Paleolandscape Reconstruction: **Cost-distance estimation method.**

Step 1: Display the DEM and zoom into the area you want to work in. Set the region extents to match the display, and make sure the raster resolution is set correctly (ie. At the same resolution as the input DEM)

Step 1: Run *r.watershed* using modern DEM. Create output option *basin* named “BASINS” at a large scale (with option *threshold*) to define the watershed boundaries.

Step 2Use *r.mask* with the resulting basins map to turn the GIS MASK on and tell GRASS that we only want to work within the watershed boundaries.

Step 3: Run *r.watershed* again at smaller scale (with option *threshold*) with output option *stream* named “STREAMS”.

Step 4: Run *r.slope.aspect* with output option *slope* named “SLOPE”.

Step 5: Use *r.mapcalc* to derive the square of slope [mapcalc syntax: “exp(SLOPE, 2)”]. Name the *output* map “FRICT”

Step 6: Run *r.cos*t to create a cost surface that will identify terraces. Use the map “FRICT” as *input* map, and the map “STREAMS” as the *start_rast* map. Name the *output* map “COST_SURF”.

Step 7: Use *r.mapcal*c to clip out unwanted areas from the DEM (ie. the “newer” areas of the landscape) by using different cutoff points in the cost surface map to set the clipping threshold. The mapcalc syntax for this looks like: “if(COST_SURF <= *x*, null(), DEM)” where *x* is the cutoff threshold. Query the cost surface map to determine this threshold, and experiment with various values. Name the *output* map “CLIPPED_DEM”.

Step 8: Use *r.random* to randomly sample some points for interpolation. Set *input* to “CLIPPED_DEM”, *n* to “20%”, *vector_output* to “INTERP_POINTS”, and check the box that says “Generate vector points as 3D points”.

Step 9: Use *v.voronoi* to create a “landscape skeleton” from these points to avoid segmentation faults during the interpolation. Set *input* to “INTERP_POINTS”, *output* to “INTERP_VORONOI”, and check the box that says “Output tessellation as a graph (lines), not areas”.

Step 10: It’s time to interpolate the paleosurface! Use *v.surf.rst* for this. Set *input* to “INTERP_VORONOI”, *elev* to “PALEOSURFACE”, *maskmap* to “MASK”, and *zcolumn* to “value”. Play around with the *tension* and *smoothing* values (start with 40 and 0.1 respectively) as well as the *npmin, segmax, dmin*, and *dmax* values. Refer to Cebecauer et. al (2002) or Neteler and Mitasova (2008) for ideas about what to set these values for.

Erosion and Human Impacts: RUSLE method.

Step 1: Figure out how much sediment has been removed in the reconstructed areas of the landscape. Subtract the current elevation (DEM) from the modeled paleosurface in the areas that we filled in. Use *r.mapcalc* with the following syntax: "if(COST_SURF <= *x*, (PALEOSURFACE – MODERNDEM), null())". Name the *output* map "ERODED_SED", where *x* is the same cutoff value used to create the clipped DEM.

Step 2: Use *r.watershed* to create an LS factor map. Use "PALEOSURFACE" as the *input* map, set option *threshold* to the same size you used when you created the "BASINS" map, and output *length.slope* as "LS".

Step 3: Calculate the rate of erosion in the clipped areas with the RUSLE equation in mapcalc. The syntax looks like this: "if(COST_SURF <= *x*, ((*R* * *K* * *C* * LS) * ((nsres()/10000) * *d*)), null()). Where *x* is the cutoff you used to create the clipped DEM, and *R, K,* and *C* are the factors of the RUSLE equation having to do with rainfall intensity, soil erodability, and vegetation protection. *R* varies from 3 to 20 (dry to wet climates), K varies from 0.05 to 0.8 (clay to loam soils), and C varies from 0.1 through 0.001 (bare land through mature forest cover). These values can also be maps, such as those output from landuse simulations. The portion of the equation "((nsres()/10000) * *d*)" turns the output of the RUSLE equation from tons/hectare/year into meters of vertical change/cell/year, where *d* is the average soil density and varies from less than 1 to more than 2 for various types of soil. Play around with different values of *R* (simulating climate change) and *C* (simulation environmental change, either human induced or natural). Name the *output* map "EROS_RATE".

Step 4: Calculate the amount of time that would have been needed to erode the amount of sediment present in "ERODED_SED". Use *r.mapcalc* with the following syntax: "if(ERODED_SED >= 0, (ERODED_SED/EROS_RATE), null())". Here, we first check to make sure the that net change between the paleosurface and the modern surface is positive (small negative changes can occur due to the nature of the interpolation process), and then we divide that net change (meters of sediment that have been eroded in the modeled area) by the erosion rate (which is in meters of erosion per year), and the output map is a map of the time it took to erode the sediment in the modeled areas, given the climatic and environmental conditions given in the RUSLE equation used to calculate the erosion rate. Name this *output* map "TIME".

Step 5: Get some stats! Use *r.uinvar* to collect some interesting statistics for *input* map TIME. This tells you how long it would have taken to erode the amount of sediment to turn the modeled paleosurface into the modern surface for the modeled erosion rate. Remember that erosion rate is based largely on climatic (rainfall intensity) and environmental (vegetation protection) conditions. Compare the outcome for a past landscape of only forest cover (*C* ~ 0.001) with that of a typical cultivated field (*C* ~ 0.05) using the same climate (*R* the same for both calculations), and then compare the resulting average times for erosion with how old you think the landscape is.