C The procedure to do a phase coherence analysis as described in the paper

C http://onlinelibrary.wiley.com/doi/10.1029/2005GL023225/full ) by Maruan and Kurths is as follows :

1. Low-pass filter the data in the spectral domain. A smooth function (arcus tangens) damping frequencies >0.7 year−1 is chosen.

2. Estimate derivatives by second order difference scheme and running mean with window width 2l + 1 = 13 months

3. Embed by Hilbert transformation with phase defined according to equation (3).

4. Unwrap the phases (add 2π after each oscillation)

A detailed MATLAB code to perform the above analysis is shown below:

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

clear all

clc

% Loading the data

ismr\_load = xlsread('ismr.xls');

nino3\_load = xlsread('nino3.xls');

for i=1:133

for j=1:12

ismr\_v((i-1)\*12+j) = ismr\_load(i,j);

nino3((i-1)\*12+j) = nino3\_load(i,j);

end

end

% Since the data on IITM's website for ISMR is a multiple of 10, we have to divide the data by 10

ismr=ismr\_v/10;

% Since our work focuses on the inference of phase relations of inter-annual

% oscillations, we low-pass filtered the data in the spectral domain by multiplying the Fourier

% transformation of the data with a hyperbolic tangent, i.e. high frequency variability with

% frequencies higher than 0.7 cycles per year is damped.

% (reference : Maruan 2006)

% Testing Hyperbolic tangens filter

% x = -5:0.01:5;

% plot(x,-tanh(x-0.7)), grid on

% 0.7 cycles per year means 0.7/12 cycles per month

% 0.0583 cycles per month

N = size(ismr,2);

x = 1:N;

years = x/12+1871;

filt = (-tanh(x-0.00583)+1)/2;

% Filtering the ISMR data

ismrt = fft(ismr);

ismrt = real(ismrt) .\* filt;

ismr\_filt = ifft(real(ismrt));

ismrt\_filt\_dtr = detrend(ismr\_filt);

% Filtering the NINO3 data

nino3t = fft(nino3);

nino3t = nino3t .\* filt;

nino3\_filt = ifft(nino3t);

nino3\_filt\_dtr = detrend(nino3\_filt);

% estimate derivatives by second order difference scheme and running

ismr\_sec\_diff(1:N-2) = 0.0;

nino3\_sec\_diff(1:N-2) = 0.0;

% Second order differencing done as per

% http://robjhyndman.com/talks/RevolutionR/8-Differencing.pdf

ismr\_sec\_diff(1:N-2) = ismrt\_filt\_dtr(1:N-2) - 2\*ismrt\_filt\_dtr(2:N-1) + ismrt\_filt\_dtr(3:N);

nino3\_sec\_diff(1:N-2) = nino3\_filt\_dtr(1:N-2) - 2\*nino3\_filt\_dtr(2:N-1) + nino3\_filt\_dtr(3:N);

% running mean with window width 2l + 1 = 13 data points

n1 = 1594;

%ismrr(1:n1) = 0.0;

%nino3r(1:n1) = 0.0;

Running = 13;

for i=1:n1-Running

ismrr(i) = sum(ismr\_sec\_diff(i:i+Running-1))/Running;

nino3r(i) = sum(nino3\_sec\_diff(i:i+Running-1))/Running;

end

n2 = 1594-13+1;

x = 1:n2;

years = x/12+1871;

% Embed by Hilbert transformation with phase defined according to Eq. (4.7).

ismr\_hilbert = hilbert(ismrr);

nino3\_hilbert = hilbert(nino3r);

ismr\_real = real(ismr\_hilbert);

ismr\_imag = imag(ismr\_hilbert);

nino3\_real = real(nino3\_hilbert);

nino3\_imag = imag(nino3\_hilbert);

% Phase unwrapped as per

% https://in.mathworks.com/matlabcentral/answers/295635-why-the-phase-obtained-with-hilbert-transform-and-phase-unwrap-is-different-from-the-actual-phase

phase\_ismr = unwrap(angle(ismr\_hilbert));

phase\_nino3 = unwrap(angle(nino3\_hilbert));

phase\_coh = phase\_ismr - phase\_nino3;

plot(years,phase\_coh);

xlabel('years');

ylabel('Phase coherence')

Title('Phase coherence between ISMR and NINO3')

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Final Phase plot https://raw.githubusercontent.com/manmeet3591/Miscellaneous/master/Phase\_Coh/abs\_phase.png

Phase coherence plot between ISMR and NINO3 https://raw.githubusercontent.com/manmeet3591/Miscellaneous/master/Phase\_Coh/phase\_coh.png