ENVS422: Data Analysis of Environmental Records Week 3

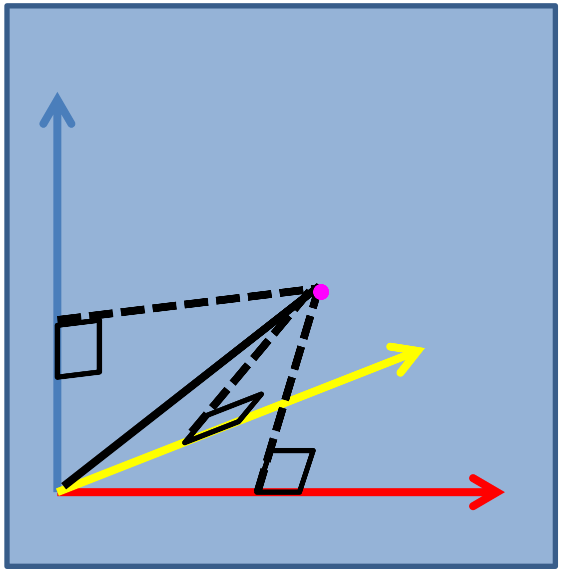
Empirical Orthogonal Functions (Principal Components)

Recap some important points

- A time series with n points is an n-dimensional vector.
- Two time series (two vectors) define a plane, and a combination of those two vectors allows you to reach all the points in that plane (all the time series which are linear combinations of the two you have, unless the two vectors happen to be parallel.
- A third time series (not in the same plane as the first 2) gives access to a 3-dimensional space.
- … n independent time series allow you to reach any point in n-dimensional space (i.e. to reproduce any time series with n points).
- If the n time series are uncorrelated, the associated vectors are all mutually orthogonal – like the axes of an n-dimensional graph.
- If the sums of the squares of the n time series are each one, they are said to be normalized. Normalized and orthogonal (orthonormal) vectors are like the unit vectors along the n different Cartesian axes. [An alternative normalization can be to give them all unit variance, i.e. standard deviation squared].
- A set of n orthonormal vectors provides a particularly simple set of basis vectors which can be used to describe any n-point time series as a sum of basis vectors times lengths.

An example with a particularly simple set of orthonormal basis functions

Time series is y = (1, 4, 3, 0, 1, 7, 2, 1, 5, 9)



Basis vector 1:	(1, 0, 0, 0, 0, 0, 0, 0, 0, 0)	× 1
Basis vector 2:	(0, 1, 0, 0, 0, 0, 0, 0, 0, 0)	× 4
Basis vector 3:	(0, 0, 1, 0, 0, 0, 0, 0, 0, 0)	× 3
Basis vector 4:	(0, 0, 0, 1, 0, 0, 0, 0, 0, 0)	× 0
Basis vector 5:	(0, 0, 0, 0, 1, 0, 0, 0, 0, 0)	× 1
Basis vector 6:	(0, 0, 0, 0, 0, 1, 0, 0, 0, 0)	× 7
Basis vector 7:	(0, 0, 0, 0, 0, 0, 1, 0, 0, 0)	× 2
Basis vector 8:	(0, 0, 0, 0, 0, 0, 0, 1, 0, 0)	× 1
Basis vector 9:	(0, 0, 0, 0, 0, 0, 0, 0, 1, 0)	× 5
Basis vector 10:	(0, 0, 0, 0, 0, 0, 0, 0, 0, 1)	× 9

Basis vectors are normalized (sum of squares of elements is 1)
Basis vectors are orthogonal (dot product of two different ones is zero)
Coefficient is given by dot product of basis function with time series (i.e. by projecting the time series onto the axes defined by the basis vectors).

There are an infinite number of such sets of orthonormal basis vectors

For example, take the first 2 vectors:

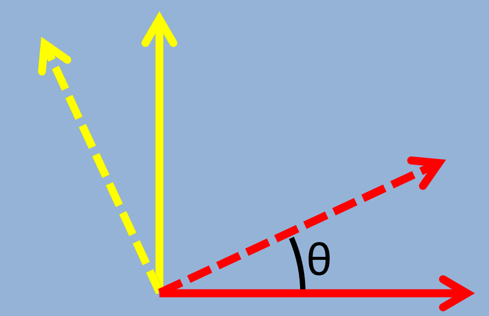
Basis vector 1: (1, 0, 0, 0, 0, 0, 0, 0, 0, 0)
Basis vector 2: (0, 1, 0, 0, 0, 0, 0, 0, 0, 0)

You can make a combination of these two which represents a rotation of the two vectors in their common plane. That plane is perpendicular to all the other vectors, so they remain orthogonal both to each other and to all the others:

Basis vector 1a: $(+\cos\theta, +\sin\theta, 0, 0, 0, 0, 0, 0, 0, 0)$
Basis vector 2a: $(-\sin\theta, +\cos\theta, 0, 0, 0, 0, 0, 0, 0, 0)$

Can repeat this as many times as you like, with different angles and different pairs of vectors, until basis vectors are completely muddled up, but remain an orthonormal set.

Looking straight down on the plane of the axes



So, there are an infinite number of possible sets of orthonormal basis vectors. How do we choose which one to use?

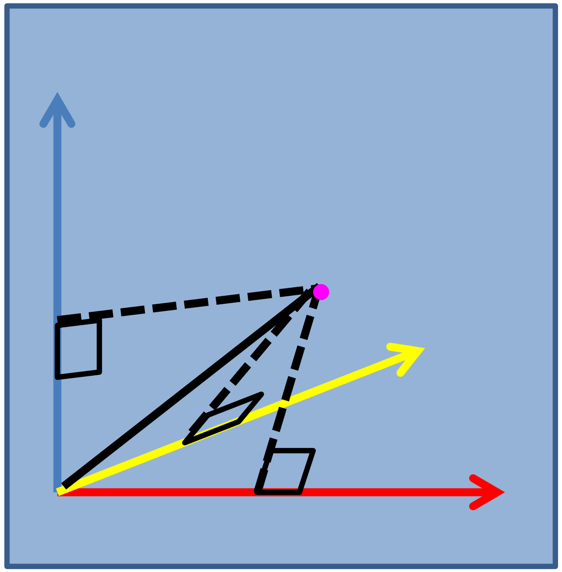
If you are interested in separating things out into different frequencies (for an equally-spaced time series), or wavelengths (when your vector is a list of values at equally spaced distances), the obvious set of basis functions is the mean, plus a series of sines and cosines. A mutually orthogonal set of these can be defined. That's Fourier analysis (which we start next week).

An alternative approach is to use the dataset itself to define the basis vectors. That is EOF analysis.

To understand how it works, it is worth introducing the concept of "explained variance"

Time series is y = (1, 4, 3, 0, 1, 7, 2, 1, 5, 9)

Basis vector 1:	(1, 0, 0, 0, 0, 0, 0, 0, 0, 0)	× 1
Basis vector 2:	(0, 1, 0, 0, 0, 0, 0, 0, 0, 0)	× 4
Basis vector 3:	(0, 0, 1, 0, 0, 0, 0, 0, 0, 0)	× 3
Basis vector 4:	(0, 0, 0, 1, 0, 0, 0, 0, 0, 0)	× 0
Basis vector 5:	(0, 0, 0, 0, 1, 0, 0, 0, 0, 0)	× 1
Basis vector 6:	(0, 0, 0, 0, 0, 1, 0, 0, 0, 0)	× 7
Basis vector 7:	(0, 0, 0, 0, 0, 0, 1, 0, 0, 0)	× 2
Basis vector 8:	(0, 0, 0, 0, 0, 0, 0, 1, 0, 0)	× 1
Basis vector 9:	(0, 0, 0, 0, 0, 0, 0, 0, 1, 0)	× 5
Basis vector 10:	(0, 0, 0, 0, 0, 0, 0, 0, 0, 1)	× 9



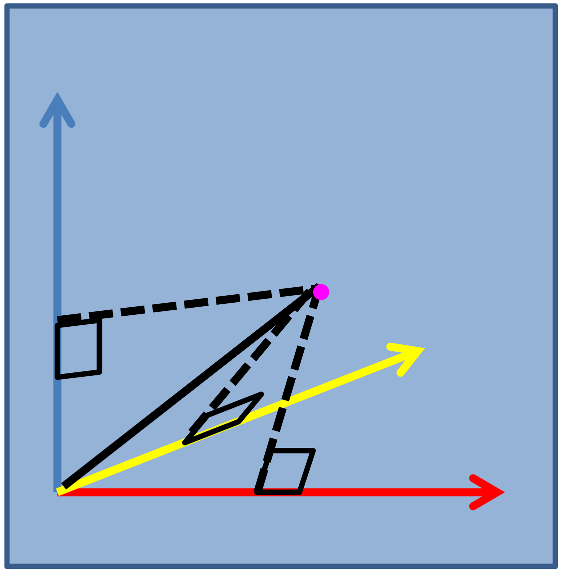
If we call the "variance" of the time series y, its mean square value, i.e. $\mathrm{Var}(y) = \frac{\sum_n {y_n}^2}{n}$

then it is 1/n time the square of the length of the vector y, since by Pythagoras' theorem, that is the sum of the squares of the (lengths of the) individual components.

The components are Basis vector 1 x coefficient 1, etc., and their lengths are simply the coefficients.

So the variance of y is also the mean of the coefficients of the basis vectors

Time series is y = (1, 4, 3, 0, 1, 7, 2, 1, 5, 9)



Basis vector 1:	(1, 0, 0, 0, 0, 0, 0, 0, 0, 0)	× 1
Basis vector 2:	(0, 1, 0, 0, 0, 0, 0, 0, 0, 0)	× 4
Basis vector 3:	(0, 0, 1, 0, 0, 0, 0, 0, 0, 0)	× 3
Basis vector 4:	(0, 0, 0, 1, 0, 0, 0, 0, 0, 0)	× 0
Basis vector 5:	(0, 0, 0, 0, 1, 0, 0, 0, 0, 0)	× 1
Basis vector 6:	(0, 0, 0, 0, 0, 1, 0, 0, 0, 0)	× 7
Basis vector 7:	(0, 0, 0, 0, 0, 0, 1, 0, 0, 0)	× 2
Basis vector 8:	(0, 0, 0, 0, 0, 0, 0, 1, 0, 0)	× 1
Basis vector 9:	(0, 0, 0, 0, 0, 0, 0, 0, 1, 0)	× 5
Basis vector 10:	(0, 0, 0, 0, 0, 0, 0, 0, 0, 1)	× 9

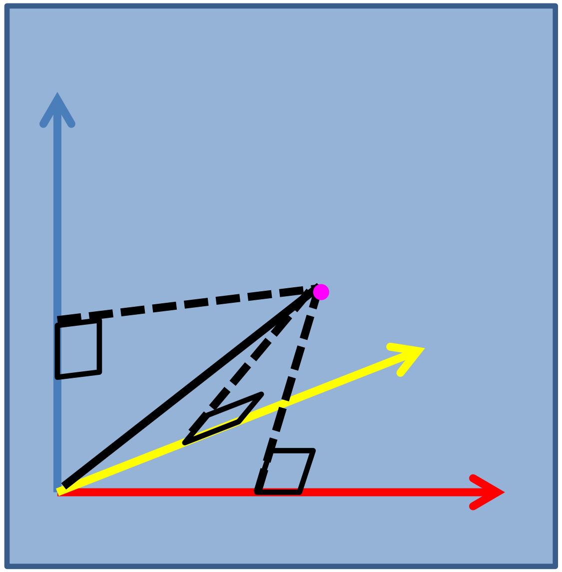
y – 1 x vector 1 = (**0**, 4, 3, 0, 1, 7, 2, 1, 5, 9)
y – 4 x vector 1 = (1, **0**, 3, 0, 1, 7, 2, 1, 5, 9)
y – 3 x vector 1 = (1, 4, **0**, 0, 1, 7, 2, 1, 5, 9) … etc.

$$\mathrm{Var}(y) = \frac{\sum_n {y_n}^2}{n}$$

So removing the contribution of one basis vector to y, removes an amount of variance from y given by (1/n) x the square of its coefficient.

This is obvious for the basis functions here, where the coefficients are simply the components of y, but also works for more general basis functions where that is not the case

Time series is y = (1, 4, 3, 0, 1, 7, 2, 1, 5, 9)



y – 1 x vector 1 = (**0**, 4, 3, 0, 1, 7, 2, 1, 5, 9)
y – 4 x vector 1 = (1, **0**, 3, 0, 1, 7, 2, 1, 5, 9)
y – 3 x vector 1 = (1, 4, **0**, 0, 1, 7, 2, 1, 5, 9)

So removing the contribution of one basis vector to y, removes an amount of variance from y given by (1/n) x the square of its coefficient.

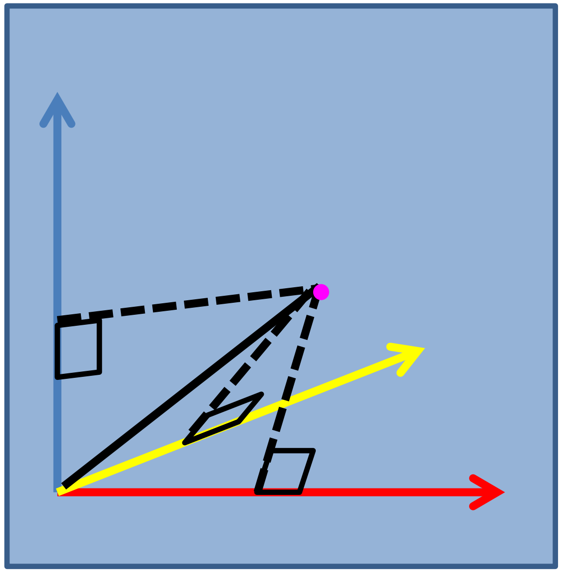
Because of this, we can say that each basis vector "explains" a certain amount of the variance in y. Or if we divide by the total variance we can say it explains a certain fraction or percentage of the variance.

This separation into variance explained by the different vectors only works if they are orthogonal. It's impossible to make a meaningful separation into variances explained by a set of non-orthogonal vectors.

One point to beware of:

I defined the variance as the mean square value:

$$\mathrm{Var}(y) = \frac{\sum_n {y_n}^2}{n}$$



Everything I said is true with this definition, but the true definition of the variance is the mean square *deviation*, i.e.

$$\mathrm{Var}(y) = \frac{\sum_n {y'_n}^2}{n}$$

where $$y'_n = y_n - \bar{y}$$

so that $\mathrm{Var}(y) = [\sigma(y)]^2$ is the square of the standard deviation

To keep the mean out of things, it is often best to subtract the mean from all your time series, and to use basis functions which have zero mean.

$$\sigma(y) = \sqrt{\frac{\sum_n (y_n - \bar{y})^2}{n}}$$

Equivalently, ensure that one of your basis vectors is constant in time, so it accounts for the mean – if the others are orthogonal to this, they must all have zero mean.

EOF analysis means choosing a set of basis vectors which account for the largest possible fraction of the variance in your time series

Time series is y = (1, 4, 3, 0, 1, 7, 2, 1, 5, 9)
Basis vector 1 is (1, 4, 3, 0, 1, 7, 2, 1, 5, 9)

Basis vector 1 explains all the variance in y. No need for any others. Job done.

But what if you have 2 time series

p = (1, 4, 3, 0, 1, 7, 2, 1, 5, 9)
q = (1, 3, 2, 9, 2, 5, 5, 1, 9, 2)

Then you could look for the basis vector which explains the largest possible fraction of the sum of the variances from both vectors, and a second vector orthogonal to the first, which explains the remaining variance.

This is an EOF analysis, and the two basis vectors are known as principal components.

Time for a concrete example

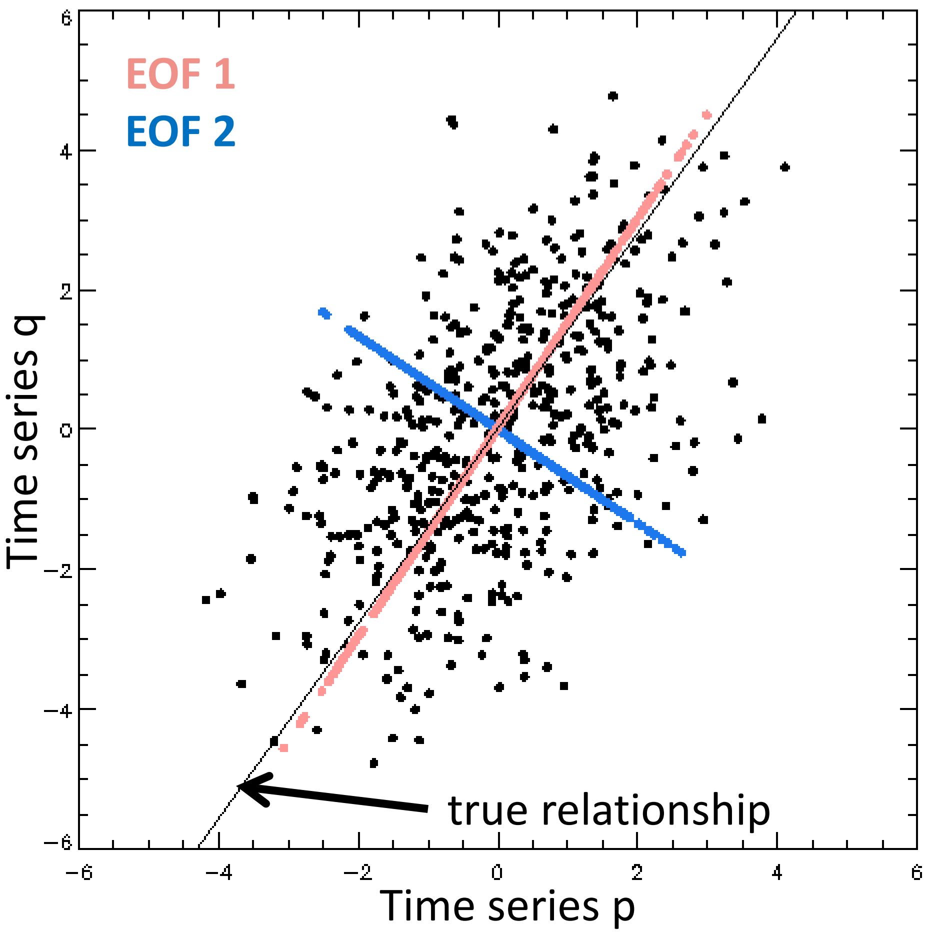
We have 2 time series: p and q

I created them so that
p = a(t) + noise
q = 1.4 a(t) + similar noise

Instead of plotting them as a function of time, plot q against p.

This gives us the scatterplot (black dots). The black line is the true underlying relationship q=1.4p.

The EOF analysis produces two pairs of time series **(p1,q1)** and **(p2,q2)**, in each of which the time series for p is proportional to the time series for q.

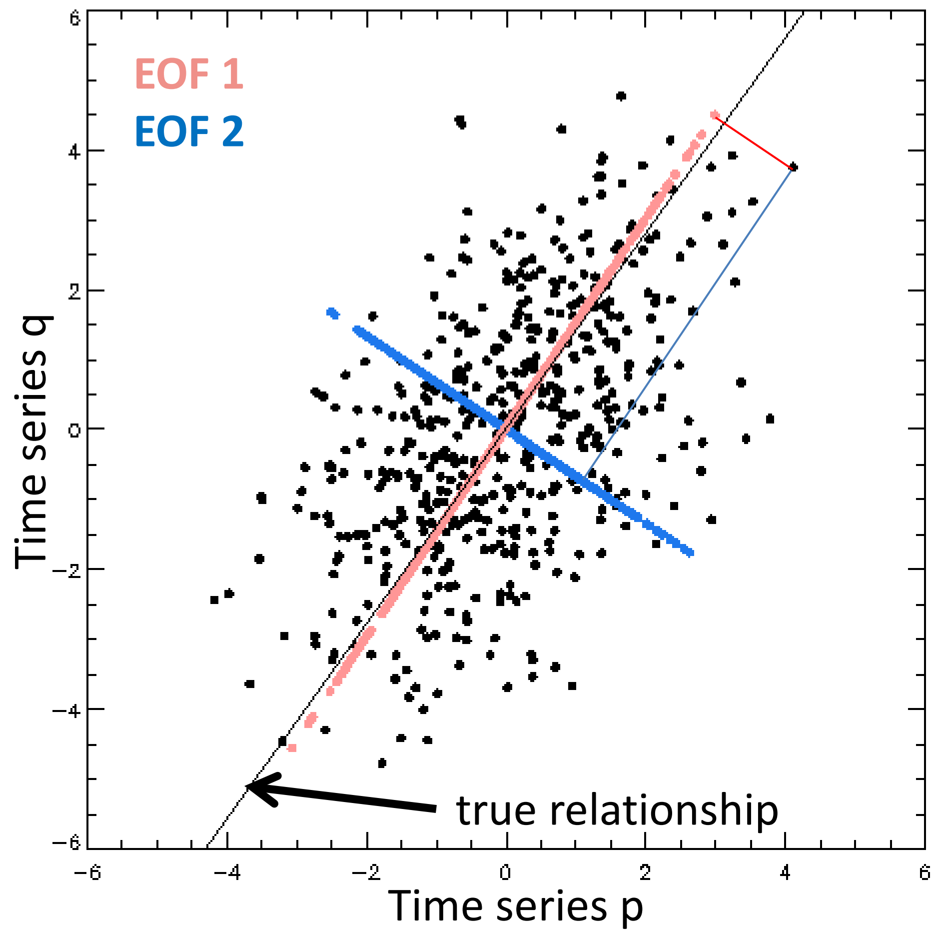


The relationship is chosen so that the sum of the squares of perpendicular distances from the pink line is minimized.

Each point (p_n, q_n) is at coordinates $(p1_n + p2_n, q1_n + q2_n)$

Instead of thinking of p and q as the coordinates, we think of orientation of the pink and blue scatter plots as the axes.

Calculating the EOFs has identified the best relationship between p and q (first EOF), and the direction of the "noise" in that relationship (2nd EOF).



If we had 3 time series, the scatterplot would be 3D. The first EOF would be a line along the longest axis of the cloud of points, defining the strongest relationship among all 3 time series. Then imagine projecting all the points onto the plane perpendicular to the first EOF. There would now be a 2D scatterplot, and the 2nd EOF would be along the longest axis of this one, and so on.

If we want a relationship between two time series, why not do a least squares fit of one on the other?

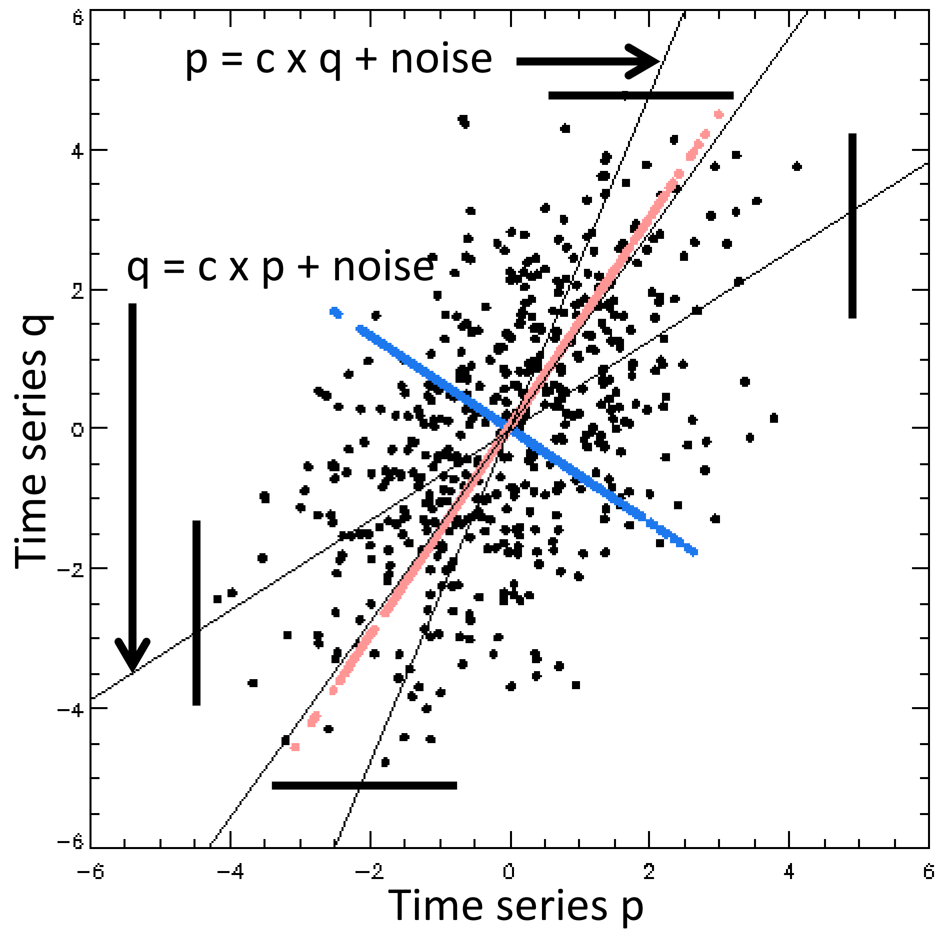
q = c x p + noise

assumes that the p value is exactly known, and minimizes the squared distances along the q axis (as if q was the measurement at the precisely known time p)

p = c x q + noise

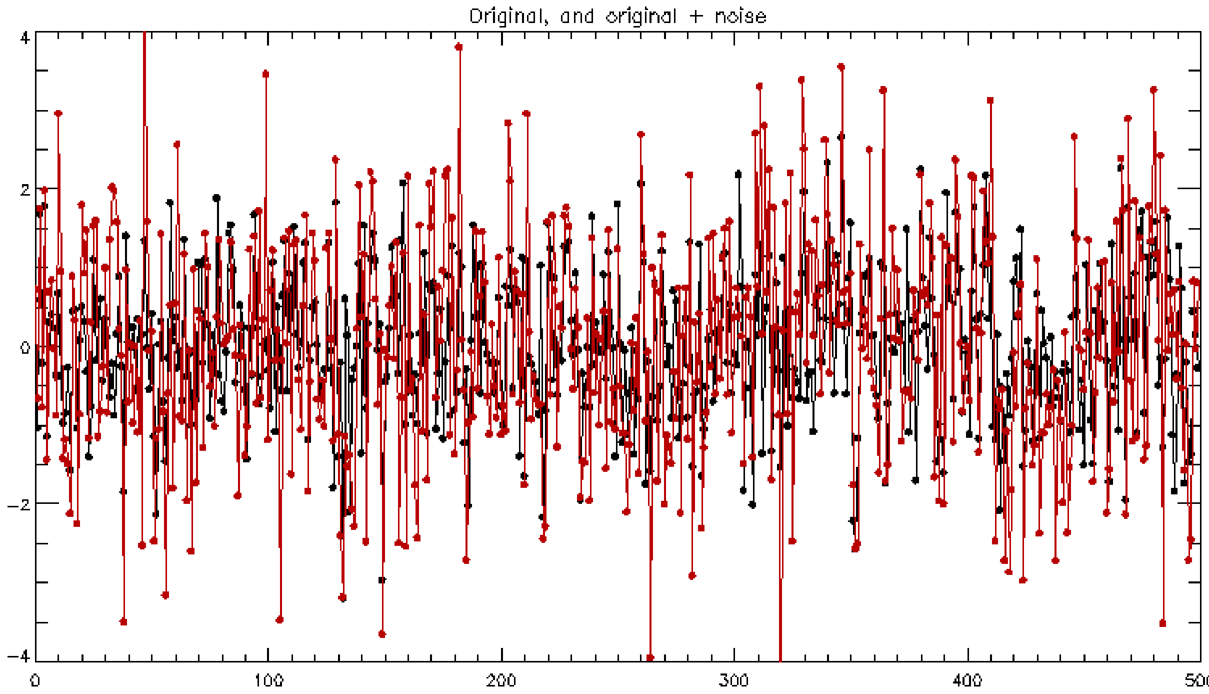
assumes that the q value is exactly known, and minimizes the squared distances along the p axis

The EOF analysis treats both time series equally – assuming “noise” appears at the same amplitude in both.



a is the underlying time series
p is a + random noise
q is const x a + random noise (same amplitude)

correlation of a with p: 0.643
correlation of a with q: 0.805
correlation of a with EOF 1: 0.847
correlation of a with EOF 2: 0.025



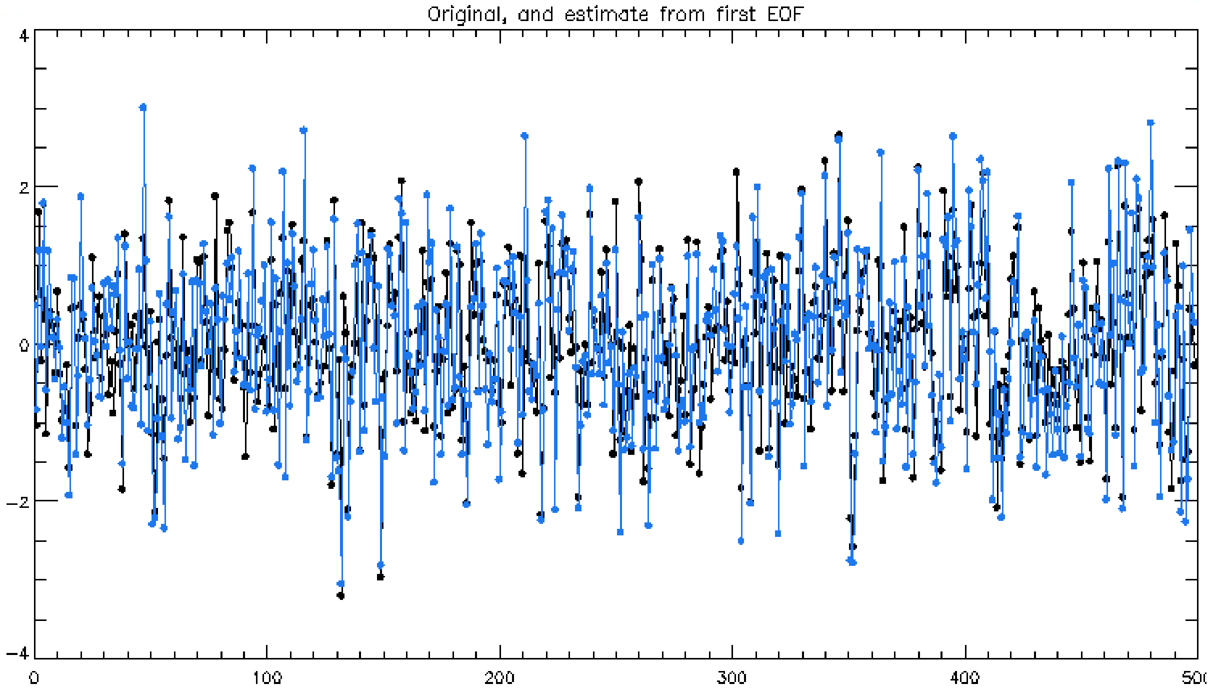
original time series a

p=a + noise

Because the "signal" is present in both p and q (with different amplitudes), but they each have different sets of noise superimposed, we can extract the signal more cleanly by using information from both p and q.

At the same time, we obtain an estimate of how much to multiply that underlying signal by for each of p and q.

i.e. if p and q are time series from different places, that means we obtain a spatial distribution to match the time series



original time series a

Estimate of original based on EOF 1

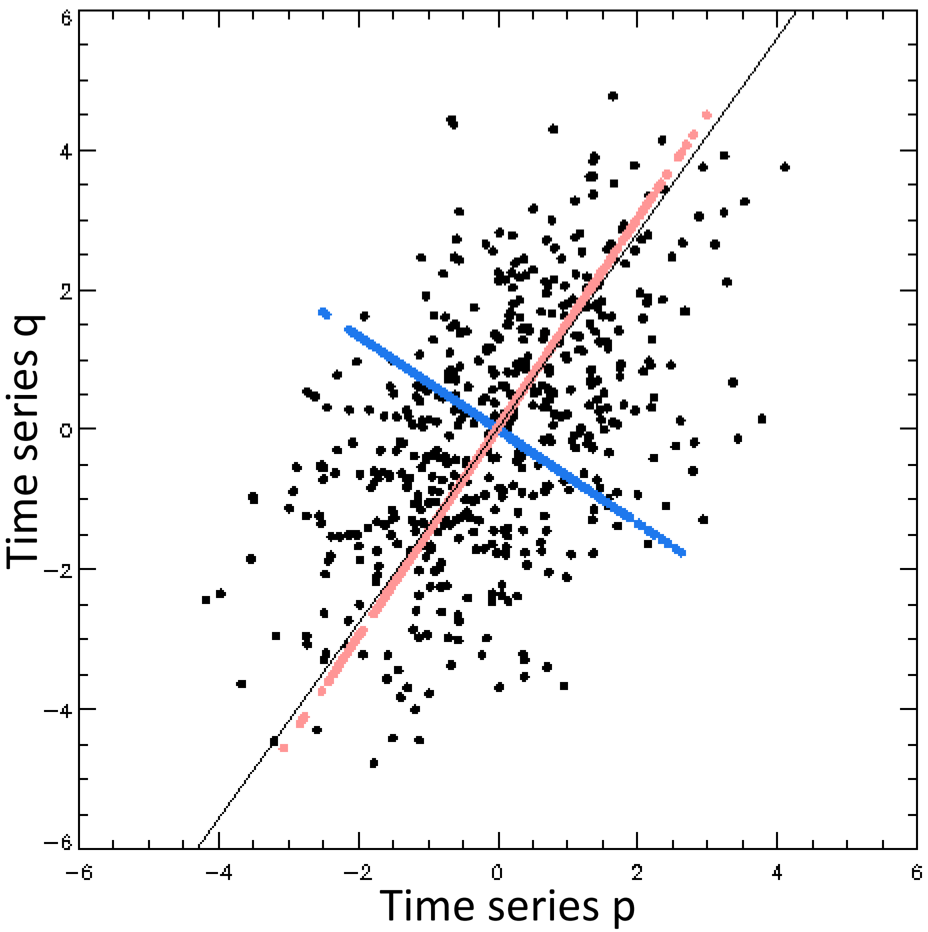
So how do we calculate these EOFs?

We want to find the (pink) direction which explains most of the variance.

This means rotating to a different set of axes, P, Q in a way which maximizes the variance along the P axis.

$$\mathrm{P} = +p\cos\theta + q\sin\theta$$

$$Q = -p\sin\theta + q\cos\theta$$



variance along P is mean of

$$P^2 = (p\cos\theta + q\sin\theta)^2 = \frac{p^2+q^2}{2} + \frac{p^2-q^2}{2}\cos 2\theta + pq\sin 2\theta$$

variance (times number of elements n) is sum:

$$\sum P^2 = \sum(p\cos\theta + q\sin\theta)^2 = \frac{\sum p^2 + \sum q^2}{2} + \frac{\sum p^2 - \sum q^2}{2}\cos 2\theta + \sum pq\sin 2\theta$$

To find the angle at which this is maximum, differentiate w.r.t. θ, and set equal to zero:

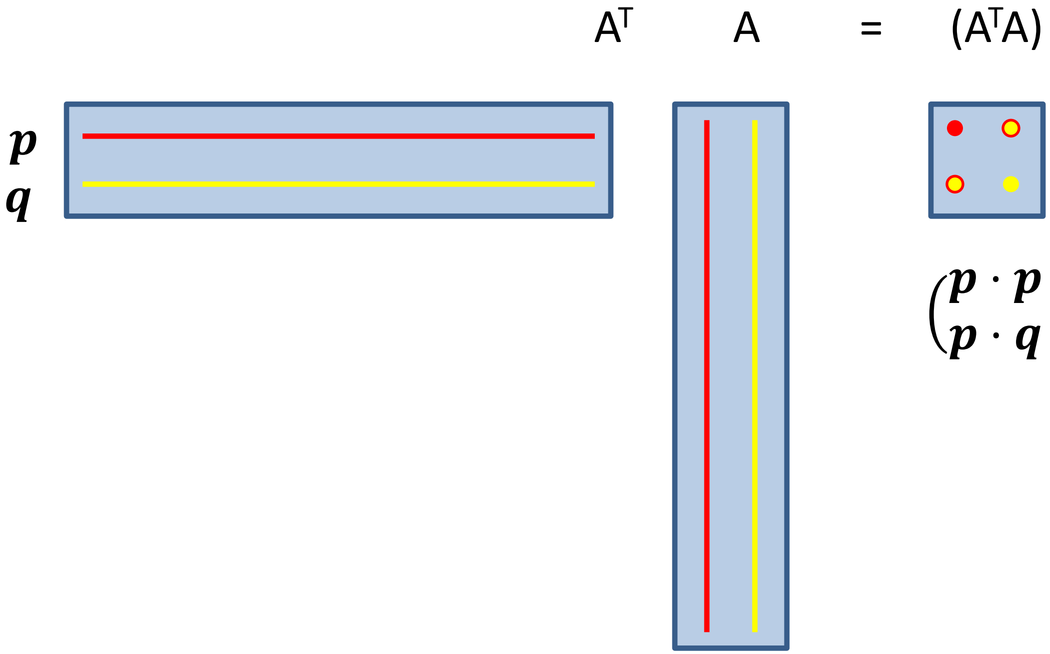
$$-2\frac{\sum p^2 - \sum q^2}{2}\sin 2\theta + 2\sum pq\cos 2\theta = 0$$

$$(\boldsymbol{p}\cdot\boldsymbol{p} - \boldsymbol{q}\cdot\boldsymbol{q})\sin 2\theta = 2\boldsymbol{p}\cdot\boldsymbol{q}\cos 2\theta$$

$$\tan 2\theta = \frac{2\sum pq}{\sum p^2 - \sum q^2}$$

$$\tan 2\theta = \frac{2\boldsymbol{p}\cdot\boldsymbol{q}}{(\boldsymbol{p}\cdot\boldsymbol{p} - \boldsymbol{q}\cdot\boldsymbol{q})}$$

Which solves the problem, but it’s interesting to see how this relates to (n times) the covariance matrix between the vectors $\boldsymbol{p}$ and $\boldsymbol{q}$…



$$\begin{pmatrix} \boldsymbol{p}\cdot\boldsymbol{p} & \boldsymbol{p}\cdot\boldsymbol{q} \\ \boldsymbol{p}\cdot\boldsymbol{q} & \boldsymbol{q}\cdot\boldsymbol{q} \end{pmatrix}$$

Now, if we write the covariance matrix for P and Q, after a bit of algebra we find that it is

$$\frac{\boldsymbol{1}}{\boldsymbol{n}}\begin{pmatrix} \frac{\boldsymbol{p}\cdot\boldsymbol{p}+\boldsymbol{q}\cdot\boldsymbol{q}}{2}+\frac{\boldsymbol{p}\cdot\boldsymbol{p}-\boldsymbol{q}\cdot\boldsymbol{q}}{2}\cos 2\theta+\boldsymbol{p}\cdot\boldsymbol{q}\sin 2\theta & \boldsymbol{p}\cdot\boldsymbol{q}\cos 2\theta-\frac{\boldsymbol{p}\cdot\boldsymbol{p}-\boldsymbol{q}\cdot\boldsymbol{q}}{2}\sin 2\theta \\ \boldsymbol{p}\cdot\boldsymbol{q}\cos 2\theta-\frac{\boldsymbol{p}\cdot\boldsymbol{p}-\boldsymbol{q}\cdot\boldsymbol{q}}{2}\sin 2\theta & \frac{\boldsymbol{p}\cdot\boldsymbol{p}+\boldsymbol{q}\cdot\boldsymbol{q}}{2}+\frac{\boldsymbol{p}\cdot\boldsymbol{p}-\boldsymbol{q}\cdot\boldsymbol{q}}{2}\cos 2\theta+\boldsymbol{p}\cdot\boldsymbol{q}\sin 2\theta \end{pmatrix}$$

$$\begin{pmatrix} \frac{\boldsymbol{p}\cdot\boldsymbol{p}+\boldsymbol{q}\cdot\boldsymbol{q}}{2}+\frac{\boldsymbol{p}\cdot\boldsymbol{p}-\boldsymbol{q}\cdot\boldsymbol{q}}{2}\cos 2\theta+\boldsymbol{p}\cdot\boldsymbol{q}\sin 2\theta & \boldsymbol{p}\cdot\boldsymbol{q}\cos 2\theta-\frac{\boldsymbol{p}\cdot\boldsymbol{p}-\boldsymbol{q}\cdot\boldsymbol{q}}{2}\sin 2\theta \\ \boldsymbol{p}\cdot\boldsymbol{q}\cos 2\theta-\frac{\boldsymbol{p}\cdot\boldsymbol{p}-\boldsymbol{q}\cdot\boldsymbol{q}}{2}\sin 2\theta & \frac{\boldsymbol{p}\cdot\boldsymbol{p}+\boldsymbol{q}\cdot\boldsymbol{q}}{2}+\frac{\boldsymbol{p}\cdot\boldsymbol{p}-\boldsymbol{q}\cdot\boldsymbol{q}}{2}\cos 2\theta+\boldsymbol{p}\cdot\boldsymbol{q}\sin 2\theta \end{pmatrix}$$

and if we substitute the relationship we found for θ, designed to maximise the top left element:

$$(\boldsymbol{p}\cdot\boldsymbol{p}-\boldsymbol{q}\cdot\boldsymbol{q})\sin 2\theta = 2\boldsymbol{p}\cdot\boldsymbol{q}\cos 2\theta$$

we note that it sets the off-diagonal elements to zero.

Since, by definition, the diagonal elements are $\boldsymbol{P}\cdot\boldsymbol{P}$ and $\boldsymbol{Q}\cdot\boldsymbol{Q}$, the covariance matrix simplifies to

$$\begin{pmatrix} \boldsymbol{P}\cdot\boldsymbol{P} & 0 \\ 0 & \boldsymbol{Q}\cdot\boldsymbol{Q} \end{pmatrix}$$

So the coordinate change which maximizes the variance $\boldsymbol{P}\cdot\boldsymbol{P}$ also sets the cross correlation between the time series P and Q to zero, and makes the covariance matrix diagonal.

For the more general (many dimensions) case, this is how we can think of EOF analysis: calculating the rotation which diagonalizes the covariance matrix.

OK – so far, so hideously complicated.

So how do we find the right rotation to diagonalize the covariance matrix?

This is a classic problem in matrix algebra, known as the eigenvalue-eigenvector problem. It can be solved in Matlab using the eig function (or eigs of you only want a few of them).

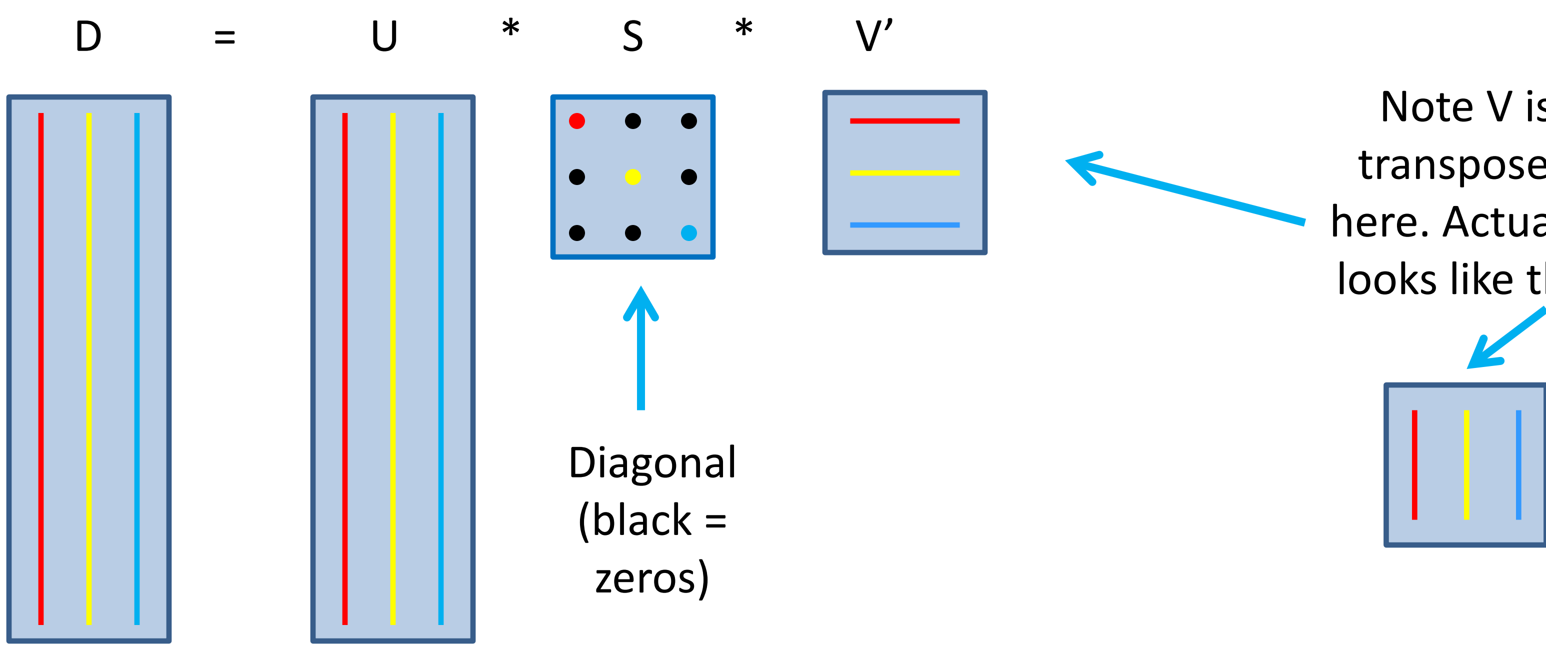
However, there's a better way to do it, which makes things very simple. It's called Singular Value Decomposition, and is an algorithm which goes straight from a set of time series to the EOFs of that time series in a very convenient way, and without the need to calculate the covariance matrix in between.

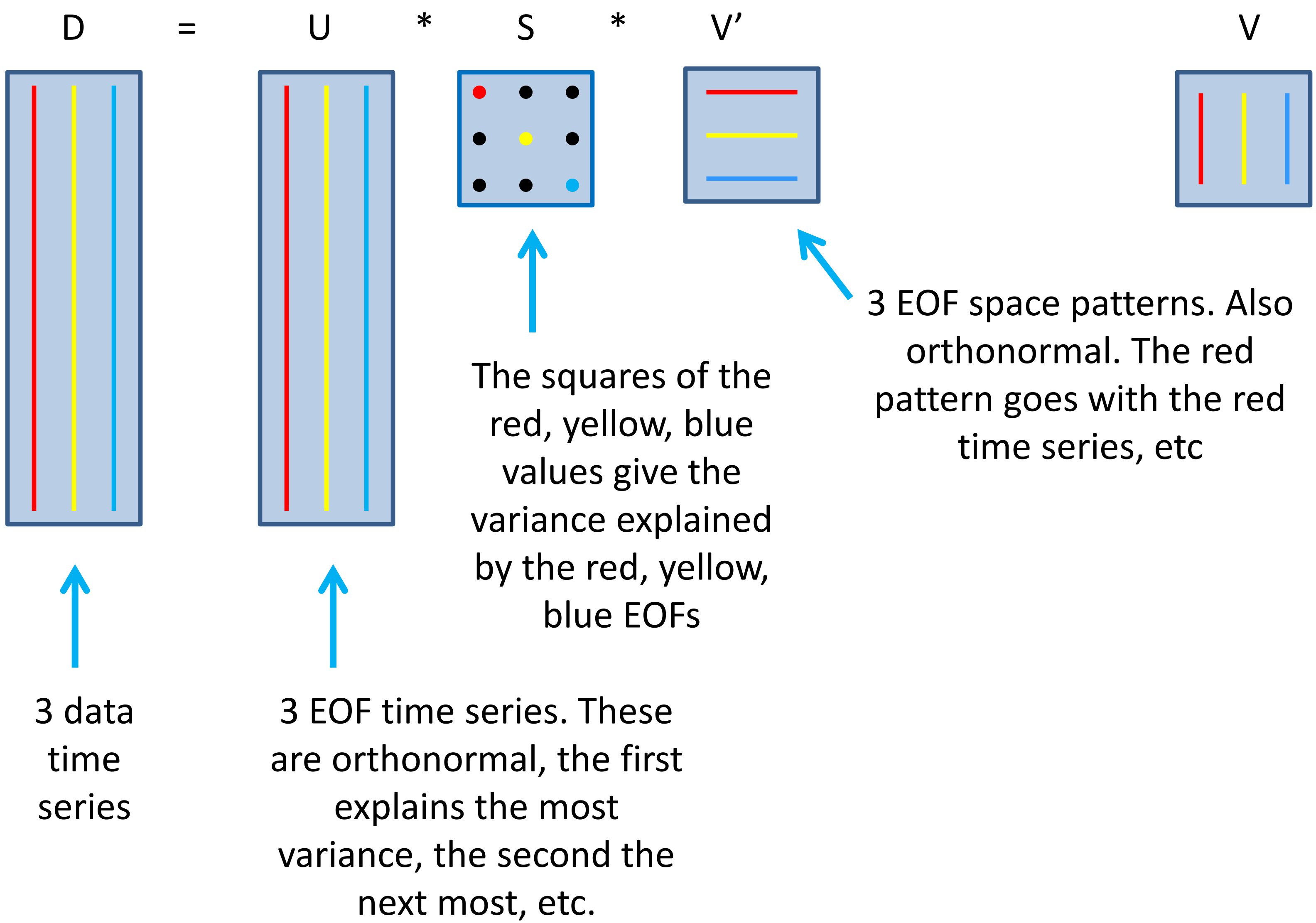
```
X;
D=detrend(X,'constant');
[U,S,V]=svd(D,0);
```

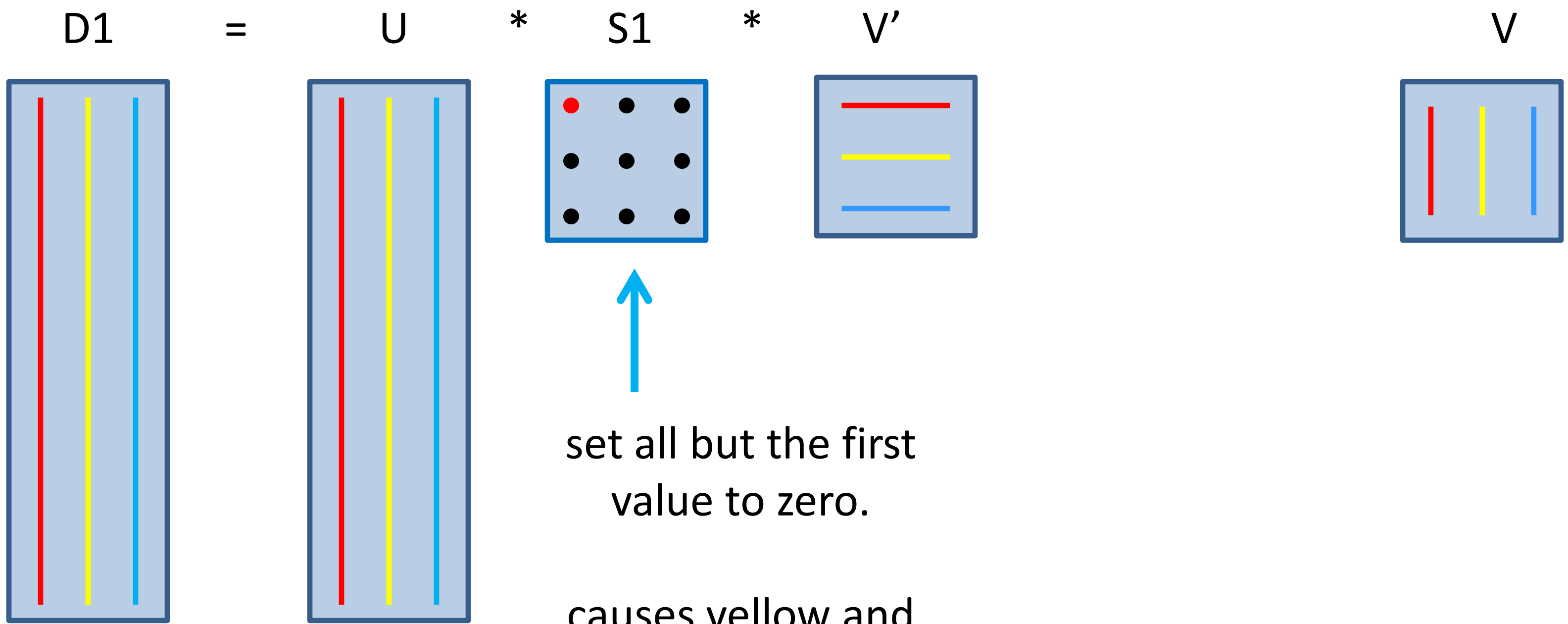
This is your 2D dataset. Let's assume it is long columns (time series with many points), and short rows (a small number or points in space, let's say 3).

This is a quick way of removing the means from each column

This calculates 3 matrices such that D=U*S*V'




D = U * S * V’
V
The squares of the red, yellow, blue values give the variance explained by the red, yellow, blue EOFs
3 EOF space patterns. Also orthonormal. The red pattern goes with the red time series, etc
3 data time series
3 EOF time series. These are orthonormal, the first explains the most variance, the second the next most, etc.



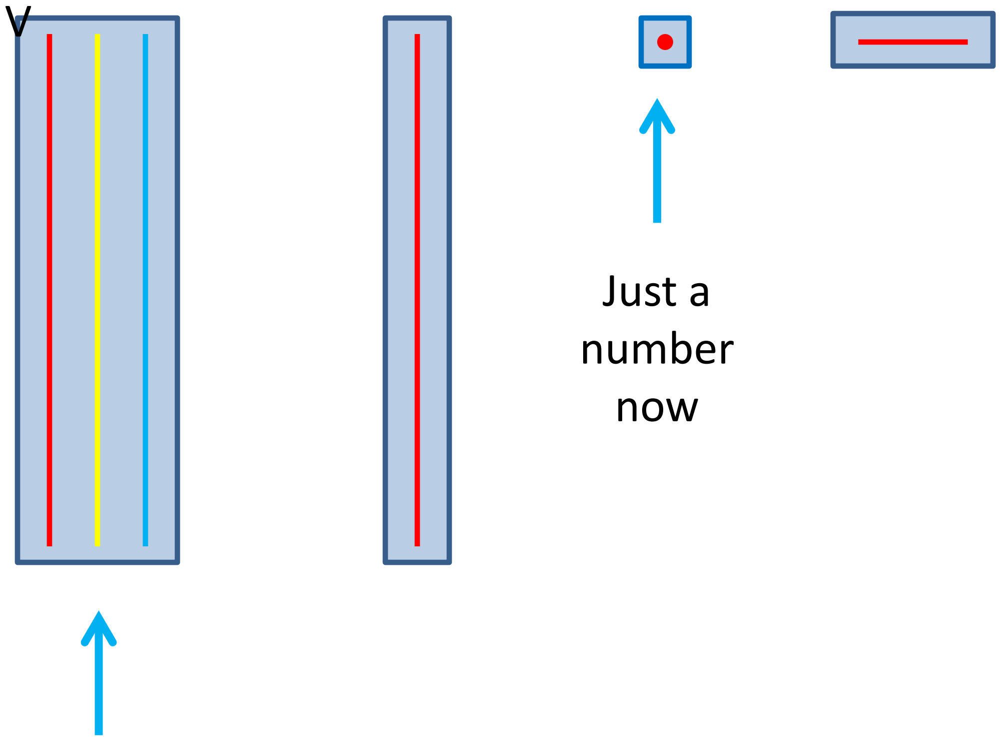
set all but the first value to zero.

causes yellow and blue vectors in U and V to be ignored

3 reconstructed data time series all proportional to the first time EOF in U.

Since the yellow and blue EOFs are ignored here, we can simplify this to….

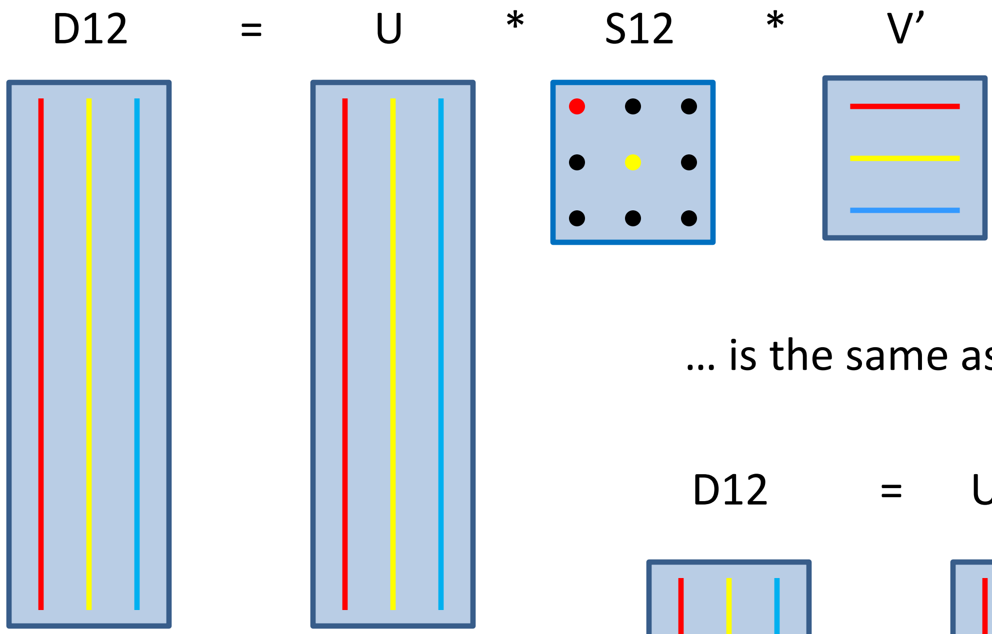
D1 = U1 * S1 * V1’



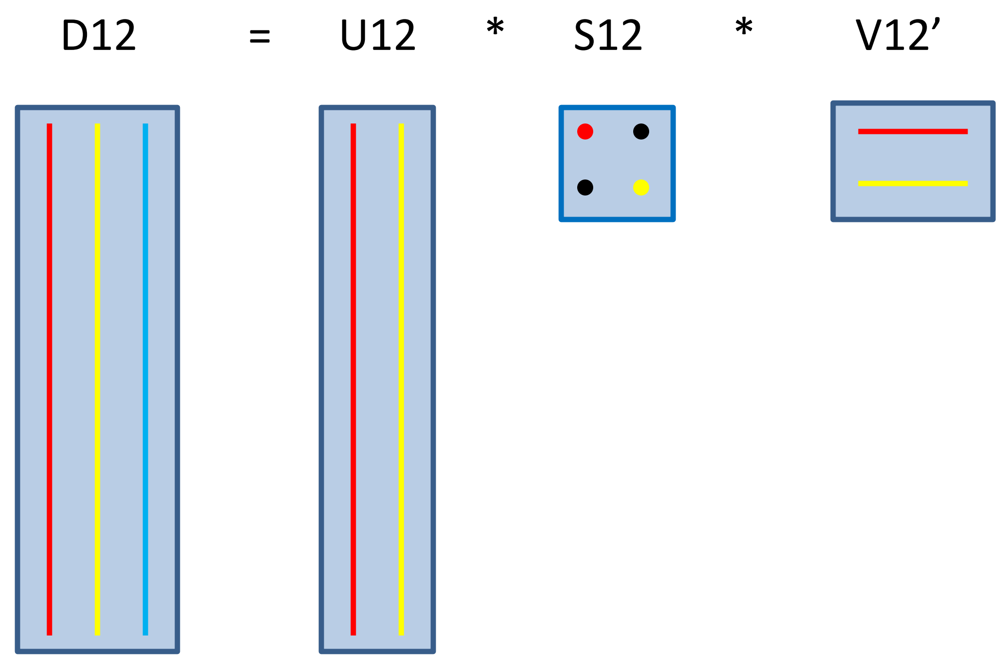
Just a number now

3 reconstructed data time series all proportional to the first time EOF in U.

Similarly, just using the first 2 EOFs …



… is the same as



Note how space and time are treated symmetrically here. Both U and V are sets of othonormal basis vectors. They have different lengths, but there are the same number of them.

Note also that this whole process doesn't care about the ordering in time or space – you can scramble all your time series (as long as you scramble them all the same way), and the covariance matrix will be the same, so the same EOFs will emerge (identical in space, and scrambled in time).

If you have more points in space than in time (e.g. a map of data with lots of grid points, and a smaller number of time steps), then simply swap space and time. The red, yellow and blue vectors each then represent a list of points from a different map (in whatever order is convenient). U then becomes the space EOF and V the time EOF. The same EOFs emerge whichever way round you do the calculation.

There's no need to worry about whether the time series are independent of each other, or whether you have reduced the degrees of freedom by subtracting means, annual cycles, etc. The SVD will simply produce some zeros along the diagonal of S if you need fewer basis vectors (EOFs) to represent the data.

The maths may look horribly complicated, but the implementation is ridiculously simple:

`X;` contains your data – time in columns, space in rows, or vice versa

```
D=detrend(X,'constant');
[U,S,V]=svd(D,0);
```

and you have your temporal EOFs in U, spatial EOFs in V, or vice versa, and the squares of the diagonal elements of S tell you how much of the total variance is explained by the corresponding space-time EOF pair.

This is a way of subtracting means all in one step. No good if your data contains NaNs – write a loop using nanmean in that case. But you'll have to fill the gaps before the next stage in any case.

```
%EOF example from altimetry
[h,hr]=readavisolat(300);
hbit=hr(:,1001:1100);
[U,S,V]=svd(hbit,0);
figure;
percVar=100.*diag(S).^2/sum(diag(S).^2);
plot(percVar, 'o');
title('Percent of variance explained by EOFs')
h1=U(:,1)*S(1,1)*V(:,1)';
h2=U(:,2)*S(2,2)*V(:,2)';
h3=U(:,3)*S(3,3)*V(:,3)';
figure
ldcol(19);
imagesc(hbit,[-0.2,0.2]); axis xy; colorbar;
title('Full dataset')
figure
ldcol(19);
imagesc(h1,[-0.2,0.2]); axis xy; colorbar;
title('1st EOF')
figure
ldcol(19);
imagesc(h2,[-0.2,0.2]); axis xy; colorbar;
title('2nd EOF')
```

reads data, removes means

extracts particular region

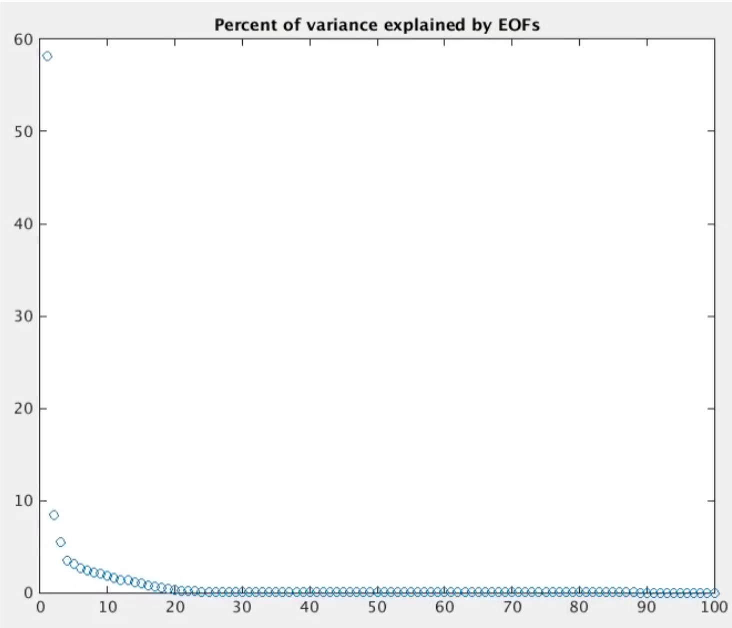
calculates EOFs

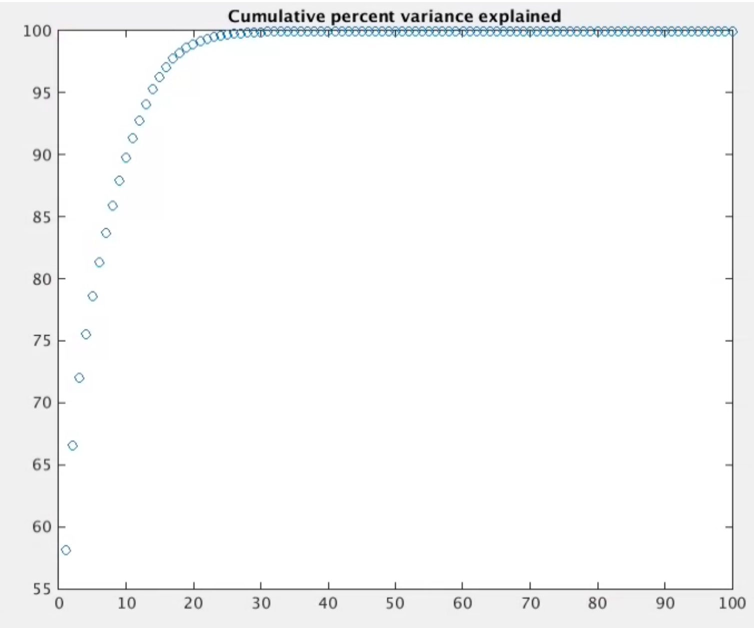
calculates % variance explained by each

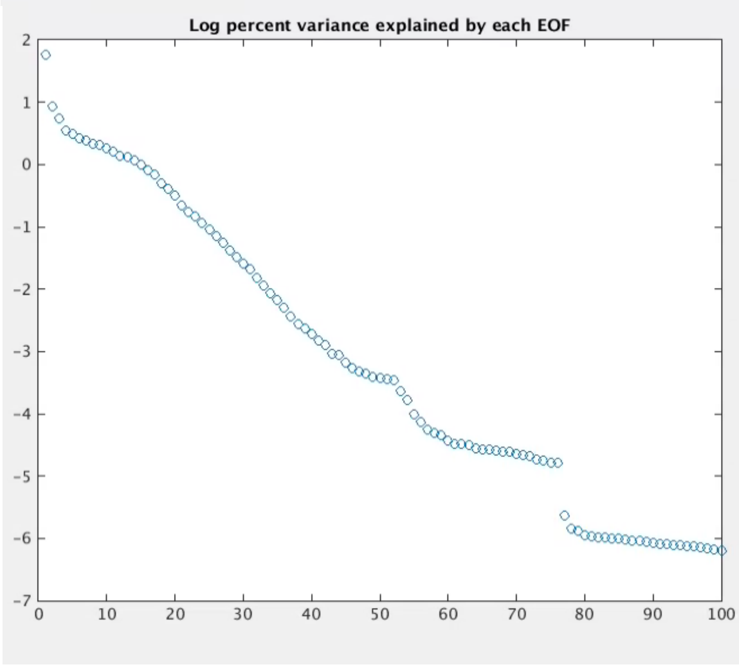
reconstructs approximation of data from EOF 1

and from EOF 2

and from EOF 3







First one stands out
2 and 3 also stand out a bit.
First 18 explain about 98% of variance (slope in log plot changes after about 18 too).

Instead of 100 time series, we only have to think about a few (1? 2? 3? 18?)

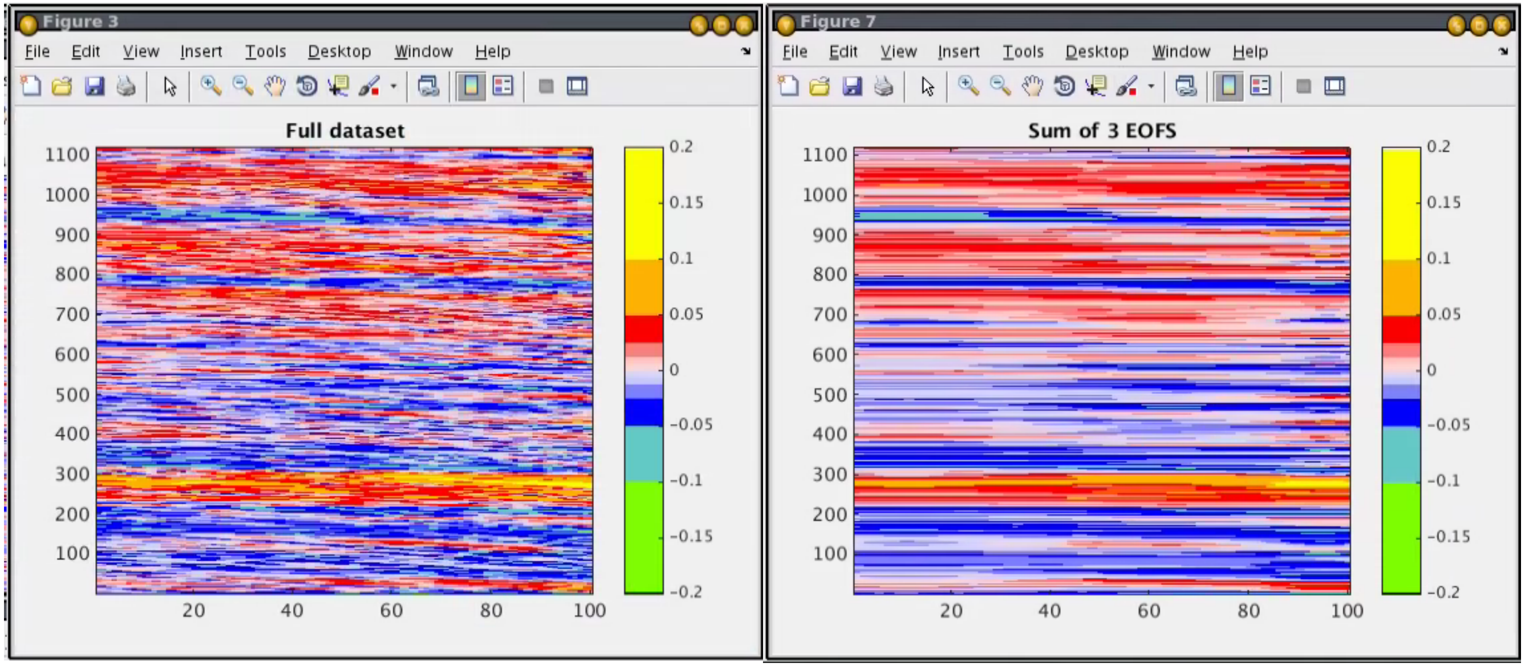


Figure 3
File Edit View Insert Tools Desktop Window Help
Full dataset
1100
1000
900
800
700
600
500
400
300
200
100
20 40 60 80 100
0.2 0.15 0.1 0.05 0 −0.05 −0.1 −0.15 −0.2
Figure 7
File Edit View Insert Tools Desktop Window Help
Sum of 3 EOFS
1100
1000
900
800
700
600
500
400
300
200
100
20 40 60 80 100
0.2 0.15 0.1 0.05 0 −0.05 −0.1 −0.15 −0.2

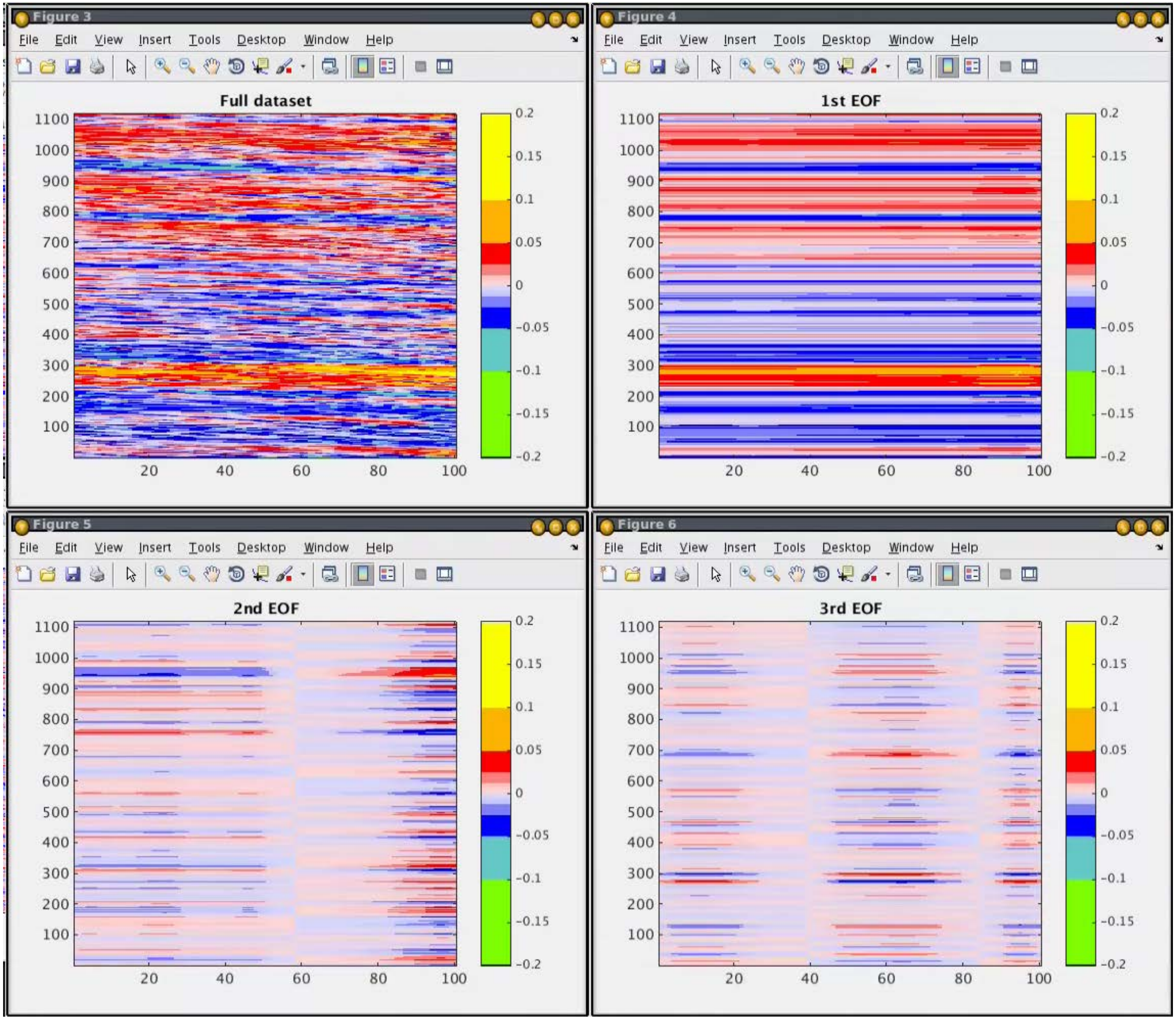


Figure 3
File Edit View Insert Tools Desktop Window Help
Full dataset
1100
1000
900
800
700
600
500
400
300
200
100
20 40 60 80 100
0.2 0.15 0.1 0.05 0 -0.05 -0.1 -0.15 -0.2
Figure 4
File Edit View Insert Tools Desktop Window Help
1st EOF
1100
1000
900
800
700
600
500
400
300
200
100
20 40 60 80 100
0.2 0.15 0.1 0.05 0 -0.05 -0.1 -0.15 -0.2
Figure 5
File Edit View Insert Tools Desktop Window Help
2nd EOF
1100
1000
900
800
700
600
500
400
300
200
100
20 40 60 80 100
0.2 0.15 0.1 0.05 0 -0.05 -0.1 -0.15 -0.2
Figure 6
File Edit View Insert Tools Desktop Window Help
3rd EOF
1100
1000
900
800
700
600
500
400
300
200
100
20 40 60 80 100
0.2 0.15 0.1 0.05 0 -0.05 -0.1 -0.15 -0.2

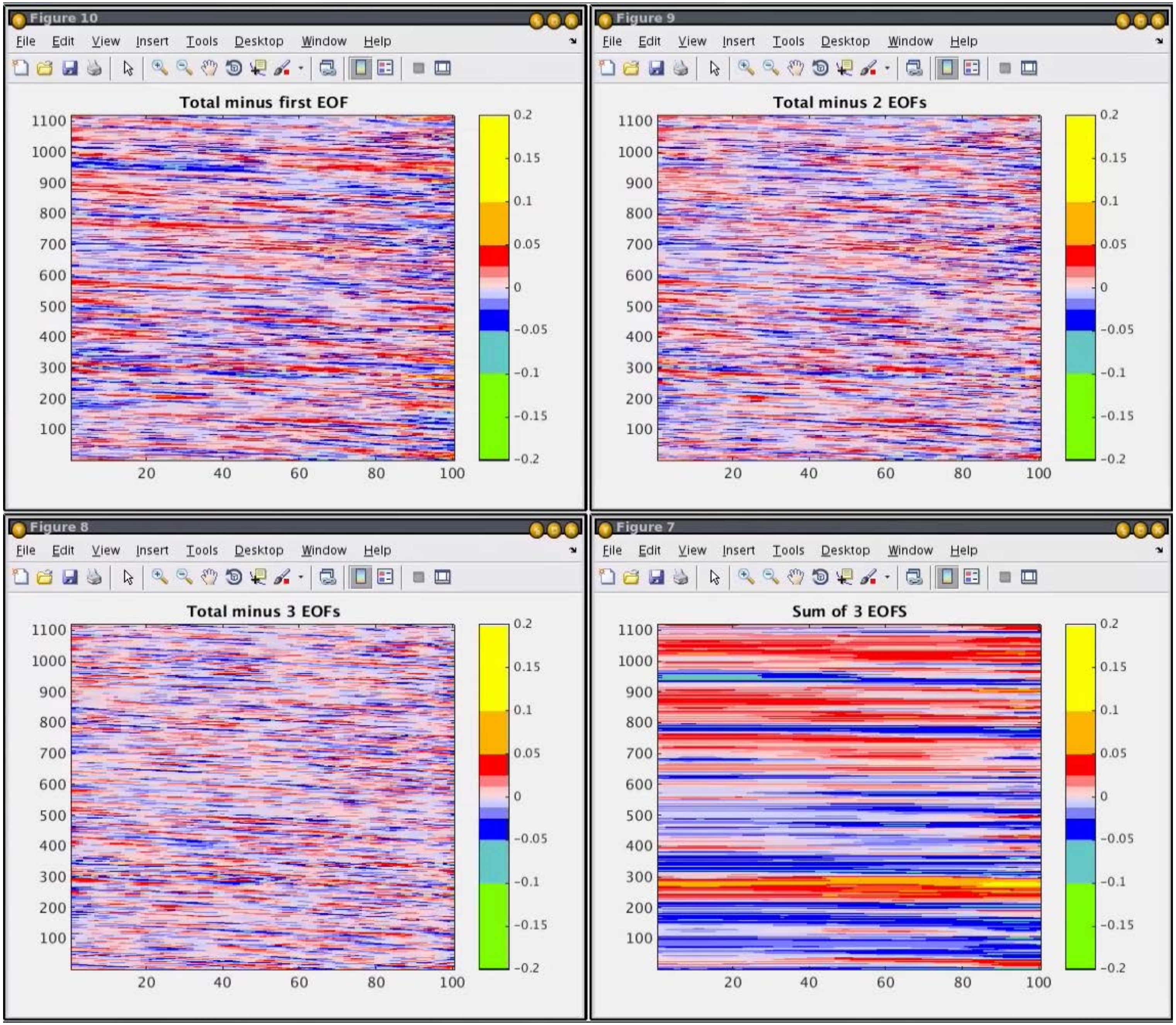
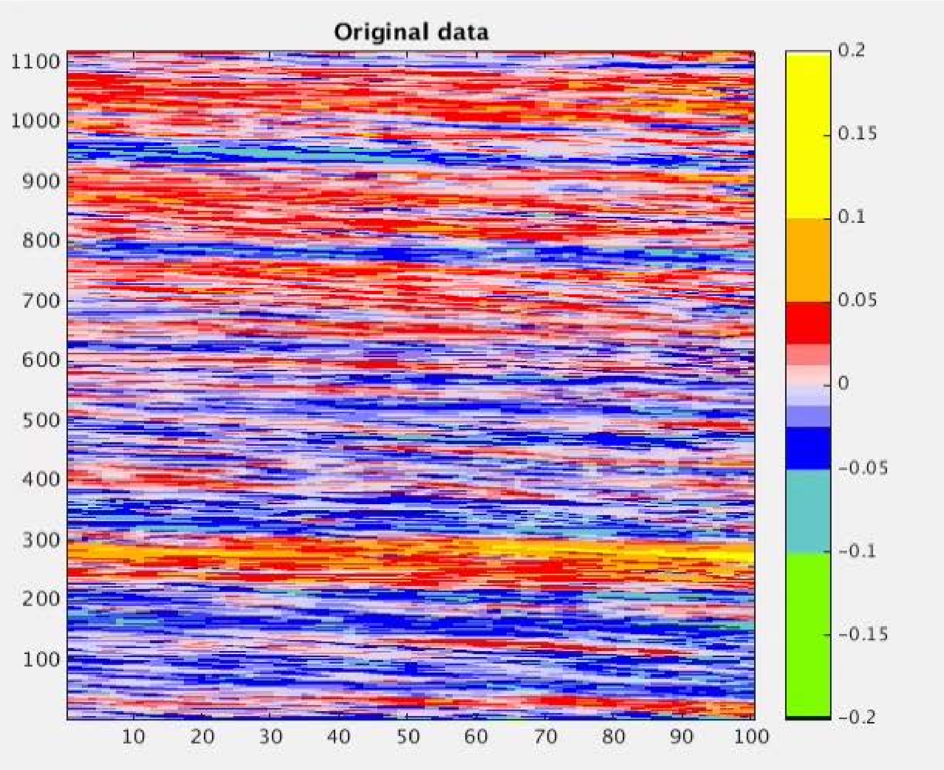
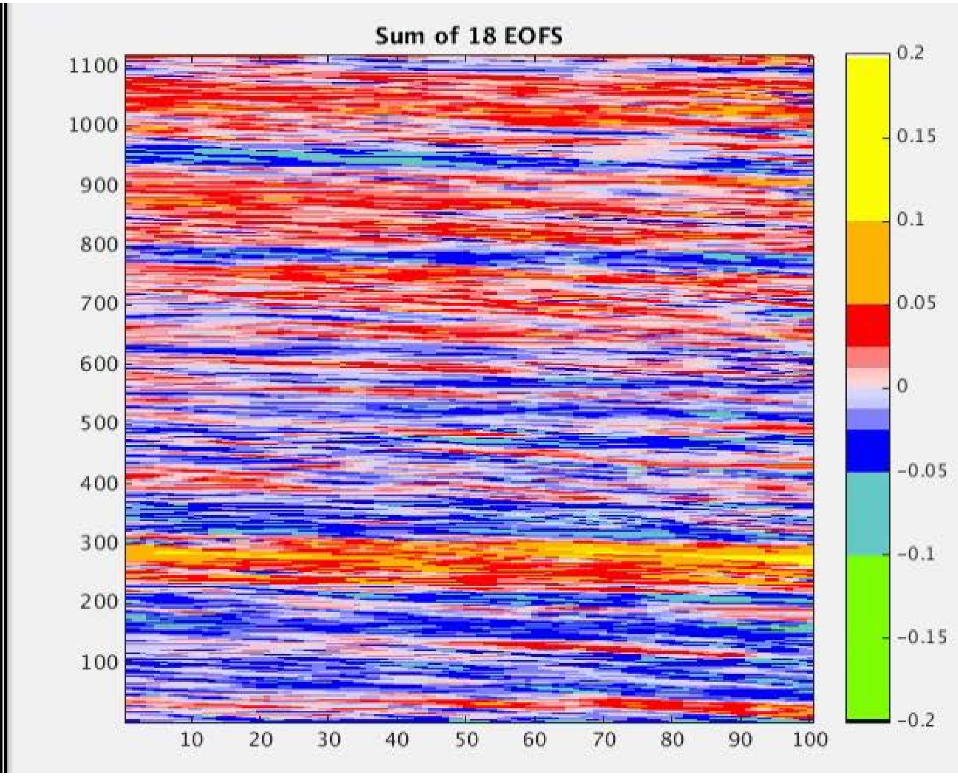


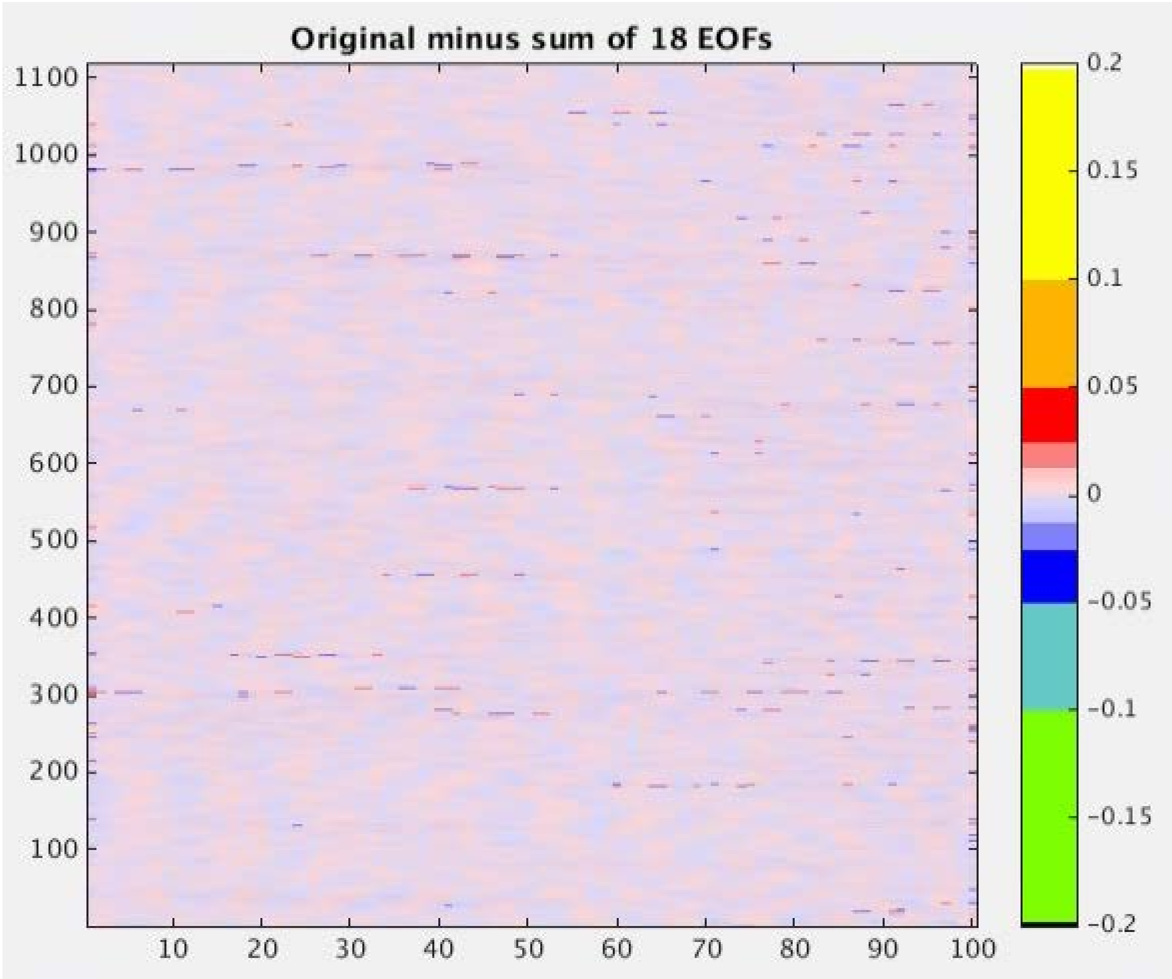
Figure 10
File Edit View Insert Tools Desktop Window Help
Total minus first EOF
Figure 9
File Edit View Insert Tools Desktop Window Help
Total minus 2 EOFs
Figure 8
File Edit View Insert Tools Desktop Window Help
Total minus 3 EOFs
Figure 7
File Edit View Insert Tools Desktop Window Help
Sum of 3 EOFS
1100 1000 900 800 700 600 500 400 300 200 100
20 40 60 80 100
0.2 0.15 0.1 0.05 0 -0.05 -0.1 -0.15 -0.2



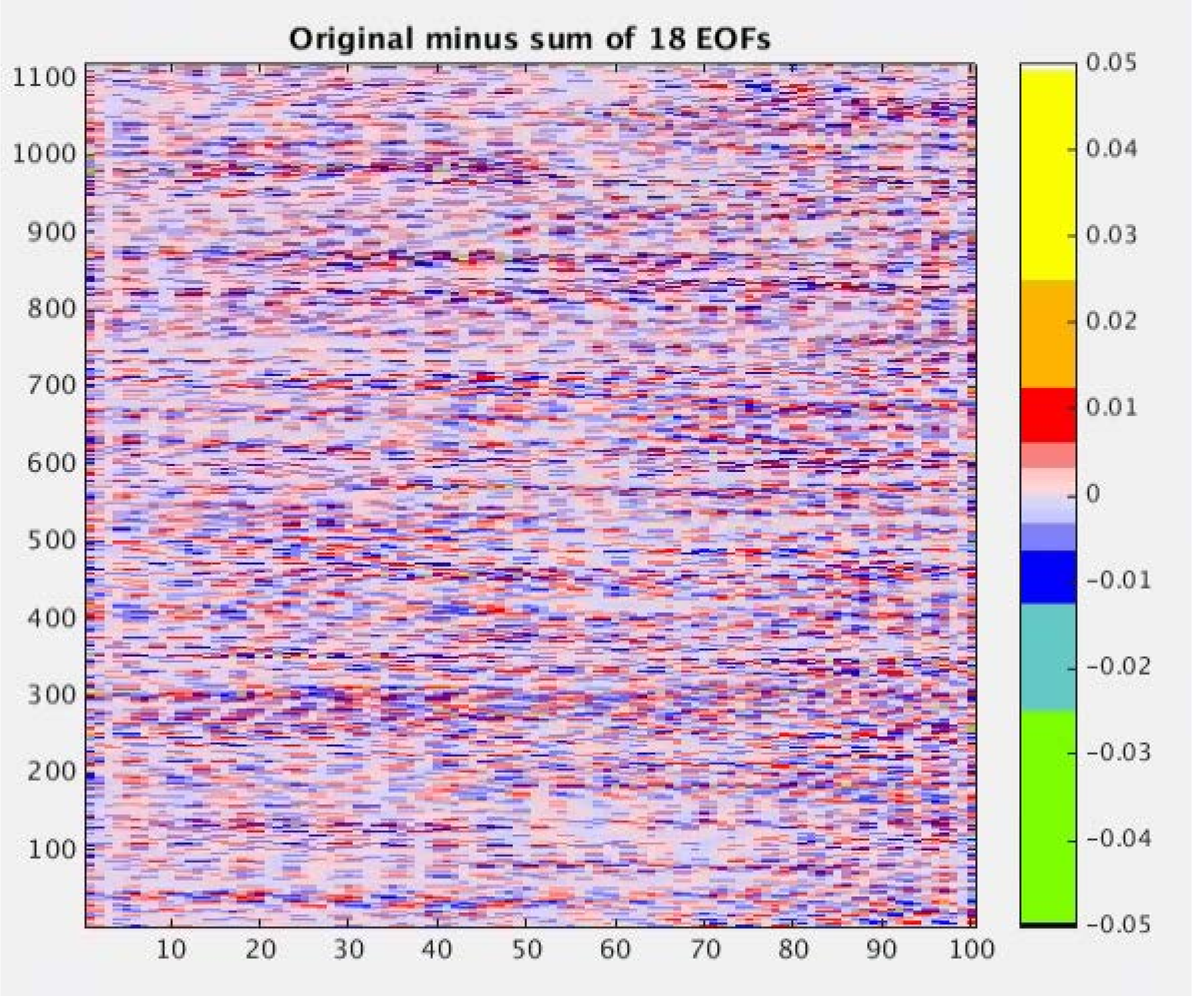
Original data
1100
1000
900
800
700
600
500
400
300
200
100
10
20
30
40
50
60
70
80
90
100
0.2
0.15
0.1
0.05
0
-0.05
-0.1
-0.15
-0.2



Sum of 18 EOFS
1100
1000
900
800
700
600
500
400
300
200
100
10
20
30
40
50
60
70
80
90
100
0.2
0.15
0.1
0.05
0
-0.05
-0.1
-0.15
-0.2



Original minus sum of 18 EOFs
1100
1000
900
800
700
600
500
400
300
200
100
10
20
30
40
50
60
70
80
90
100
0.2
0.15
0.1
0.05
0
-0.05
-0.1
-0.15
-0.2

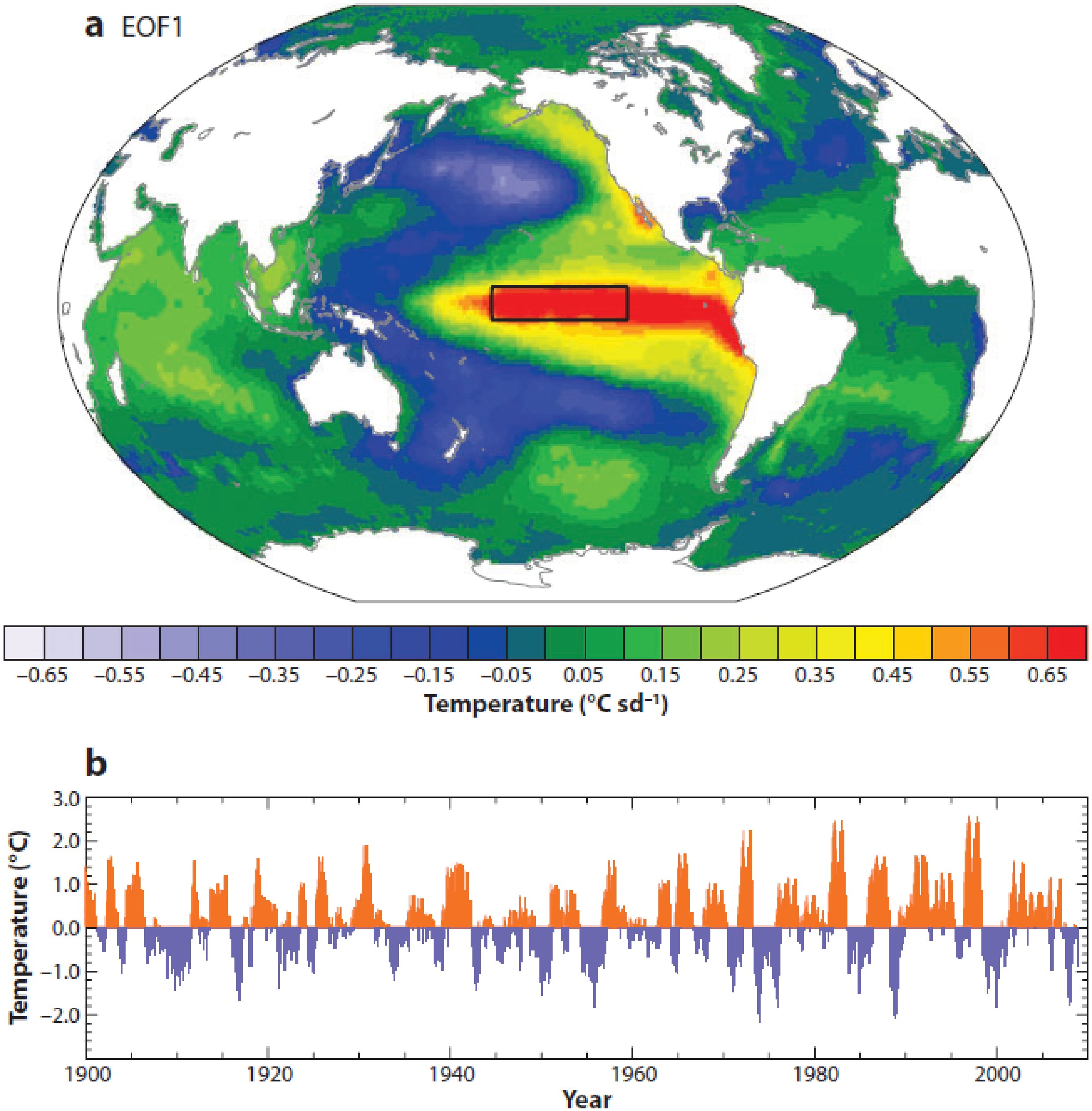

Original minus sum of 18 EOFs
1100
1000
900
800
700
600
500
400
300
200
100
10
20
30
40
50
60
70
80
90
100
0.05
0.04
0.03
0.02
0.01
0
−0.01
−0.02
−0.03
−0.04
−0.05

Deser et al., Annu. Rev. Mar. Sci. 2010

“The Niño-3.4 SST index is nearly identical to the PC time series associated with the leading EOF: Their correlation coefficient is 0.93”

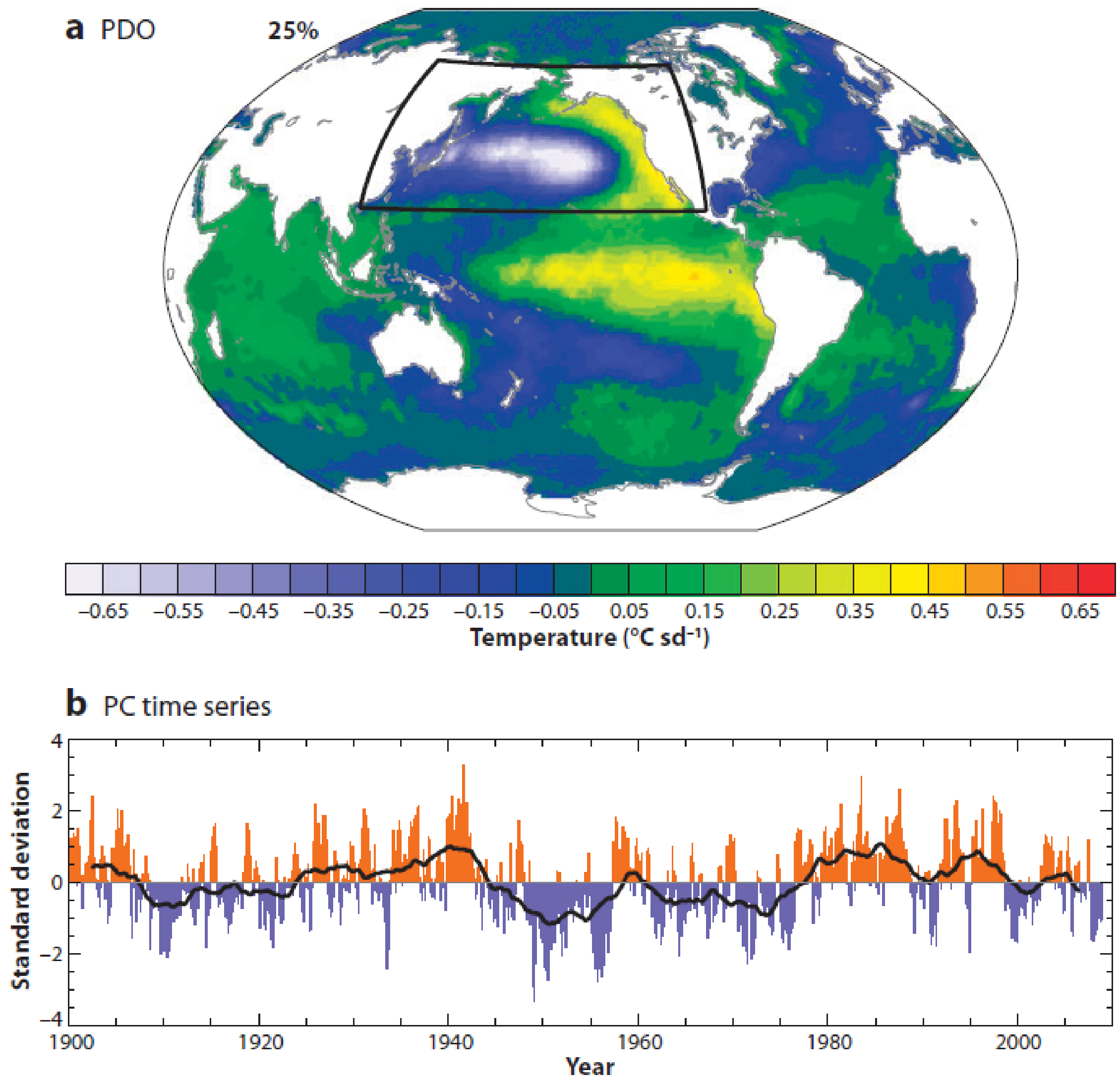
First EOF of global sea surface temperature

Note, this is an example where there are more points in space than in time.



Deser et al., Annu. Rev. Mar. Sci. 2010

"The leading EOF of monthly SST anomalies over the North Pacific (after removing the global mean SST anomaly) and its associated PC time series are termed the Pacific Decadal Oscillation (PDO) after Mantua et al. (1997)."

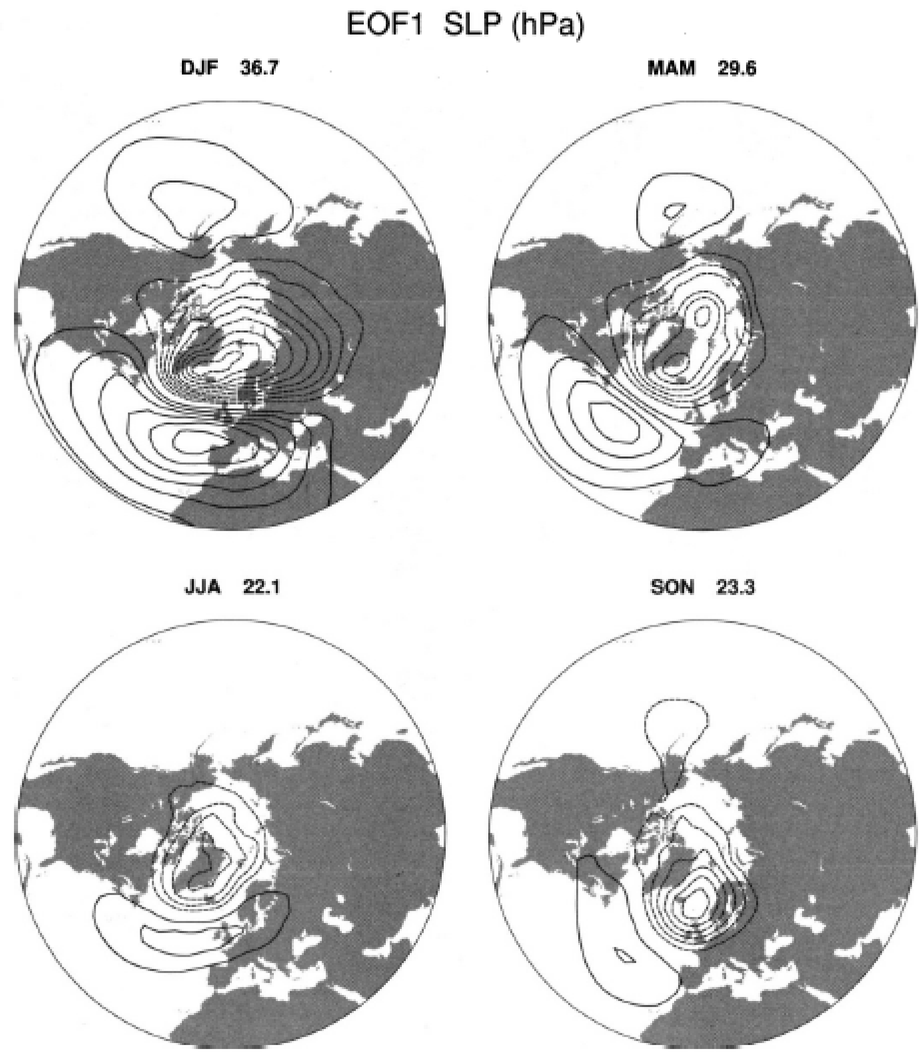


Hurrell et al., 2003:
The North Atlantic Oscillation:
Climatic Significance and Environmental Impact
Geophysical Monograph 134
doi: 10.1029/134GM01

"the NAO dominates the leading EOF of the NH SLP field"

First EOF of northern hemisphere sea level pressure (time series of 3-month averages for different seasons):

There are lots of different NAO definitions, all looking rather like this

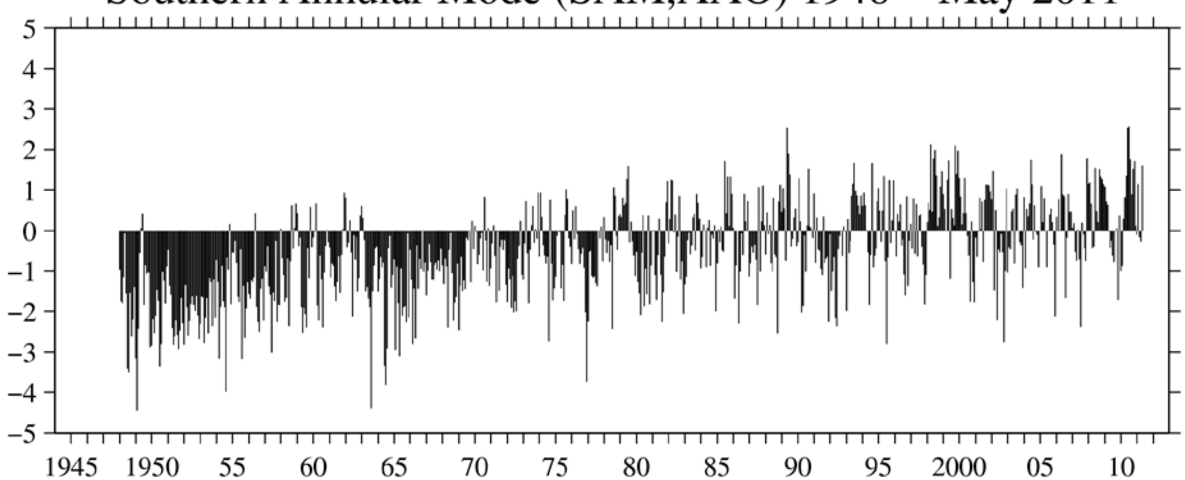


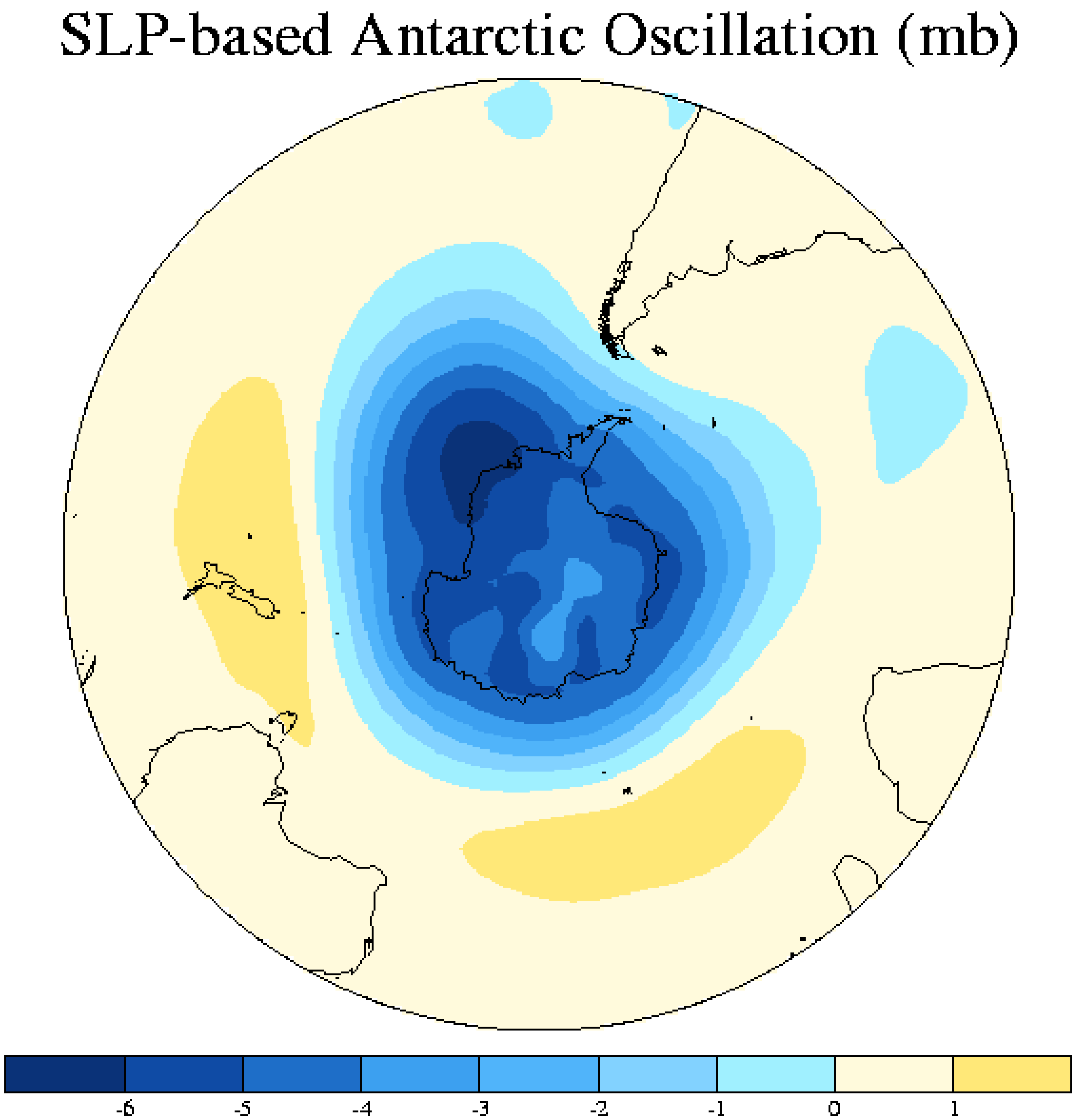
Thompson and Wallace, 2000:
Annular Modes in the Extratropical Circulation. Part I: Month-to-Month Variability.
J. Climate,**13**, 1000–1016.
doi: 10.1175/1520-0442(2000)013<1000:AMITEC>2.0.CO;2

The Antarctic Oscillation (AAO or Southern Annular Mode SAM)

“This SH mode emerges as **the leading EOF of** zonally varying geopotential height based on station data (Szeredi and Karoly 1987), gridded analyses of SLP (Rogers and van Loon 1982; Gong and Wang 1999), 500-hPa height (Rogers and van Loon 1982; Kidson 1988b), and 300-hPa height (Karoly 1990). It has also been identified as the leading EOF of the SH zonally averaged geopotential height field at 1000 hPa (Shiotani 1990), the zonally averaged zonal wind at 500 hPa (Kidson 1988a), and in the 1000–100- hPa layer (Yoden et al. 1987; Hartmann and Lo 1998)

Southern Annular Mode (SAM,AAO) 1948 – May 2011





see JISAO web pages for more information

… and so on

… and so on

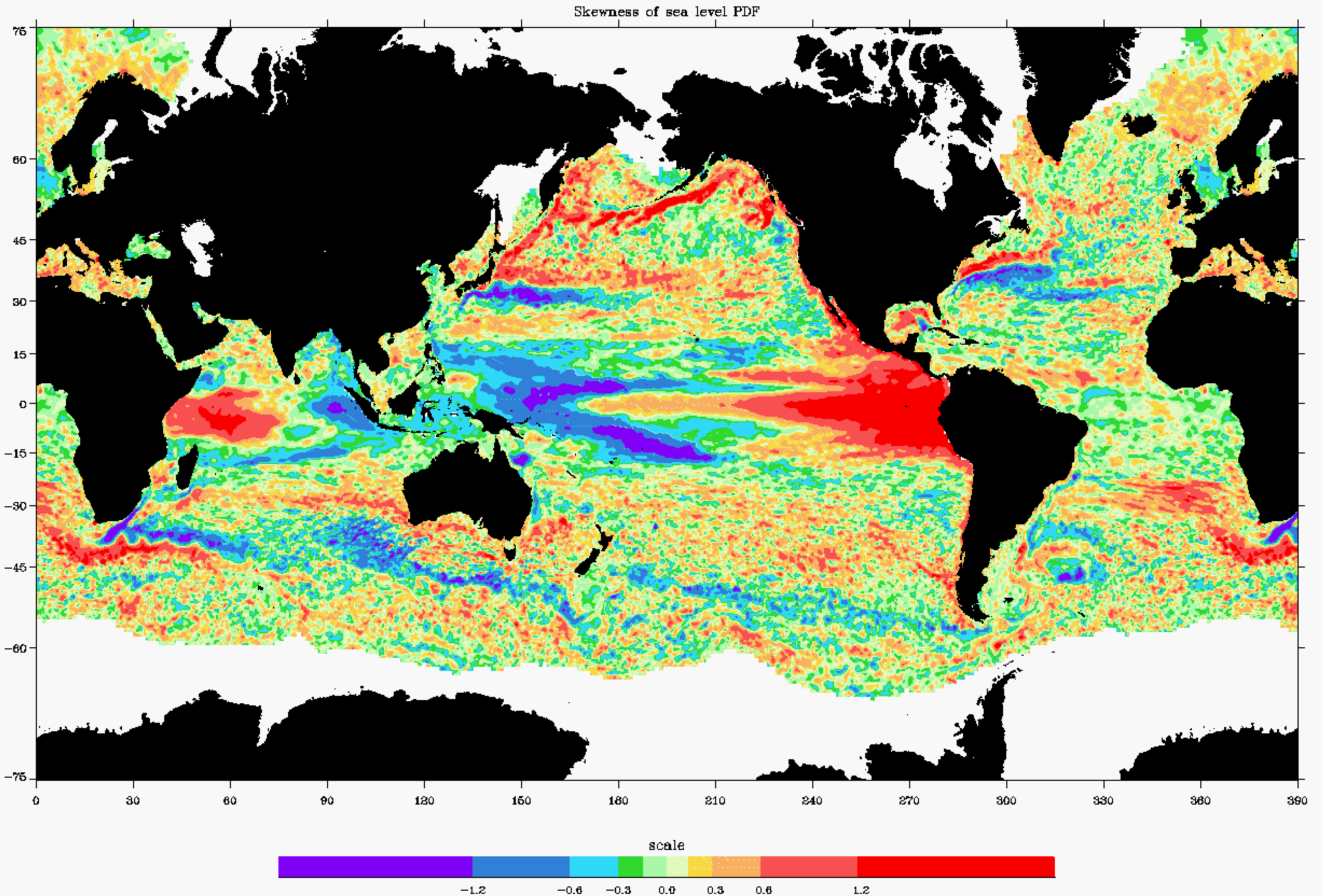
Hurrell et al., 2003:
The North Atlantic Oscillation:
Climatic Significance and Environmental Impact
Geophysical Monograph 134
doi: 10.1029/134GM01

Pacific North American pattern

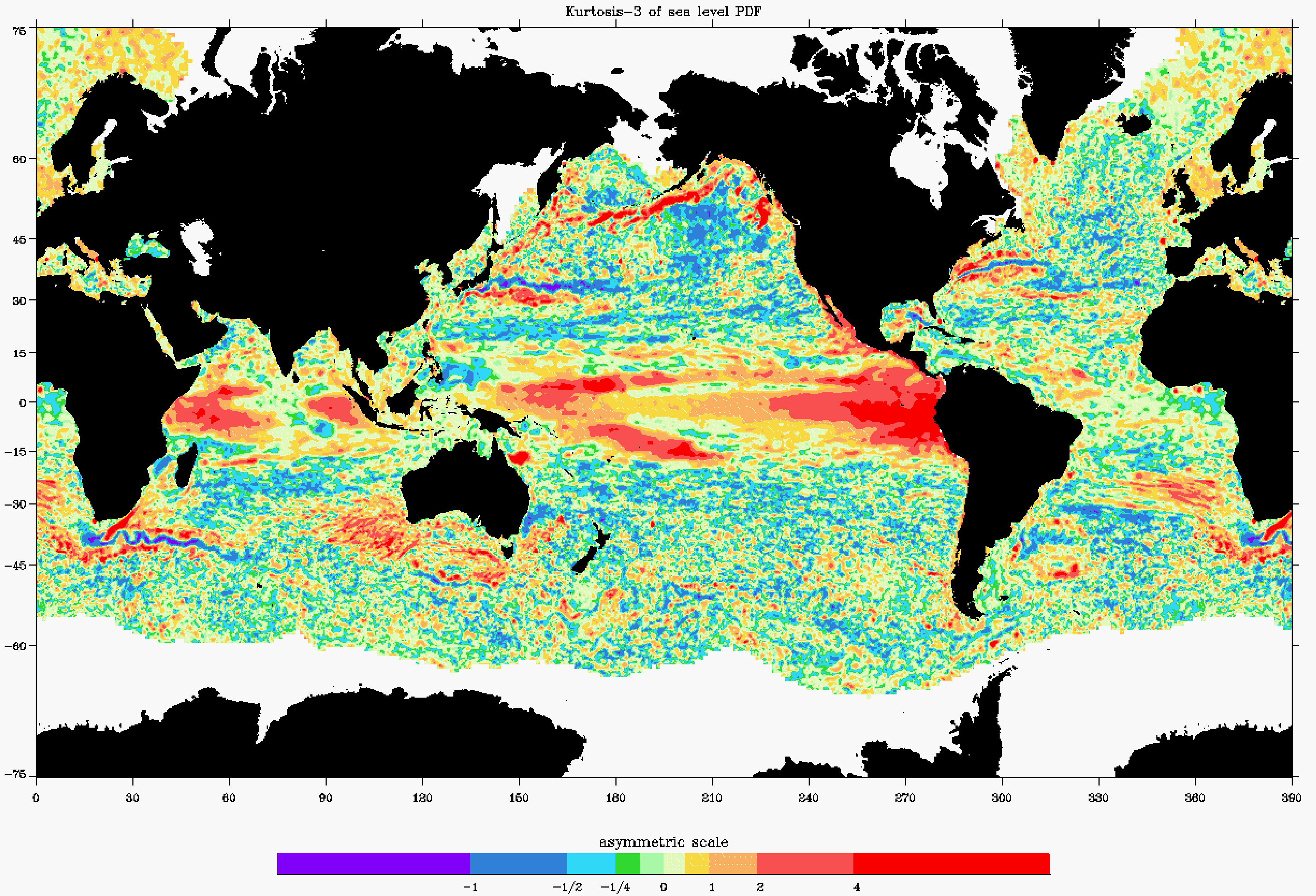
"Both the NAO and PNA are also reflected in the spatial patterns of the two leading empirically-determined orthogonal functions (EOFs) of NH boreal winter 500 hPa height (not shown), **but in order to see them clearly it is necessary to rotate (i.e., to form linear combinations of) the EOFs** in a manner that tends to simplify their spatial structure"

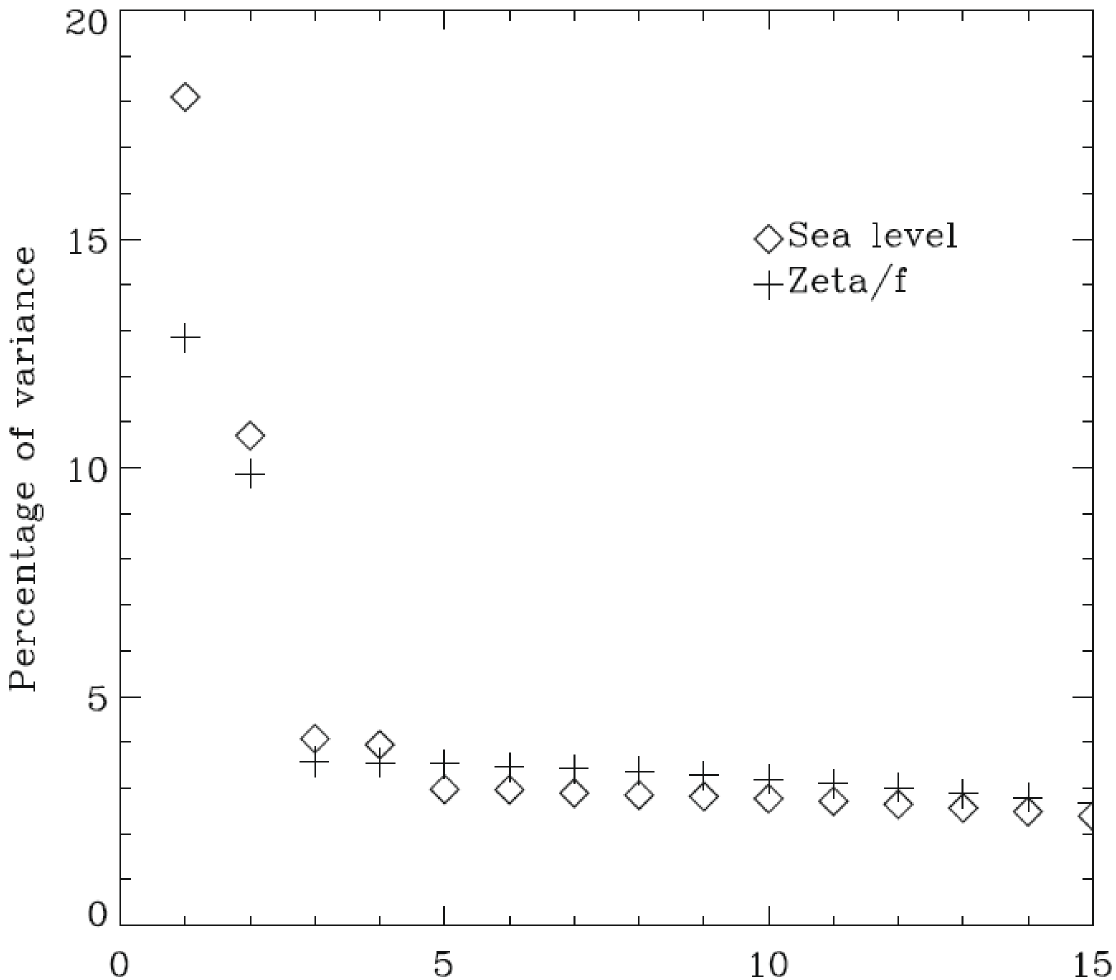
… so what's that about?

Sea level skewness (annual cycle removed first)



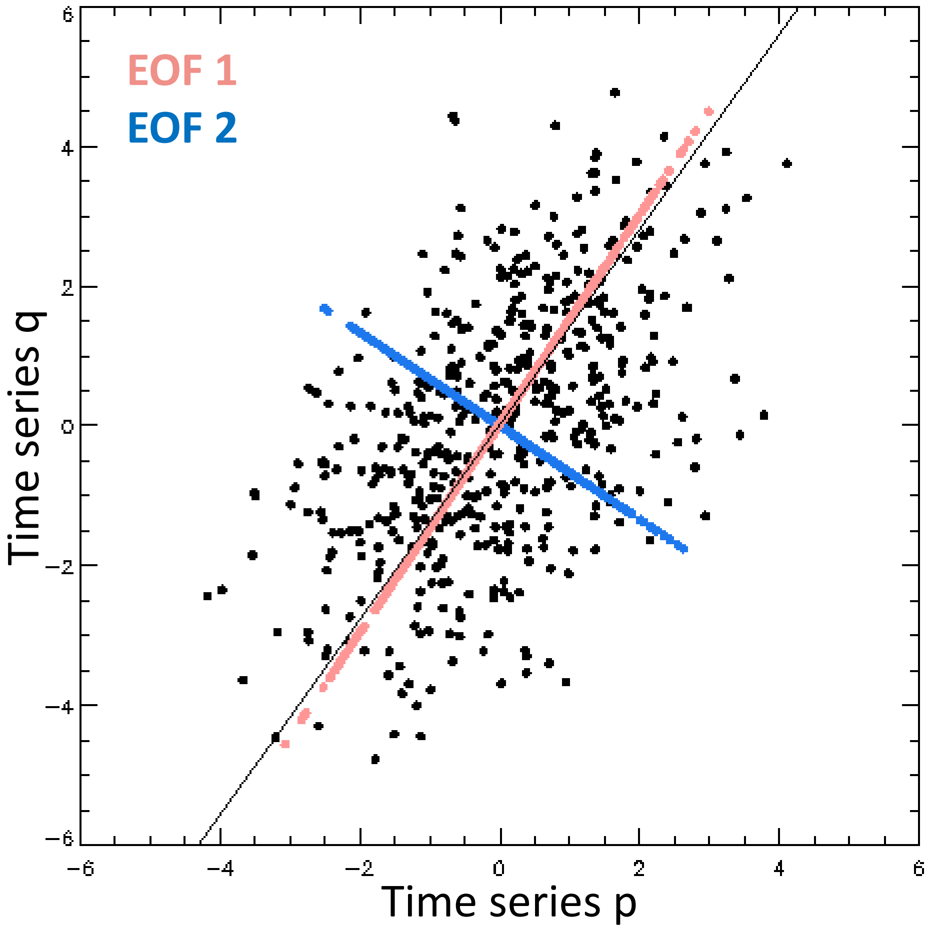
Sea level kurtosis-3 (annual cycle removed first). Notice the scale bar.





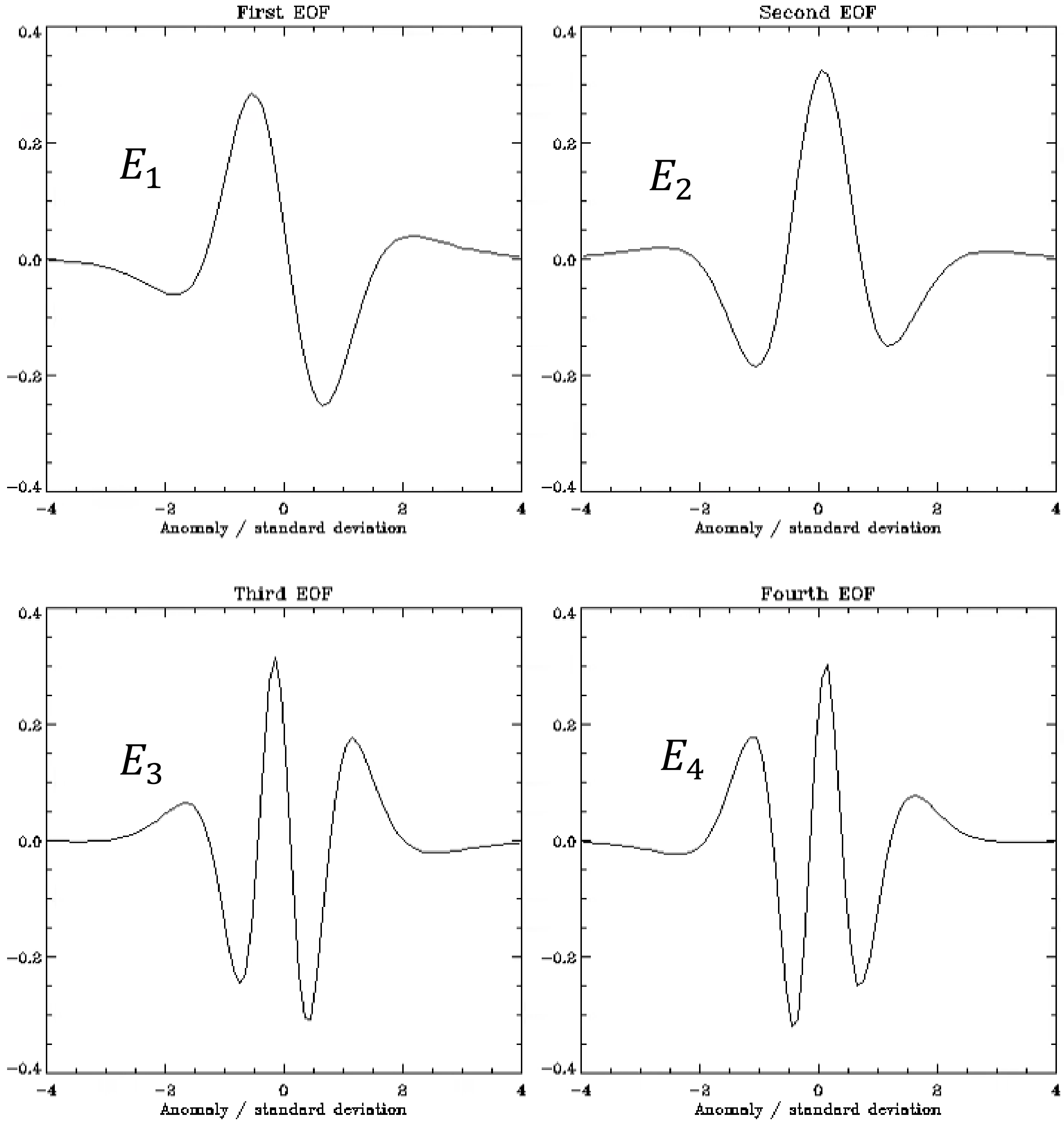
Notice how EOFs appear in pairs (especially 3rd and 4th explain nearly the same percentage of variance)

When this happens, it means the two EOFs are not clearly distinguished – a rotated combination of them would work almost as well.



Similar variance explained means the cloud of points is almost symmetrical.

When it is symmetrical, any pair of orthogonal axes would be equally useful.



Any 2 EOFs can be "rotated":

$$R_1 = E_1 \cos\theta + E_2 \sin\theta$$

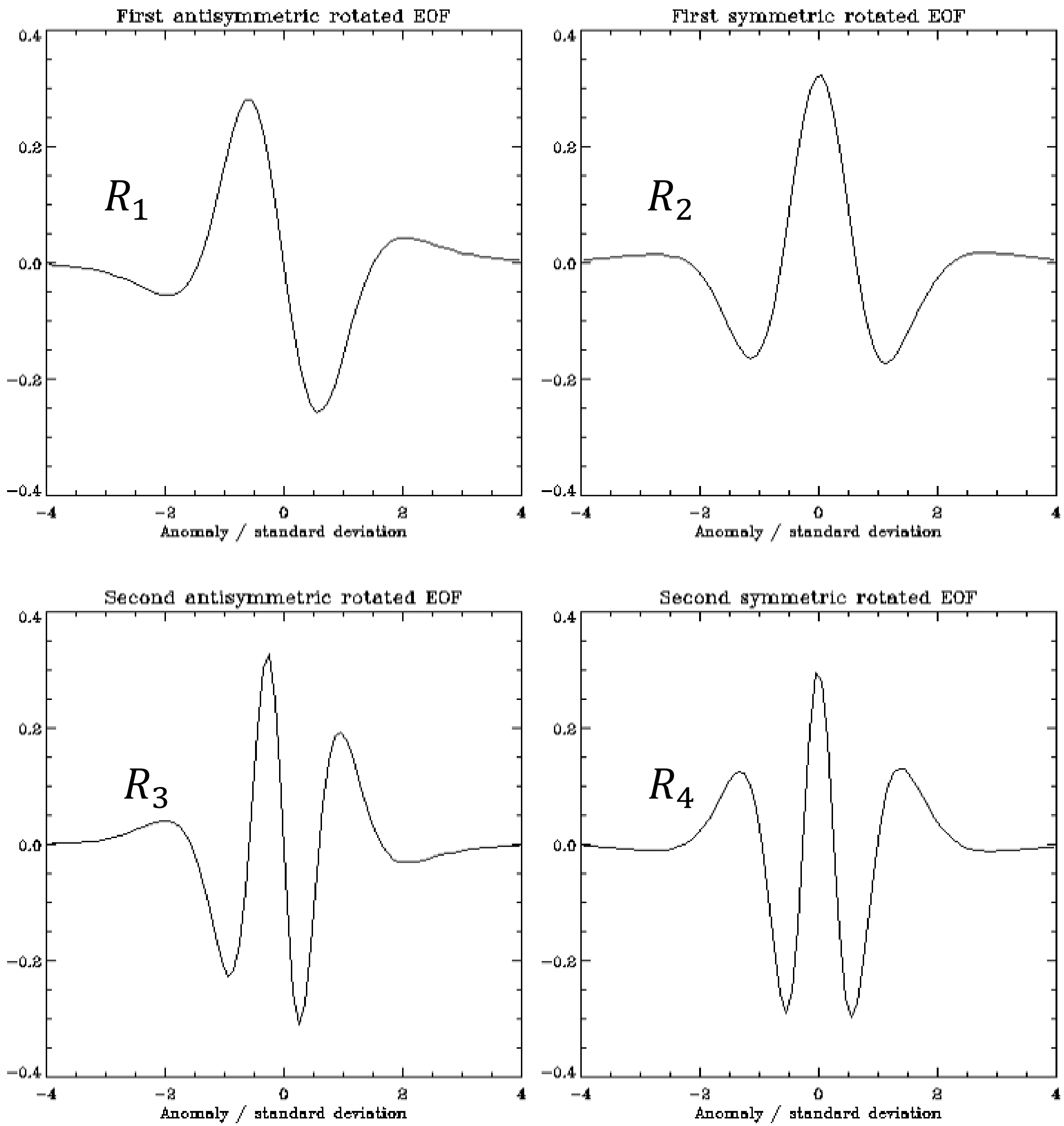
$$R_2 = E_2 \cos\theta - E_1 \sin\theta$$

They remain orthogonal to each other and, because they are in the same plane as before, they remain orthogonal to all other EOFs.

$$R_3 = E_3 \cos\varphi + E_4 \sin\varphi$$

$$R_4 = E_4 \cos\varphi - E_3 \sin\varphi$$

This rotates the "time" or "space" component – some work remains to calculate the complementary one (project the original data onto the new components)



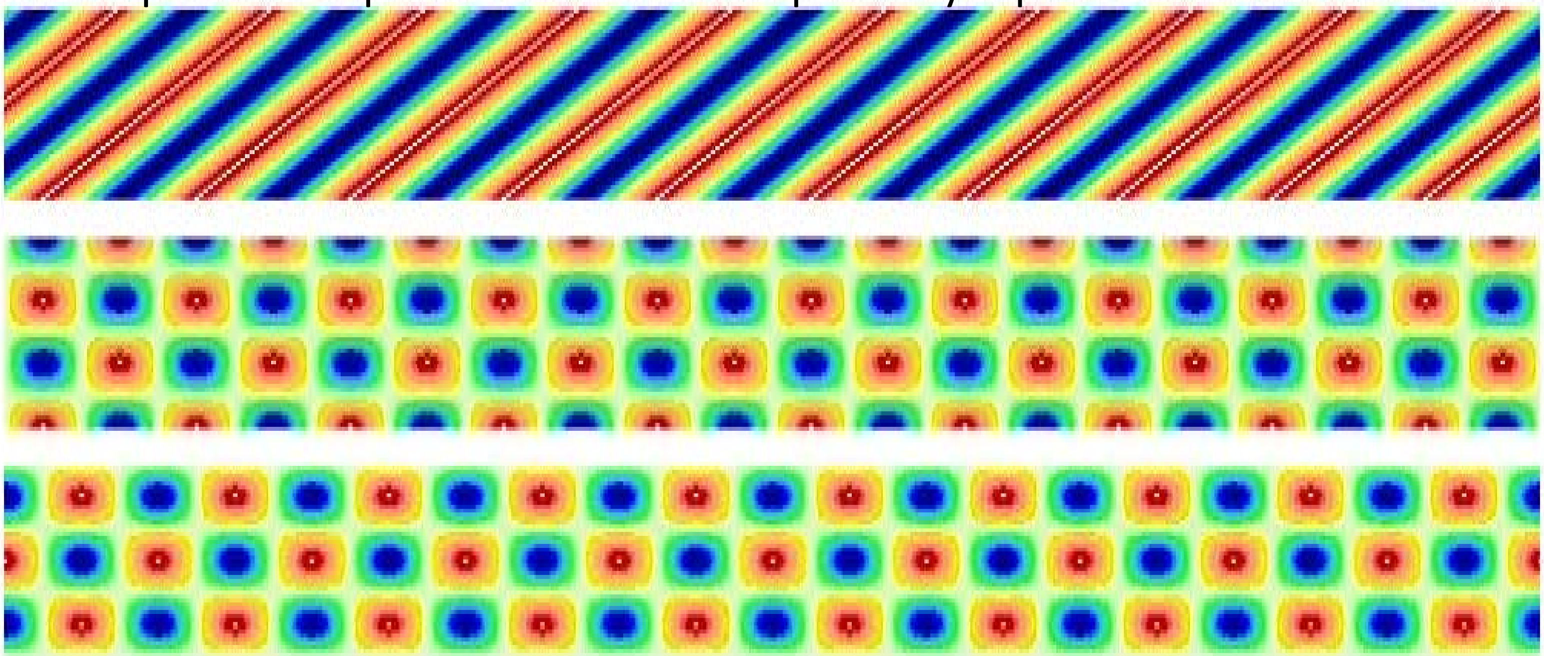
Because I was interested in skewness (asymmetry) and kurtosis (symmetric), I chose to rotate the pairs into the most symmetric and most antisymmetric combinations possible.

The first 2 were close to this already (and were quite distinct)

Numbers 3 and 4 were not so distinct, and the rotation made a big difference

Another way pairs of EOFs arise is from propagating patterns:

This pattern in space and time can be perfectly reproduced from 2 EOFs:



EOF 1
50% of
variance

EOF 2
50% of
variance

Rotation of EOF1 and EOF2 simply shifts them
up/down or left/right by the same amount