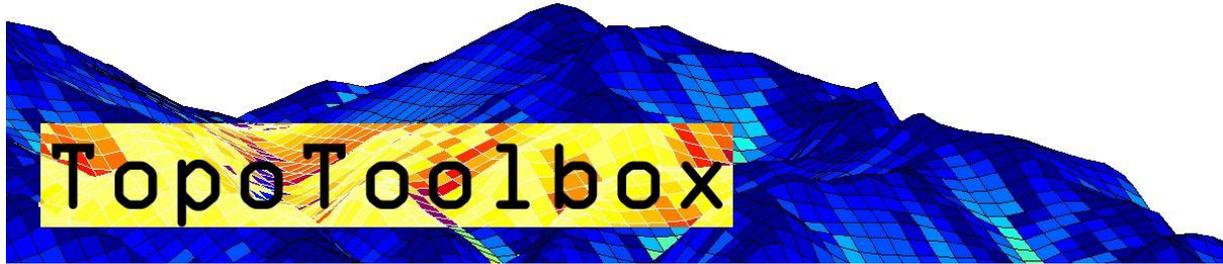


## User Guide to TopoToolbox - Plotting ksn-values



Background.....	1
Load and generate required data.....	2
Calculating the concavity index.....	4
Calculate and plot ksn values .....	4
Aggregate ksn values in stream segments .....	5
Plot stream segments.....	5
Export as shapefile .....	6

### Background

Stream profiles carry information on tectonic and/or climatic perturbations in the form of knickpoints migrating in upstream direction. Plotting ksn-values provides a means to quantify deviations from the logarithmic channel length profile and thus enables to detect and visualize these locations in the stream network.

The basic assumption behind ksn-values is that the stream power model of sediment transport/incision applies, e.g., sediment transport or incision are functions of the energy expenditure of water when moving downstream. In a landscape evolution context, the stream power model is written in following (simplified) form

$$\frac{dz}{dt} = U - kA^m \left( \frac{dz}{dx} \right)^n$$

where  $U$  refers to the uplift rate,  $A$  is the upslope area,  $\frac{dz}{dx}$  is the slope and  $k$ ,  $m$  and  $n$  are coefficients. In steady state the processes uplift and incision are in equilibrium such that  $\frac{dz}{dt} = 0$  at each point along the profile. Hence, the equation is rearranged

$$\left( \frac{dz}{dx} \right)^n = \frac{U}{kA^m}$$

$$\frac{dz}{dx} = \left( \frac{U}{k} \right)^{1/n} A^{-n/m}$$

The terms are rewritten so that

$$S = k_{sn} A^{-\theta}$$

and

$$k_{sn} = \frac{S}{A^{-\theta}}$$

where  $k_{sn}$  refers to the normalized channel steepness and  $\theta$  to the channel concavity.  $\theta$  ranges between 0.3 and 0.8 and often takes on the value 0.45. It can be determined either based on slope-area plots (see function `slopearea`) or  $\chi$ -plots (see function `chiplot`).

## Load and generate required data

```
DEM = GRIDobj('srtm_bigtujunga30m_utm11.tif');
FD = FLOWobj(DEM, 'preprocess', 'carve');
A = flowacc(FD);
```

The quality of the DEM is relatively bad, so we correct it by imposing a downward minimum gradient.

```
DEM = imposemin(FD, DEM, 0.0001);
```

Subsequently, we extract a stream network by thresholding flow accumulation. In this example, we simply assume that channels occupy cells that have a minimum upslope area of 1000 pixels.

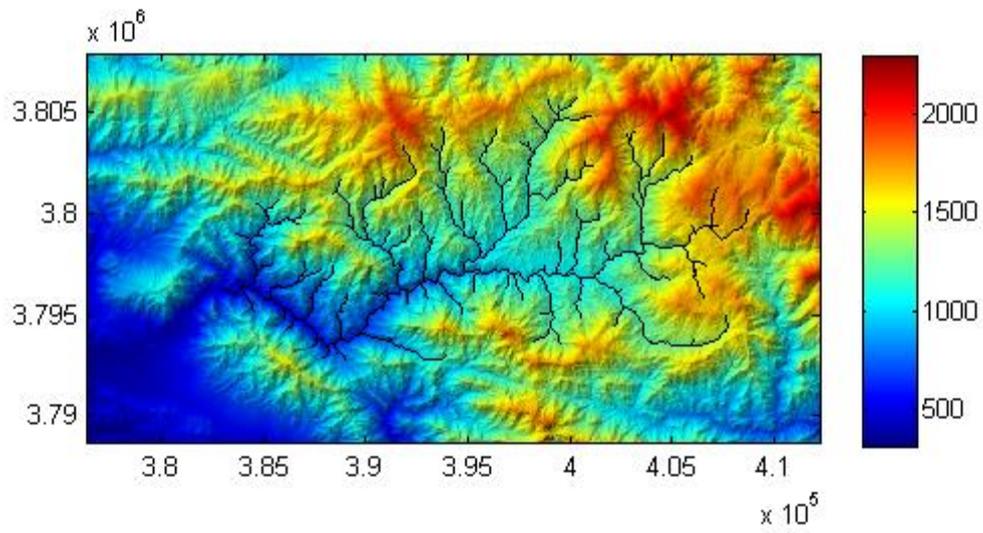
```
S = STREAMobj(FD, 'minarea', 1000);
S = klargestconncomps(S, 1);
```

We modify the stream network somewhat so that we avoid to include the alluvial areas in the most downstream portion of the profile.

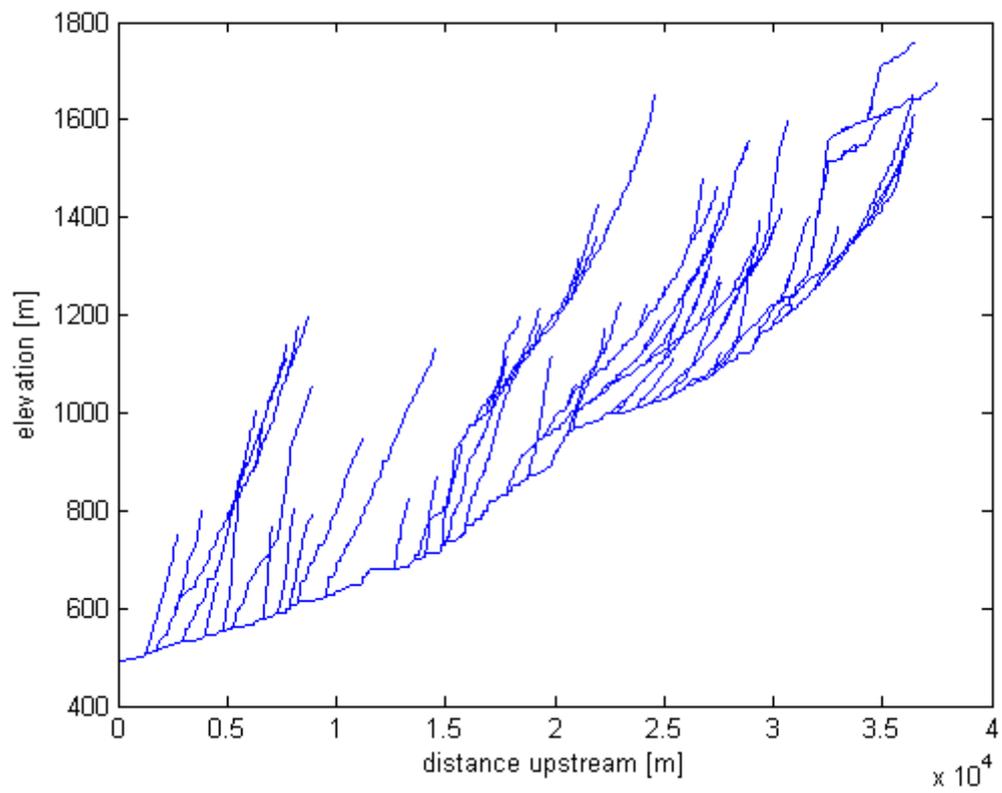
```
S = modify(S, 'distance', 10000);
```

In order to remove small stream sections we again extract the largest connected component of the network.

```
S = klargestconncomps(S, 1);
imageschs(DEM);
hold on
plot(S, 'k')
hold off
```



```
plotdz(S,DEM)
```



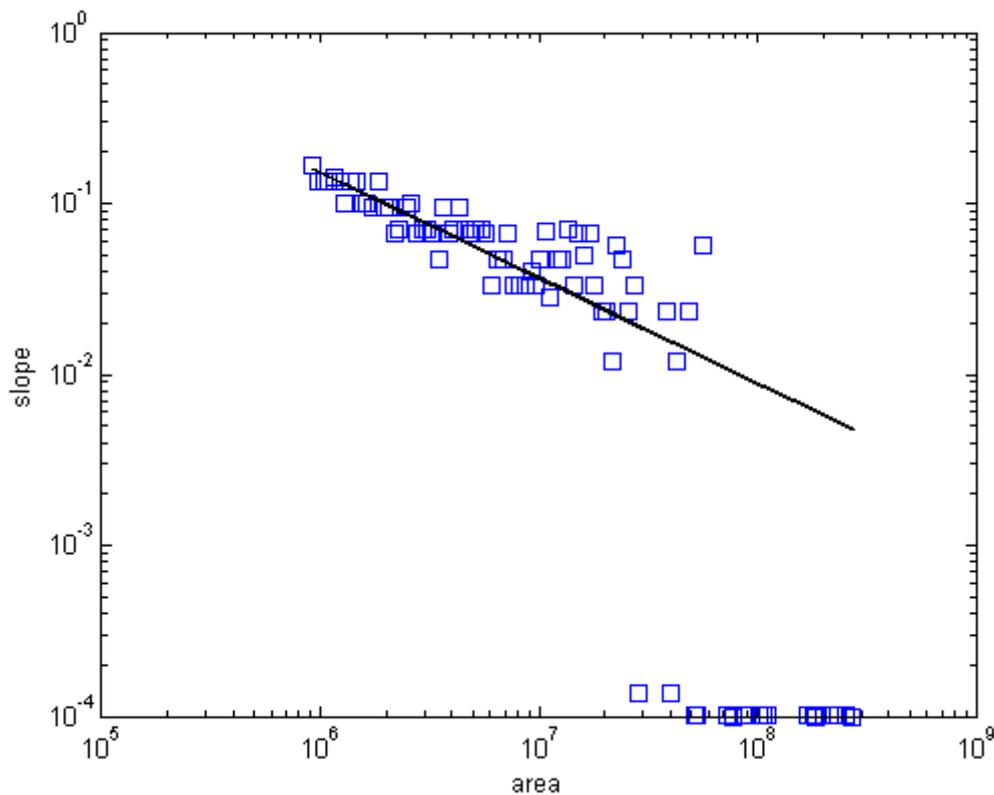
## Calculating the concavity index

After extracting the channel network of interest, we calculate the concavity index  $\theta$  using the slopearea function. **Note that** slopearea returns the concavity index  $\theta$  including its negative sign!

```
STATS = slopearea(S,DEM,A)
```

```
Exiting: Maximum number of function evaluations has been exceeded  
- increase MaxFunEvals option.  
Current function value: 0.022585
```

```
STATS =  
    a: [84x1 double]  
    g: [84x1 double]  
hPoints: 174.0205  
    ks: 743.7756  
    theta: -0.6156  
    hLine: 178.0057
```



## Calculate and plot ksn values

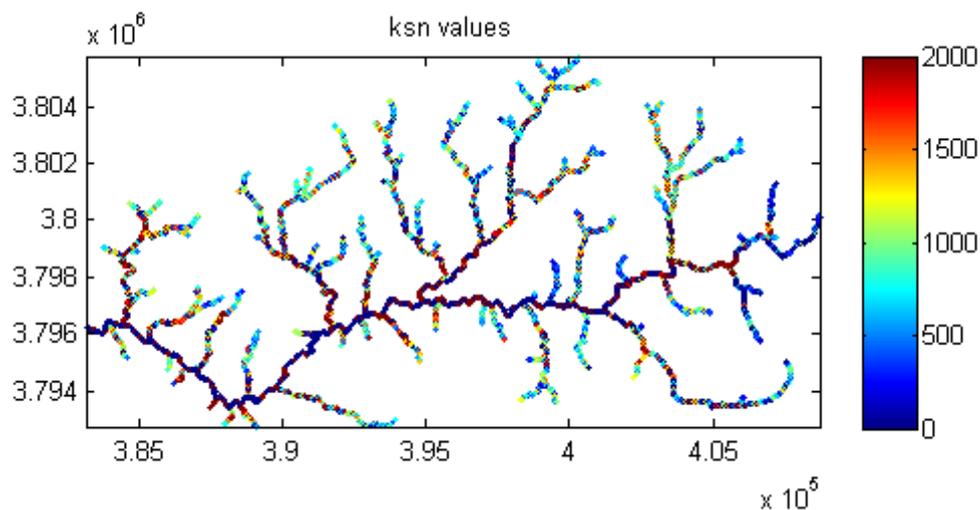
Subsequently, we calculate the ksn-values for the entire grid. While this won't make sense for hillslope sections, our aim is to generate a grid that we can later reference with the stream network (STREAMObj).

```

G = gradient8(DEM);
KSN = G./(A.*(A.cellsize^2)).^ASTATS.theta;

[x,y,ksn] = STREAMobj2XY(S,KSN);
scatter(x,y,5,ksn,'filled')
caxis([0 2000])
title('ksn values')
box on
colorbar
axis image

```



### Aggregate ksn values in stream segments

Note that in the above figure each channel location in the grid is plotted. There are two problems arising from doing so. First, owing to the high point density in the xy plane some of the points and their color may not be visible due to overlapping. Second, owing to the poor quality of the DEM the ksn-values are strongly variable at short distances. Note the large variability in the main river trunk! To avoid both problems, ksn values should be generalized along stream segments. This is achieved with the function `STREAMobj2mapstruct`. Here we use a stream segment length of 1 km (1000 m).

```

MS = STREAMobj2mapstruct(S,'seglength',1000,'attributes',...
    {'ksn' KSN @mean ...
    'uparea' (A.*(A.cellsize^2)) @mean ...
    'gradient' G @mean});

```

### Plot stream segments

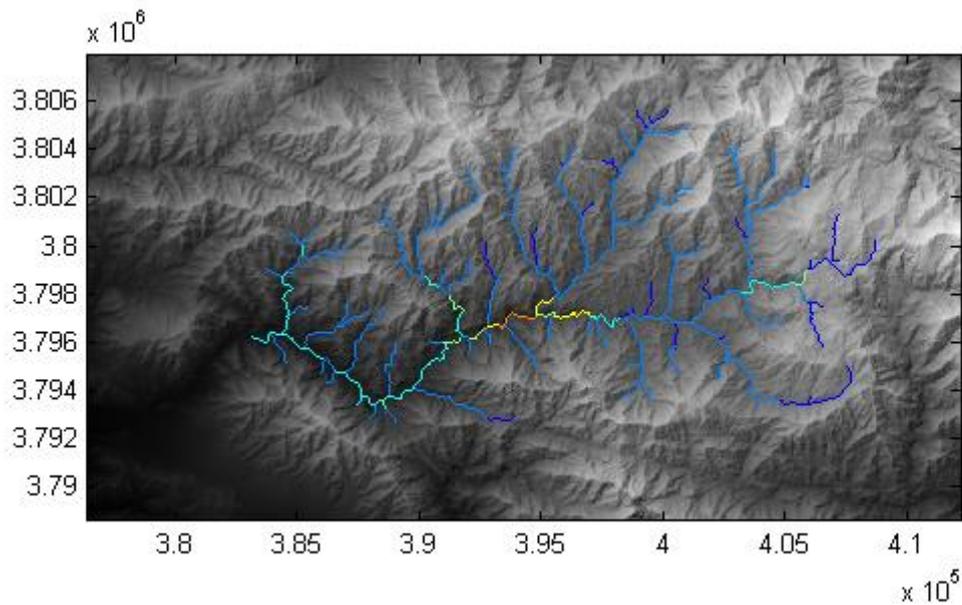
MS is a mapstruct that you can plot using `mapshow` and `makesymbolspec`. Both functions are available with the mapping toolbox. We use a coloring based on percentiles of the data.

```

symbolspec = makesymbolspec('line',...
    {'ksn' [min([MS.ksn]) max([MS.ksn])] 'color' jet(6)});

```

```
imageschs(DEM,DEM,'colormap',gray,'colorbar',false);  
mapshow(MS,'SymbolSpec',symbolSpec);
```



### Export as shapefile

Alternatively, you may want to choose another software to visualize the stream network together with ksn values. In order to do so, you will need the function `shapewrite` available with the mapping toolbox (in case you don't have access to the toolbox, check the Mathworks fileexchange for functions with same functionality).

```
shapewrite(MS,'ksn_testshape.shp');
```