This paper published in:
Hickey, R, A. Smith, and P. Jankowski, 1994, Slope length calculations from a DEM within ARC/INFO GRID: *Computers, Environment and Urban Systems*, v. 18, no. 5, pp. 365 - 380.

Slope Length Calculations from a DEM within ARC/INFO GRID

Robert Hickey, Alan Smith, and Piotr Jankowski
Department of Geography
University of Idaho
Moscow, ID 83843

Abstract:

The Universal Soil Loss Equation has been used for a number of years to predict soil erosion rates. One of the required inputs to this model is the cumulative uphill slope length. Calculating it has been the largest problem in using the USLE. The only necessary data for this calculation is a digital elevation model (DEM). DEMs have been available for years and are frequently used within Geographical Information Systems (GIS), but until recently, the GIS software has not been robust enough to support the necessary calculations. This paper describes a method for calculating slope length from a digital elevation model within the ARC/INFO GRID raster system.

Keywords: Erosion, geographical information systems, slope, slope length, Digital elevation model.

Erosion Modeling

The USLE (universal soil loss equation) has been used for a number of years to predict soil erosion rates. In its traditional form, the USLE is given by the following equation:

$$A = RKLSCP$$

where A is the average annual soil loss per unit area, R is the rainfall and runoff factor, K is the soil erodability factor, L is the slope length factor, S is the slope steepness factor, C is the cover and management factor, and P is the support practice factor (8). This paper will focus on generating the L and S factors from a DEM (digital elevation model) within GRID, the raster module of the GIS software package ARC/INFO. The two inputs to the LS factor are cumulative slope length and slope angle.

In recent years, soil erosion models more advanced than the USLE have been created, including ANSWERS (1), AGNPS (9), and the USLE's replacement, RUSLE (6). It is important to note here that RUSLE and AGNPS include both a slope and a slope length component in their equations.

Generating the LS values poses the largest problem in using the USLE (4, 5, 6), especially when applying it to real landscapes within a GIS (4). Traditionally, the best estimates for L were obtained from field measurements, but these are not always available or practical. In addition, until recently, GIS packages have not been able to support the algorithms necessary for slope length calculations. This software limitation can be overcome using ARC/INFO GRID, a raster-based modeling environment and ARC/INFO's Arc Macro Language (AML) programming environment (2).

Calculating slope from a DEM is relatively simple, but care must be taken when selecting an algorithm. For example, the ARC/INFO GRID command, **SLOPE**, calculates maximum slope from a 3X3 cell neighborhood, not a maximum downhill slope (2). The AML program **dn_slope.aml** given at the end of this paper calculates maximum downhill slope.

Algorithm Description:

The overall methodology is described in Figure 1. The first requirement for the algorithm is a depressionless DEM. We have found that the GRID command **FILL** does not eliminate all depressions. Therefore, we have written a small AML (**fil.aml**) which is similar to the **FILL** command (2) but will entirely fill all the depressions in a DEM.

Once this has been completed, the maximum downhill slope and the flowdirection can be calculated using the AML **dn_slope.aml** and the GRID command **FLOWDIRECTION**. It is important to note here that the flowdirection and the direction of maximum downhill slope are the same.

High points are designated (**high_pts.aml**) by selecting those cells which have either no flow entering them or in cases where both the cell in question and its input cell have a slope angle of zero.

Non-cumulative slope length (NCSL) is then calculated for each cell within **sl.aml**. In short, the distance to each cell from its input cell is calculated using the following equations:

if the cell being calculated is a high point

$$NCSL = 0.5(\text{cell resolution})(\text{atan}\beta).$$

if the input cell is in a cardinal direction (N, S, E, W),

$$NCSL = (\text{cell resolution})\ (\text{atan}\ \beta)$$

otherwise,

$$NCSL = 1.4142(\text{cell resolution})(\text{atan}\beta).$$

(β is the slope angle)

This is based on the assumption that the calculations for slope length are from the center of the cell to the center of its input cell. Therefore, as high points do not have an input cell, the 0.5 value represents only the erosion occurring within the half of that cell which is uphill of the center point.

At this point, all the input coverages are present to calculate cumulative slope length (NCSL, slope angle, high points, and flowdirection). This is done by simply summing the non-cumulative slope lengths along flowdirection beginning at the high points (**s_length.aml**). There are a number of assumptions built into this calculation. The first is that in areas of converging flows, the highest cumulative slope length takes precedence. The second is in finding cells in which there is no erosion. The assumption here is that if the slope angle decreases by 50% (or more) from one cell to the next along flowdirection (i.e. 10° to 5°), there is net deposition rather than erosion. This is similar to assumptions made in other studies (4 and 7), although in these cases, the cutoffs were a change of 50% from the average uphill slope angle and a change of 50% from the maximum uphill slope angle, respectively. While we do not consider more than the nearest upslope cell in our cutoff calculations, the program allows the user to specify any cutoff value (50% will be the default).

The final steps in this analysis are to convert the cumulative slope length values into feet (if necessary) and to calculate the LS values. The USLE equation for LS values is:

$$LS = (\lambda/72.6)^m (65.41 \sin^2\beta + 4.56\sin\beta + 0.065)$$

where λ is the cumulative slope length, β is the downhill slope angle, and m is a slope contingent variable--0.5 if the slope angle is greater than 2.86°, 0.4 on slopes of 1.72° to 2.86°, 0.3 on slopes of 0.57° to 1.72°, and 0.2 on slopes less than 0.57° (8).

The ARC/INFO AML program that calculates the LS values requires a DEM coverage and two input values: the cell resolution units (feet or meters) and the decrease in slope angle required for identifying cells with net deposition rather than erosion (a 50% decrease is the default). A simple user interface (**slmenu.aml**) has been programmed to assist the user in entering these required values. The final outputs from this program are three coverages: ls_values (USLE LS values), slope_angle (downhill slope angles), and slope_len (the cumulative slope lengths).

The overall flow of the AMLs begins as follows: **slmenu.aml** calls the user interface AML **execute.aml**. Once the user has input the appropriate information, **sl.aml** takes over. For the rest of the LS calculations, **sl.aml** acts as the core program and calls the other AMLs as necessary. It is important to note here that the only programs that are designed to act independently are those that effectively replace ARC/INFO commands (**fil.aml** and **dn_slope.aml**).

Model Limitations

The first problem associated with using a DEM to calculate slope length is the typically low resolution (i.e. 30m for USGS DEMs). Microfeatures which slow runoff, and therefore erosion, are lost. Thus, LS estimates will be higher than actual values. As DEM resolutions get higher, the landscape will be more accurately modeled, and erosion estimates will approach actual values. However, in cases where detailed field studies are impractical, this method offers a solution with available data.

The second limitation is directly associated with the model presented. For accurate cumulative slope length values, the high points calculated by the model must correspond to true terrain high points. To accomplish this, this model must be run on a watershed scale, not on partial watersheds. If this is not done, slope lengths will be underestimated because the high points calculated may be located on the sides of slopes, not on ridgelines. It is important to note here that the Arc/Info GRID command **WATERSHED** will delineate watersheds from a DEM (2).

Conclusions

While field estimates of cumulative slope length may be more accurate than this model, for larger areas they are typically not practical nor affordable. In addition, trained personnel may not be available. Thus, cumulative slope length calculations from a DEM may be the only option if erosion rates are to be modeled. The advantages of this model are:

1) Using the USLE, erosion rates can be calculated entirely within a GIS environment.

2) The cumulative slope length output can be used in a number of different erosion models, including the USLE, RUSLE, and AGNPS (8,6,9).

3) The erosion rates coverage can be used within the GIS as an input to land suitability analysis problems. For example, erosion rates may be one factor considered when deciding which parts of a large superfund site should be reclaimed.

4) Erosion rates can be calculated for large areas without time-consuming and costly slope length field surveys.

Finally, we would like to state that the AMLs provided at the end of this paper represents a prototype implementation of the method for calculating slope length from a DEM within a GIS environment, not a definitive solution.

References:

1) Beasley, D.B. and Huggins, L.F., 1991, ANSWERS users manual: Department of Agr. Eng., Purdue Univ., West Lafayette, IN.

2) ESRI, Environmental Systems Research Institute. Arc/Info, version 6.1, Users Guide, 1992.

3) DeRoo, A.P.J., Hazelhoff, L., and Burrough, P.A., 1989, Soil erosion modelling using 'ANSWERS and geographical information systems: Earth Surface Processes and Landforms, v. 14: 517-532.

4) Griffin, M.L., D.B. Beasley, J.J. Fletcher, and G.R. Foster, 1988, Estimating soil loss on topographically nonuniform field and farm units. J. Soil and Water Cons. 43: 326-331.

5) Moore, I.D., and J.P. Wilson, 1992, Length-slope factors for the revised universal soil loss equation: simplified method of estimation. J. Soil and Water Cons., 47(5): 423-428.

6) Renard, K., G.R. Foster, G.A. Weesies, and J.P Porter, 1991, RUSLE Revised universal soil loss equation. J. Soil and Water Cons. 46: 30-33.

7) Wilson, J.P., 1986, Estimating the topographic factor in the universal soil loss equation for watersheds. J. Soil and Water Cons. 41: 179-184.

8) Wischmeier, W.H. and D.D. Smith, 1978, Predicting rainfall erosion losses--A guide to conservation planning: Agricultural Handbook no. 537, Sci. and Educ. Admin., U.S. Dept. Agr., Washington, D.C.

9) Young, R.A., C.A. Ontsad, D.D. Bosch, and W.P. Anderson, 1987, AGNPS, Agricultural Non-Point-Source Pollution Model. A Watershed Analysis Tool. U.S. Department of Agriculture, Conservation Research Report 35, 80 p.

slmenu.aml

```
/* execute.aml *********************************************************************

&terminal 9999
&menu sl.menu &position &UL &stripe 'Slope Length'
&return
```

execute.aml

```
/* execute.aml *********************************************************************

/* Set up error checking **********************************************************
&severity &warning &routine error
&severity &error   &routine error
&sv error = .FALSE.

/* Check for DEM ******************************************************************
&sv test = [exist %.DEM% -grid]
&if %error% &then
   &do
   &sv choice = ~
      [getchoice OK -prompt 'You must choose a DEM.']
   &return
   &end

/* Check for UNITS ****************************************************************
&sv test = [null %.unit%]
&if %error% &then
   &do
   &sv choice = ~
      [getchoice OK -prompt 'You must choose a unit of measure.']
   &return
   &end

/* Check for CUTOFF **************************************************************
&sv test = [calc %.cutoff% * 1]
&if %error% &then
   &sv .cutoff = 50

/* Execute and return to GRID ****************************************************
&severity &warning &ignore
&severity &error   &fail
&run s.aml %.DEM% slope_len ls_values %.unit% %.cutoff% slope_angle
&delvar .DEM
&delvar .unit
&delvar .cutoff
&stop
```

```
/* error ****************************************************************
&routine error
&sv error = .TRUE.
&return
```

sl.aml

```
/* sl.aml ****************************************************************
/* The input : a grid of elevations.
  /* The elevations must be in the same units as the horizontal distance.
        /* The unit of measurement for the elevation grid.
        /* The change in slope(as a %) that will cause the slope length
        /* calculation to stop and start over.
/* The output: a grid of cumulative slope lengths,
       /*: a grid of LS values for the soil loss equation,
       /*: an optional grid of down hill slope angle.
/* Usage: sl <elevation grid> <slope length grid> <LS value grid>
      /* {FEET | METER} {cutoff value} {slope angle grid}

&args sl_elev sl_out LS_out grd_units cutoff_value sl_angle

/* Convert user input to capital letters ************************************
&sv .grid_units = [translate %grd_units%]

/* Set default cutoff value if necessary ************************************
&if [null %cutoff_value%] or [index %cutoff_value% #] eq 1 &then
   &sv .slope_cutoff_value = .5
&else
   &sv .slope_cutoff_value = [calc [value cutoff_value] / 100]

/* Set the grid environment *************************************************
setcell   %sl_elev%
setwindow %sl_elev%
&describe %sl_elev%

/* Create a depressionless DEM ********************************************
&run fil.aml %sl_elev% sl_DEM

/* Create an outflow direction grid ****************************************
sl_outflow = flowdirection(sl_DEM)

/* Create a possible inflow grid *******************************************
sl_inflow = focalflow(sl_DEM)

/* Calculate the degree of the down slope for each cell ********************
&run dn_slope.aml sl_DEM sl_slope
```

```
/* Calculate the slope length for each cell *******************************
&sv cell_size       = [show scalar $$cellsize]
&sv diagonal_length = 1.414216 * %cell_size%
   /* Convert to radians for cos--to calculate slope length
if (sl_outflow in {2, 8, 32, 128})
   sl_length = %diagonal_length% div cos(sl_slope div deg)
else
   sl_length = %cell_size% div cos(sl_slope div deg)
endif

/* Set the window with a one cell buffer to advoid NODATA around the edges **
setwindow [calc [show scalar $$wx0] - [show scalar $$cellsize]] ~
        [calc [show scalar $$wy0] - [show scalar $$cellsize]] ~
        [calc [show scalar $$wx1] + [show scalar $$cellsize]] ~
        [calc [show scalar $$wy1] + [show scalar $$cellsize]]

/* Create a new flow direction grid with a one cell buffer *****************
sl_flow = sl_outflow
kill sl_outflow
sl_outflow = con(isnull(sl_flow), 0, sl_flow)
kill sl_flow

/* Create a grid of the high points and NODATA *****************************
   /* The high points will have 1/2 their cell slope length for VALUE ******
&run high_pts.aml

/* Create a grid of high points and 0's ************************************
   /* This will be added back in after the slope lengths for all other *****
   /* cells has been determined for each iteration ************************
sl_high_pts = con(isnull(sl_cum_l), 0, sl_cum_l)

/* Calculate the cumulative slope length for every cell *******************
&run s_length.aml

/* Calculate the LS value for the soil loss equation **********************
&run ls.aml

/* Reset window and mask **************************************************
setwindow %sl_elev%
setmask   %sl_elev%
setmask   off

/* Kill temporary grids ***************************************************
kill sl_(!DEM outflow inflow length high_pts!)

/* Set the output grid names to the user input ****************************
rename sl_cum_l  %sl_out%
rename ls_amount %LS_out%
```

```
&if [null %sl_angle%] &then
   kill sl_slope
&else
   rename sl_slope %sl_angle%
&return
```

fil.aml

```
/* fil.aml ****************************************************************
/* The input : a grid of elevations
/* The output: a depressionless elevation grid.
/* Usage: fil <elevation grid> <depressionless grid>

&args DEM_grid fil_DEM

/* Copy original elevation grid *********************************************
%fil_DEM% = %DEM_grid%

/* Create a depressionless DEM grid ****************************************
finished = scalar(0)
&do &until [show scalar finished] eq 1
   finished = scalar(1)
   rename %fil_DEM% old_DEM
   if (focalflow(old_DEM) eq 255) {
      %fil_DEM% = focalmin(old_DEM, annulus, 1, 1)
      test_grid = 0
   }
   else {
      %fil_DEM% = old_DEM
      test_grid = 1
   }
   endif
   kill old_DEM
      /* Test for no more sinks filled ************************************
   docell
      finished {= test_grid
   end
   kill test_grid
&end

&return
```

dn_slope.aml

```
/* dn_slope.aml ***********************************************************
/* The input : a grid of elevations with no depressions
/* The output: a grid of down slopes in degrees.
/* Usage: dn_slope <elevation grid> <down slope grid>
```

```
&args DEM_grid down_slope

/* Compute the outflow direction for each cell ******************************
dn_outflow = flowdirection(%DEM_grid%)

/* Set the window with a one cell buffer *************************************
&describe %DEM_grid%
setwindow [calc [show scalar $$wx0] - [show scalar $$cellsize]] ~
        [calc [show scalar $$wy0] - [show scalar $$cellsize]] ~
        [calc [show scalar $$wx1] + [show scalar $$cellsize]] ~
        [calc [show scalar $$wy1] + [show scalar $$cellsize]]

/* Create a DEM with a one cell buffer **************************************
   /* This prevents NODATA being assigned to the edge cells that flow
   /* off the DEM. Cells that flow off the DEM will get 0 slope ************
dn_buff_DEM = con(isnull(%DEM_grid%), focalmin(%DEM_grid%), %DEM_grid%)

/* Calculate the down slope in degrees **************************************
&sv cell = [show scalar $$cellsize]
   /* The () pervent problems that occur with using whole numbers **********
&sv cell_size       = (1.00 * %cell%)
&sv diagonal_length = (1.414216 * %cell_size%)
   /* find down slope cell and calculate slope ****************************
if (dn_outflow eq 64)
   %down_slope% = deg * atan((dn_buff_DEM - dn_buff_DEM(0, -1)) div ~
              %cell_size%)
else if (dn_outflow eq 128)
   %down_slope% = deg * atan((dn_buff_DEM - dn_buff_DEM(1, -1)) div ~
              %diagonal_length%)
else if (dn_outflow eq 1)
   %down_slope% = deg * atan((dn_buff_DEM - dn_buff_DEM(1, 0)) div ~
              %cell_size%)
else if (dn_outflow eq 2)
   %down_slope% = deg * atan((dn_buff_DEM - dn_buff_DEM(1, 1)) div ~
              %diagonal_length%)
else if (dn_outflow eq 4)
   %down_slope% = deg * atan((dn_buff_DEM - dn_buff_DEM(0, 1)) div ~
              %cell_size%)
else if (dn_outflow eq 8)
   %down_slope% = deg * atan((dn_buff_DEM - dn_buff_DEM(-1, 1)) div ~
              %diagonal_length%)
else if (dn_outflow eq 16)
   %down_slope% = deg * atan((dn_buff_DEM - dn_buff_DEM(-1, 0)) div ~
              %cell_size%)
else if (dn_outflow eq 32)
   %down_slope% = deg * atan((dn_buff_DEM - dn_buff_DEM(-1, -1)) div ~
              %diagonal_length%)
```

```
else
   %down_slope% = 0.00
endif

/* Reset old settings *****************************************************
setwindow %DEM_grid%

/* Clip the output grid ***************************************************
dn_slope = %down_slope%
kill %down_slope%
rename dn_slope %down_slope%

/* Kill the temporary grids *************************************************
kill dn_buff_DEM
kill dn_outflow

&return
```

high_pts.aml

```
/* high_pts.aml ************************************************************
/* This is not a stand alone AML ********************************************
   /* Grids used from sl.aml:
      /* sl_outflow
      /* sl_inflow
      /* sl_slope
   /* Grid produced for sl.aml:
      /* sl_cum_l

/* Find the high points and set value to half their own slope length ********
/* A high point is a cell that has no points flowing into it or if the only
/* cells flowing in to it are of equal elevation. **************************
if ((sl_inflow && 64) and (sl_outflow(0, -1) eq 4))
   sl_cum_l = setnull(1 eq 1)
else if ((sl_inflow && 128) and (sl_outflow(1, -1) eq 8))
   sl_cum_l = setnull(1 eq 1)
else if ((sl_inflow && 1) and (sl_outflow(1, 0) eq 16))
   sl_cum_l = setnull(1 eq 1)
else if ((sl_inflow && 2) and (sl_outflow(1, 1) eq 32))
   sl_cum_l = setnull(1 eq 1)
else if ((sl_inflow && 4) and (sl_outflow(0, 1) eq 64))
   sl_cum_l = setnull(1 eq 1)
else if ((sl_inflow && 8) and (sl_outflow(-1, 1) eq 128))
   sl_cum_l = setnull(1 eq 1)
else if ((sl_inflow && 16) and (sl_outflow(-1, 0) eq 1))
   sl_cum_l = setnull(1 eq 1)
else if ((sl_inflow && 32) and (sl_outflow(-1, -1) eq 2))
```

```
   sl_cum_l = setnull(1 eq 1)
   /* Flat high points get 0 instead of 1/2 slope length ******************
else if (sl_slope eq 0)
   sl_cum_l = 0.0
else
   sl_cum_l = 0.5 * sl_length
endif

&return
```

s_length.aml

```
/* s_length.aml *********************************************************
/* This is not a stand alone AML *******************************************
   /* Grids used from sl.aml:
      /* sl_inflow
      /* sl_outflow
      /* sl_slope
      /* sl_length
      /* sl_high_pts
      /* sl_DEM
   /* Grid produced for sl_aml:
      /* sl_cum_l

/* Pervents the testing of the buffer cells ********************************
setmask sl_DEM

/* Calculate the cumulative slope length for each cell *********************
nodata_cell  = scalar(1)
&sv finished = .FALSE.
&do &until %finished%
   rename sl_cum_l sl_out_old
   &sv counter = 0
   &do counter = 1 &to 8

      /* Set the varibles for the if that follows
   &select %counter%
      &when 1
         &do
         &sv from_cell_grid          = sl_north_cell
         &sv from_cell_direction     = 4
         &sv possible_cell_direction = 64
         &sv column                  = 0
         &sv row                     = -1
         &end
      &when 2
         &do
```

```
   &sv from_cell_grid         = sl_NE_cell
   &sv from_cell_direction    = 8
   &sv possible_cell_direction = 128
   &sv column                 = 1
   &sv row                    = -1
   &end
&when 3
   &do
   &sv from_cell_grid         = sl_east_cell
   &sv from_cell_direction    = 16
   &sv possible_cell_direction = 1
   &sv column                 = 1
   &sv row                    = 0
   &end
&when 4
   &do
   &sv from_cell_grid         = sl_SE_cell
   &sv from_cell_direction    = 32
   &sv possible_cell_direction = 2
   &sv column                 = 1
   &sv row                    = 1
   &end
&when 5
   &do
   &sv from_cell_grid         = sl_south_cell
   &sv from_cell_direction    = 64
   &sv possible_cell_direction = 4
   &sv column                 = 0
   &sv row                    = 1
   &end
&when 6
   &do
   &sv from_cell_grid         = sl_SW_cell
   &sv from_cell_direction    = 128
   &sv possible_cell_direction = 8
   &sv column                 = -1
   &sv row                    = 1
   &end
&when 7
   &do
   &sv from_cell_grid         = sl_west_cell
   &sv from_cell_direction    = 1
   &sv possible_cell_direction = 16
   &sv column                 = -1
   &sv row                    = 0
   &end
&when 8
   &do
```

```
        &sv from_cell_grid         = sl_NW_cell
        &sv from_cell_direction    = 2
        &sv possible_cell_direction = 32
        &sv column                 = -1
        &sv row                    = -1
        &end
&end

   /* Test for possible flow source cell
if (not(sl_inflow && %possible_cell_direction%))
   %from_cell_grid% = 0
   /* Test for flow source cell
else if (sl_outflow(%column%, %row%) <> %from_cell_direction%)
   %from_cell_grid% = 0
  /* Test flow source cell for nodata
else if (isnull(sl_out_old(%column%, %row%)))
   %from_cell_grid% = setnull(1 eq 1)
   /* Test current cell slope against cutoff value
else if (sl_slope >= (sl_slope(%column%, %row%) * %.slope_cutoff_value%))
   %from_cell_grid% = sl_out_old(%column%, %row%) + ~
                   sl_length(%column%, %row%)
else
   %from_cell_grid% = 0
endif
&end

   /* Select the longest slope length
sl_cum_l = max(sl_north_cell, sl_NE_cell, sl_east_cell, sl_SE_cell, ~
              sl_south_cell, sl_SW_cell, sl_west_cell, sl_NW_cell, ~
              sl_high_pts)

   /* Kill the temporary grids
kill (!sl_north_cell sl_NE_cell sl_east_cell sl_SE_cell ~
      sl_south_cell sl_SW_cell sl_west_cell sl_NW_cell!)
kill sl_out_old

   /* Test for the last iteration filling in all cells with data
&sv no_data = [show scalar nodata_cell]
&if %no_data% eq 0 &then
   &sv finished = .TRUE.

   /* Test for any nodata cells
if (isnull(sl_cum_l) and not isnull(sl_outflow))
   sl_nodata = 1
else
        sl_nodata = 0
endif
nodata_cell = scalar(0)
```

```
   docell
      nodata_cell }= sl_nodata
   end
   kill sl_nodata

&end

/* Reset original window and clip the cumulative slope length grid **********
setwindow sl_DEM
rename sl_cum_l sl_out_old
sl_cum_l = sl_out_old
kill sl_out_old

&return
```

ls.aml

```
/* ls.aml ***************************************************************
/* This is not a stand alone AML *********************************************
   /* Grids used from sl.aml:
      /* sl_cum_l
      /* sl_slope
   /* Grid produced for sl.aml:
      /* ls_amount

/* Convert meters to feet if necessary **************************************
&if %.grid_units% eq METERS &then
   ls_length = sl_cum_l div 0.3048
&else
   ls_length = sl_cum_l

/* Calculate LS for the soil loss equation *********************************
   /* For cells of depostion
if (ls_length eq 0)
   ls_amount = 0
   /* For slopes 5% and over
else if (sl_slope >= 2.862405)
   ls_amount = pow((ls_length div 72.6), 0.5) *    ~
         (65.41 * pow(sin(sl_slope div deg), 2) + ~
          4.56 * sin(sl_slope div deg) + 0.065)
   /* For slopes 3% to less than 5%
else if ((sl_slope >= 1.718358) and (sl_slope < 2.862405))
   ls_amount = pow((ls_length div 72.6), 0.4) *    ~
         (65.41 * pow(sin(sl_slope div deg), 2) + ~
          4.56 * sin(sl_slope div deg) + 0.065)
   /* For slopes 1% to less than 3%
else if ((sl_slope >= 0.572939) and (sl_slope < 1.718358))
   ls_amount = pow((ls_length div 72.6), 0.3) *    ~
```

```
          (65.41 * pow(sin(sl_slope div deg), 2) + ~
           4.56 * sin(sl_slope div deg) + 0.065)
    /* For slopes under 1%
else
    ls_amount = pow((ls_length div 72.6), 0.2) *    ~
          (65.41 * pow(sin(sl_slope div deg), 2) + ~
           4.56 * sin(sl_slope div deg) + 0.065)
endif

/* kill temporary grids *****************************************************
kill ls_length

&return
```