed.; 1974), Applied Optimal Estimation, MIT Press, Cambridge, Mass., London

Golub, G.H., Van Loan, C. (1983), Matrix Computations, John Hopkins University Press, Baltimore

Goodrich, R.L. (1978) , Stochastic Models of Human Growth, Ph.D. Thesis, Division of Applied Sciences, Harvard University, Cambridge, MA

Hamerle, A., Nagl, W., Singer, H. (1991a), Problems with the estimation of stochastic differential equations using structural equations models, Journal of Mathematical Sociology, 16, 3, 201-220

Hamerle, A., Nagl, W., Singer, H. (1991b), Parameter Estimation in Continuous Time Dynamic Systems using Panel Data, submitted to Econometric Theory.

Hansen, L.P., Sargent, T.J. (1983), The dimensionality of the aliasing problem in models with rational spectral density, Econometrica, 51, 377-387

Hosking, J.R.M. (1980), The Multivariate Portmanteau Statistic, Journal of the American Statistical Society, 75, 371, 602-608

Jagger, G. (1977), Discussion on the papers by Mr. Campbell and Prof. Walker, Dr. Morris and Dr. Tong, Journal of the Royal Statistical Society A, 140, 4, 454-455

Jazwinski, A.H. (1970), Stochastic Processes and Filtering Theory, Academic Press, New York

Jennrich, R.I., Bright, P.B. (1976), Fitting Systems of Linear Differential Equations Using Computer Generated Exact Derivatives, Technometrics, 18, 4, 385-392

Jöreskog, K., Sörbom, D. (1981), LISREL V, National Educational Resources, Chicago

Jones, R.H. (1981), Fitting a continuous time autoregression to discrete data, in: Findley, D.F. (ed., 1981), Applied Time Series Analysis, II, Academic Press, 651-682

Jones, R.H. (1984), Fitting Multivariate Models to Unequally Spaced Data, in: Parzen, E. (ed.; 1984), Time Series Analysis of Irregularly Observed Data, Springer, New York, 158-188

Jones, R.H., Tryon, P.V. (1987), Continuous time series models for unequally spaced data applied to modeling atomic clocks, SIAM J. Sci. Stat. Comput., 8, 1, 71-81

Kalbfleisch, J.D., Lawless, J.F. (1985), The analysis of panel data under a Markov assumption, JASA, 80, 863-871

Kappler, E. (1931), Versuche zur Messung der Avogadro-Loschmidtschen Zahl aus der Brownschen Bewegung einer Drehwaage, Annalen der Physik, 11, 233-256

Koehler, A.B., Murphree, E.S. (1988), A Comparision of the Akaike and Schwarz Criteria for selecting Model Order, Applied Statistics, 37, 2, 187-195

Koopmans, T.C. (1950), Models involving a continuous time variable, in: Koopmans, T.C. (ed.), Statistical inference in dynamic economic models

Langevin, P. (1908), Sur la theorie du mouvement brownien, C.R. Acad. Sci. Paris, 146, 530-533

Le Breton, A. (1976), On continuous and discrete sampling for parameter estimation in diffusion type processes, Mathematical Programming Study 5, 124-144

Lewis, F.L. (1986), Optimal Estimation, Wiley, New York

Liptser, R.S., Shiryayev, A.N. (1977, 1978), Statistics of Random Processes, Band I und II, Springer, New York, Heidelberg, Berlin

Louis, T.A. (1982), Finding the Observed Information Matrix when Using the EM Algorithm, Journal of the Royal Statistical Association B, 44, 2, 226-233

Mardia, K.V., Kent, J.T., Bibby, J.M. (1979), Multivariate Analysis, Academic Press, London

McDonald, R.P., Swaminathan, H. (1973), A simple matrix calculus with applications to multivariate statistics, General Systems, XVIII, 37-54

Merton, R.C. (1990), Continuous-time finance, Basil Blackwell, Cambridge, Mass.

Möbus, C. Nagl, W. (1983), Messung, Analyse und Prognose von Veränderungen. In: Hypothesenprüfung, Band 5 der Serie Forschungsmethoden der Psychologie der Enzyklopädie der Psychologie, Hogrefe, Göttingen

Otter, P.W. (1986), Dynamic structural systems under indirect observation: Identifiability and estimation aspects from a system theoretic perspective, Psychometrica 51, 415-428

Phadke, M.S., Wu, S.M. (1974), Modeling of Continuous Stochastic Processes from Discrete Observations with Application to Sunspots Data , Journal of the American Statistical Association, 69, 346, 325-329

Phillips, A.W. (1959), The estimation of parameters in systems of stochastic differential equations, Biometrika, 46, 76-76.

Phillips, P.C.B. (1972), The Structural Estimation of a Stochastic Differential Equation System, in: Bergstrom (1976).

Phillips, P.C.B. (1973), The Problem of Identification in Finite Parameter Continuous Time Models, in: Bergstrom (1976).

Priestley, M.B. (1988), Non-linear and Non-stationary Time Series Analysis, Academic Press, London

Rao, C.R. (1973), Linear Statistical Inference and its Applications, 2. edition, Wiley, New York

Rauch, H.E., Tung, F., Striebel, C.T. (1965), Maximum Likelihood Estimates of Linear Dynamic Systems, AIAA Journal, 3, 8, 1445-1450

Rissanen, J. (1978), Modeling by shortest data description, Automatica, 14, 465-471

Sargan, J.D. (1976), Some Discrete Approximations to Continuous Time Stochastic Models, in: Bergstrom (1976).

SAS Institute Inc. (1989), SAS/IML Software: Usage and Reference, Version 6, Cary, NC

Schwarz, G. (1978), Estimating the dimension of a model, Annals of Statistics, 6, 2, 461-464

Schweppe, F. (1965), Evaluation of Likelihood Functions for Gaussian Signals, IEEE Transactions on Information Theory, 11, 61-70

Simon, H.A. (1952), A formal theory of interaction in social groups, American Sociological Review, 17, 202-211

Singer, H. (1988), Parameter Estimation in Stochastic Differential Equations: Comparision of Some Estimators, Diskussionsbeiträge 112, Universität Konstanz

Singer, H. (1990), Parameterschätzung in zeitkontinuierlichen dynamischen Systemen, Hartung-Gorre-Verlag, Konstanz.

Singer, H. (1991b), Continuous Time Dynamical Systems with Sampled Data, Errors of Measurement and Unobserved Components, Journal of Time Series Analysis, in print

Singer, H. (1991c), The Aliasing Phenomenon in Visual Terms, Journal of Mathematical Sociology, in print

Singer, H. (1992), Zeitkontinuierliche Dynamische Systeme, Campus, Frankfurt

Singer, H., Hautzinger, M. (1988), Kausalattributionen, Depressivität und kritische Lebensereignisse als Stochastischer Prozeß, in: Kammer, D., Hautzinger, M. (Hrsg., 1988), Kognitive Depressionsforschung, Huber, Bern, Stuttgart, Toronto

Sinha, N.K., Lastman, G.J. (1982), Identification of continuous-time multivariable systems from sampled data, International Journal of Control, 35, 1, 117-126

Soong, T.T. (1973), Random Differential Equations in Science and Engineering, Academic Press, New York, London

Thomas, E.A.C., Martin, J.A. (1976), Analyses of Parent-Infant-Interaction, Psychological Review, 83, 2, 141-156

Tuma, N.B., Hannan, M. (1984), Social Dynamics, Academic Press, New York

Uhlenbeck, L.S., Ornstein, L.S. (1930), On the Theory of Brownian Motion, in: Wax, N. (ed., 1954), Selected Papers on Noise and Stochastic Processes, Dover, New York

Van Kampen, N.G. (1981), Stochastic Processes in Physics and Chemistry, North Holland, Amsterdam, New York Oxford

Waldmeier, M. (1961), The Sunspot-Activity in the Years 1610-1960, Schulthess, Zürich

Watson, M.W., Engle, R.F. (1983), Alternative Algorithms for the Estimation of Dynamic Factor, Mimic and varying Coefficient Regression Models, Journal of Econometrics, 23, 385-400

Wilcox, R.M. (1967), Exponential Operators and Parameter Differentiation in Quantum Physics, Journal of Mathematical Physics, 8, 962

Yule, G.U. (1927), On a method for investigating periodicities in disturbed series with special reference to Wolfer`s sunspot numbers, Philos. Trans. Roy. Soc. London, Ser. A, 226, 267-298

INDEX