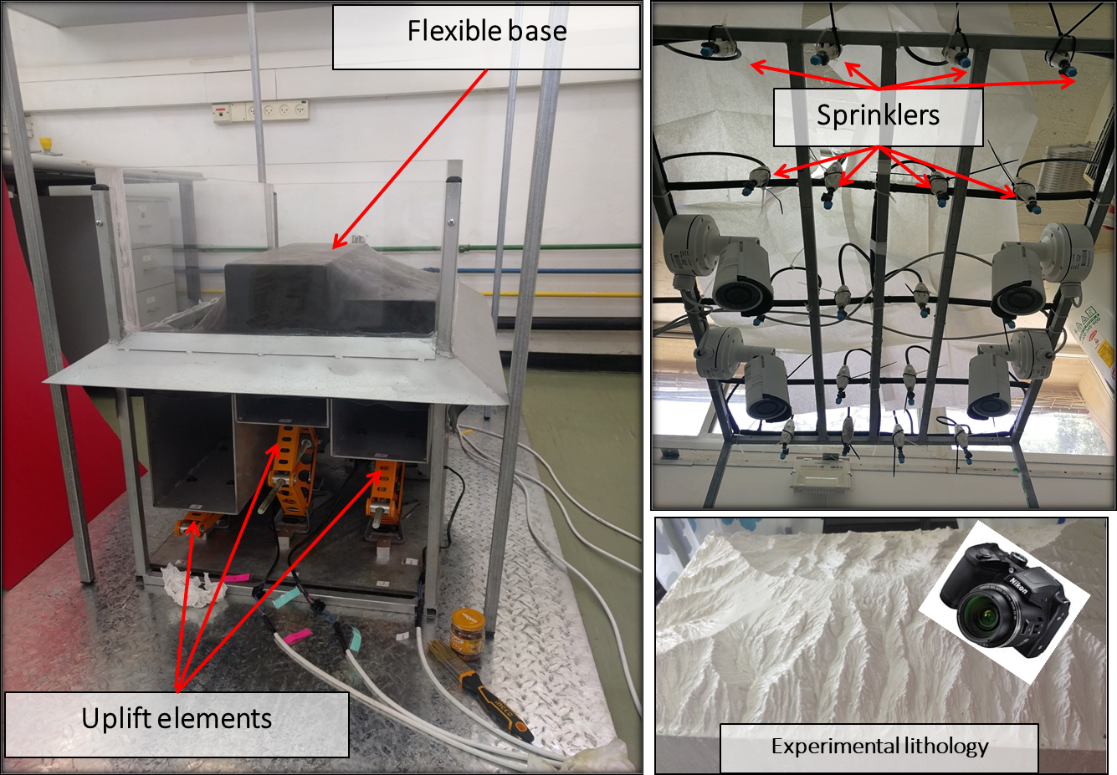
**Use of Chi analysis in experimental landscapes (DULAB)**

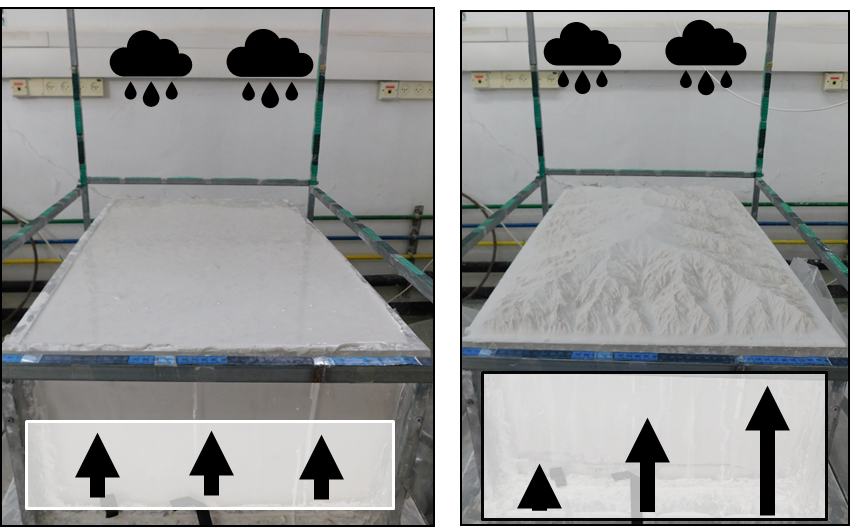
**Settings**

In DULAB (Differential Uplift Landscape-evolution Box), we create experimental landscape by applying tectonic uplift and precipitation in a controlled laboratory environment, in order to test the fluvial response of differential tectonic uplift on landscape evolution. Our system consists of the following:

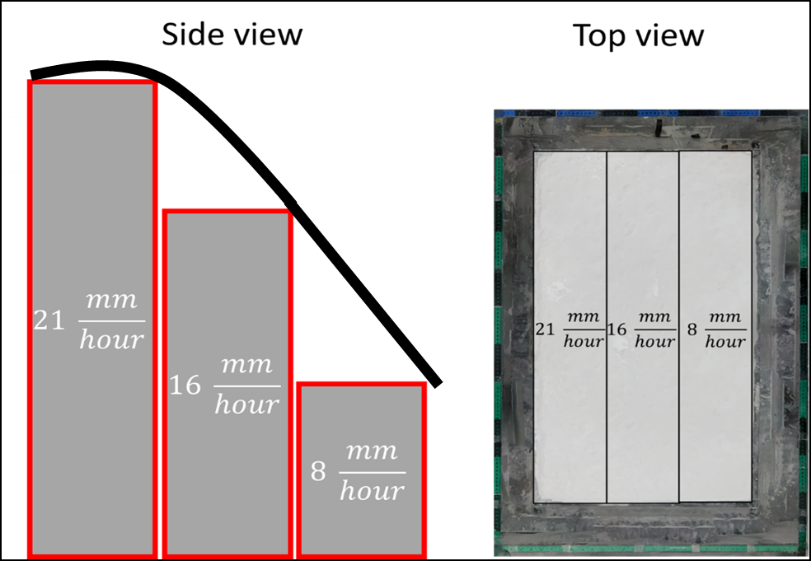


In all experiments, the experimental lithology is uniform (silica paste) and precipitation is constant.

The experimental scheme of the experiment presented here consists of two distinct stages: (1) uniform uplift for 13 hours in which the drainage network forms and a main divide is established, and (2) differential uplift for 6 hours during which the landscape responds to the spatial gradient in uplift.

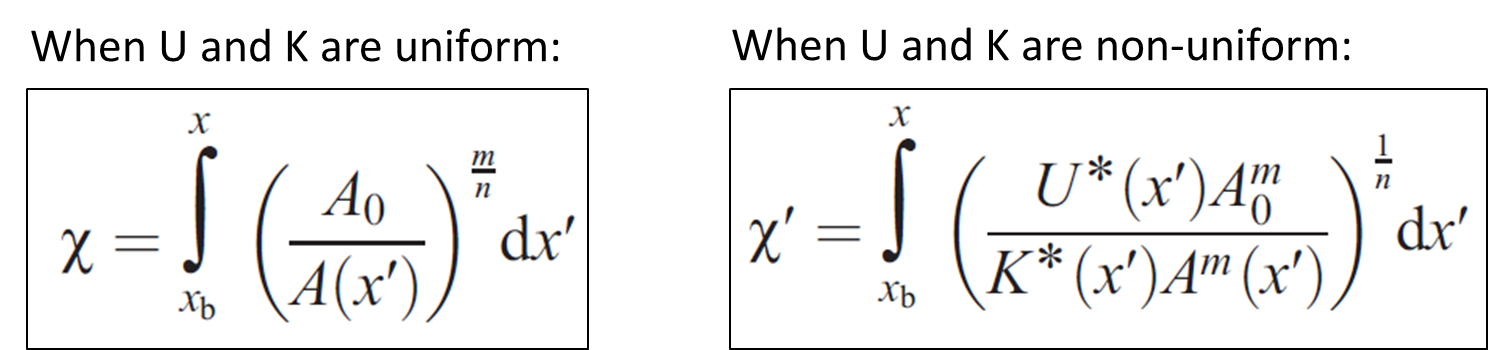


The spatial pattern of uplift is as follows:



**Chi analysis**

One of many ways to learn about the stability of divides and basin geometry is by looking at across divide contrasts in the χ metric (Willett et al. 2014).



Here I show how to perform χ analysis on a DEM from the differential uplift stage of the experiment using TopoToolBox and some functions modified from it.

First, let’s visualize the landscape and the drainage network by deriving flow directions using D8 flow routing algorithm:

DEM = GRIDobj('Diff\_EXP\_17hr.tif') % load DEM.

DEM.Z(DEM.Z<-9998)=NaN; % Convert all values outside the borders of the box into NaN

FD = FLOWobj(DEM,'preprocess','carve');

ST = STREAMobj(FD,'minarea',5000);

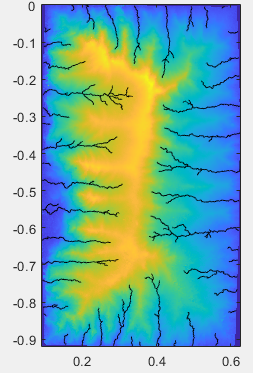
figure

imagesc(DEM)

hold on

plot(ST,'color','k')

hold off



Next, we derive the drainage divide network and calculate its attributes (divide order and distance) and plot it with line-width proportional to divide distance (for more detailed information - [https://topotoolbox.wordpress.com/2020/05/29/introduction-to-divideobj](https://topotoolbox.wordpress.com/2020/05/29/introduction-to-divideobj/)/):

D = DIVIDEobj(FD,ST);

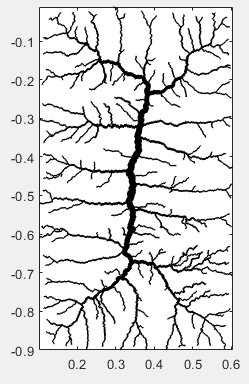
D2 = cleanedges(D,FD);

D = divorder(D2,'topo');

plot(D,'limit',[0.02 inf],'color','k')

axis image

box on



As stated before, lithology and precipitation are uniform throughout the experiment, so we assume that K=1 when calculating χ’. Since U is not uniform and varies according to the spatial gradient shown above, we need to take that into account when calculating χ’. To do so, we generate a new GRIDobj named ‘U’ that contains values of uplift rates at each pixel. Note: Since we assume that K=1 in this case, ‘U’ can be regarded as U over K (U/K).

Now let’s plot the ‘U’ grid to see the spatial gradient of uplift.

Ufast = 0.021; % Fast uplift rate is 0.021 m/hr.

Umid = 0.016; % Medium uplift rate is 0.016 m/hr.

Uslow = 0.008; % Slow uplift rate is 0.008 m/hr.

U = DEM;

U.Z = NaN(DEM.size);

columns = U.size(2);

Third = floor(columns/3); % represents 1/3 of the width of the box.

U.Z(:,1:Third) = Ufast;

for i=Third+1:2\*Third

U.Z(:,i) = ((i-Third)/(Third))\*(Umid-Ufast)+Ufast;

end

for i=2\*Third+1:columns

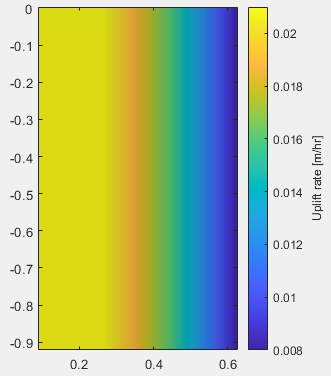
U.Z(:,i) = ((i-2\*Third)/(Third))\*(Uslow-Umid)+Umid;

end

imageschs(U)

hc = colorbar;

hc.Label.String = 'Uplift rate [m/hr]'



Now, we can calculate χ’ and take into account the differential uplift pattern (Willett et al. 2014), using the function ‘ChiPrimeTransform’ (modified from the function ‘chitransform’ in TopoToolbox) that generates a node-attribute list of χ’ values along the drainage network. Note: DULAB channels are characterized by low concavity, therefore we use m/n value of 0.15. In natural landscapes, this might not be the case.

A = flowacc(FD);

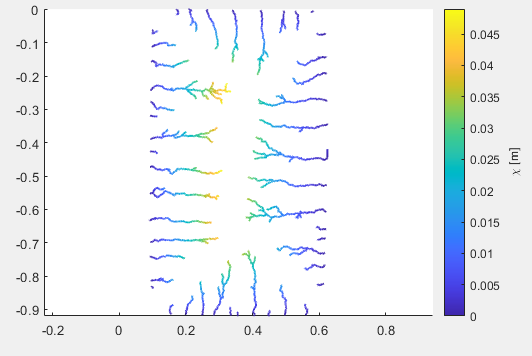
chi = ChiPrimeTransform(ST,A,'mn',0.15, 'a0',10000,'UoverK', U);

plotc(ST,chi)

axis equal

hc = colorbar;

hc.Label.String = '\chi [m]'



At this point we can already see how χ’ values are generally higher on the higher uplift rate side. Now, let’s look more closely at across-divide contrasts along the main divide. First, we use the function ‘ChiAtNearestStream’ (modified from the function ‘vertdistance2stream’ in TopoToolbox) to generate a GRIDobj that contains χ’ values of the nearest stream at each pixel. Next, we use the function ‘ChiAsymmetry’ (modified from the function ‘asymmetry’ in TopoToolbox) that computes the across divide contrast in χ and yields a mapping structure with the divide network. Finally, the figure shows the divide network on top of a hillshade image, with the divides colored by the contrast in χ’ values, and with arrows that point to the direction of expected divide migration.

C = ChiAtNearestStream(FD,ST,DEM,chi);

C.Z(isinf(C.Z)) = nan;

[MS,S] = ChiAsymmetry(D,C);

for i = 1 : length(S)

S(i).length = max(getdistance(S(i).x,S(i).y));

end

figure

imageschs(DEM,[],'colormap',[.9 .9 .9],'colorbar',false);

hold on

plotc(D,vertcat(S.rho),'caxis',[0 max(vertcat(S.rho))],'limit',[0.05 inf])

colormap(gca,flipud(pink))

axis image

hc = colorbar;

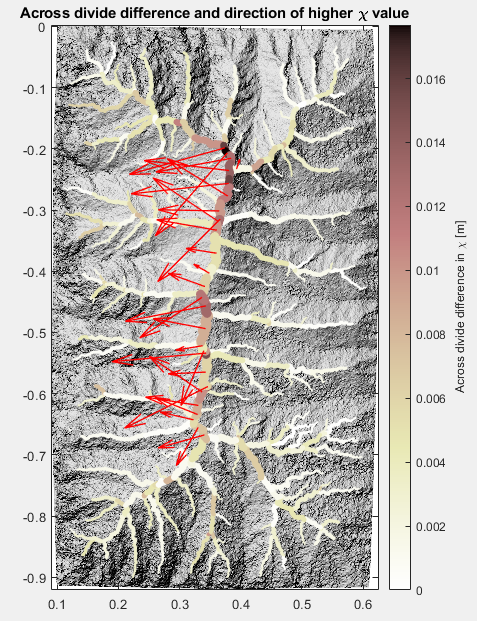
hc.Label.String = 'Across divide difference in \chi [m]';

ix = [MS.order]>15;

quiver([MS(ix).X],[MS(ix).Y],[MS(ix).u],[MS(ix).v],2,...

'color','r','linewidth',1)

title('Across divide difference and direction of higher \chi value')



Here I chose to plot arrows on divide segments from order 15 and above, in order to show the expected divide migration direction along the main divide only. As can be clearly seen, the arrows pointing to the left suggest a prediction of divide migration to the left, and this is exactly what we observe as the experiment continues – the main divide migrates toward the higher uplift rate side, as basins shrink on the left size and lengthen on the right size.