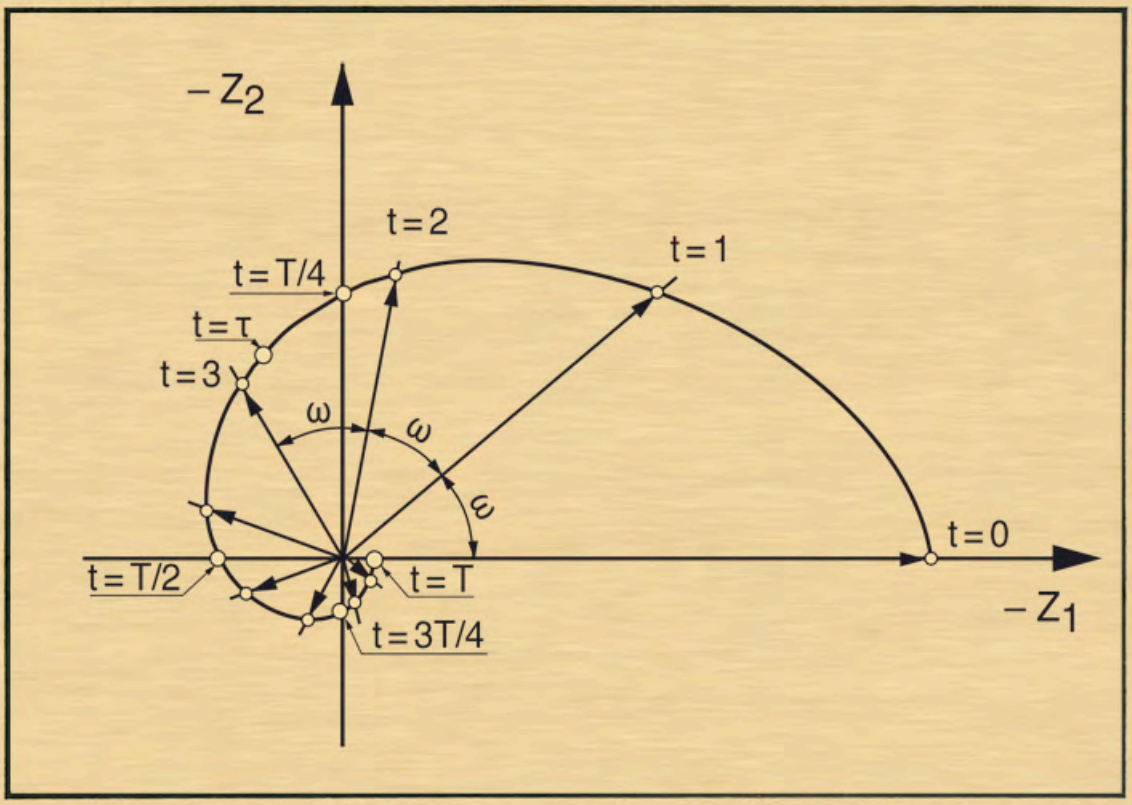
DEUTSCHES KLIMARECHENZENTRUM

TECHNICAL REPORT No.1



The Pop Manual
(POPs=Principal Oscillation Patterns)

by

Fergus Gallagher * Hans von Storch

Reiner Schnur * Gerhard Hannoschöck

Edited by: Modellberatungsgruppe

Hamburg, August 1991

THE POP MANUAL

(POPs = PRINCIPAL OSCILLATION PATTERNS)

by

Fergus Gallagher, Hans von Storch, Reiner Schnur

Max-Planck-Institut für Meteorologie, Hamburg

and

Gerhard Hannoschöck

Hanno-Consult, Hamburg

Version 3.0
August 1991

ISSN 0940-9327

Contents

1 INTRODUCTION

1.1 ON POPs AND ON THIS MANUAL

Principal Oscillation Patterns (POPs) are a technique for identifying the characteristic time-scales and characteristic spatial patterns of a multi-component vector time-series. POPs may be seen as the normal modes of a linear dynamic system whose system matrix is estimated from the data.

The POP technique was developed at the *Max-Planck-Institut fuer Meteorologie* (Hasselmann (1988) and Storch et al. (1988)[1]) and is nowadays a routinely used tool. In Appendix 6, a list of applications of the POP method is given. A standard POP program has been implemented at the *Deutsches Klimarechenzentrum (DKRZ)*. It is the purpose of the present manual to describe this POP program. See Section 1.3 on how to use this manual.

The POP approach has been used to design forecast schemes (Xu and Storch, 1990, Storch and Xu, 1990); recently two generalizations were proposed: *cyclostationary* POPs (Blumenthal, 1991) and *complex* POPs (Bürger, 1991). These aspects are not covered by the POP program and by this manual.

---

[1] For the references see Appendix 6.

1.2 VERSION OF CODE

The POP program this manual refers to is "Version 3". It has been installed in January 1991 and differs from Version 2 by

- a completely new user interface reducing the parameter input to one line of text,
- the processing of data gaps in the calculation of moments,
- the smallest possible covariance matrix for EOF expansion,
- the exclusive use of *Extra Code* format in input and output files making the POPs program open to the world of *Extra-Code modular system*,
- dynamically declared array dimensions, thus the user has no problems with setting sufficient array bounds any more,
- the explained local variance is computed to give an overview where a cer-tain POP explains a significant fraction of the variability,
- the FORTRAN code has become more readable and updatable, e.g. the sub-routines are in a *top-down* order and there is more modular structure. But for reasons of limited time, a complete re-engineering could not be done, so the well-known principles of modular programming (e.g., minimality, program-independent interface, single purpose) was brought only in parts to the program.

1.3 HOW TO USE THIS MANUAL

This section is intended to give a short guideline on how to use this manual. The flow chart on the next page should contain the most important routes on your way through the POP manual.

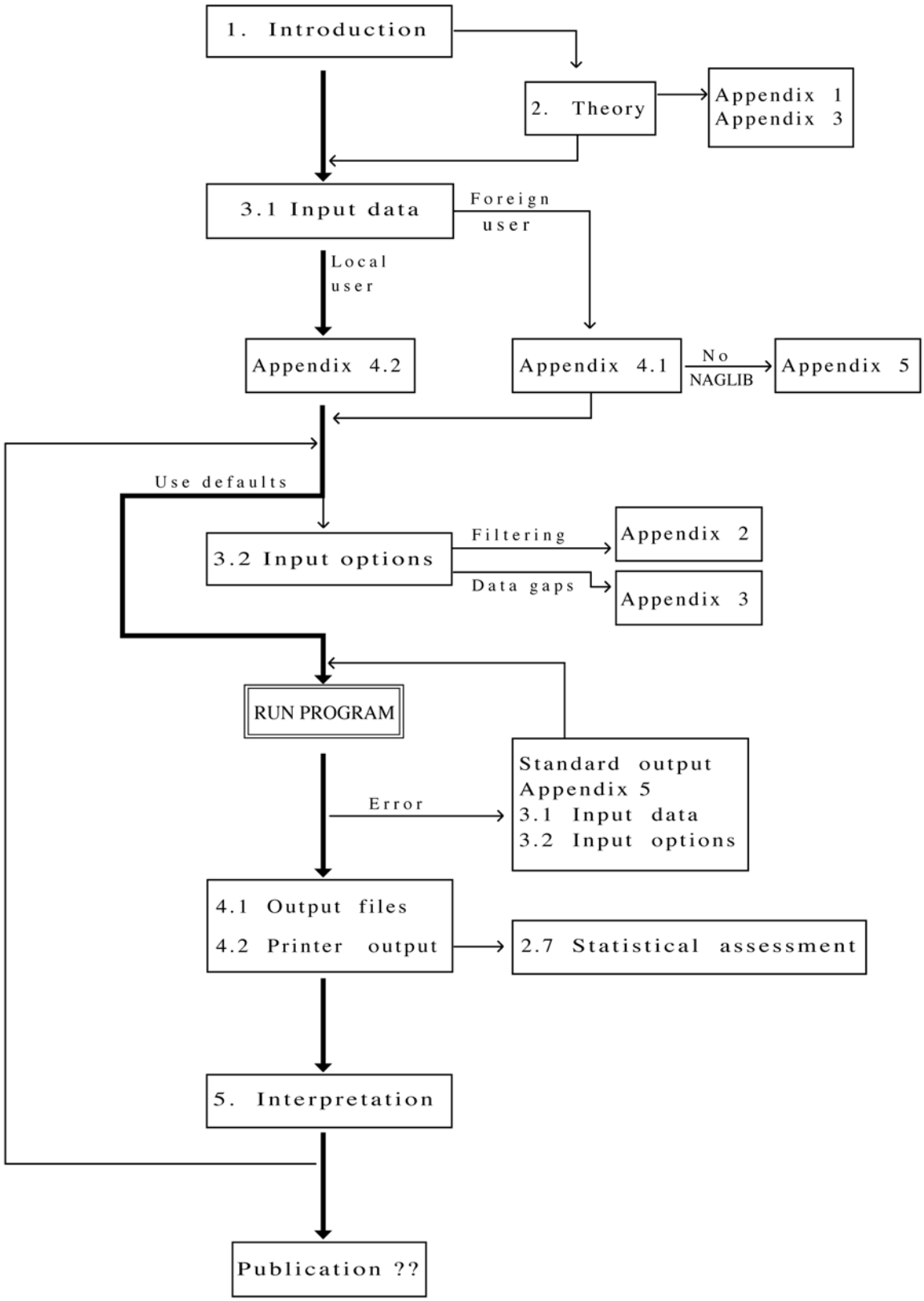
Are you unpatient - or do you belong to the group of users of computer software that first try out a program and then, after the program crashed, read the manual? Then, if you are a user at the *DKRZ*, the quickest way to run the POP program (thick arrows in diagram) is to read how the input data are supplied (Section 3.1) and how to access the executable program (Local user's guide, Appendix 4.C). All options have defaults so you may just start the program and look at the printer output (Section 4.1) and the output files (Section 4.2).

If you are a foreign user without account on the computers at the *DKRZ* you will first have to build the executable from source before you can start the POP program. Read Appendix 4.B on how to do this. The POP program has been written in standard FORTRAN so that it can be used on any computer.

If you want to specify options to the POP program others than the defaults then you have to read Section 3.2 and possibly Appendix 2 for time filtering and Appendix 3 for the case that your data contain gaps.

If the POP program abnormally aborts execution then, hopefully, the internal checking has produced an adequate error message. If one of the NAGLIB routines caused the termination then see Appendix 5 for the meaning of the error codes.

Last but not least, if you want to know what POPs are and how a POP analysis is carried out by the POP program then read Section 2. An outline of how to interpret the results of a POP analysis and an example are given in Section 5. Appendix 6 contains a list of publications in which the POP analysis technique has been applied.


1. Introduction
2. Theory
Appendix 1
Appendix 3
3.1 Input data
Foreign user
Local user
Appendix 4.2
Appendix 4.1
No NAGLIB
Appendix 5
Use defaults
3.2 Input options
Filtering
Appendix 2
Data gaps
Appendix 3
RUN PROGRAM
Standard output
Appendix 5
3.1 Input data
3.2 Input options
Error
4.1 Output files
4.2 Printer output
2.7 Statistical assessment
5. Interpretation
Publication ??

2 OUTLINE THEORY.

The POP method is the linear approximation of the more general 'Principal Interaction Patterns (PIPs)' (see Hasselmann, 1988) method to estimate the normal modes of a system with (possibly) unknown dynamics, or very many degrees of freedom. The theory given in this section is considered to be the essential 'core' necessary for using the POP program - in particular, the theory is given *as used by the program.*

2.1 EOF EXPANSION

If we have a system X(r,t) with n degrees of freedom (i.e., number of data points), where r is a spatial index $1 \leq r \leq n$ and t is a time index, then this system will have n normal modes. The use of the term 'spatial' is only used for convenience as most uses of the program are expected to be for calculating spatial POPs. The space dimension here means all data points at a given time whether they be real-space data, Fourier coefficients, etc. Also, this index does not have to indicate the physical organization of the data - it is just an arbitrary label.

For a large system, these normal modes represent a huge amount of data from which the important characteristics, i.e. the dominant normal modes, may be hard to extract. The method employed is to transform the data to Empirical Orthogonal Function (EOF) space which is simply a set of (time-invariant) orthonormal functions which completely spans the real space. The EOFs are defined as the eigenvectors of the covariance matrix of X(r,t) i.e. the eigenvectors of

$$C(r,r') = \langle X(r,t), X(r',t) \rangle$$

where $< \cdot >$ denotes a time average (see Appendix 3). The EOFs are ordered such that the first EOF accounts for the most variance, the second EOF accounts for most of the residual variance of the data which is not accounted for by the first, and so on. If $F^i(r)$ are the EOFs, then the data set X(r,t) may be expanded as:

$$X(r,t) = \sum_{i=1}^{n} \bar{x}_i(t) F^i(r) \qquad (1)$$

where $\bar{x}_i(t)$ are a set of coefficients, called *Principal Components*, given by

the projection of X(.,t) onto the EOFs F^i:

$$\bar{x}_i(t) = \sum_{r=1}^{n} X(r,t)F^i(r).$$

Thus, $\bar{x}(t)$ is the same vector as X but expressed in EOF-space coordinates and not in Euclidean coordinates . In the following, all quantities with a bar are in EOF-space.

We may now truncate the EOF-space to m-dimensions ($m \leq n$) with a judicious choice of m (the fraction of the variance of the original data set accounted for by the first m EOFs is the sum of the first m eigenvalues). In doing so we hope to preserve most, if not all, of the signal(s) we are interested in while removing noisy components. With this reduction we have,

$$X(r,t) \longrightarrow x(r,t) = \sum_{i=1}^{m} \bar{x}_i(t)F^i(r) \qquad (2)$$

i.e. we now consider as data space the m-dimensional subspace spanned by the EOFs $F^1,...,F^m$ with coordinates $\bar{x}_1,...,\bar{x}_m$.

Note that if no EOF expansion is performed to reduce the dimension of the problem the POP analysis is done in the original space and in the next sections the parts concerning EOFs can just be skipped.

2.2 THE LINEAR VECTOR PROCESS

We now model the time evolution of the system by the linear vector process:

$$\frac{d}{dt}\,\mathbf{x}(t) = \mathbf{A}\mathbf{x}(t) + \xi(t) \tag{3}$$

where **A** is some constant (n x n) matrix, **x**(t) is the n-dimensional column vector formed from x(r,t) and $\xi(t)$ is some noise vector driving the system.
Discretization of (3) leads to

$$\mathbf{x}(t+1) = \mathbf{B}\mathbf{x}(t) + \xi(t) \tag{4}$$

with some (n x n) matrix **B** depending on the matrix **A** and the details of the fine discretisation.
Using the transformation (2) we can transform equation (4) to truncated EOF-space,

$$\bar{\mathbf{x}}(t+1) = \bar{\mathbf{B}}\,\bar{\mathbf{x}}(t) + \bar{\xi}(t) \tag{5}$$

where $\bar{\mathbf{B}}$ is the matrix **B** transformed to the truncated EOF-space (i.e. $\bar{\mathbf{B}} = \mathbf{F}^T\mathbf{B}\mathbf{F}$ with F the (n x m) matrix with $F^1,...,F^m$ as columns, F^T the transpose matrix of F) and $\bar{\xi}(t)$ is $\xi(t)$ in the truncated EOF-space.

For the case $\bar{\xi}(t) = \mathbf{O}$, the above normally has a in the most cases complete set of linearly independent, but not necessarily orthogonal, eigenmodes with eigenvectors $\bar{\mathbf{p}}^k$, k=1,...m, and corresponding eigenvalues λ_k such that:

$$\bar{\mathbf{B}}\,\bar{\mathbf{p}}^k = \lambda_k\,\bar{\mathbf{p}}^k \tag{6}$$

The eigenvectors of $\bar{\mathbf{B}}$ are called **Principal Oscillation Patterns**, **POP**s. The matrix $\bar{\mathbf{B}}$ is, in general, not symmetric so that some of the eigenvalues may be complex. Since $\bar{\mathbf{x}}$ and $\bar{\mathbf{B}}$ are real, any complex eigenvalues and corresponding eigenvectors occur in conjugate pairs. In the rest of this manual we refer to POPs with real eigenvalues as 'real POPs' and to 'complex POPs' as those associated with complex eigenvalues.
Thus we may reconstruct the time series in terms of the eigenvectors,

$$\bar{\mathbf{x}}(t) = \sum_{k=1}^{m} a_k(t)\bar{\mathbf{p}}^k \quad . \tag{7}$$

If we transform the POPs to (complex) Euclidean space using,

$$\mathbf{p}^k = \sum_{i=1}^{m} \overline{p}_i^k \, F^i \,, \tag{8}$$

we obtain the representation of the time series in Euclidean space

$$\mathbf{x}(t) = \sum_{k=1}^{m} a_k(t)\mathbf{p}^k \,. \tag{9}$$

In the remainder of this manual we refer to these POPs (though of course the same applies to EOF-space POPs).
The noise $\xi_k(t)$ driving POP k is defined by

$$\xi_k(t) = \sum_{k=1}^{m} \xi_k(t)\mathbf{p}^k \,. \tag{10}$$

2.3 ESTIMATING THE PROCESS MATRIX $\bar{B}$

If the dynamics of the system are known, then matrix $\bar{B}$ may be calculated analytically but, as this is generally not the case, it may be estimated by fitting it to the process (5). It can be shown that,

$$\bar{B} = \bar{C}_1 \bar{C}_0^{-1} \tag{11}$$

where C_0, C_1 are the lag-0 and lag-1 covariance matrices:

$$\bar{C}_0 = \langle \bar{x}(t)\ \bar{x}^T(t) \rangle \tag{12}$$

$$\bar{C}_1 = \langle \bar{x}(t+1)\ \bar{x}^T(t) \rangle \tag{13}$$

with $\bar{x}$ written as column vector and ^{T} indicating transposition.

2.4 MAKING THE POPS UNIQUE

The complex eigenvectors $\mathbf{p}^k$ are only defined to an arbitrary factor $m_k exp\ i\theta_k$ i.e., if $\mathbf{p}_k$ is an eigenvector, then so is $m_k\ exp\ i\theta_k\ \mathbf{p}^k$. We need to choose some standard normalization for the POPs so that POPs from different systems may be compared directly. The POPs cannot be defined to be mutually orthonormal (i.e. $\mathbf{p}^j \cdot \mathbf{p}^k = \delta^k_j$, for some suitably defined scalar product[1]) because the POP matrix $\bar{B}$ was not symmetric, but each POP can itself be normalized, choosing m_k such that $\mathbf{p}^k \cdot \mathbf{p}^k = n$. This normalization ensures that the POPs have similar magnitudes independent of the choice of n, in the sense that at any particular location the POP has the same magnitude - the more obvious normalization $\mathbf{p}^k \cdot \mathbf{p}^k = 1$ would mean that the magnitudes of POP patterns would depend on n.

The phase angle θ_k may be chosen in two ways in the present version of the POPs program. The first choice is to require that the real and imaginary components of each POP ($\mathbf{p}^k = \mathbf{p}^{k'} + i\mathbf{p}^{k''}$) be orthogonal with respect to Euclidean geometry. Having found one orthogonal pair, we would get another one by multiplying $\mathbf{p}^k$ by i which results in the POP $-\mathbf{p}^{k''} + i\mathbf{p}^{k'}$. This indeterminacy is obviously resolved (up to a minus sign) by defining $\mathbf{p}^{k'}$ as being the component with the larger vector norm, i.e.

$$\mathbf{p}^{k'} \cdot \mathbf{p}^{k''} = 0 \tag{14}$$

$$\| \mathbf{p}^{k'} \| \geq \| \mathbf{p}^{k''} \| \ . \tag{15}$$

The second choice of θ_k is to rotate the POPs so that the real and imaginary components of the forcing are statistically orthogonal,

$$< \xi^{k'}(t)\ \xi^{k''} > = 0 \tag{16}$$

$$< \xi^{k'}(t)\ \xi^{k'}(t) > \geq < \xi^{k''}(t)\ \xi^{k''}(t) > \tag{17}$$

with condition (17) corresponding to condition (15). In general, the rotation angles corresponding to the two normalization choices will differ.

---

[1] A suitable definition is $\mathbf{a} \cdot \mathbf{b} = \Sigma\ a_i b_i^*$ where $\mathbf{a}$ and $\mathbf{b}$ are two vectors of order n and * denotes the complex conjugate of a complex number.

2.5 REAL VECTOR PRESENTATION OF POPS

The complex conjugate of a POP describes the same signal as the POP itself. Thus, if it is assumed that the complex POP matrix $\mathbf{P} = (\mathbf{p}^1 | \ldots | \mathbf{p}^n)$ has m complex eigenvectors (which occur in complex conjugate pairs) and n - 2m real eigenvectors, the real matrix **R** contains the same information as **P** does:

$$\mathbf{R} = (\mathbf{r}^1 | \ldots | \mathbf{r}^n)$$

with

$$\left.\begin{array}{ll} \mathbf{r}^{2j-1} = Re(\mathbf{p}^{2j-1}) \\ \mathbf{r}^{2j} = Im(\mathbf{p}^{2j-1}) \end{array}\right\} \quad \text{for } j = 1, \ldots, m$$
$$\mathbf{r}^j = \mathbf{p}^j \qquad \text{for } j = 2m+1, \ldots, n \quad , \tag{18}$$

The **Principal Adjoint Patterns** (PAPs) $\mathbf{q}^k$ are the eigenvectors of the transposed matrix $\mathbf{B}^T$. In Appendix 1 adjoint patterns and some of their properties are introduced.

The adjoint patterns may be calculated from **R** using the following relationship:

if $\mathbf{S} = (\mathbf{s}^1 | \mathbf{s}^2 | \ldots | \mathbf{s}^n) = (\mathbf{R}^T)^{-1}$ then

$$\left.\begin{array}{ll} \mathbf{q}^{2j-1} = \frac{1}{2}(\mathbf{s}^{2j-1} + i\mathbf{s}^{2j}) \\ \mathbf{q}^{2j} = \frac{1}{2}(\mathbf{s}^{2j-1} - i\mathbf{s}^{2j}) \end{array}\right\} \quad \text{for } j = 1, \ldots, m$$
$$\mathbf{q}^j = \mathbf{s}^j \qquad \text{for } j = 2m+1, \ldots, n \tag{19}$$

Proof:

The vectors $\mathbf{q}^j$ are the unique solution of $\mathbf{p}^k \cdot \mathbf{q}^{j*} = \delta_{kj}$ (k=1,n) (cf. Appendix 1, (A.5)). It has to be shown that this relation holds if $\mathbf{q}^j$ is replaced by using (18):

$$\mathbf{p}^{2k-1} \cdot \mathbf{q}^{2j-1*} = \frac{1}{2}\left[\mathbf{r}^{2k-1} + i\mathbf{r}^{2k}\right] \cdot \left[\mathbf{s}^{2j-1} - i\mathbf{s}^{2j}\right]$$
$$= \frac{1}{2}(\mathbf{r}^{2k-1} \cdot \mathbf{s}^{2j-1} + \mathbf{r}^{2k} \cdot \mathbf{s}^{2j} + i[\mathbf{r}^{2k} \cdot \mathbf{s}^{2j-1} - \mathbf{r}^{2k-1} \cdot \mathbf{s}^{2j}])$$
$$= \frac{1}{2}(\delta_{kj} + \delta_{kj} + i[0 - 0]) = \delta_{kj}$$

since **S** is the inverse matrix of $\mathbf{R}^T$.

Similarly, it is shown that $\mathbf{p}^{2k} \cdot \mathbf{q}^{2j*} = \delta_{kj}$. ■

If a state $\mathbf{x}$ is expanded by the POPs, $\mathbf{r}^k$, in real vector representation,

$$\mathbf{x} = \sum_{k=1}^{n} b_k \mathbf{r}^k \quad , \tag{20}$$

the real coefficients b_k are given by

$$\left.\begin{aligned} b_{2j-1} &= 2Re(a_{2j-1}) \\ b_{2j} &= -2Im(a_{2j-1}) \end{aligned}\right\} \quad \text{for } j = 1, \ldots, m \tag{21}$$

$$b_j = a_j \quad \text{for } j = 2m+1, \ldots, n$$

and may be calculated from

$$b_k = \mathbf{x} \cdot \mathbf{s}^k \quad . \tag{22}$$

Proof:

From (7) we have

$$\begin{aligned} \mathbf{x} &= \sum_{k=1}^{n} a_k \mathbf{p}^k = \sum_{k=1}^{m} (a_{2k-1}\mathbf{p}^{2k-1} + a_{2k}\mathbf{p}^{2k}) + \sum_{k=2m+1}^{n} a_k \mathbf{p}^k \\ &= \sum_{k=1}^{m} (a_{2k-1}\mathbf{p}^{2k-1} + a_{2k-1}^{*}\mathbf{p}^{2k-1^{*}}) + \sum_{k=2m+1}^{n} a_k \mathbf{p}^k \\ &= \sum_{k=1}^{m} 2\, Re(a_{2k-1}\mathbf{p}^{2k-1}) + \sum_{k=2m+1}^{n} a_k \mathbf{p}^k \\ &= \sum_{k=1}^{m} (2\, Re(a_{2k-1})\, \mathbf{r}^{2k-1} - 2\, Im(a_{2k-1})\, \mathbf{r}^{2k}) + \sum_{k=2m+1}^{n} a_k \mathbf{r}^k \\ &= \sum_{k=1}^{n} b_k \mathbf{r}^k \end{aligned}$$

with b_k defined as in (19).

For complex POPs we have (cf. Appendix 1, (A.6))

$$\begin{aligned} b_{2j-1} - i b_{2j} &= 2a_{2j-1} = 2\mathbf{x} \cdot \mathbf{q}^{2j-1^{*}} \\ &= \mathbf{x} \cdot [\mathbf{s}^{2j-1} - i\mathbf{s}^{2j}] = \mathbf{x} \cdot \mathbf{s}^{2j-1} - i\mathbf{x} \cdot \mathbf{s}^{2j} \quad . \end{aligned}$$

Thus $b_{2j-1} = \mathbf{x} \cdot \mathbf{s}^{2j-1}$ and $b_{2j} = \mathbf{x} \cdot \mathbf{s}^{2j}$. ■

2.6 THE POP MODEL

The pop coefficients $a_k(t)$ obey, from (4), (9) and (10),

$$a_k(t+1) = \lambda_k a_k(t) + \xi_k(t) \tag{23}$$

where $\xi_k(t)$ is the noise driving POP k (10) and $|\lambda_k| \leq 1$ (an amplifying, and therefore non-physical, solution would result from $|\lambda_k| > 1$).

In the absence of noise (i.e. $\xi_k(t) \equiv 0$) equation (23) becomes

$$a_k(t) = \lambda_k^t \, a_k(0) \tag{24}$$

If we express λ_k as $|\lambda_k| \cdot exp(i\omega_k)$ (24) may be written as

$$a_k(t) = exp(-t/\tau_k) \; exp(i \frac{2\pi}{T_k} t) \; a_k(0) \tag{25}$$

with an e-folding (decay) time

$$\tau_k = - 1 / \log_e |\lambda_k| \; > \; 0 \tag{26}$$

and an oscillation period T_k

$$T_k = 2\pi / \omega_k \quad . \tag{27}$$

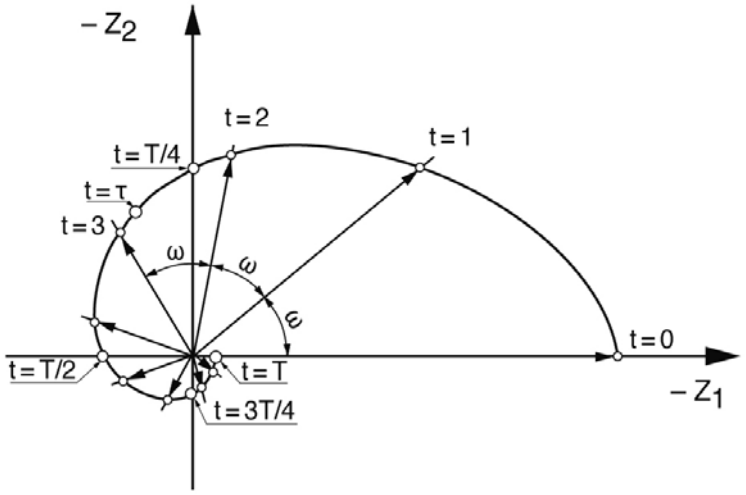
We now drop, for the sake of simplicity, the index k and use the real vector representation (18) and (21) for a particular POP

$$p = p^1 + i \, p^2$$
$$2 \, a(t) = z_1(t) - i \, z_2(t) \quad .$$

Then (25) is equivalent to

$$\begin{pmatrix} z_1(t) \\ z_2(t) \end{pmatrix} = exp(-t/\tau) \cdot \begin{pmatrix} \cos(\omega t) & \sin(\omega t) \\ -\sin(\omega t) & \cos(\omega t) \end{pmatrix} \begin{pmatrix} z_1(0) \\ z_2(0) \end{pmatrix} . \tag{28}$$

If we assume that $z_1(0)=1$, $z_2(0)=0$, i.e. the initial noise is driving the real part of the POP only, the time evolution of the bivariate index (z_1,z_2) for the duration of one period can be represented in a dial diagram as follows:



where we have assumed that λ falls into the 1st quadrant of the complex plane.

The interpretation of the particular POP **p** may be made clearer by showing the time evolution for the POP (driven by an impulse δ-function) at t=0:

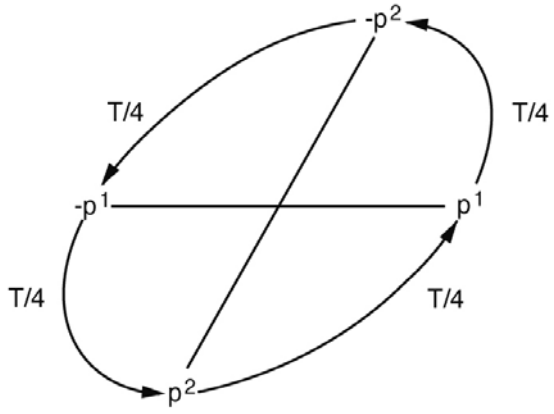
$$\mathbf{P}(t) = \text{Observed pattern at time t corresponding to the POP}$$

$$= a(t)\,\mathbf{p} + a^{*}(t)\,\mathbf{p}^{*} = 2\,Re\,[a(t)\,\mathbf{p}\,]\;(\;= 2\,\mathrm{Re}\,[\mathbf{p}\,\lambda^{t}\,a(0)]\;)$$

$$= z_1(t)\,\mathbf{p}^1 + z_2(t)\,\mathbf{p}^2$$

$$= [\;\mathbf{p}^1 \cos\frac{2\pi}{T}t - \mathbf{p}^2 \sin\frac{2\pi}{T}t\;]\;exp\;-t/\tau$$

where again only the real part is driven by the initial noise ($z_2(0)=0$).
Thus at t = 0 the observed pattern is purely due to the real part of the POP. One quarter of a period later the observed pattern is that due to minus the imaginary part, followed, a further quarter of a period later, by minus the real part, then (plus) the imaginary part and finally back to (plus) the real part again. The magnitude of the observed pattern continually decays by a factor 1/e in time τ. The following diagram illustrates this *POP cycle*:



The elliptic form of the diagram reflects the fact that the trajectory of $\mathbf{P}(t)$ in the $\mathbf{p}^1/\mathbf{p}^2$-phase space is, apart from the damping, an ellipse because, in general, $\mathbf{p}^1$ and $\mathbf{p}^2$ have to be neither orthogonal nor of equal length.

For a real POP **p** (24) is equivalent to

$$z_1(t) = exp\ -t/\tau \cdot z_1(0)$$

and the corresponding observed pattern is just this POP damped with the e-folding time τ.

The time series of POP coefficients a(t) for each POP **p** may be computed either by inverting equation (9) or by calculating the scalar product of the field **x** with the corresponding Principal Adjoint Pattern **q** :

$$a(t) = \mathbf{x}(t)\cdot\mathbf{q}\ , \tag{30}$$

similarly, the real coefficients can be computed from (20) or from (22):

$$\begin{aligned} z_1(t) &= \mathbf{x}(t)\cdot\mathbf{s}_1 \\ z_2(t) &= \mathbf{x}(t)\cdot\mathbf{s}_2 \end{aligned} \tag{31}$$

where $\mathbf{s}_1$ and $\mathbf{s}_2$ represent **q** in real vector representation (19). For a real POP only the first equation in (31) has meaning.

In view of (30) a significant contribution a(t) of a certain state is not necessarily found if **x** has a large projection on the POP **p** but only if the projection on the corresponding PAP **q** is large. Note that it is because the POPs are, in general, not orthogonal to each other that the adjoints can be different from the POPs and the coefficients are not simply the projection of the data onto the pattern itself (as in EOF analysis). Since **q** is subject to the constraint that is has to be orthogonal to all POPs except the one under consideration the coefficient time series of a certain POP depends on all other POPs, in particular those reflecting noise. Therefore, it may sometimes be advisable to compute the coefficient time series of a POP **p** by a least square fit to the data:

$$\| \mathbf{x}(t) - a(t)\ \mathbf{p} - a^*(t)\ \mathbf{p}^* \| = \| \mathbf{x}(t) - z_1(t)\ \mathbf{p}^1 - z_2(t)\ \mathbf{p}^2 \| = \text{Min !} \tag{32}$$

The POP program has a switch to choose between the two approaches (31) and (32) (see section 3.2).

2.7 STATISTICAL ASSESSMENT

In most cases, we interpret one or a few POPs as modelling the original multivariate data time series. In this section a means is provided in order to aid the assessment of the 'usefulness' of a given POP.
For the following explanations we define three quantities $\rho(t)$, $\mathbf{y}(t)$, $\mathbf{r}(t)$ where $\rho(t)$ is some *predictor* attempting to model the *predictand* $\mathbf{y}(t)$. We define the error in the prediction, $\mathbf{r}(t)$, such that

$$\mathbf{y}(t) = \rho(t) + \mathbf{r}(t) \ . \tag{33}$$

Explicitly, there are two relevant cases to be considered,

1. POPs predicting original data set, i.e.

$$\mathbf{y}(t) = \mathbf{x}(t) \tag{34}$$

$$\rho(t) = \sum_{\Delta} a_k(t)\ \mathbf{p}^k \tag{35}$$

   where the sum is taken over some subset Δ of POPs, as disscussed below.
2. POPs predicting the rate of change of data,i.e.

$$\mathbf{y}(t) = \mathbf{x}(t+1) - \mathbf{x}(t) \tag{36}$$

$$\rho(t) = \sum_{\Delta} (a_k(t+1) - a_k(t))\ \mathbf{p}^k \tag{37}$$

We define the set Δ for various different cases;

1. The independent single prediction assessment variables, - for each predictor (POP) separately.

2. The sequential incremental prediction assessment variables, - for each predictor separately applied to the predictand,

$$\mathbf{y}' = \mathbf{y} - \mathbf{s}_{n-1}$$

   where $\mathbf{s}_{n-1}$ is the sum of all predictors *up* to the one in question. This implies a certain ordering of the POPs in some *hierarchical model* such that the *first* POP has the lowest relative error, the second POP has the lowest relative error using the first POP as it's sub-predictor set, etc.

3. The complementary incremental prediction assessment variables, - for each predictor separately applied to the predictand,

$$\mathbf{y}'' = \mathbf{y} - \mathbf{s}'_n$$

where $\mathbf{s}'_n$ is the sum of all predictors *except* the one in question. These variables thus assess how well each POP predicts the data not accounted for by the other POPs.

4. Total prediction assessment variables. One pair of variables for all predictors which together model the predictand.

In order to assess the modelled quantities in terms of second moments we now define two numbers

$$\varepsilon = \sqrt{\frac{< \mathbf{r}^2(t) >}{< \mathbf{y}^2(t) >}} \tag{38}$$

$$\xi = \sqrt{\frac{< \rho(t) \cdot \mathbf{y}(t) >}{< \mathbf{y}^2(t) >}} \tag{39}$$

where again <.> is expectation resp. time average and · is the Euclidian dot product in n-dimensional space.

ε is the *relative error* made by predicting $\mathbf{y}$ by ρ, and ξ is the *biasing factor*. Of course, for ρ to be a 'good' prediction ε should be small. If additionally, ξ is large this means that the component of ρ which is statistically orthogonal to $\mathbf{y}$ is small.

The *explained variance* of a POP, that is, the amount of variance the POP explains as fraction of the variance of the original data, is given by

$$\text{Expl. var.} = 1 - \varepsilon^2 \tag{40}$$

with ε evaluated for the independent single prediction case. In the sequential incremental prediction case, this quantity defines the *cumulative explained variance*, i.e. the variance explained by all POPs up to the one in question using the hierarchical ordering described above.

3 INPUT TO POP PROGRAM

3.1 INPUT DATA.

The input time series DAT(r,t) is read from an unformatted file. The file name is given by the user on the command line. In the FORTRAN context, the unit number is defined as a fixed parameter called JPUNIR.

The data are expected in the *Extra Code* format (see Borgert and Welke, 1991). That means, they have been previously written by a FORTRAN program of the following form, e.g.,

```
      DIMENSION DATA(p,q)                                  with  p ≥ <number of data points>
                                                                 q ≥ <number of time steps>
          .
          .
          .   (Any user pre-processing of data)
          .
          .

      ILEN = <number of data points>
      DO 1 JTIME = 1,<number of time steps>
           WRITE(<unit>) IDATE, IVAR, ILEV, ILEN
           WRITE(<unit>)
     &           (DATA(JSPACE,JTIME),JSPACE=1,ILEN)
    1      CONTINUE
          .
          .
          .
      END
```

For the header parameters, IDATE, IVAR, ILEV and ILEN, the only obligation is that ILEN is always equal to the length of the following record. Header parameters are generally not conserved in the output files. See the output documentation for details. Note that the output file with the filtered time series may be used as input file for another POP experiment, with conserved calendar date IDATE and recoded data gaps.

Data gaps are processed by the POP program if coded by a numerical value of 9.E+99 (FORTRAN parameter PPGAP defined in the include file "scalars.i").

The order of points in the 'space' dimension does not matter since the method uses correlations between 'points' in EOF-space and has no knowledge of their spatial organization.

3.2 PARAMETERS SET BY THE USER.

It is quite simple to run the POP program for the user has to type just one line to set all his or her parameters. The program will, after being started, read this line from standard input or from a special parameter file POP.PAR consisting of that line. See the "User's Guide" in Appendix 4 of this manual for setting up this behaviour of the POP program and for practical hints to deal with the parameter line in a UNIX environment.
In this section a standard FORTRAN 77 implementation is assumed where the parameters are read from standard input (unit '*' in a FORTRAN read statement).

Let's look at the syntax for the parameter line as expected by the FORTRAN program. It is borrowed from the UNIX command line syntax but it is easy to understand even if you are not familiar with UNIX.
The parameters have to be specified on one line of at most 120 characters. The general form is

```
[-ECTdrupfl] [-n <NTS>] [-t <NTO>] [-e <NEOF>] [-c <NC>]                ..
[-s <DT>] [-m <FACT>] [-i <infile>] [-o <outfile>]                      ..
[-- <PMIN> <P2> <P1> <PMAX>] ,
```

where '..' means that the line is continued on the next line.

The brackets [] mean that the contents is optionally given. In the first bracket, each one of the letters E,C,T,... may be typed or be omitted. In all other brackets, the contents is given completely or is omitted as a whole. Angle brackets <> are values defined by the user to fill the corresponding variables or file names in the FORTRAN code. All parameters are optional and have default values, but an empty parameter line is not meaningful.

The single letters preceded by a minus sign are switches corresponding to logical FORTRAN variables in the following manner.

Option letter	Effect	FORTRAN variable
-E	An EOF analysis is performed (no POP analysis)	LEOFS
-C	A Complex EOF analysis is made (no POP analysis)	LCEOF
-T	Time filtering of data is enabled (see Appendix 2)	LTFILT
-d	Trend filtering of data: the linear trend is subtracted from the data after reading (least square fit)	LTREND
-r	Normalisation of data: at each point the time series is divided by it's (local) standard deviation	LNORM
-u	Turning of pop pairs to statistical orthogonality (see section 2.4)	LTURN
-p	Large printout	LEVPRI
-f	Use of smaller covariance matrix is forced even in case of data gaps (see Appendix 3)	LFLIP
-l	Pop coefficients are computed by a least square fit to the data (otherwise, by projection onto the adjoint POPs) (see section 2.6)	LLSF

The default value of all these logical values is .false.

The options	-n, -t, -e, -c	require (positive) integer arguments,
the options	-s, -m	require real arguments,
the options	-i, -o	require strings (max. length 60) and
the delimiter	--	is followed by 4 (real or integer) numbers

with the following meanings:

Name	Type	Default	Description
NTS	Integer	As found in data.	The number of time series. i.e. the number of 'spatial' points in the analysis (if smaller than found in data only the first NTS values are read from each record)
NTO	Integer	As found in data	The length of time series, i.e. the number of time steps (if smaller than found in data only the first NTO records are read from the data file). The number might be reduced slightly during run time if the prime number decomposition is improper for the NAGLIB routine which is used for filtering (see Appendix 5).
NEOF	Integer	20	Number of EOFs in POP/EOF analysis or number of complex EOFs in CEOF analysis. The default will in most cases retain a sufficient amount of the data information for calculating the POPs. If NEOF is set to zero no EOF analysis is performed prior to the POP analysis, i.e. the POPs are computed from the original data without reduction of dimensions.
NC	Integer	10	The number of 'chunks' for the spectral analysis of the data by using *Bartlett*'s procedure. The number of chunks should be set to any 'natural' time period represented in the data e.g. if the data comprises 7 years of daily measurements of the number of Patagonian squirrels, then an appropriate value for NC would be 7.
DT	Real	1.0	Length of time intervals in data.
FACT	Real	1.0	Factor for multiplication of the data after reading.
infile	String	POP.IN	Name of input file.
outfile	String	POP	Name base for all output data files. If "ANTARC" is chosen, the names will then be, ANTARC.filt, ANTARC.pop, etc. (see section 4.2).
PMIN, P2, P1, PMAX	Real or Integer	see description of time filter parameters in Appendix 2	Filter characteristics; a default value is forced by specifying -1 at the corresponding place.

Here is an example for starting the POP program if it is assumed that the file name of the executable is "pop.x":

```
pop.x        <cr>
-d -p -e 18 -c 3 -i ENSO.DAT        <cr>
```

The input may be abbreviated up to the limit:

```
pop.x        <cr>
-dp -e18 -c3 -iENSO.DAT        <cr>
```

If you want to use the defaults for all options (i.e. data in POP.IN, output files POP.*, 20 EOFs, 10 chunks, etc.) the command line has the very simple form

```
pop.x        <cr>
<cr>
```

For time filtering with non-default filter parameters, the command is e.g.

```
pop.x        <cr>
-Tdp -e 18 -c 3 -i ENSO.DAT  --  2. 2. 8.0 10.        <cr>
```

All the above used syntax features are supported by the syntax check within the FORTRAN program by which the parameter line is processed. However, the -- is expected before the filter arguments which are not preceded by an option letter like -e.

4 POP PROGRAM OUTPUT

4.1 PRINTER OUTPUT

In this section, an explanation of the printer output from the POP program is given, together with some sample output from an actual run.

The example shown is the analysis of geopotential height data of three winters, 1984/85, 1985/86 and 1986/87 (Schnur et al., 1991b). The time step is 12 hours yielding observations at 540 times. Spatially resolved are latitudes from 85°S up to 10°S with a distance of 2.5 degrees and heights of 1000, 850, 700, 500, 300, 200 and 100 hPa. In the zonal direction, the data were Fourier transformed, and only wave number 8 is taken. By that, the length of each record of data is

434 = 31 latitudes × 7 heights × 2 zonal Fourier coefficients

The POP run was started by entering the command line,

```
pop.x        <cr>
-pd -e18 -c3 -ifort.12 -o3dSH8487          <cr>
```

Generally, the first page of output shows the values of the input parameters. The values have first been chosen by the user on the parameter line. The subroutine PARSIN has supplied default values if necessary. Default or adjusted values have also been found by pre-reading the input data file looking for the number and length of records.

The program then checks these values for validity generating 'help' messages and aborting if necessary. Even in case of an abort the values are printed out first to make debugging easier for the user.

In our example we got the following output:

```
POP script started at Thu Mar 7 14:54:32 MET 1991

NUMBER OF TIME SERIES IN FILE:        434
LENGTH OF TIME SERIES IN FILE:        540

****************************************
********** INPUT PARAMETERS ************
****************************************
* NUMBER OF TIME SERIES              434 *
* LENGTH OF TIME SERIES              540 *
* NUMBER OF CHUNKS                     3 *
* MULTIPLICATION FACTOR        .1000E+01 *
* PRINTER OUTPUT                    HIGH *
* NORMALISATION                      OFF *
* TREND FILTERING                     ON *
* TIME FILTERING                     OFF *
* STATISTICALLY ORTHOGONAL TURNING   OFF *
* NUMBER OF EOFS                      18 *
****************************************

*  INPUT NAME: fort.12
*  LOGICAL UNIT NO.    1 - SYSTEM NAME:
*  3dSH8487.filt
*  LOGICAL UNIT NO.    2 - SYSTEM NAME:
*  3dSH8487.var
*  LOGICAL UNIT NO.    3 - SYSTEM NAME:
*  3dSH8487.eof
*  LOGICAL UNIT NO.    4 - SYSTEM NAME:
*  3dSH8487.eofc
*  LOGICAL UNIT NO.    5 - SYSTEM NAME:
*  3dSH8487.eofsp
*  LOGICAL UNIT NO.    6 - SYSTEM NAME:
*  3dSH8487.pop
*  LOGICAL UNIT NO.    7 - SYSTEM NAME:
*  3dSH8487.pope
*  LOGICAL UNIT NO.    8 - SYSTEM NAME:
*  3dSH8487.popc
*  LOGICAL UNIT NO.    9 - SYSTEM NAME:
*  3dSH8487.popsp
*  LOGICAL UNIT NO. 10 - SYSTEM NAME:
*  3dSH8487.apop
*  LOGICAL UNIT NO. 11 - SYSTEM NAME:
*  3dSH8487.apope

TOTAL NUMBER OF GAPPY DATA:                    0
NUMBER OF SPACE POINTS WITH NO OBSERVATION: 0
MAX. ABS. VALUE OF NON-GAP DATA:               0.889E+02
```

The next page of output is generated by the filtering subroutines. The output comprises the frequency, ν the corresponding period $P = 1/\nu$ and the filter characteristic, $G(\nu)$, for all frequencies 1/2 -> $1/(2 \times \mathrm{NTO})$. However, only those frequencies are shown for which $G(\nu) \neq 0$, and to save repetition, only frequencies at either end of the 'flat' part of the window.

```
**********************************************
*************** FILTERING ****************
**********************************************
```

In our special example no time filtering was chosen in the POP run.

The first 51 equivalent digital filter weights are then listed - these have already been described above.

Subroutine VARIAN calculates the time mean and variance *of the filtered data* at each data point. The mean should be equal to zero to within machine precision since, regardless of whether filtering has actually been performed, the mean has been subtracted by the filtering subroutines. Trend-filtering may slightly alter the mean from zero because it is carried out after subtracting the mean.

If the user has selected 'NORMALISATION ON' then all the variances should equal 1 after normalization but, since this is not very useful output, the variances *before normalization* are given.

```
SUBROUTINE VARIAN: COMPUTES VARIANCE OF FILTERED DATA (BEFORE ANY
NORMALISATION)

ESTIMATION OF 1ST AND 2ND MOMENT:

COMPONENT        MEAN          VARIANCE
        1   -0.359679E-05    0.121965E-01
        2   -0.120354E-04    0.171476E-01
        3   -0.506777E-04    0.100975E+01
        4   -0.102398E-03    0.148924E+01
        5   -0.139198E-03    0.522246E+01
        .         .               .
        .         .               .
        .         .               .
        .         .               .
        .         .               .
      430    0.440594E-03    0.870012E+01
      431    0.109303E-03    0.822323E+01
      432    0.377837E-03    0.699672E+01
      433    0.169178E-03    0.666123E+01
      434    0.300494E-03    0.559445E+01
```

Subroutine TSEOFS calculates the EOFs and transforms the data into EOF-space. The trace (the sum of the eigenvalues of the covariance matrix $C(\mathbf{r},\mathbf{r}')$ is given, followed by the eigenvalues themselves, in decreasing order.

It can be shown that the eigenvalue of an EOF is equal to the variance explained by that EOF. Since the EOFs are orthogonal, the cumulative variance, i.e., the variance explained by the first k EOFs, is simply the sum of the variances explained by the EOFs individually. Note that this is not true for the case of the POPs, since they are not orthogonal.

The percentages are obvious but the user should note how much percentage variance is explained by the first NEOF EOFs and perhaps change NEOF on input. It is suggested that this number of EOFs should explain at least 95 of the total variance.

SUBROUTINE TSEOFS: TRANSFORM TIME SERIES IN EOF SPACE.

EIGENVALUE DIAGNOSIS

TRACE = 0.3581E+05

EIGENVALUES

0.1162E+05	0.1137E+05	0.3183E+04	0.2594E+04	0.1046E+04	0.8157E+03
0.6020E+03	0.4638E+03	0.3611E+03	0.3498E+03	0.3259E+03	0.2722E+03
0.2497E+03	0.2248E+03	0.1965E+03	0.1799E+03	0.1390E+03	0.1235E+03
.	.	.	.	.	.
.	.	.	.	.	.
.	.	.	.	.	.

CUMULATIVE VARIANCES

0.1162E+05	0.2299E+05	0.2617E+05	0.2877E+05	0.2981E+05	0.3063E+05
0.3123E+05	0.3169E+05	0.3205E+05	0.3240E+05	0.3273E+05	0.3300E+05
0.3325E+05	0.3348E+05	0.3367E+05	0.3385E+05	0.3399E+05	0.3412E+05
.	.	.	.	.	.
.	.	.	.	.	.
.	.	.	.	.	.

PERCENTAGES OF EIGENVALUES

32.5	31.7	8.9	7.2	2.9	2.3
1.7	1.3	1.0	1.0	0.9	0.8
0.7	0.6	0.5	0.5	0.4	0.3
.	.	.	.	.	.
.	.	.	.	.	.
.	.	.	.	.	.

PERCENTAGES OF CUMULATIVE VARIANCES

32.5	64.2	73.1	80.3	83.3	85.5
87.2	88.5	89.5	90.5	91.4	92.2
92.9	93.5	94.0	94.5	94.9	95.3
.	.	.	.	.	.
.	.	.	.	.	.
.	.	.	.	.	.

The covariance matrices $\bar{C}_0$, $\bar{C}_1$, and $\bar{B}$, as defined in Section 2, are output for diagnostic purposes (not all the elements are given below). Since the Principal Component (EOF coefficient) time series are statistical orthogonal the covariance matrix $\bar{C}_0$ has to be of diagonal form within machine precision (trend-filtering may again slightly increase the error in non-diagonal elements, see above).

```
****************************************
********** CALCULATION OF POPS *********
****************************************

LAG-0 COVARIANCE MATRIX (C0)
PRINTOUT OF MATRIX

TRACE:        0.3412E+05

    0.1162E+05  -0.3152E-09  -0.1018E-09   0.8623E-11   0.1089E-10 .....

   -0.3152E-09   0.1137E+05   0.1768E-10   0.2167E-10   0.1822E-10 .....

   -0.1018E-09   0.1768E-10   0.3183E+04  -0.1552E-10   0.1078E-10 .....

    0.8623E-11   0.2167E-10  -0.1552E-10   0.2594E+04   0.3676E-10 .....

    0.1089E-10   0.1822E-10   0.1078E-10   0.3676E-10   0.1046E+04 .....

        .            .            .            .            .          .


LAG-1 COVARIANCE MATRIX (C1)
PRINTOUT OF MATRIX

TRACE:        0.2479E+05

    0.8262E+04  -0.7103E+04   0.8735E+03   0.5893E+03   0.5636E+02 .....

    0.7185E+04   0.7862E+04  -0.1319E+04   0.5539E+03  -0.8827E+02 .....

    0.8129E+03   0.1317E+04   0.2472E+04  -0.1282E+04   0.1104E+03 .....

   -0.6031E+03   0.6291E+03   0.1293E+04   0.2013E+04  -0.3130E+03 .....

    0.2937E+02   0.1732E+03   0.1242E+03   0.3052E+03   0.8359E+03 .....

        .            .            .            .            .          .
```

ESTIMATED SYSTEM MATRIX (STRUMA)
PRINTOUT OF MATRIX

TRACE: 0.1390E+02

0.7109E+00	-0.6249E+00	0.2744E+00	0.2272E+00	0.5390E-01	.....
0.6183E+00	0.6917E+00	-0.4144E+00	0.2135E+00	-0.8441E-01	.....
0.6995E-01	0.1158E+00	0.7764E+00	-0.4943E+00	0.1055E+00	.....
-0.5189E-01	0.5535E-01	0.4063E+00	0.7759E+00	-0.2993E+00	.....
0.2527E-02	0.1524E-01	0.3901E-01	0.1176E+00	0.7994E+00	.....
.	.	.	.	.	.

Subroutine POPS is the main subroutine for calculating the POPs and associated quantities.

The eigenvalues of the matrix $\bar{\mathbf{B}} = \bar{\mathbf{C}}_1 \bar{\mathbf{C}}_0^{-1}$ are given, followed by the matrix of eigenvectors $\bar{\mathbf{p}}^k$. Note, however, that the eigenvectors are not stored as complex matrix $\bar{\mathbf{P}}$ (EOF space) and $\mathbf{P}$ (Euclidean space) but as real matrix $\bar{\mathbf{R}}$ and $\mathbf{R}$ (see Section 2.5). From now on, whenever 'POP *k* (in Euclidean space)' is referred to, it means $\mathbf{r}^k$ which is a composite of the two POPs $\mathbf{p}^k$, $\mathbf{p}^{k*}$ (see equation 18). 'POP *k* in EOF-space' means the same with all entities given a bar. From a complex conjugate pair of POPs the program always selects the one with positive imaginary part of the corresponding eigenvalue, so that the period T is always positive.

```
COMPLEX   EIGENVALUES   OF   MATRIX   STRUMA

(0.5708E+00;  0.7613E+00) (0.5708E+00; -0.7613E+00) (0.6884E+00;  0.5876E+00)

(0.6884E+00; -0.5876E+00) (0.7192E+00;  0.5636E+00) (0.7192E+00; -0.5636E+00)

(0.7448E+00;  0.4951E+00) (0.7448E+00; -0.4951E+00) (0.8266E+00;  0.3715E+00)

(0.8266E+00; -0.3715E+00) (0.8254E+00;  0.3040E+00) (0.8254E+00; -0.8266E+00)
      .           .           .           .           .           .
      .           .           .           .           .           .
      .           .           .           .           .           .
```

In this step the paired POPs are rotated so that the real and imaginary components are orthogonal, even when option -u has been specified (see Section2.4):

```
LENGTH OF RAW EIGENVECTOR: 1.0
SCALAR PRODUCTS: PP, QQ, PQ:0.510638707,     0.489361292,    -1.100902295E-3

NEW VECTORS ARE:

P(NEU) =    1.00*P -   0.05*Q
Q(NEU) =    0.05*P +   1.00*Q

LENGTH OF RAW EIGENVECTOR: 1.0
SCALAR PRODUCTS: PP, QQ, PQ:0.520498439,     0.47950156,     -1.354687558E-3

NEW VECTORS ARE:
    .     .     .     .
    .     .     .     .
    .     .     .     .
```

Under the headline "POP MATRIX IN EOF SPACE" the POPs $\mathbf{r}^k$ are given in EOF space coordinates.
In this example, we have eight pairs of POPs, i. e., complex POPs and two single POPs, i.e., real POPs. The first pair is marked by "1R" and "I" in the headline and its EOF space coordinates are given in columns 2 and 3. "R" and "I" refer to the real and to the imaginary component. The entries "9" and "10" in columns 18 and 19 of the headline refer to the two real POPs of this example.

POP MATRIX IN EOF SPACE

COMP.	NR. OF POP 1R	I	2R	I	3R	I	4R	I	5R	I	6R	I	7R	I	8R	I	9	10
1	-0.63	13.26	10.73	0.35	9.19	4.64	-10.23	8.48	9.49	6.08	-6.65	6.66	-3.29	-2.80	8.54	5.58	-2.60	2.98
2	14.02	0.72	0.49	-9.88	3.71	-9.54	9.59	9.30	6.02	-8.62	7.11	6.66	-0.87	2.61	7.51	-3.83	-3.77	-4.64
3	1.00	-5.77	3.25	0.28	1.86	3.92	0.24	1.19	7.65	4.06	-2.48	3.91	-2.94	-0.66	8.29	2.60	-8.06	7.23
.	.		.		.		.		.		.		.		.		.	.
.	.		.		.		.		.		.		.		.		.	.
.	.		.		.		.		.		.		.		.		.	.

In the same way, the principal adjoint patterns, PAPs, are printed in EOF space. Note, however that the PAPs have not been subject to any rotation as they are completely specified by the POPs (see Appendix 1, (A.4)).
The printout of he "APOP matrix in EOF space" is analogous to the former one.

The POP eigenvalue is then expanded to give the e-folding time τ_k and the rotation period T_k for each POP. Note that for a real POP the rotation period is given as zero and not as infinity as might be expected. Due to the way the POPs are stored, stated above, real POPs are distinguished from complex ones by the fact that real POPs have T_k=0.

Next, the POPs $\mathbf{r}^k$ and the PAPs $\mathbf{s}^k$ are given in EOF-space and in Euclidean space. If the POPs have been rotated to obtain statistically independent noise compo-nents (the variable LTURN set 'true' by the user option '-u', see Section 2.5), the EOF space coordinates given above under the headline "POP MATRIX IN EOF SPACE" are different from the EOF space coordinates given below.

```
POP GROUP NUMBER        1
EIGENVALUE NO. 1     E-FOLDING TIME   0.2013E+02     OSCILLATION PERIOD   0.6775E+01

REAL PART IN EOF-SPACE
      -0.6341E+00  0.1402E+02   0.1000E+01 -0.4672E+01 -0.3091E+00 -0.1881E+00
      -0.4842E+00 -0.3281E+00   0.6855E+00  0.6886E+00 -0.5666E+00  0.2468E+00
      -0.2759E+00 -0.1027E+00   0.8721E-01 -0.4081E-01 -0.7250E-02 -0.1269E+00
              IN EUCLIDEAN SPACE
      -0.8222E-04 -0.9871E-03  0.2093E-02 -0.6021E-02  0.8536E-02 -0.7010E-02
       0.8106E-02  0.2286E-01 -0.2137E-01  0.4271E-01 -0.4543E-01 -0.2573E-02
      -0.1644E-01 -0.6755E-01  0.6541E-01 -0.1246E+00  0.2109E+00 -0.2143E+00
       0.4136E+00 -0.3422E+00  0.6140E+00 -0.4810E+00  0.7964E+00 -0.6204E+00
       0.9800E+00 -0.7413E+00  0.1134E+01 -0.8436E+00  0.1187E+01 -0.9433E+00
       0.1107E+01 -0.1007E+01  0.9023E+00 -0.1014E+01  0.6344E+00 -0.9486E+00
       0.3833E+00 -0.8017E+00  0.1962E+00 -0.6257E+00  0.1055E+00 -0.4560E+00
         .           .           .           .           .           .
         .           .           .           .           .           .
         .           .           .           .           .           .

IMAGINARY PART IN EOF-SPACE
       0.1326E+02  0.7234E+00 -0.5766E+01 -0.8733E+00  0.5740E+00 -0.2622E+00
      -0.6987E+00  0.1620E+00  0.3371E+00 -0.5126E+00  0.4343E+00  0.1608E+00
      -0.2627E+00 -0.3764E+00  0.1888E+00  0.2903E+00  0.1416E+00 -0.1148E+00
                    IN EUCLIDEAN SPACE
      -0.9787E-03 -0.2673E-03 -0.1344E-01 -0.3071E-02 -0.3365E-01 -0.1519E-01
      -0.2275E-02 -0.1641E-01  0.2825E-01  0.9726E-02 -0.7553E-01 -0.1256E-01
      -0.1703E+00 -0.9817E-01 -0.1982E+00 -0.1564E+00 -0.3308E+00 -0.2200E+00
      -0.4830E+00 -0.3684E+00 -0.5588E+00 -0.5941E+00 -0.6304E+00 -0.8357E+00
      -0.7300E+00 -0.1029E+01 -0.8454E+00 -0.1144E+01 -0.9264E+00 -0.1155E+01
         .           .           .           .           .           .
         .           .           .           .           .           .
         .           .           .           .           .           .

POP GROUP NUMBER        2
EIGENVALUE NO. 3     E-FOLDING TIME   0.1003E+02     OSCILLATION PERIOD   0.8892E+01

REAL PART IN EOF-SPACE
       0.1073E+02  0.4935E+00  0.3250E+01  0.1694E+01 -0.8646E+01  0.1171E+01
      -0.1597E+01  0.2119E+01  0.1417E+01 -0.3684E+00 -0.1113E+01 -0.1911E+00
      -0.1073E+00  0.3188E+01  0.8978E-02 -0.5047E+00 -0.1022E+00 -0.8416E-01
              IN EUCLIDEAN SPACE
       0.7625E-03 -0.7719E-02  0.1355E-01 -0.7137E-01  0.4127E-01 -0.1396E+00
         .           .           .           .           .           .
         .           .           .           .           .           .
         .           .           .           .           .           .
```

... and so on ...

Next the POPs are tested for orthogonality. The orthogonality is defined by $\mathcal{E}_{ij} = \mathbf{p}^i \cdot \mathbf{p}^j = 0$. This matrix is symmetric so only the lower half is printed. As stated before, the complex POPs are not stored individually, but as a complex pair so that the diagonal elements of matrix $\mathcal{E}_{ij}$ are not equal to NTS (434 in the given example) but rather that the elements occur in pairs which should add up to NTS (sum over $p^{k'2} + p^{k''2}$ = NTS for each k). For real POPs, however, the diagonal element should be exactly NTS.

If the parameter '-u' has n o t been selected (as here), the POPs have been rotated such that $\mathbf{p}^{k'} \cdot \mathbf{p}^{k''} = 0$ and this should be manifest in the corresponding off-diagonal elements.

```
CHECK OF GEOMETRIC ORTHOGONALITY
 1R 221.64
  I    0.00 212.36
     -------------
 2R    -0.99 117.53!225.92
  I -111.78    0.32    0.00 208.08
     ---------------------------
 3R    28.11 108.10!  88.52 -92.61!205.23
  I -115.86  34.55 100.86  88.74    0.00 228.77
     -----------------------------------------
 4R 124.75-129.69!104.47 -88.41!-51.28-168.60!238.67
  I 128.74 109.34  77.52 -86.59 133.92 -44.11    0.00 195.33
     -------------------------------------------------------
 5R   62.18  82.69!147.01 -76.23!123.77   4.40!-24.94 127.42!233.00
  I  -88.45  56.92  69.24 126.18   5.23 135.93-149.72 -13.81    0.00 201.00
     -----------------------------------------------------------------
 6R   81.34 -72.62!-82.37 -91.97! -3.23-126.98!157.42   9.39!-43.16-125.25!230.78
  I   94.11  67.07  89.24 -73.24  98.40  -1.65  -9.95 139.46 119.78 -16.47    0.00 203.22
     -----------------------------------------------------------------------------
 7R   -8.48 -28.63!-74.68  47.99!-41.30 -56.84! 38.50 -17.62!-85.14  -7.74! 40.80  -9.28!196.63
  I   20.54 -31.25 -72.30 -71.50  19.37 -47.60  45.01  -0.97  -5.81 -94.68  -2.86  -4.25    0.00 237.37
     -----------------------------------------------------------------------------------------
 8R   74.07  70.06!  86.28 -97.61!144.35 -48.65!  8.73 137.08!217.03 -42.05 -14.31 112.14!-39.60  85.27!323.78
  I  -41.78  57.45  66.99  27.73  44.84  98.78-105.57  20.83  34.59  96.96 -61.85  33.86 -61.86 -29.38    0.00 110.22
     -----------------------------------------------------------------------------------------------------------
 9    -86.00   2.94!  -9.68  23.84!-59.79  -9.12!  -5.99-107.42!-54.96 -75.57 -13.98-101.37!  14.34  20.05!-20.93 -68.32!434.00
     -----------------------------------------------------------------------------------------------------------------------
10    -51.77   6.52!   5.71  57.34!  -8.51  33.13!-70.63 -34.44! 50.97  90.47 107.49  -1.35!  10.33 -38.52!  87.23 -15.90!  -3.72!434.00
     -----------------------------------------------------------------------------------------------------------------------
```

A summary of the statistics for the time series of POP coefficients is then given for each POP pair.

The e-folding time and oscillation period are repeated. For real POPs the e-folding time has also been estimated from the time series. The standard deviation (STD. DEV.) of the time series is given.

The other information given is for debugging purposes and are not generally useful but for completeness they are :- ACF(1) -- the lag-1 auto-correlation function; TAU - the estimated decorrelation time; ANGLE -- the difference be-tween the angles for the two POP rotation choices; NOISE VARIANCE -- the variance of the 'EOFs' defined by equations (23) and (24).

```
****************************************
******** TIME SERIES STATISTICS ********
****************************************
POP   E-FOLDING TIME OSC.PERIOD STD.DEV.   ACF(1) TAU     ANGLE      NOISE VAR.

NR.    POP      EST.    POP      ESTIM.    EST.   EST.     DEG.      EOF1   EOF2
 1R     20.13            6.77     5.74*
  I     20.13            6.77     5.78*                             -35.0    80.5   79.8
 2R     10.03            8.89     2.80*
  I     10.03            8.89     2.78*                              44.0    12.2   12.1
 3R     11.08            9.45     2.62
  I     11.08            9.45     2.59                               75.7     9.8    9.6
 4R      8.95           10.71     3.46*
  I      8.95           10.71     3.60*                             -45.5    15.1   14.3
 5R     10.16           14.87     3.51*
  I     10.16           14.87     3.48*                              21.3     9.0    8.9
 6R      7.80           17.80     1.98*
  I      7.80           17.80     1.97                               38.5     2.3    2.3
 7R      5.13           25.51     1.52*
  I      5.13           25.51     1.51*                             -10.9     1.1    1.1
 8R      9.19           36.47     2.79*
  I      9.19           36.47     3.13*                             -80.5     2.9    2.1
 9      11.46   11.97            1.23      0.9    22.9
10       7.03    7.55            1.16      0.9    14.1
```

The correlation coefficient is calculated between all pairs of time series coefficients (b_i, b_j). This is defined as

$$c = \frac{\mathrm{cov}\{b_i(t) b_j(t)\}}{\sqrt{\mathrm{var}\{b_i(t)\}\mathrm{var}\{b_j(t)\}}}$$

This is obviously symmetric with respect to i,j so only the 'lower diagonal' is printed. Note that $c_{ii} = 1$.

CORRELATION MATRIX

1R	1.00																		
I	0.00	1.00																	
2R	-0.07	0.07	1.00																
I	-0.07	-0.07	0.00	1.00															
3R	-0.06	-0.09	0.07	0.33	1.00														
I	0.08	-0.06	-0.34	0.07	0.01	1.00													
4R	0.07	0.05	0.04	-0.15	0.00	0.29	1.00												
I	-0.07	0.07	0.14	0.02	-0.28	-0.03	0.01	1.00											
5R	-0.02	-0.02	0.02	0.17	0.12	-0.02	-0.12	0.09	1.00										
I	0.02	-0.01	-0.17	0.03	-0.02	0.09	-0.13	-0.14	-0.01	1.00									
6R	-0.01	0.02	0.09	0.03	0.02	-0.05	-0.20	0.11	0.36	0.02	1.00								
I	0.00	0.00	-0.05	0.09	0.05	0.03	-0.10	-0.21	0.03	0.37	0.00	1.00							
7R	0.02	0.00	0.00	-0.07	0.03	-0.05	0.01	-0.02	0.04	-0.17	-0.08	-0.26	1.00						
I	-0.01	0.02	0.08	0.00	0.07	-0.04	-0.03	0.08	0.18	0.00	0.33	-0.11	-0.04	1.00					
8R	0.00	0.01	-0.03	0.02	-0.04	0.03	-0.07	-0.03	-0.17	0.29	-0.03	0.14	-0.15	-0.34	1.00				
I	-0.01	0.00	0.03	-0.04	0.01	-0.04	0.04	-0.06	-0.25	-0.15	-0.21	-0.01	0.35	-0.16	0.12	1.00			
9	0.00	-0.01	-0.02	0.05	0.05	-0.01	0.06	0.03	0.15	-0.05	0.11	0.05	0.04	0.06	-0.19	-0.08	1.00		
10	-0.02	0.00	0.06	-0.07	0.02	0.00	0.07	-0.03	-0.14	-0.10	-0.05	-0.15	0.13	0.08	0.10	0.55	-0.23	1.00	

Next a set of statistics is printed in order to aid the assessment of the 'usefulness' of a given POP. The notation has been introduced in Section 2.7. Note that the POPs have been ordered as explained there and thus the POPs are not listed in the same order as given previously.

```
************************************************************
******************** STATISTICS PACKAGE ********************
************************************************************

                    - PLEASE SEE MANUAL FOR EXPLANATION OF QUANTITIES

EPS - RELATIVE ERROR
XI - BIAS FACTOR
EV - EXPLAINED VARIANCE
CEV - CUMULATIVE EXPLAINED VARIANCE

3 CASES :--
*1* - INDEPENDENT SINGLE PREDICTION ASSESSMENT
*2* - SEQUENTIAL INCREMENTAL PREDICTION ASSESSMENT
*3* - COMPLEMENTARY INCREMENTAL PREDICTION ASSESSMENT


****STATISTICS FOR POPS PREDICTING DATA****

                  *1*                *2*                *3*
POP NO      EPS       XI       EPS       XI       EPS       XI       EV       CEV
```

POP NO	*1* EPS	*1* XI	*2* EPS	*2* XI	*3* EPS	*3* XI	EV	CEV
1	0.780	0.397	0.780	0.397	0.324	0.895	0.391	0.391
5	0.941	0.131	0.895	0.221	0.491	0.759	0.114	0.512
8	0.975	0.075	0.904	0.195	0.565	0.681	0.049	0.602
4	0.943	0.131	0.877	0.305	0.488	0.762	0.112	0.694
3	0.995	0.046	0.888	0.239	0.604	0.635	0.010	0.758
2	0.961	0.086	0.805	0.371	0.578	0.666	0.077	0.843
6	0.990	0.034	0.837	0.301	0.706	0.501	0.021	0.890
7	0.993	0.021	0.864	0.253	0.793	0.370	0.015	0.918
9	0.991	0.018	0.881	0.224	0.849	0.279	0.017	0.936
10	0.994	0.014	0.862	0.257	0.862	0.257	0.012	0.953
TOTAL	0.217	0.953						

****STATISTICS FOR POPS PREDICTING RATE OF CHANGE OF DATA****

	1		*2*		*3*			
POP NO	EPS	XI	EPS	XI	EPS	XI	EV	CEV
1	0.634	0.618	0.634	0.618	0.237	0.944	0.598	0.598
4	0.966	0.104	0.889	0.281	0.460	0.788	0.067	0.682
3	1.000	0.042	0.884	0.241	0.558	0.689	0.000	0.751
2	0.962	0.092	0.831	0.372	0.511	0.739	0.075	0.828
5	0.982	0.066	0.886	0.385	0.535	0.714	0.036	0.865
8	1.010	0.010	0.863	0.274	0.702	0.508	-0.019	0.900
6	1.004	0.011	0.846	0.294	0.745	0.445	-0.008	0.928
7	0.998	0.012	0.848	0.282	0.807	0.348	0.004	0.948
10	1.000	0.004	0.920	0.153	0.910	0.172	0.000	0.956
9	1.000	0.003	0.933	0.129	0.933	0.129	0.000	0.962
TOTAL	0.195	0.962						

The "*explained local variance*" is related to this but is not contained in the printer output. Instead it is output on one of the output files (see Section 4.2). These are numbers defined on each original (Euclidean) point and for each POP (pair) giving a measure for the variance explained by that POP (pair) as a spatial pattern. They are defined in the same way as the explained (total) variance given above, (39), with the only difference that (37) is evaluated at every Euclidean point and the product is the product of real numbers.

4.2 DOCUMENTATION OF OUTPUT FILES

The POP program writes the output data on 12 different files with logical numbers 1,2,...,12. They may well be merged together later on to one file. This is supported by the fact that the first three header fields are a unique key for the data record, i.e. no two headers are the same on all output files.

General description of the *Extra Code* header:

KDATE Calendar date or some other number describing the sequence of a fixed variable. For the POPs, KDATE is the number of a POP or of a pair of real POPs. Note that a negative KDATE denotes a set of eigenvalues, periods or similar items.

KVAR Code for the variable. In the POP context, this is simply the logical unit number added to 500.

KLEV Additional sequence information if KDATE is not unique for the variable under KVAR. For a complex POP, the real part has KLEV = 1, the imaginary part has 2.

KLEN Always the length of the following data record.

The file name extensions are given in the table on the next two pages.

Note: In the filtered data all data gaps are re-coded and all calendar dates KDATE are restored in the headers from the original data file.
All spatial points where no observations are available at all, i.e. for each time step the value at these points is *PPGAP*, are also coded as being gappy in the output of all EOFs and of all POPs and APOPs in Euclidean space.
Time steps at which no observation is available, i.e. for each spatial point the value at these time steps is *PPGAP*, should be interpolated prior to the POP analysis because this is not done in the program.

Log. unit number / file ext.	KDATE	KVAR	KLEV	KLEN	Data description
1 .filt	Calendar date	501	--	NTS	Data prepared by the filter algorithms (data gaps are re-coded)
2 .var	-1	502	--	NTS	Local variances
	POP No. (group)		--	NTS	Explained local variances for a POP group
3 .eof	-1	503	--	NEOF	EOF eigenvalues
	EOF No.		--	NTS	EOFs
4 .eofc	-1	504	--	NEOF	EOF eigenvalues
	EOF No.		--	NTO	Principal Components time series
5 .eofsp	-1	505	--	NEOF	EOF eigenvalues
	-2		--	depends	Frequencies
	EOF No.		1	depends	Variance spectra of Principal Components
6 .pop	-1	506	--	NEOF	e-folding times
	-2		--	NEOF	POP periods
	POP No. (group)		1	NTS	POP in Euclidean space: real part
	POP No.		2	NTS	imaginary part
7 .pope	-1	507	--	NEOF	e-folding times
	-2		--	NEOF	POP periods
	POP No. (group)		1	NEOF	POP in EOF space: real part
	POPNo.		2	NEOF	imaginary part
8 .popc	-1	508	--	NEOF	e-folding times
	-2		--	NEOF	POP periods
	POP No. (group)		1	NTO	POP coefficient time series: real parts
	POP No.		2	NTO	imaginary parts

Log. unit number / file ext.	KDATE	KVAR	KLEV	KLEN	Data description
9 .popsp	-1	509	--	NEOF	e-folding times
	-2		--	NEOF	POP periods
	-3		2*NC	depends	Frequencies (NC is the number of chunks used in the *Bartlett* procedure to compute the spectra, 2*NC is the number of degrees of freedom which may be used to compute confidence intervals)
	POP No.		1	depends	POP coeff. time series: Variance spectra, real parts
	POP No.		2	depends	Variance spectra, imaginary parts
	POP No.		3	depends	Phase spectra
	POP No.		4	depends	Squared coherence spectra
10 .apop	-1	510	--	NEOF	e-folding times
	-2		--	NEOF	POP periods
	APOP No.		1	NTS	APOP in Euclidean space real part
	APOP No.		2	NTS	imaginary part
11 .apope	-1	511	--	NEOF	e-folding times
	-2		--	NEOF	POP periods
	APOP No.		1	NEOF	APOPs in EOF space: real parts
	APOP No.		2	NEOF	imaginary parts
12 .ceof	-1	512	--	NCEOF	CEOF eigenvalues
	CEOF No.		1	NCEOF	CEOF: real parts
	CEOF No.		2	NCEOF	CEOF: imaginary parts

5. INTERPRETATION OF OUTPUT.

As with all statistics, the user should be wary of the results produced by the program. To give a long list of significance tests would probably overwhelm the user and thus make more difficult the assessment of the results. The following are guidelines for determining whether a particular set of POPs are useful, based on the experiences of certain members of the Max-Planck-Institut. No one piece of information can alone determine whether a POP is 'significant'.

- Look at the output from the 'Statistics Package' to see how much of the variance in the original data each POP explains (plus the other associated quantities). A good POP will probably have ε small and ξ large.
- Look at the e-folding time for each POP. If this is less than one (time step) then that POP is not useful. The ratio of the e-folding time to the oscillation period should also be not too small (greater than 1/4, say) - this ratio shows how much the POP will 'develop' before becoming damped out.
- Look at the variance of the complex POPs as given in the *Time Series Statistics* section of the output. If the POP model (..) is to be valid then the variance of the real and the imaginary part of a POP have to be of about equal magnitude.
- Perform a *Cross Spectral Analysis* of the time series of POP coefficients - in particular the variance, relative phase and coherence as a function of frequency (only the first is defined for real POPs). Jenkins and Watts (Jenkins, G., M., and Watts, D., G., ,1968: *Spectral Analysis and its Applications*, Holden Day, San Francisco) is a standard work on this subject.

  If the POP model is good then the maximum variance should be near the POP frequency, but be careful that the maximum is not a false one produced by the filter window. Also the phase near the POP frequency should be close to -90° and the coherence should be high. The phase of -90° indicates that the imaginary part of the coefficient time series leads the real part for a quarter of a period thus reflecting the evolution equation (18). Any other phase is not consistent with the POP model.

- Plot any POPs which look 'interesting' under the above definitions. It is probably most useful to reconstruct the POPs in some physical space which can be readily interpreted. Real POPs represent no oscillation but a continuously damped pattern. Real and imaginary parts of complex POPs should be plotted together if possible. Interpret the evolution of these in the light of the known cycle

  IMAG ⟶ REAL ⟶ -IMAG ⟶ -REAL ⟶ IMAG ... etc.

  If any maxima in the real and imaginary part are displaced relative to each other and if the phase is -90° then the POP can be said to be a 'travelling wave'. If the maximum/minimum of either the real or the imaginary part is close to zero then the pattern of the other part describes a 'standing oscillation'. If the amplitude of one part is significantly smaller than that of the other part the POP can be thought of as a superposition of a travelling and a standing oscillation.

  You may also make a plot of the local explained variances to see for which points or in which regions your POP explains most of the total variance.

To make things clearer we now go through the above procedure for the case of the example given before - the 'Southern Hemisphere/Wavenumber 8 baroclinic waves':

- Statistic Package.

  Here we see that the most outstanding POP is POP No. 1 which accounts for 39% of the total variance in the original data set. But the relative error (ε_{in}) for the POP predicting the data is about 0.78 and the parallel component (ξ_{in}) is only 0.39. This indicates that the POP has a fairly large component which is statistically orthogonal to the data. However these numbers change to (ξ_{in}) = 0.89, (ε_{in}) = 0.32 for the POP predicting the data after all the other POPs have been subtracted. The total prediction variables (ξ_T,ε_T) = (0.95, 0.27) are also good and the statistical variables for the POP predicting the rate of change of data is even better.

  All other POPs have higher relative errors and lower bias factors.

- E-folding time. (Output shown previously)

  Here we see that the first POP has an e-folding time of 20 time steps. The ratio of the e-folding time to the oscillation period is about 3 so that about 3 full oscillations will be 'seen' before the POP dies away (in the absence of subsequent driving impulse, that is).

- Time Series Statistics.
  It can be seen that the variance of the coefficient time series driving the POP are about the same for the real and imaginary parts, confirming that the evolution of the signal is in fact described by the POP cycle (25).
- Cross Spectral Analysis
  Figure 3 shows the results produced by a spectral analysis package at the Max-Planck-Institut. It used a Bartlett procedure (*q.v.*) with 3 chunks.
  Note the logarithmic scales and be aware that the filter window 4/6-50/60 (i.e. PMIN/P2 - P1/PMAX) automatically produces a decrease of variance for periods larger than about 60 time steps.
  In the variance plot the solid line represents the variance associated with the real part of the POP and the dashed represents the imaginary part. The variance has a maximum around the POP frequency (indicated by the vertical dashed line) and is fairly flat in this period band.
  The relative phase near the POP frequency is close to -90° and the coherence (squared) is also well above the 99% coherence squared level.

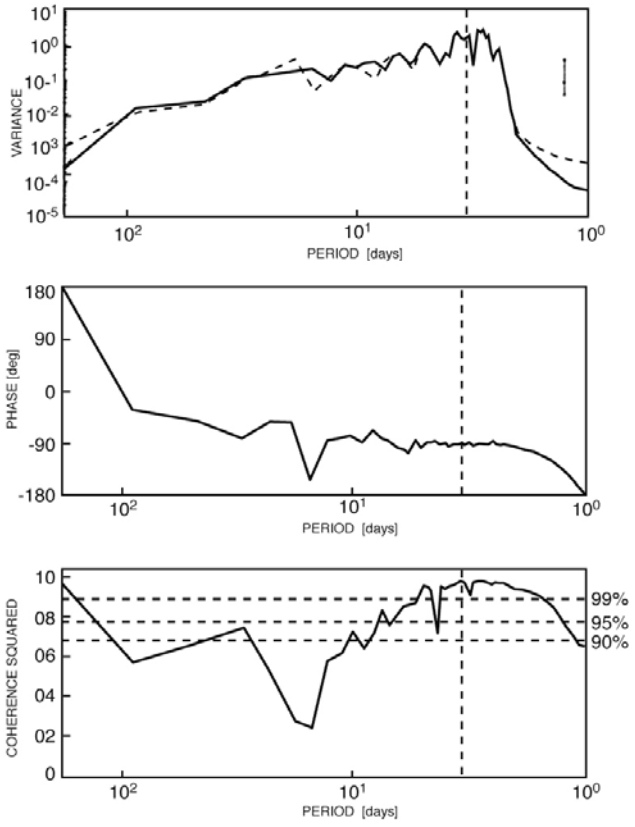


Figure 3 Example Cross Spectral Analysis for *Southern Hemisphere / Wavenumber 8 baroclinic wave.*

- Plot of POP.

The data analysed comprised cosine- and sine-coefficients of zonal wave-number 8 geopotential height for 31 latitudes and 7 levels. Thus, a (y,z) contour plot of amplitude and phase is adequate for reconstructing the data in real space. Figure 4 shows the selected POP 1.

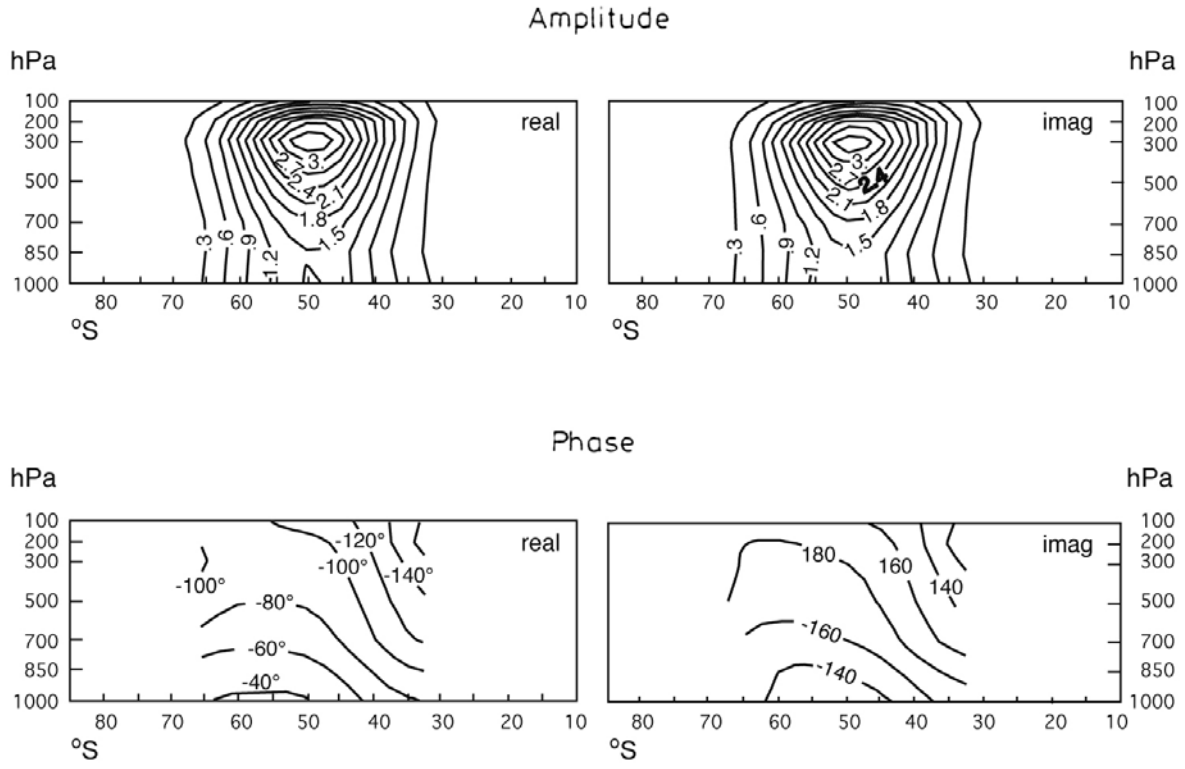


Figure 4: Plot of POP for *Southern Hemisphere / Wavenumber 8 baroclinic wave.*

The horizontal scale is the latitude and the vertical scale is the vertical level in a logarithmic pressure coordinate system. The two panels on the left show the amplitude/phase representation of the real part, on the right the imaginary part is shown. The phases are only plotted where the amplitude is at least 5% of it's maximum value.
It can be seen that the amplitudes of both patterns are of equal magnitude and structure. The structure of the two phase plots is also very similar with a phase difference (real - imaginary) of about 90° so that, having already noted that the relative phase is about -90°, this POP represents a smooth eastward propagation.

APPENDIX 1: PRINCIPAL ADJOINT PATTERNS

In this appendix we briefly put together the algebra of adjoint patterns. For this purpose the Euclidean dot product in a n-dimensional space is considered. For any two complex n-dimensional vectors $\mathbf{x}$ and $\mathbf{y}$, the dot product (scalar product) $\mathbf{x}\cdot\mathbf{y}$ is defined as

$$\mathbf{x}\cdot\mathbf{y} = \sum_{k=1}^{n} x_k \, y_k^* = \mathbf{y}^{*T}\,\mathbf{x}$$

where * denotes the complex conjugate and ^{T} transposition. Note that $\cdot$ is linear in the first and conjugate linear in the second argument.

Theorem: If $\mathbf{B}$ is a real matrix with transposed matrix $\mathbf{B}^T$ the following statements hold:

(A.1) $(\mathbf{Bx})\cdot\mathbf{y} = \mathbf{x}\cdot(\mathbf{B}^T\mathbf{y})$ for all $\mathbf{x}$ and $\mathbf{y}$.

(A.2) $\mathbf{B}^T$ and $\mathbf{B}$ have identical eigenvalues.

(A.3) If $\mathbf{p}^k$ is an eigenvector of $\mathbf{B}$ with eigenvalue λ^k and $\mathbf{q}^j$ an eigenvector of $\mathbf{B}^T$ with eigenvalue λ^j, then

$$\mathbf{p}^k\cdot\mathbf{q}^{j*} = 0$$

if

$$\lambda^k \neq \lambda^j \quad .$$

Proof:

(A.1) $$(\mathbf{Bx})\cdot\mathbf{y} = \sum_k \Big(\sum_j b_{kj}x_j\Big)\, y_k^* = \sum_j x_j \Big(\sum_k b_{kj}y_k\Big)^* = \mathbf{x}\cdot(\mathbf{B}^T\mathbf{y})$$

(A.2) $$det(\mathbf{B} - \lambda\mathbf{I}) = det(\mathbf{B} - \lambda\mathbf{I})^T = det(\mathbf{B}^T - \lambda\mathbf{I})$$

(A.3) $$\lambda^k(\mathbf{p}^k\cdot\mathbf{q}^{j*}) = (\lambda^k\mathbf{p}^k)\cdot\mathbf{q}^{j*} = (\mathbf{Bp}^k)\cdot\mathbf{q}^{j*} = \mathbf{p}^k\cdot(\mathbf{B}^T\mathbf{q}^{j*}) = \mathbf{p}^k\cdot(\lambda^{j*}\mathbf{q}^{j*}) = \lambda^j(\mathbf{p}^k\cdot\mathbf{q}^{j*})$$

if $\lambda^k \neq \lambda^j$, it follows that $(\mathbf{p}^k\cdot\mathbf{q}^{j*}) = 0$. ■

Generally, an operator $\mathcal{B}^*$ satisfying $\langle\mathcal{B}\mathbf{x},\mathbf{y}\rangle = \langle\mathbf{x},\mathcal{B}^*\mathbf{y}\rangle$ is called to be adjoint to $\mathcal{B}$. In view of (A.1), the transpose matrix $\mathbf{B}^T$ is the adjoint matrix of $\mathbf{B}$.

In the POP concept, the POPs are the eigenvectors of the matrix $\mathbf{B}$ (see (3)). The *principal adjoint patterns (PAPs)* are the eigenvectors of the adjoint matrix $\mathbf{B}^T$. The significance of the PAPs in the POP concept originates from the following.

Theorem: Let all eigenvalues of **B** be mutually different. If

$$\mathbf{P} = (\mathbf{p}^1 \mid \mathbf{p}^2 \mid \ldots \mid \mathbf{p}^n)$$

is the (complex) matrix of eigenvectors of **B**, i. e. the *POP matrix*, the matrix of eigenvectors

$$\mathbf{Q} = (\mathbf{q}^1 \mid \mathbf{q}^2 \mid \ldots \mid \mathbf{q}^n)$$

of $\mathbf{B}^T$, i e. the *PAP matrix*, is given by

(A.4) $$\mathbf{Q} = (\mathbf{P}^T)^{-1}$$

The columns $\mathbf{q}^k$ of **Q** are the unique solutions of

(A.5) $$\mathbf{p}^j \cdot \mathbf{q}^{k^*} = \delta_{jk} \qquad ,j=1,n.$$

For the columns $\mathbf{q}^k$ of **Q** the following relation holds:

(A.6) $$\mathbf{x} = \sum_k (\mathbf{x} \cdot \mathbf{q}^{k^*}) \mathbf{p}^k$$

for all **x**.

Proof:

(A.4) If **P** is the POP-matrix of **B** and Λ the matrix with the eigenvalues in the main diagonal:

$$\mathbf{P}^{-1}\mathbf{B}\mathbf{P} = \Lambda \quad \Rightarrow (\mathbf{P}^{-1}\mathbf{B}\mathbf{P})^T = \Lambda^T = \Lambda \quad \Rightarrow \mathbf{P}^T\mathbf{B}^T(\mathbf{P}^{-1})^T = \Lambda \, .$$

Because of $(\mathbf{P}^{-1})^T = (\mathbf{P}^T)^{-1}$ after setting $\mathbf{Q} = (\mathbf{P}^T)^{-1}$:

$$\mathbf{Q}^{-1}\mathbf{B}^T\mathbf{Q} = \Lambda$$

showing that **Q** is the eigenvector matrix of $\mathbf{B}^T$.

(A.5) is equivalent to $\mathbf{P}^T\,\mathbf{Q} = \mathbf{1}$ which follows immediately from $\mathbf{Q} = (\mathbf{P}^T)^{-1}$.

(A.6) Since (A.4) is equivalent to $\mathbf{P}\mathbf{Q}^T = \mathbf{1}$

$$\mathbf{x} = (\mathbf{P}\mathbf{Q}^T)\mathbf{x} = \mathbf{P}(\mathbf{Q}^T\mathbf{x}) = \sum_k (\mathbf{q}^k\mathbf{x}^*)\,\mathbf{p}^k = \sum_k (\mathbf{x}\cdot\mathbf{q}^{k^*})\,\mathbf{p}^k \quad . \quad \blacksquare$$

Relation (A.6) allows the calculation of the POP coefficients a_k (see (10)) of a certain state **x** by projecting this state on the adjoint pattern $\mathbf{q}^k$:

(A.7) $$a_k(t) = \mathbf{x}(t)\cdot\mathbf{q}^{k^*} = \mathbf{q}^k\cdot\mathbf{x}(t) \quad .$$

APPENDIX 2: TIME FILTER PARAMETERS

The POP program has an option which allows the user to time filter the given time series before processing thus removing features on 'uninteresting' time scales. For example, suppose the given time series contains daily values of some quantity and the user wishes to look for the important oscillations on time scales of the order of a few weeks. Now if this time series has a very strong annual cycle then this may dominate in the POP analysis, so much so that the features of interest may be completely lost. The filtering subroutine would enable the user to remove all variations with periodicity greater than, say, one month.

The filtering technique is Fourier filtering whereby the data set X(r,t) is transformed into Fourier space by the finite Fourier transform,

$$\tilde{X}(r,\nu) = \frac{1}{\sqrt{n}} \sum_{t=1}^{n} X(r,t) \exp(-i2\pi\nu t)$$

where ν is a frequency component. The filter window $G(\nu)$ is defined by the four parameters PMAX, P1, P2, PMIN such that,

$$G(\nu) = \begin{cases} 0, & \text{if } \nu < \nu_{MAX}; \\ \frac{1}{2}\left[1 + \cos\left(\frac{\pi}{\nu_1 - \nu_{MAX}} (\nu - \nu_1) \right) \right], & \text{if } \nu_{MAX} \le \nu \le \nu_1; \\ 1, & \text{if } \nu_1 \le \nu \le \nu_2; \\ \frac{1}{2}\left[1 - \cos\left(\frac{\pi}{\nu_{MIN} - \nu_2} (\nu - \nu_{MIN}) \right) \right], & \text{if } \nu_2 \le \nu \le \nu_{MIN}; \\ 0, & \text{if } \nu_{MIN} < \nu; \end{cases}$$

where $\nu_{MIN} = \frac{1}{PMIN}$ etc. Figure (1) may make this clearer.

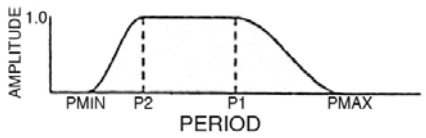


Figure 1: Filter window with cosine tails.

The reason for the introduction of the cosine 'tails' is to suppress the secondary maxima that might occur for a pure 'rectangular' filter, particularly for the case of the response to a pulse (Figure 2) which may thenbe falsely interpreted as an true oscillation of the data set. This problem is

particularly important when a very narrow filter window is requested. To help the user detect any such 'false signals' the program output includes the 'Equivalent Digital Filter Weights' which effectively shows how the filter transforms a pulse at t=1.

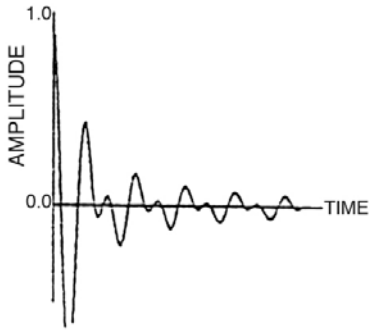


Figure 2: Response to unit impulse at t=0 after filtering with rectangular window.

The user may select the following special types of window;

General Rectangular filter;	P1=PMAX, P2=PMIN
High Pass filter;	P2=PMIN=1 (default)
Low Pass filter;	P1=PMAX=2×<length of time series>
Default values;	P1=PMAX=2×<length of time series>, P2=PMIN=2

Note that in the user parameter line, the filter parameters have to be typed in the order PMIN, P2, P1, PMAX, separated by one or more spaces. A default value is inserted by the program if the value -1.0 is typed at the corresponding place. For instance, in the second case when the program is given P1=PMAX=-1, it sets P1=PMAX=2×<length of time series> because this corresponds to the lowest frequency represented in the data. If all four parameters take their default values, then no filtering is performed. This can better be achieved by not setting the -T option in the input parameter line of the POP program (see input description).

The time series is then reconstructed using the inverse Fourier transform with each frequency weighted by the window,

$$X'(r,t) = \frac{1}{\sqrt{n}} \sum_{\nu=1}^{n} G(\nu)\tilde{X}(r,\nu)\exp(i2\pi\nu t)$$

APPENDIX 3: DATA GAPS AND THE ESTIMATION OF MOMENTS, ESPECIALLY EOFS

Usually, the length of the multivariate time series will be larger than the number of time series. However, if the spatial dimension n of the problem is much larger than the number m of samples a considerable amount of computing time can be saved by an alternative approach. It turns out that the EOFs, i.e. the eigenvectors of the n×n-covariance matrix, can equally well be obtained by computing the eigenvalues and eigenvectors of a (smaller) m×m-matrix which is related to the transposed ('tilted') data matrix. For details of theory and computation rules of this procedure see

Hans von Storch and Gerhard Hannoschoeck, "Comments on 'Empirical Orthogonal Function Analysis of Wind Vectors Over the Tropical Pacific Region'", Bull. Am. Met. Soc., Vol. 65 (1984), page 162.

In the POP program it is automatically decided weather to use the untilted or the tilted data matrix. If no gaps are contained in the data set the results are identical.

A special problem arises when computing EOFs from the tilted data matrix with data gaps included. When moments are estimated using the input data set, there is generally the convention that a sum of quantities is divided by the number of terms. This number is reduced if data gaps occur for the program will not interpolate from existing data then. Considering the covariance between the two points with numbers *i* and *j*, the number to divide the sum by is

$$n_{ij} := \text{number of times with existing data at i } \textit{and} \text{ j.}$$

This nonlinear dependence cannot be exactly transferred to the tilting case computation. But a good approximation is the following numbers assuming that all spatial points have about the same 'no-data times'. Instead of n_{ij} we use

$$\sqrt{n_i}\sqrt{n_j}$$

with n_i := number of times with existing data at i.
The program uses this approximation in the tilting case because then the new summations can be split up between pairs of *i* and *j*.

However, because of the assumption made with this approximation the POP program does not use the tilted data matrix by default if it has detected data gaps. The user can force the use of the smaller matrix in the computation of the EOFs by giving the option "-f" to the program (see section 3.2). So it is ensured that no false results are obtained 'by accident' if there are many data gaps which are irregularly spread over space and time.

APPENDIX 4: USER'S GUIDE

A INTRODUCTION

This section is intended to provide a guide to install and use the POP program in a general environment. Additionally, it gives hints for local users at the *Deutsches Klimarechenzentrum DKRZ* for accessing the program and for post-processing.

B RUNNING THE POP PROGRAM

The main program is the FORTRAN file "pop.f". The input data must be in *EXTRA Code* format, or be converted to that format. See section 3.1 and

Borgert and Welke, 1991: *Description of Programs Handling Files in EXTRA and GRIB Format*, Max Planck Institut für Meteorologie, Hamburg, FRG

for details of the Extra Code.

B.1 Generating executable program from source

To get an executable POP program the FORTRAN source code in file "pop.f" has first to be compiled with your FORTRAN compiler, e.g. cft77 in the UNICOS operating system on Cray machines. You need two additional files in your current working directory, "scalars.i" and "arrays.i", which contain the variable declarations for scalars and arrays and which are automatically included into the source code in the appropriate places during compilation time. If you have to make changes to any declaration you only have to change these files. If your FORTRAN compiler doesn't support the INCLUDE-statement you have to edit "pop.f" and insert the two files manually, look for statement INCLUDE.

The second step consists of linking all subroutines with the loader, e.g. segldr in UNICOS. The only external subroutines which have to be linked to the program are from the NAGLIB mathematical library and are listed in Appendix 5. If you don't have this library you will have to substitute them by own versions with the same effect.

B.2 Switching dynamical field lengths on and off

In the Cray version, the array dimensions *NSER*, *NTSDIM* and *NEODIM* (length of time series, number of time series and number of EOFs) are defined dynamically at run time as possible on the Cray. This is done in subroutine *SETDIM* (U0101). In this case the PARAMETER statement in the following piece of code from "scalars.i" has to be blinded to comments by the user, and the COMMON statement is read by the compiler.
Since not all Fortran compilers (e.g. f77 on Sun workstations) allow adjustable declaration of non-parameter arrays in subroutines one can switch to standart FORTRAN with fixed field lengths by de-commenting the PARAMETER line and hiding the COMMON statement. Also, you have to make *SETDIM* an empty subroutine (e.g. by comment C's).
Example (hide one of the following two statements by comment C's):

```
      COMMON /DIMS/    NSER,   NTSDIM,   NEODIM

      PARAMETER ( NSER = 1500, NTSDIM = 1400, NEODIM = 18 )


C     VARIABLE   TYPE       PURPOSE.
C     --------   ----       --------

C     NSER       INTEGER    MAXIMAL LENGTH OF TIME SERIES.
C     NTSDIM     INTEGER    MAXIMAL NUMBER OF TIME SERIES (IN EUCLIDEAN
C                           SPACE).
C     NEODIM     INTEGER    MAXIMAL NUMBER OF EOF'S.
```

B.3 Running the POP program on different operating systems

Non-UNIX system:
In the default implementation the POP program reads the parameter line from standard input (e.g. terminal). This is done by setting variable *LPINPUT* to *.TRUE.* in include file "scalars.i". If, for some reason, you want to change this behaviour so that the parameters are read from a file "POP.PAR" just set *LPINPUT* to *.FALSE.* in "scalars.i". This input file then consists of one line containing the parameters as described in section 3.1. Don't forget the delimiter -- before the filter parameters and don't exceed 120 characters.
Example:

```
      LOGICAL     LPUNIX

      PARAMETER( LPUNIX = .TRUE. )
```

UNIX-system:

If you are lucky to be in a UNIX environment you should read the parameters from standard input. The syntax of the parameter line is like in shell commands. The output from the UNIX shell command 'getopt' is a standard form of command parameters which is fully accepted by this program. Therefore, the most comfortable way is to let the POP program be called from a command file (shell script) to gain all shell syntax features including syntax check. An example of such a (Bourne) shell script is given below.

```
#!/bin/sh
#
# All lines beginning like this one are comments from here on!
#
set -- 'getopt ECThdrupfln:t:e:c:s:m:i:o:  $*'
#
# Give user help for syntax error or if called with option -h
#
err=$?
if [ $err -ne 0  -o "$1" = "-h" ]
then
  echo "Usage: $0 [-ECThdrupfl] [-n NTS] [-t NTO] [-e NEOF] "
  echo "........... [-c NC] [-s DT] [-m FACT]"
  echo "........... [-i infile] [-o outfile] -- [PMIN P2 P1 PMAX]"
       .
       .   (more help messages)
       .
  exit 1
fi
#
echo $* | <whereever the executable is located>/pop.x
#
exit
```

With this shell script the last example from section 3.2 would be run by just typing

```
pop -Tdp -e18 -c3 -iENSO.DAT -- 2 2 8 10        <cr>
```

C LOCAL USER'S GUIDE

C.1 Accessing the POP program at the *DKRZ*

The program will generally be run on one of the Cray computers by calling the shell script "pop" as described in B.3. At login every user has set a global environment variable

EXTRA=/pool/EXTRA/programs

which leads to the directory that contains "pop". Thus, to run the POP program you either have to

add this directory to your PATH variable
for the Bourne shell in ".profile" by
PATH = $PATH:...:$EXTRA
and for the C-shell in ".cshrc" by
set path="$path $EXTRA"
and just type
pop .... <cr>

or type

$EXTRA/pop .... <cr> .

If you enter "pop -h" you will get a help message. If you specify "pop" with no options a POP analysis will be performed with all options set to the default values (see Section 3.2).

C.2 Post-Processing

The output from the program probably contains much information that is not of use to a particular user, for example, you may only be interested in the POPs themselves, and don't care about the time series etc. Then you will keep only the corresponding output files in permanence. To process the resulting data further you may use the fact that all are in the standard format Extra-Code which is supported by numerous data manipulation and plot routines available and documented at the MPI (Borgert and Welke, 1991).

APPENDIX 5: EXTERNAL SUBROUTINES

The POP program uses several *NAGLIB* routines to perform various operations. Users who do not have access to this mathematical library will have to supply their own versions. The following is a brief description of the functions performed by these subroutines. In addition, the error codes are explained which are generated by the NAGLIB routines in case of abnormal program termination (FORTRAN variable IFAIL).

A) Fourier transforms.

Subroutines C06FAF, C06FBF and C06FCF calculate the discrete Fourier transform $\hat{z}_k$ of a sequence of N data values x_j

$$\hat{z}_k = \frac{1}{\sqrt{N}} \sum_{j=1}^{N} x_j \, exp(-i \, \frac{2\pi \, j \, k}{N}) \;, \quad k=1,...,N$$

in 3 different cases:

C06FAF (Subroutine FILTTS): x_j is a sequence of real data values.

```
DIMENSION X(p), WORK(q)                                  with p,q ≥ N
      .
      .
CALL C06FAF(X,N,WORK,IFAIL)
      .
      .
END
```

X is the array, of dimension at least N, containing the real data. Then, $\hat{z}_k$ is a Hermitian sequence of complex values (i.e. $\hat{z}_{N-k}$ is the complex conjugate of $\hat{z}_k$). On exit, X contains these transformed data in Hermitian form, i.e. for $\hat{z}_k = a_k + ib_k$

$$X(k) = \begin{cases} a_k = a_{N-k} \;, & \text{for } 1 \le k \le N/2 \\ b_{N-k} = -b_k \;, & \text{for } N/2 < k \le N \end{cases} .$$

C06FBF (Subroutines FILTTS and FILGEW): x_j is a Hermitian sequence of N complex data values.

```
DIMENSION X(p), WORK(q)                                    with p,q ≥ N
      .
      .
      .
CALL C06FAF(X,N,WORK,IFAIL)
      .
      .
      .
END
```

Before entry, X contains the x_j stored in Hermitian form (see above). The transforms $\hat{z}_k$ are purely real and are stored in X on exit.
In particular, C06GBF followed by C06FBF forms the inverse Fourier transform to that performed by C06FAF.

C06FCF (Subroutines SPEC and CEOFAN): x_j is a sequence of N complex data values.

```
DIMENSION X(p), Y(p'), WORK(q)                             with p,p',q ≥ N
      .
      .
      .
CALL C06FAF(X,Y,N,WORK,IFAIL)
      .
      .
      .
END
```

Before entry, X(j) contains the real part and Y(j) contains the imaginary part of the data (j=1,...,N). On successful exit, the two arrays contain the real and imaginary part of the transformed data.
In particular, to compute the inverse Fourier transform C06FCF should be preceded and followed by a call to a routine which computes the conjugate of a complex sequence.

In all three program fragments above WORK is a workspace array of dimension of at least N and IFAIL is a parameter which determines how errors are treated. On exit, IFAIL=0 if no errors occurred. If the routine has not been successful, probably for reasons unique to the NAGLIB, then IFAIL contains the following:

1 N has at least one prime factor greater than 19. NAGLIB is unable to compute the transformation in this instance.
2 The number of prime factors of N is greater than 20. This can only occur for $N > 10^6$.
3 $N \leq 1$.

B) Subroutine **C06GBF** (Subroutines FILTTS and FILGEW) calculates the complex conjugate of a Hermitian sequence of N complex data values, i.e. forms the conjugate of the sequence formed by C06FAF.

```
DIMENSION X(p), WORK(q)                                    with p,q ≥ N
      .
      .
      .
CALL C06GBF(X,N,IFAIL)
      .
      .
      .
END
```

with X containing a Hermitian sequence before entry and it's conjugate on exit. IFAIL = 1 indicates that N < 1.

C) Eigenvalues and Eigenvectors.

F02ABF (Subroutine TSEOFS) calculates all the eigenvalues and eigenvectors of a real symmetric matrix:

```
DIMENSION X(NXD,p), EV(q), EVEC(NED,r), WORK(s)        with p,q,r,s ≥ NXO
      .                                                      NXD,NED ≥ NXO
      .
      .
CALL F02ABF(X,NXD,NXO,EV,EVEC,NED,WORK,IFAIL)
      .
      .
      .
END
```

X is the given symmetric matrix, with first dimension NXD as defined in the calling subprogram, and order NXO. Only the lower triangle is used. On exit, EV contains the eigenvalues in ascending order and the columns of EVEC contain the normalised eigenvectors , where (EVEC(j,i),j=1,N) corresponds to EV(i).
WORK is used as workspace. IFAIL = 0 on successful exit and IFAIL = 1 otherwise (NAGLIB fails if more than 30·NXO iterations are required to isolate all the eigenvalues).

F02AGF (Subroutine POPS) calculates all the eigenvalues and eigenvectors of a real unsymmetric matrix.

```
      DIMENSION  X(NXD,p),  EVR(q),  EVI(r),                    with  p,q,r,s,t,u ≥ NXO
     &           EVECR(NERD,s),  EVECI(NEID,t),                       NXD,NERD,NEID ≥ NXO
     &           INTGER(u)
                .
                .
                .
      CALL  F02AGF(X,NXD,NXO,EVR,EVI,EVECR,NERD,EVECI,NEID,INTGER,IFAIL)
                .
                .
                .
      END
```

X is again the given matrix, with dimensions as above. EVR, EVI are arrays which on exit contain the real and imaginary parts of the eigenvalues of X. The columns of EVECR and EVECI contain the real and imaginary parts of the eigenvectors corresponding to the eigenvalues. The eigenvectors are normalised such that the sum of the squares of the components is one and the eigenvector is rotated so that the component of largest magnitude is real (this also ensures that eigenvectors associated with real eigenvectors are purely real). INTGER is a workspace array which on exit contains the number of interations used to find the eigenvalues. IFAIL has the same meaning as above.

F02AXF (Subroutine CEOFAN) is used only in a CEOF analysis to compute the eigenvalues and eigenvectors of a complex Hermitian matrix. Only the calling sequence is given here, the notation follows the previous cases:

```
      DIMENSION  XR(NXRD,p),  XI(NXID,q),  EV(r)              with  p,q,r,s,t,u,v,w ≥ NXO
     &           EVECR(NERD,s),  EVECI(NEID,t),               NXRD,NXID,NERD,NEID ≥ NXO
     &           WORK1(u),  WORK2(v),  WORK3(w)
                .
                .
                .
      CALL  F02AXF(XR,NXRD,XI,NXID,NXO,EV,EVECR,NERD,EVECI,NEID,
     &                                        WORK1,WORK2,WORK3,IFAIL)
                .
                .
                .
      END
```

Additionally, IFAIL = 2 on exit means that the diagonal elements of XI are not all zero, i.e. the complex matrix is not Hermitian. The normalisation is as for F02AGF.

D) Inverse

F01AAF (Subroutine POPS) computes the approximate inverse of a real matrix.

```
DIMENSION A(NAD,p), B(NBD,q), WORK(r)
      :
      .
CALL F01AAF(A,NAD,NO,B,NBD,WORK,IFAIL)
      :
      .
END
```

with $p,q,r \geq NO$

$NAD,NBD \geq NO$

Before entry, A contains a real matrix of order NO and on exit B contains it's inverse provided A is non-singular. In this case, IFAIL = 0. If A is singular or almost singular, possibly due to rounding errors, IFAIL = 1.

APPENDIX 6: APPLICATIONS OF THE POP METHOD

The POP method is nowadays a diagnostic tool that is routinely used at the Max-Planck-Institut für Meteorologie. In the following a list of publications is given. In these publications the POP technique has been used.

Blumenthal, B., 1991: *Predictability of a coupled ocean-atmosphere model.* J. Climate, in press

Bürger, G., 1991: *Complex Principal Oscillation Patterns.* In preparation

Hasselmann, K..H., 1988: *PIPs and POPs: The Reduction of Complex Dynamical Systems Using Principal Interaction and Oscillation Patterns.* - Geophys. Res. 93, 11.015-11.021

Latif, M. and, M. Flügel, 1990: *An investigation of short range climate predictability in the tropical Pacific.* J. Geophys. Res. 96, 2661-2673

Latif, M., A. Sterl and E. Maier-Reimer, 1991: Climate variability in a coupled GCM. Part I: The tropical Pacific. Submitted to J. Climate

Latif, M., and A. Villwock, 1989: Interannual variability in the tropical Pacific as simulated in Coupled ocean-atmosphere models. - J. Marine Sys. 1, 51-60

Schnur, R. and G. Schmitz, 1991: *POP Analysis - an application to midlatitude baroclinic waves.* - In preparation

Schnur, R. and H.v.Storch, 1991: *POP-Analysis of global large-scale travelling Rossby waves.* - Max Planck Institut für Meteorologie, unpublished manuscript

Storch, H.v., T. Bruns, I. Fischer-Bruns and K. Hasselmann, 1988: *Principal Oscillation Pattern Analysis of the 30- to 60-Day Oscillation in General Circulation Model Equatorial Troposphere.* - J. Geophys. Res. 93, 11.022-11.036

Storch, H. von, and J. Xu, 1990: *Principal Oscillation Pattern Analysis of the tropical 30- to 60-day oscillation. Part I: Definition on an index and its prediction.* - Climate Dyn. 4, 175-190

Storch, H. von; U. Weese and J. Xu, 1990: *Simultaneous analysis of space-time variability: Principal Oscillation Patterns and Principal Interaction Patterns with applications to the Southern Oscillation.* - Z. Meteor. 40, 99-103

Storch, H. von, and D. Baumhefner, 1991: *Principal Oscillation Pattern Analysis of the tropical 30- to 60-days oscillation. Part II: The prediction of equatorial velocity potential and its skill.* - Climate Dynamics 5, 1-12

Xu, J., 1991: *On the relationship between the stratospheric QBO and the tropospheric SO.* - J. Atmos. Sci., in press

Xu, J., 1990: *Analysis and prediction of the El Nino Southern Oscillation phenomenon using Principal Oscillation Pattern Analysis.* Max Planck Institut für Meteorologie Examensarbeiten 4 (Max-Planck-Institut für Meteorologie; Bundesstrasse 55; 2000 Hamburg 13, Germany)

Xu, J. and H. von Storch, 1990: *"Principal Oscillation Patterns"-prediction of the state of ENSO.* - J. Climate 3, 1316-1329