ADVANCED ENVIRONMENTAL DATA ANALYSIS

HOMEWORK on EOF/PCA ANALYSIS

The empirical orthogonal function (EOF) analysis, also known as the principal component analysis (PCA), is a statistical technique used primarily to find relevant structures in a dataset and/or reduce its dimensionality. In this homework we explore this technique through two applications. Please, discuss your results and produce easy-to-interpret figures with appropriate scales.

[1] EOF Analysis as Tool for Climate Dynamics Studies

The MATLAB file *Pacif_SST.mat* contains the monthly mean values of the Pacific sea surface temperature (SST) for the period 1959-2000.

(a) Load the SST data and plot its temporal mean and standard deviation (std).

Code hint:

```
load Pacific_SST
pcolor(SST.lon,SST.lat,SST.fld(:,:,1)) %SST.fld[lat,lon,time]
hold on
world_coastp('k','linewidth',2)
shading interp
```

(b) **Data anomalies.** Compute monthly mean anomalies and plot a new map for the std. Discuss briefly the difference between this std map and the one obtained in (a).

Hint:

To compute the monthly mean anomalies you need to first compute a 12-months climatology, which is a 2D mean field for each month. The monthly mean anomalies are then computed by subtracting to each monthly record the corresponding monthly mean climatology.

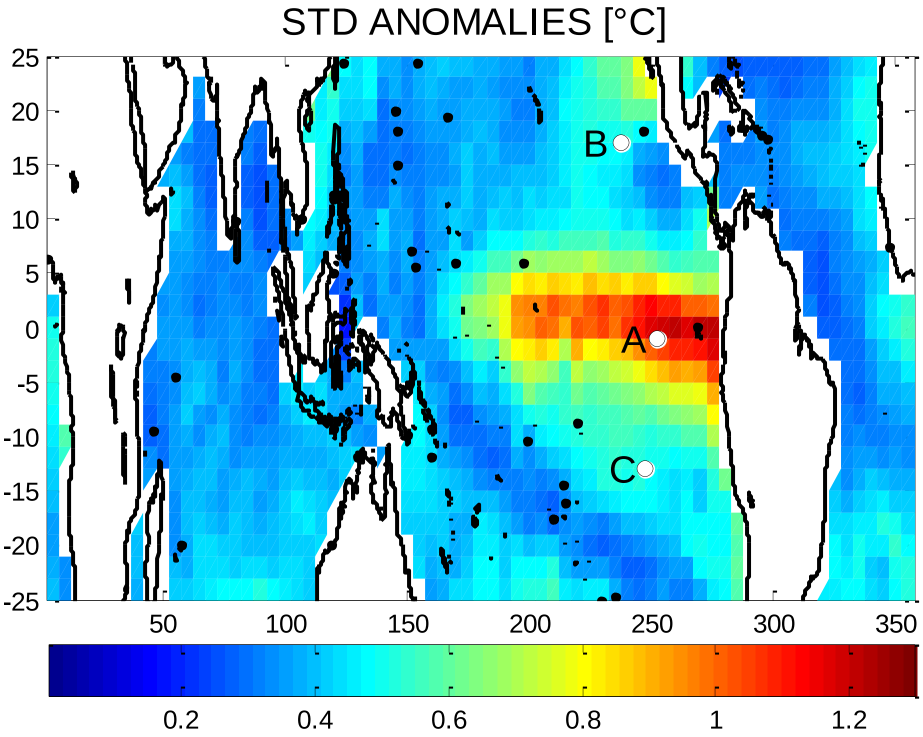
(c) EOF decomposition. Perform EOF decomposition and plot the first three modes (EOF pattern + PC timeseries). Indicate the percentage of variance explained by each mode. Renormalize the EOFs so that the PC are in units of STD and the EOFs are in units of regression.

Code hint:

```
clear eofs pcs

[I,J,T]=size(SST.ano); % get dataset dimensions
eofs=zeros(I,J); % initialize eofs arrays
dnew=reshape(SST.ano,[I*J,T]); %reshape dataset (I,J,T) >> (I*J,T)
in=find(isnan(dnew(:,1)) == 0); %find sea points
dnew=dnew(in,:); %take only the sea points

dd=(dnew'*dnew); %compute the covariance-like matrix (not scaled!)

[A,D] = eig(dd); % get eigenvalues(D) and eigenvectros(A)

E=dnew*A; %Compute eofs/pcs, project data onto eigenvectors

lambda=diag(D); % Get eigenvalues
modes=lambda/sum(lambda)*100; %Compute explained variance

im=numel(lambda);

%Sort EOFs/PCs from largest to smallest
im=??; %What’s “im”? You should able to figure it out!
for i=length(im):-1:1
m1=zeros(I,J); m1(:)=NaN;
m1(in)=E(:,im(i));
eofs(:,:,-i+1+length(im))=m1;
pcs(:,-i+1+length(im))=A(:,im(i));
varexp(-i+1+length(im))=modes(im(i));
end
```

(d) **Eigenvalue Spectrum**. Plot the eigenvalue spectrum (λ_k-k) in percentage of explained variance with the confidence limits computed using the North's rule (North at al., 1982). In the computation of the confidence limits use 12 as number of degrees of freedom. How many statistical independent modes do you see?

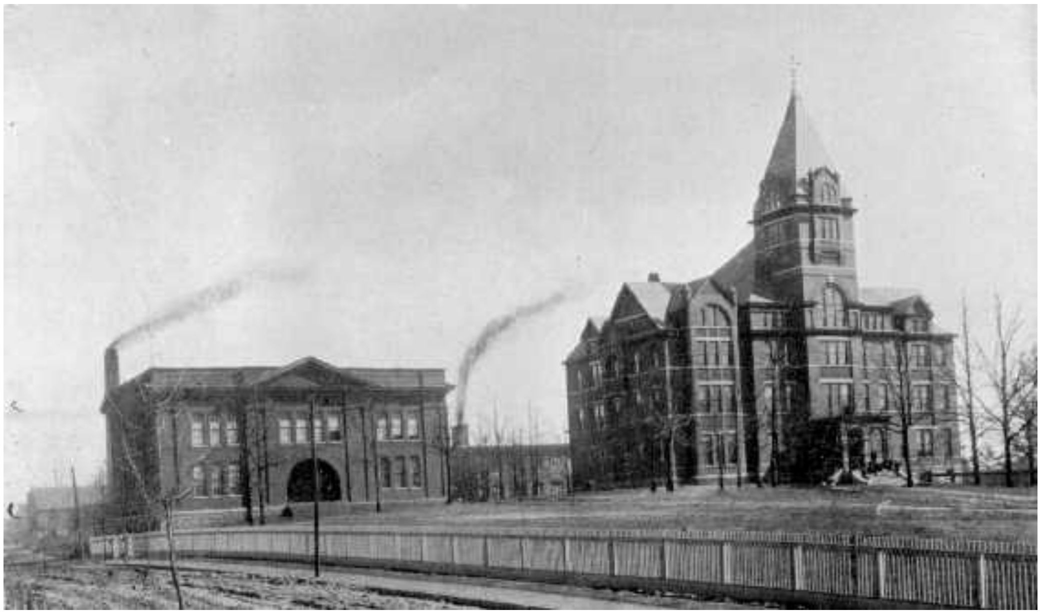
(e) **EOF reconstruction.** Reconstruct the SST anomaly using the first 3 and 7 modes. Plot the std map of the reconstructed SST anomalies along with the std map obtained in (b). Plot the time series of the reconstructed and observed SST anomalies for locations indicated with A, B, and C in the map below. Use the correlation coefficient between observed and reconstructed data to evaluate the quality of the EOF reconstruction. How does the reconstruction change among the locations?



Code hint:

```
Yp(1)=13; Xp(1)=51; % Point A
Yp(2)=22; Xp(2)=48; % Point B
Yp(3)=7; Xp(3)=50; % Point C
figure
pcolor(SST.lon,SST.lat, SST.fld(:,:,1))
hold on
world_coastp('k','linewidth',2)
shading interp
lon=SST.lon;
lat=SST.lat;

for i=1:3;
plot(lon(Yp(i),Xp(i)),lat(Yp(i),Xp(i)),'o','MarkerFaceColor','w')
end
```

[2] Image Compression Using Principal Component Analysis (PCA)



The black-and-white picture displayed above has dimension of 331x565 pixels that is an array of 187015 elements. Use the PCA technique to reduce at about 20% the size of the array you need to store in order to recover the picture in its original size. This means that the sum of EOF and PC elements that you need to use to reconstruct the picture (EOFsubset + PCsubset) should be ~36000.

Hint (pseudo-code):

```
FIELD=EOF*PC’;   FIELD_approx=EOFsubset*PCsubset’;
R=numel(mean(FIELD)) + numel(EOFsubset) + numel(PCsubset))/numel(FIELD)
%R should be about 0.2
```

How many EOF/PC components do you need to retain? What is the amount of variance explained by this EOF/PC subset? Show the picture that one would recover using this EOF/PC subset.

Code hint:

```
[TMP]=imread('Tech_TowerAndShop_1899.gif');
imshow(TMP),shg
IMG=double(TMP(:,:,1)); % Convert from uint8 to double
%... Write your own code to generate the IMG_rec
imshow(uint8(IMG_rec)),shg %Plot the picture reconstruction

% (1) Compute IMG anomalies (IMGa) (i.e., each column of IMGa
      has 0 mean)
% (2) Use the IMGa to compute the EOF decomposition (find
      EOFs/PCs)
% (3) Reconstruct the anomalies using an EOF/PC subset
% (4) Recover the picture adding to the anomalies reconstructed
      in (3) the mean removed in (1)
```