A MatLab scripts for the polarization analysis

The polarization analysis described in section 2.3 was performed using the following MatLab scripts.

```
load('DATA');
%%% which contain  'dt'      the sampling time in seconds,
%%%                'ns'      number of samples,
%%%                'ntr'     number of traces,
%%%                'rcvx','rcvy','rcvz'  receiver coordinates,
%%%                'tracex','tracey','tracez'   the data

rectilinP=zeros(ns,ntr);
azimuthP=zeros(ns,ntr);
dipP=zeros(ns,ntr);

%%%% CAREFULLY
%%%% assign half-length of the moving time window
halfInterval=0.01; % in seconds
w= round(halfInterval/dt);

%%%% define start and end time for the analysis
trstart=w+1;
trend=ns-w;

%%% loop over traces
for h=1:ntr
    %%%% loop over moving time window
    for k=trstart:trend
        clear xp yp zp MP pP DP
        xP=tracex(k-w:k+w,h);
        yP=tracey(k-w:k+w,h);
        zP=tracez(k-w:k+w,h);
        MP=covariance(xP,yP,zP);
        [pP,DP] = eig(MP);
        %%% DP contains the eigenvalues of the covariance matrix, with
        %%% DP(3,3)>DP(2,2)>DP(1,1)
        %%% pP contains the eigenvectors, where the first column is the
        %%% eigenvector that corresponds to the smallest eigenvalue, the
        %%% second one to the intermedian eigenvalue and the third one to
        %%% the largest eigenvalue (this one shows the dominant particle motion)
        rectilinP(k,h)=1-((DP(1,1)+DP(2,2))/(2*DP(3,3)));
        azimuthP(k,h)=atan(pP(2,3)/pP(1,3))*180/pi;
        dipP(k,h)=atan(pP(3,3)/sqrt(pP(2,3)^2+pP(1,3)^2))*180/pi;
```

```
    end;
end;
```

This program calls the function $covariance(xP, yP, zP)$ which calculates the covariance matrix of the analyzed data.

```
function M=covariance(x,y,z)
%%%%%% Function to sum up the auto- and crossvariance of 3C-data
%%%%%% after Jurkevics, 1988, BSSA

M=zeros(3,3);

%%%% calculate mean values to remove possible DC shifts
mx=0;
my=0;
mz=0;
for k=1:length(x)
     mx=mx+x(k);
     my=my+y(k);
     mz=mz+z(k);
end
mx=mx/length(x);
my=my/length(y);
mz=mz/length(z);

for k=1:length(x)
    %%%%% remove possible mean shifts
    x(k)=x(k)-mx;
    y(k)=y(k)-my;
    z(k)=z(k)-mz;
    %%%%% calculate covariance matrix
    M(1,1)=M(1,1)+x(k)*x(k);
    M(1,2)=M(1,2)+x(k)*y(k);
    M(1,3)=M(1,3)+x(k)*z(k);
    M(2,1)=M(2,1)+y(k)*x(k);
    M(2,2)=M(2,2)+y(k)*y(k);
    M(2,3)=M(2,3)+y(k)*z(k);
    M(3,1)=M(3,1)+z(k)*x(k);
    M(3,2)=M(3,2)+z(k)*y(k);
    M(3,3)=M(3,3)+z(k)*z(k);

end

M=M./length(x);
```

All pictures that show the polarization analysis (calculated with the scipts above) in a time window moving over the data are plotted with the following routine.

```
%%%%%%% Plotting routines
figure;
imagesc(azimuthP(:,:));
set(gca,'ytick',(0:0.1/dt:ns));
yyy=num2cell([0:0.1:ns*dt]);
set(gca,'yticklabel',yyy');
xlabel('Receiver number', 'fontsize', 20);
ylabel('Time (s)', 'fontsize', 20);
c=colorbar; set(get(c,'Title'),'String',{'Azimuth','ccw from East'},'Fontsize',20);
set(c, 'fontsize', 20);
achse=get(gcf,'CurrentAxes');
set(achse, 'fontsize', 20);
```

```
figure;
imagesc(dipP(:,:));
set(gca,'ytick',(0:0.1/dt:ns));
yyy=num2cell([0:0.1:ns*dt]);
set(gca,'yticklabel',yyy');
xlabel('Receiver number', 'fontsize', 20);
ylabel('Time (s)', 'fontsize', 20);
c=colorbar; set(get(c,'Title'),'String',{'Dip'},'Fontsize',20);
set(c, 'fontsize', 20);
achse=get(gcf,'CurrentAxes');
set(achse, 'fontsize', 20);

figure;
imagesc(rectilinP( :,:));
colormap (flipud(gray));
set(gca,'ytick',(0:0.1/dt:ns));
yyy=num2cell([0:0.1:ns*dt]);
set(gca,'yticklabel',yyy');
xlabel('Receiver number', 'fontsize', 20);
ylabel('Time (s)', 'fontsize', 20);
c=colorbar; set(get(c,'Title'),'String',{'Rectilinearity'},'Fontsize',20);
set(c, 'fontsize', 20);
achse=get(gcf,'CurrentAxes');
set(achse, 'fontsize', 20);
```