2D Roughness Parameter Algorithm (plot_2dParams_roughness_GaussianFilt_alt.m)

Kevin Fan

1) Specify parameters:

a) Dimensions and horizontal resolution of DTM (assumed to be symmetrical in x/y directions)
b) 2D cell radius (analogous to the half-width of a 1D profile)
c) Maximum 2D lag distance for variogram best-fitting line calculation of RMS Slope and Hurst Exponent
d) Sigma (ie. # of standard deviations) for 2D high-pass Gaussian filter (proportional to the cut-off wavelength, at which 50% of signal is transmitted and 50% attenuated)

2) Gaussian Filter:

For each choice of Gaussian filter (as specified by sigma), the DFT of the Gaussian kernel is point-wise multiplied by the DFT of the surface heights (ie. DEM), and then inverse Fourier Transformed back to the Space Domain (ie. Frequency-domain approach). This splits up the DEM into a 2D waviness profile and a 2D roughness profile, representing the long-wavelength and short-wavelength components, respectively.
[NB: This is equivalent to convolution of the DEM with the Gaussian Kernel in the Space Domain (ie. Space-domain approach)]

3) Calculate Variogram:

For each pixel in the DEM, the RMS Slope and Hurst exponent are calculated by constructing a variogram (ie. a 2000x2000 DEM will require 4 million variograms to calculate its roughness parameters). Thus, for each pixel, the following process occurs:

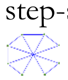
a) A 2D cell is 'partitioned' out of the larger DEM, stored in the array 'prison' in the script (101x101 elements in this case)

b) For the mini-cell within this larger cell (9x9 elements in this case), all combinations (in the combinatorics sense) of lag positions (p,q) [NB: where (3,1) denotes the position 3 rows above and 1 column right with respect to the centre] relative to the centre are determined, and the diagonals are supplemented (ie. For this case, these are (0,1), (0,2), (0,3), (0,4), (1,2), (1,3), (1,4), (2,3), (2,4), (3,4), with diagonals (1,1), (2,2), (3,3), (4,4)), making for 14 combinations in this case.

c) For each of these combinations, the RMS deviation (ie. square root of the Allen Variance) is calculated for each 2D step-size, using the same implicit low-pass filter as was applied in the 1D case), based on 1 of 3 configurations that arise:

i) **X** (diagonals: 4-connected) → (eg. p=q=1; p=q=2; …) The heights of 4 'arms' are extracted (ie. 2 diagonals, meeting at the cell centre) for each step-size, and the squared differences between adjacent points on each arm are calculated.

ii) **+** (cross: 4-connected) → (eg. p=0, q=1; p=0, q=2; …) The heights of 4 heights of 4 'arms' are extracted (ie. a horizontal and vertical band, meeting at the cell centre) for each step-size, and the squared differences between adjacent points on each arm are calculated.

iii)  (spider: 8-connected) → (eg. p=2, q=1; p=3, q=1; …) The heights of 8 'arms' are extracted (ie. a vertically-elongated X superimposed with a horizontally-elongated X) for each step-size, and the squared differences between adjacent points on each arm are calculated.

d) The squared differences *dif* from all 'arms' are combined, and the RMS Deviation (ie. square-root of the Allan Variance) is calculated as **sqrt[sum(all elements in dif) / (# elements in dif)]** for the particular combination (p,q), with associated 2D step-size **sqrt[(p*resolution)^2 + (q*resolution^2)]**.

e) The Allen deviation and associated step-sizes are calculated from each of the combinations are combined into 1 vector each, allowing for plotting. The function *polyfit* is applied to calculate the least-squares linear fit* to the line log(step_size**) vs. log(RMS deviation), assuming the slope to be the Hurst exponent and the y-intercept to be the y-intercept. Then, associated errors are calculated: the standard error in the calculated polynomial coefficients (denoting the 50% confidence interval), and the coefficient of determination (ie. r^2).

**Alternatively, lscov for a weighted linear least-squares fit, in which a 'weight' is assigned to each point on a particular variogram, where this weight is proportional to the number of point pairs used to calculate the RMS Deviation. This meshes with the intuition that, for instance, the RMS deviation for the case p=q=1 should have many more point pairs than the case p=q=4, and thus bear proportionately more weight during the calculation of the least squares fit.*

***step_size = **Δ**d = sqrt((**Δ**x)^2 + (**Δ**y)^2), ie. the Euclidean distance, where spherical curvature effects are assumed to be negligible; alternatively, the Haversine distance could be employed should this assumption be too coarse*

<u>4) Save Arrays:</u>

-Having calculated the RMS Slope and error in such, the Hurst exponent and error in such, and r^2, for each and every point of the DEM (with the exception of the edges), save these 5 arrays for later plotting.