**#Method of critical fluctuations**

function F = Frequency(A)

F = zeros(1,numel(A));

Dt = 0;

for ii = 1:numel(A)

 if A(ii) == 1

 Dt = Dt + 1;

 else

 if Dt ~= 0

 F(Dt) = F(Dt) + 1;

 Dt = 0;

 else

 continue;

 end

 end

end

if A(end) == 1

 F(Dt) = F(Dt) + 1;

end

end

**#Haar wavelet analysis**

function wavelets\_corr(f\_ratioLambda, f\_correl, numDataToRead)

 Nmax = floor(numDataToRead);%floor is used in case a double is given

 if ( Nmax < 0 )

 disp('numDataToRead must be a positive integer');

 end

 % open files to write

 f\_ratioL = fopen(f\_ratioLambda, 'w');

 %disp(f\_ratioL)

 %disp(f\_ratioLambda)

 if ( f\_ratioL < 3 )

 disp('Error opening ratioLambda file for write');

 return;

 end

 %read data from file

 [y, err\_occ] = read\_correl\_file(f\_correl, Nmax);

 if ( err\_occ )

 fclose(f\_ratioL);

 return;

 end

 % calculations

 j = 0;

 k = 0;

 sumcritical = 0;

 lcount = 0;

 for N = (Nmax - 9): Nmax % initial fortran code : do N=771,Nmax

 u1=floor(k\*N/(2^j)+N/(2^(j+1)));

 u2=floor(k\*N/(2^j));

 u3=floor((k+1)\*N/(2^j));

 q1=floor(k\*N/(2^(j+1))+N/(2^(j+2)));

 q2=floor(k\*N/(2^(j+1)));

 q3=floor((k+1)\*N/(2^(j+1)));

 r1=floor(k\*N/(2^(j+2))+N/(2^(j+3)));

 r2=floor(k\*N/(2^(j+2)));

 r3=floor((k+1)\*N/(2^(j+2)));

 %fprintf('%3d %3d %3d %3d %3d %3d %3d %3d %3d %3d\n',N,u1,u2,u3,q1,q2,q3,r1,r2,r3);

 sum1=0;

 for i = u2:u1 % initial fortran code : do i=u2,u1,1

 sum1 = sum1 + get\_data\_from\_pos(y, i);

 end

 sum2=0;

 for i = u1:u3 % initial fortran code : do i=u1,u3,1

 sum2 = sum2 + get\_data\_from\_pos(y, i);

 end

 sum3=0;

 for i = q2:q1 % initial fortran code : do i=q2,q1,1

 sum3 = sum3 + get\_data\_from\_pos(y, i);

 end

 sum4=0;

 for i = q1:q3 % initial fortran code : do i=q1,q3,1

 sum4=sum4 + get\_data\_from\_pos(y, i);

 end

 sum5=0;

 for i = r2:r1 % initial fortean code : do i=r2,r1,1

 sum5 = sum5 + get\_data\_from\_pos(y, i);

 end

 sum6=0;

 for i = r1:r3 % initial fortran code : do i=r1,r3,1

 sum6 = sum6 + get\_data\_from\_pos(y, i);

 end

 %fprintf('%3d %3d %3d %3d %3d %3d %3d %3d %3d %3d %5.0f %5.0f %5.0f %5.0f %5.0f %5.0f\n',N,u1,u2,u3,q1,q2,q3,r1,r2,r3,sum1,sum2,sum3,sum4,sum5,sum6);

 % write data to files

 R1=((sum1-sum2)/(sum3-sum4))/(2^0.5);

 R2=((sum3-sum4)/(sum5-sum6))/(2^0.5);

 fprintf(f\_ratioL,'%d %20.9f %20.9f %20.9f\n', N, R1, R2, R1/R2); % initial code : write(2,14)N,R1

 sumcritical = sumcritical+ (1-(R1/R2))^2;

 lcount=lcount+1;

 end

 fprintf('Q: %f\n', sumcritical/lcount); % initial code : print\*,sumcritical/lcount

 %fprintf('%d\n', lcount); %initial code : print\*,lcount

 fclose(f\_ratioL);

end

function val = get\_data\_from\_pos(y, pos)

 if ( pos > 0 )

 val = y(pos);

 else

 val = 0;

 end

end

function [y, err\_occ] = read\_correl\_file(f\_correl, Nmax)

 y = dlmread(f\_correl);

 if ( length(y) < Nmax )

 disp('Less values are read from correl file');

 err\_occ = 1;

 return;

 end

 %for i = 1:Nmax

 %fprintf('Data %d %f\n',i,y(i));

 %end

 err\_occ = 0;

end