```
% programmed by Pierre Henry&Sibel Bulkan, Oct 2019. Modified by JPM to sample↙
material derivative
% at the resolution of the 'finer' topographic grid -- although still
% interpolating from velocities calculated on the coarser grid. 2 Nov 2019
clear all
close all
load('/Users/sibel/Desktop/France_STSM/Matlab-France/gridded_calc/gridded_data.mat');


x0=515;
y0=-325;
xlims=[0.05 0.5];
ylims=[0.015 0.13];

itopo=4;
defstep=[num2str(5*(itopo-1)) '-' num2str(5*itopo) ' mm']

if itopo>1 & itopo<=14

i0=3*(itopo-1)+1
icount=0
for i = i0:3*itopo
    filename2=['modelraw_00' num2str(i,'%02i') '.txt'];
    ModelMatrix=importdata(filename2,',',3);
    ModelMatrix_3Row=ModelMatrix.data(:,3);
    ModelMatrix_4Row=ModelMatrix.data(:,4);
    if icount==0
        ModelMatrix_1Row=ModelMatrix.data(:,1);
        ModelMatrix_2Row=ModelMatrix.data(:,2);
        xaxisvalues = unique(ModelMatrix_1Row);
        yaxisvalues = unique(ModelMatrix_2Row);
        nxaxis = length(xaxisvalues);
        nyaxis = length(yaxisvalues);
        xx=repmat(xaxisvalues',nyaxis,1);
        yy=repmat(yaxisvalues,1,nxaxis);

        Dx=[ 0 0 0 0 0;
        1 1 0 -1 -1;
        0 0 0 0 0]
        Dy=[0 1/2 1 1/2 0;
        0 0 0 0 0;
        0 -1/2 -1 -1/2 0]
        dx=conv2(xx,Dx,'valid');
        dy=conv2(yy,Dy,'valid');

        ucM = reshape((ModelMatrix_3Row),[nyaxis nxaxis]);
        vcM = reshape((ModelMatrix_4Row),[nyaxis nxaxis]);
    else
        ucM = ucM + reshape((ModelMatrix_3Row),[nyaxis nxaxis]);
        vcM = vcM + reshape((ModelMatrix_4Row),[nyaxis nxaxis]);
    end
    icount=icount+1
end
ucM=ucM/icount;
vcM=vcM/icount;

gradux=conv2(ucM,Dx,'valid')./dx;
gradvx=conv2(vcM,Dx,'valid')./dx;
graduy=conv2(ucM,Dy,'valid')./dy;
gradvy=conv2(vcM,Dy,'valid')./dy;

%set dimension
```

```
[nrows,ncols] = size(gradux);
xn=xx(2:1+nrows,3:2+ncols);
yn=yy(2:1+nrows,3:2+ncols);

exy = 0.5*(graduy+gradvx);
eyx = exy;
exx = gradux;
eyy = gradvy;

%Angular velocity
cav = curl(xx,yy,ucM,vcM);

Arealchange = eyy+exx;
Shear1 = exx - eyy;
Invariant2 = ((exx .* eyy)-(exy.*eyx));
Jnvariant2 = (0.5*Shear1).^2 + 0.5*exy.^2;
Shearrate = graduy;
Angvel = (cav);


dt=15*60 % 15 minutes between topographic scans

% xb=1000*(xx-ucM*dt/2)+x0;
% yb=-1000*(yy-vcM*dt/2)+y0;
%
% xf=1000*(xx+ucM*dt/2)+x0;
% yf=-1000*(yy+vcM*dt/2)+y0;

% find backtrack points on coarse grid

xb=  1000*(xx-ucM*dt)+x0;
yb= -1000*(yy-vcM*dt)+y0;

xf=  1000*(xx)+x0;
yf= -1000*(yy)+y0;

zf=interp2(xgrid,ygrid,zgrid(:,:,itopo),xf,yf);
zb=interp2(xgrid,ygrid,zgrid(:,:,itopo-1),xb,yb);

% write output (Shear)
filenameout=['Shear' num2str(i) '.txt'];
dlmwrite(filenameout, Shearrate);

% backtrack on fine grid

% find grid points of fine grid inside the region of the coarse grid
% (remember they are offset by x0,y0)
min_xf = min(min(xf));
max_xf = max(max(xf));
min_yf = min(min(yf));
max_yf = max(max(yf));

xgrid_allpts = unique(xgrid); % list of all xgrid points
ygrid_allpts = unique(ygrid); % list of all ygrid points

% extract list of all xgrid points between min_xf and max_xf
xgrid_axis = ...
  xgrid_allpts((xgrid_allpts >= min_xf) & (xgrid_allpts <= max_xf));


% extract list of all ygrid points between min_yf and max_yf
ygrid_axis = ...
```

```
  ygrid_allpts((ygrid_allpts >= min_yf) & (ygrid_allpts <= max_yf));

% now make a list of all points in the mesh made up of xgrid_pts and
% ygrid_pts
xgrid_pts = repmat(xgrid_axis',length(ygrid_axis),1);
ygrid_pts = repmat(ygrid_axis,1,length(xgrid_axis));

zfgrid_pts = interp2(xgrid,ygrid,zgrid(:,:,itopo),xgrid_pts,ygrid_pts);



% redefine (xgrid_pts,ygrid_pts) to be in 'original' coordinate system for plots
xx_pts =  (xgrid_pts - x0)/1000;
yy_pts = -(ygrid_pts - y0)/1000;
% xf=1000*(xx)+x0;
% yf=-1000*(yy)+y0;

ucM_pts = interp2(xx,yy,ucM,xx_pts,yy_pts);
vcM_pts = interp2(xx,yy,vcM,xx_pts,yy_pts);

% xb_pts=  1000*(xx_pts+ucM_pts*dt)+x0;
% yb_pts= -1000*(yy_pts+vcM_pts*dt)+y0;
xb_pts=  1000*(xx_pts-ucM_pts*dt)+x0;
yb_pts= -1000*(yy_pts-vcM_pts*dt)+y0;

zbgrid_pts = interp2(xgrid,ygrid,zgrid(:,:,itopo-1),xb_pts,yb_pts);

zf=interp2(xgrid,ygrid,zgrid(:,:,itopo),xf,yf);
zb=interp2(xgrid,ygrid,zgrid(:,:,itopo-1),xb,yb);

% % CHECK PLOT comparing velocity components sampled on coarse and fine grids
%
% % 1st subplot
% figure(itopo+102);
% subplot(2,1,1);
% %surface(xgrid/1000,ygrid/1000,zgrid(:,:,itopo),'edgecolor','none');
% %surface(xx,yy,ucM,'edgecolor','none');
% surface(xx,yy,vcM,'edgecolor','none');
% shading(gca,'interp');
% %caxis([-1.5 1.5]);
% axis('equal');
% view(0,-90);
% colorbar;
% %title(['ucM (coarse mesh) ' defstep],'FontSize',14 )
% title(['vcM (coarse mesh) ' defstep],'FontSize',14 )
% xlim(xlims)
% ylim(ylims)
%
%
% % 2nd subplot
% figure(itopo+102);
% subplot(2,1,2);
% %surface(xgrid/1000,ygrid/1000,zgrid(:,:,itopo),'edgecolor','none');
% %surface(xx_pts,yy_pts,ucM_pts,'edgecolor','none');
% surface(xx_pts,yy_pts,vcM_pts,'edgecolor','none');
% shading(gca,'interp');
% %caxis([-1.5 1.5]);
% axis('equal');
% view(0,-90);
% colorbar;
% %title(['ucM (fine mesh) ' defstep],'FontSize',14 )
% title(['vcM (fine mesh) ' defstep],'FontSize',14 )
```

```
% xlim(xlims)
% ylim(ylims)

% CHECK PLOT comparing topography sampled on coarse and fine grids

% 1st subplot
figure(itopo+101);
subplot(2,1,1);
%surface(xgrid/1000,ygrid/1000,zgrid(:,:,itopo),'edgecolor','none');
%surface(xx,yy,zf,'edgecolor','none');
%surface(xx,yy,zb,'edgecolor','none');
surface(xx,yy,(zf-zb),'edgecolor','none');
shading(gca,'interp');
%caxis([-1.5 1.5]);
axis('equal');
view(0,-90);
colorbar;
colormap parula
title(['Material Derivative of Topography (coarse mesh) ' defstep],'FontSize',14 )
xlim(xlims)
ylim(ylims)



% 2nd subplot
figure(itopo+101);
subplot(2,1,2);
%surface(xgrid/1000,ygrid/1000,zgrid(:,:,itopo),'edgecolor','none');
%surface(xx_pts,yy_pts,zfgrid_pts,'edgecolor','none');
%surface(xx_pts,yy_pts,zbgrid_pts,'edgecolor','none');
surface(xx_pts,yy_pts,(zfgrid_pts-zbgrid_pts),'edgecolor','none');
shading(gca,'interp');
%caxis([-1.5 1.5]);
axis('equal');
view(0,-90);
colorbar;
colormap parula
title(['Material Derivative of Topography (fine mesh) ' defstep],'FontSize',14 )
xlim(xlims)
ylim(ylims)


cl = caxis;  % save caxis for check


%-----------
% Make Plots
%-----------
% 1st subplot
figure(itopo+100);
subplot(4,1,1);
surface(xn,yn,-Shearrate,'edgecolor','none');
shading(gca,'interp');
colormap(jet);
axis('equal');
caxis([0 4e-4]);
view(0,-90);
colorbar;
colormap parula;
title(['Shear Rate ' defstep,],'FontSize',14 );
xlim(xlims)
ylim(ylims)
```

```
% 2nd subplot
figure(itopo+100);
subplot(4,1,2);
surface(xx,yy,Angvel,'edgecolor','none');
shading(gca,'interp');
caxis([0 2e-4]);
axis('equal');
view(0,-90);
colorbar;
colormap parula;
title(['Angular velocity ' defstep],'FontSize',14);
xlim(xlims)
ylim(ylims)

% 3rd subplot
figure(itopo+100);
subplot(4,1,3);
surface(xn,yn,Arealchange,'edgecolor','none');
shading(gca,'interp');
caxis([-1e-4 1e-4]);
axis('equal');
view(0,-90);
colorbar;
colormap parula;
title(['Areal Strain ' defstep],'FontSize',14 )
xlim(xlims)
ylim(ylims)


% 4th subplot (hi-res)
figure(itopo+100);
subplot(4,1,4);
%surface(xgrid/1000,ygrid/1000,zgrid(:,:,itopo),'edgecolor','none');
%surface(xx,yy,(zf-zb),'edgecolor','none'); % mat. der. on low-res mesh
surface(xx_pts,yy_pts,(zfgrid_pts-zbgrid_pts),'edgecolor','none');% mat. der. on hi-↙
res mesh
shading(gca,'interp');
%caxis([-1.5 1.5]);
%caxis('auto')
caxis(cl)
colormap parula;
axis('equal');
view(0,-90);
colorbar;
title(['Topographic change ' defstep],'FontSize',14 ); % material derivative plot
xlim(xlims)
ylim(ylims)

end
```