**2D Roughness Parameter Algorithm**

**(plot\_2dParams\_roughness\_GaussianFilt\_interleaved.m)**

Kevin Fan

**1) Specify parameters:**

 a) Dimensions and horizontal resolution of DTM (latter assumed to be symmetrical in

 x/y directions)

 b) 2D cell length (eg. 10 produces local cells of size 10x10)

 c) Maximum 2D lag distance (eg. 4 pixels) 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: Mathematically, this is equivalent to convolution of the DEM with the Gaussian Kernel in the Spatial Domain (ie. Space-domain approach)]. Specifically, it is a high-pass Gaussian Filter that is employed, producing the ‘roughness surface’ defined in metrology, in which only the high-frequency components remain. It should be noted, however, that by the nature of the Gaussian filter, the frequency cut-off is smooth, not abrupt, such that even if the cut-off wavelength were to be 10cm, spatial wavelengths of 20cm would still be retained (albeit much less so than the 50% of information retained at the 10cm wavelength; and much less so for spatial wavelengths of 30cm, etc.). The precise amount of spatial information transmitted varies as a function of wavelength, and is defined by the Gaussian Filter’s transfer function.

**3) Calculate Variogram:**

For each pixel in the DEM, the RMS Slope and Hurst exponent are calculated by constructing a variogram (eg. 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). Such a cell may also be referred to as a ‘local cell’, centred about each individual

 pixel in the DEM (excepting those close to the DEM edges).

 b) For the mini-cell within this larger cell (10x10 in this example), all combinations (in the combinatorics

 sense) of lag vectors (*p*,*q*) relative to the centre are determined\* and the diagonals are supplemented,

 where p and q are integer pixel lag distances in the y and x directions, respectively. For this case, in which

 the maximum lag distance is 4 pixels, these combinations are (0,1), (0,2), (0,3), (0,4); (1,1), (1,2), (1,3),

 (1,4); (2,1), (2,2), (2,3), (2,4); (3,1), (3,2), (3,3), (3,4); (4,1), (4,2), (4,3), (4,4), resulting in 24 combinations.

 Note that the zero vector (0,0) is skipped, as we are concerned with comparing pixel values.

\*NB: where (3,1) denotes the position 3 rows above and 1 column right with respect to the centre

c) For each of these combinations of lag vector direction/magnitude, all pairs of points within the local cell

spatially separated by this lag vector are determined. For each of these pairs, the difference is calculated and

squared.

d) These squared differences *dif^2* (a vector) are summed, and the RMS Deviation (ie. square-root of the Allan Variance) is calculated as

**sqrt[ sum(*dif^2*) / (# of point pairs) ]**

for the particular combination (p,q), with associated 2D step-size

**sqrt[ (*p* \* pixel resolution)^2 + (*q* \* pixel resolution^2) ]**

e) The RMS Deviation and scalar step-sizes\*\* associated with them are calculated for each and every

combination of lag vector, then plotted as a variogram on a log-log plot. The function *polyfit* is applied to

calculate the least-squares linear fit\*\*\* for this log-log plot, assuming the slope to be the Hurst exponent and

the RMS Slope to be the y-intercept. Formally, a weighted-least squares regression is performed (utilizing

the MATLAB function lscov in conjunction with polyfit), where the weighting accounts for the fact that,

combinatorically, the calculated RMS Deviation for small lag vectors (eg. (p,q) = (1,1)) results from the

combination of many more point differences than that which is calculated for large lag vectors (eg. (p,q) =

(4,4))\*\*\*\*. The weighting function is proportional, such that a point on the variogram calculated via the

combination of 100 point pairs would be weighted twice as much as another point on the variogram

calculated via the combination of only 50 point pairs.

*\*\* 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*

*\*\*\* 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.*

\*\*\*\* This discrepancy arises due to the edge effects. For instance, in a 3x3 local cell, there are 4 combinations of pair points separated by the lag vector (1,1); meanwhile, in the same cell, there exists only 1 combination of pair points separated by the lag vectors (2,2).

f) Finally, associated errors are calculated: the standard error in each of the calculated polynomial

coefficients (ie. corresponding to the RMS Slope and Hurst Exponent), denoting the 50% confidence

interval; and the coefficient of determination (ie. r^2).

The process is then repeated for every pixel in the DEM, excepting those on its edge, for which the local cell would go beyond the borders of the DEM.

**4) Save Arrays:**

-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.