Supporting Information for "Bayesian spectral likelihood for hydrological parameter inference"

Bettina Schaefli[1] and Dmitri Kavetski[2]

Contents of this file

1. Matlab implementation of BSL for AR1 error models

D. Kavetski, School of Civil, Environmental and Mining Engineering, University of Adelaide, SA, Australia

Corresponding author: B. Schaefli, Institute of Earth Surface Dynamics, Faculty of Geosciences and Environment, University of Lausanne, 1015 Lausanne, Switzerland (bettina.schaefli@unil.ch)

[1]Institute of Earth Surface Dynamics,
Faculty of Geosciences and Environment,
University of Lausanne, Switzerland

[2]School of Civil, Environmental and
Mining Engineering, University of Adelaide,
SA, Australia

Introduction

This supporting information presents a Matlab implementation of BSL for AR1 error models. This pdf has been generated from a latex file; some signs do thus not copy well to another text format. In particular the sign "∼" will likely not print and any combinations of "fl" and "ff" will result in a line break (manual correction after copying required).

Matlab implementation of BSL for AR1 error models.

```
function [logL,resPeriodo,errPDS]=bslAR1(xobs,xsim,arcoef,varinnov,meaninnov)


% input:
% xobs, xsim: observed and simulated time series of same length (size n x 1)
% arcoef: lag-1 autoregressive parameter of the AR1 error model (size 1 x 1)
% varinnov: variance of the innovations of the error model(size 1 x 1)
% meaninnov: mean of the innovations of the error model (size 1 x 1)
%
% output:
% logL: log-likelihood, log-BSL (size 1 x 1)
% resPeriodo: periodogram of the residuals, where residuals = xobs - xsim (size n/2 x 1)
% errPDS: power density spectrum of the AR1 error model (size n/2 x 1)


if nargin<5
   meaninnov=0; % if the mean is not given, assume it is zero
end
nmax=length(xsim);
```

```
if nmax~=length(xobs) % make sure you have the "is not equal" sign here
   error('xobs and xsim data should have the same length');
end
res=xobs-xsim;
if floor(nmax/2)==nmax/2
   nhalfm=nmax/2;
else
   nhalfm=floor((nmax-1)/2);
end
errPDS=pdsAR1M(arcoef,meaninnov,varinnov,nmax);
% the power density spectrum of an AR1 process with mean different from zero
resfft=fft(res); % Fast Fourier transform of the residuals
resPeriodo=abs(resfft.^2)./nmax;
% not the same result as per=periodogram(z), which is normalized by pi

resPeriodo=resPeriodo(1:nhalfm); % include all frequencies up to the fundamental fre-
quency
spectratio=resPeriodo./errPDS;

logL=-sum(log(errPDS))-sum(spectratio)-0.5*(log(2*pi*spectratio(1)))+0.5*spectratio(1);

% explanation:
% full equation would read as
```

```
% logL=-log(errPDS(1)-sum(log(errPDS(2:end)))

% -0.5*(log(2*pi*spectratio(1)))-0.5*spectratio(1)-sum(spectratio(2:end));


% Attention: different programming environments use different normalizations

% of the Fourier transform, see

% http://mathworld.wolfram.com/FourierTransform.html, eqns 15 and 16.

% to check the normalization used in Matlab:

% This is simple to check:

%

% x=random('Normal',0,1.5,100,1);

% s=(abs(fft(x))).2̂;

% mean(s)/length(x)

% var(x)


function [pds,fsp]=pdsAR1M(arcoef,meaninnov,varinnov,nmax)


% power density spectrum for non-zero mean AR1 process

% input:

% arcoef: lag-1 autoregressive parameter of the AR1 error model (size 1 x 1)

% varinnov: variance of the innovations of the error model(size 1 x 1)

% meaninnov: mean of the innovations of the error model (size 1 x 1)

% nmax: number of time steps of the time series
```

```
% output:
% pds of the AR1 process (size n x 1)
% fsp: spectral profile function of the AR1 process (size nmax/2 x 1)

if varinnov==0
   warning('You have a zero variance process')
end
if round(nmax/2)==nmax/2
   nhalfm=nmax/2;
else
   nhalfm=floor((nmax-1)/2);
end

fsp=fspAR1(arcoef,nhalfm); % equals one at all freq if arcoef =0
pds=fsp*varinnov;

% add the spike for the non-zero mean at 0th frequency
mproc=meaninnov/(1-arcoef); % process mean
pds(1)=pds(1)+nmax*(mproc^2);

function fspec=fspAR1(arcoef,nhalfm)

% Compute spectral profile function for AR1 process
```

```
% defined such that pds = varinnov * fsp;
% input:
% arcoef: lag-1 autoregressive parameter of the AR1 error model (size 1 x 1)
% nhalfm: number of frequencies(size 1 x 1)
% output:
% fspec: spectral profile function (size nhalfm x 1)
nmax=2*nhalfm;
if arcoef ~=0
% make sure you have the "is not equal" sign here
  x=([1:nhalfm])';
  x=2*pi/nmax*x; % Fourier frequencies
  cosar=cos(x);
  sinar=sin(x);
  Rar=cosar*arcoef;
  Iar=sinar*arcoef;
  far=(1-Rar).^2+Iar.^2;
  fspec=far.^(-1);
else
  fspec=ones(nhalfm,1);
end
```