ELSEVIER
Computers & Geosciences 30 (2004) 1043–1053
COMPUTERS & GEOSCIENCES
www.elsevier.com/locate/cageo

Computing the LS factor for the Revised Universal Soil Loss Equation through array-based slope processing of digital elevation data using a C++ executable☆

R.D. Van Remortel[a,*], R.W. Maichle[a], R.J. Hickey[b]

[a] *Lockheed Martin Environmental Services, Las Vegas, NV 89119, USA*
[b] *Central Washington University, Ellensburg, WA 98926, USA*

Received 18 November 2003; received in revised form 3 July 2004; accepted 10 August 2004

Abstract

Until the mid-1990s, a major limitation of using the Universal Soil Loss Equation and Revised Universal Soil Loss Equation erosion models at regional landscape scales was the difficulty in estimating LS factor (slope length and steepness) values suitable for use in geographic information systems applications. A series of ArcInfo™ Arc Macro Language scripts was subsequently created that enabled the production of either USLE- or RUSLE-based LS factor raster grids using a digital elevation model input data set. These scripts have functioned exceptionally well for both single- and multiple-watershed applications within targeted study areas. However, due to the nature and complexity of flowpath processing necessary to compute cumulative slope length, the scripts have not taken advantage of available computing resources to the extent possible. It was determined that the speed of the computer runs could be significantly increased without sacrificing accuracy in the final results by performing the majority of the elevation data processing in a two-dimensional array framework outside the ArcInfo environment. This paper describes the evolution of a major portion of the original RUSLE-based AML processing code to an array-based executable program using ANSI C++™ software. Examples of the relevant command-line arguments are provided and comparative results from several AML-vs.-executable time trials are also presented. In wide-ranging areas of the United States where it has been tested, the new RUSLE-based executable has produced LS-factor values that mimic those generated by the original AML as well as the RUSLE Handbook estimates. Anticipated uses of the executable program include water quality assessment, landscape ecology, land-use change detection studies, and decision support activities. This research has now given users the option of either running the executable file alone to process a single watershed reporting unit or running a supporting AML shell program that calls upon the executable file as necessary to perform automated processing for a user-specified number of watersheds.

Keywords: Soil erosion; RUSLE; GIS applications; Watershed modeling; Landscape ecology

1. Introduction

For nearly 40 years, the Universal Soil Loss Equation (USLE) model (Wischmeier and Smith, 1978) and its

---

☆Code on server at http://www.iamg.org/CGEditor/index.htm.

*Corresponding author. Tel.: +1 702 897 3295; fax: +1 702 897 6640.

E-mail address: rvanremo@lmepo.com (R.D. Van Remortel).

0098-3004/$ - see front matter © 2004 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cageo.2004.08.001

principal derivative, the Revised Universal Soil Loss Equation (RUSLE) model (Renard et al., 1997) have been used throughout the world to estimate average annual soil loss per unit land area resulting from rill and sheet (interrill) erosion. When applying the USLE or RUSLE models, five component factors (R, K, LS, C, and P) are multiplied together to compute the average annual sheet and rill erosion per unit area. Traditionally, the two models have been used mostly for local conservation planning at an individual farmstead scale because the USLE model was originally developed for gently sloping cropland applications. Recent research leading to the RUSLE model has broadened the applicability of the models somewhat to allow limited soil loss estimation for rangeland, forests, disturbed sites, and steeper slopes. It should be noted here that the term *soil loss* should not be construed to mean that all eroded soil is lost from the spatial land unit being investigated, as it is possible for eroded soil to be subsequently re-deposited downslope on lesser sloping surfaces (Haan et al., 1994). In this sense, USLE and RUSLE are primarily erosion models with some limited linkages to sediment yield models.

It has been demonstrated that increases in slope length and slope steepness can produce higher overland flow velocities and correspondingly higher erosion (Haan et al., 1994). Moreover, gross soil loss is considerably more sensitive to changes in slope steepness than to changes in slope length (McCool et al., 1987). Slope length has been broadly defined as the distance from the point of origin of overland flow to the point where either the slope gradient decreases enough that deposition begins, or the flow is concentrated in a defined channel (Wischmeier and Smith, 1978). The specific effects of topography on soil erosion are estimated by the dimensionless LS factor as the product of the slope length (L) and slope steepness (S) constituents converging onto a point of interest, such as a farm field or a cell on a GIS raster grid.

Until recently, the use of USLE and RUSLE for regional landscape ecology modeling has been limited by an inability to generate reliable estimates of the LS factor. Although the LS factor is usually either estimated or manually calculated from actual field measurements of slope length and steepness for local conservation planning purposes, labor-intensive field measurements are generally not feasible for modeling soil erosion at significantly larger spatial scales. However, newly developed procedures allow users of geographic information system (GIS) technology to generate both USLE- and RUSLE-based raster grids of the LS factor for various site characterization and landscape ecology applications. A thorough review of available GIS-based methods for calculating the LS factor is included in papers by (Dunn and Hickey (1998); Hickey (2000)). Various approaches and algorithms for quantifying slope length are available, including raster grid cumulation, unit stream power theory, contributing area, and network triangulation techniques. There are also several methods for estimating slope steepness including neighborhood, quadratic surface, maximum slope, and maximum downhill slope techniques. The algorithms described in this paper use the raster grid cumulation and maximum downhill slope methods.

The previous work of Hickey et al. (1994), Hickey (2000)) resulted in the production of ArcInfo™ Arc Macro Language (AML) programs for creating a USLE-based LS factor grid from an input digital elevation model (DEM) data set. Subsequent RUSLE-based amendments were added by Van Remortel et al. (2001) to the USLE-based code and involved the substitution of several recently developed RUSLE algorithms and the modification of a few assumptions in the AML program concerning the treatment of high points, flat areas, slope breaks, and other specific slope criteria. The RUSLE algorithms derived by McCool et al. (1987, 1989) utilized the results of statistical analysis applied over a much broader range of slope configurations, gradients, and cover types than those modeled for USLE, so the new algorithms are generally considered to be more comprehensive than those of the earlier model (Renard et al., 1997).

Detailed descriptions of the computational basis for our particular RUSLE-based approach for generating an LS factor surface are included in Van Remortel et al. (2001), Hickey (2000). Instead of duplicating that background information within this paper, the authors have chosen to focus on the mechanisms involved in extracting key flowpath-based portions of the original AML program and converting the extracted code to run in a more robust ANSI C++™ executable program. Consequently, supporting information from these previous manuscripts has been brought forward only where needed to understand or clarify a step in the program execution. Users will be shown that it is now possible to compute the RUSLE LS factor either by running the new executable file by itself from a command-line prompt to process a single watershed reporting unit, or by running a supporting AML shell program that calls the executable file as necessary to perform iterative processing for a specified group of watersheds. When the supporting AML shell is utilized, the Arc and Grid modules from the ArcInfo™ Workstation Version 8.2 software for PC/Windows platforms (a product of ESRI, Redlands, California, USA) are called upon to perform the analysis.[1]

---

[1] ESRI ArcGIS products home page. http://www.esri.com/

2. Assumptions and pre-processing caveats

The RUSLE algorithm for calculating slope length (i.e., the L constituent of the LS factor) serves to reference the erosion estimate for a horizontally projected slope length (hpsl) to the experimentally measured erosion for a 22.1-m (72.6-foot) reference slope length (rsl), raised to the power of a designated slope-length exponent (m) value that addresses the ratio of rill-to-interrill erosion, or $L = (\text{hpsl}/\text{rsl})^m$ as set forth in the RUSLE Handbook (McCool et al., 1997; Renard et al., 1997). The dominant land cover types for our study areas are assumed to be rangeland or woodland with a low susceptibility to rill erosion, so a graduated range of RUSLE slope length exponents has been adopted from look-up tables in the RUSLE Handbook to model a low ratio of rill-to-interrill erosion over a wide range of slope gradients (McCool et al., 1989). It is also assumed that actual slope lengths are always longer than 4.6 m (15 ft) such that rilling is likely to be an active component of the erosion. This presumption allows a single L-constituent algorithm with multiple exponents to be applied across the entire slope range (McCool et al., 1987). As required for use in USLE and RUSLE, slope length values are assigned with respect to the x,y horizontal projection of an array matrix, not the true x,y,z surface of a natural landscape.

The RUSLE algorithm for calculating slope steepness (i.e., the S constituent of the LS factor) is calculated directly from a slope angle matrix using two equations (McCool et al., 1987, 1997) that are differentially applied according to a break point at the experimentally modeled 9% gradient (Wischmeier and Smith, 1978). For slopes of less than 9% gradient, $S=10.8*\sin(\text{slope\_angle}+0.03)$. For slopes of 9% or steeper, $S=16.8*\sin(\text{slope\_angle}-0.50)$. The LS factor is subsequently calculated as the product of the L and S constituents.

The primary input needed to generate an LS-factor matrix is a DEM data set of suitable scale (resolution) that has been clipped to encompass the zone of interest. In our studies, a hydrologically or topographically defined *catchment* watershed is the spatial unit best suited for this type of flowpath-based analysis. The geographic projection of the input DEM must be known so that the DEM reporting units and other projection parameters can be identified and tracked through the watershed processing. The raw DEM data set and the output from any calculations upon it should be closely examined to ensure that there are no significant format problems or internal clusters of missing data within the raw DEM data and that the processing algorithms are being applied properly.

As a rule, DEM product suppliers do not necessarily attest to the significance of decimal digits within their data sets. If processing difficulties or disk storage limitations occur with the use of a floating-point storage format, it is possible that truncating or rounding input DEM elevation values from floating-point to integer format (e.g., nearest whole-meters) may be necessary to ensure successful computer runs. Unfortunately, such conversions can produce undesirable *stair-step* features (i.e., wedding-cake effect). The presence of horizontal or vertical stippling, corn-rowing, or edge-matching errors in the DEM can also result in erratic or discontinuous slope length features. Available smoothing algorithms may essentially *correct* some of the DEM irregularities, but can also result in unwarranted smoothing or generalization of other adjacent DEM elevation cells. If utilized, supplemental DEM-enhancement algorithms must be well documented and applied with caution to avoid a gross over-extension of slope lengths (Van Remortel et al., 2001).

The relationships between real-world slope microrelief and DEM cell size are not examined in this paper, so it is assumed that, for erosion and deposition purposes, a cell resolution of 100 or 900 m^2 area (i.e., 10-m or 30-m DEM data, respectively) accurately represents the natural microrelief of the slopes being modeled. If this assumption is erroneous and the actual topography reflects significant slope breaks that are more or less frequent than the fixed cell size, then any estimates of the LS factor from this DEM analysis can be expected to deviate accordingly. Although previous research has suggested that most measured slope lengths are less than 120 m and that slope lengths generally do not exceed 300 m (McCool et al., 1997), very little research on slope length has been conducted in mountainous terrain to confirm the validity of this guideline for such diverse and complex landscapes (Van Remortel et al., 2001). Historically, erosion models have typically been very good at deriving patterns of erosion but not necessarily the actual rates of erosion, so it is recommended that erosion data be evaluated with an emphasis on qualitative, not quantitative, interpretation and assessment.

3. Slope processing methodology

Our LS factor framework and methodology uses DEM data analyzed in accordance with RUSLE criteria and was primarily derived from Version 3 of the RUSLE-based AML code (Van Remortel et al., 2001) that was in turn developed from Version 2 of the USLE-based AML code (Hickey, 2000). Interested users are directed to *Bob's Slope Page* on the Internet for more information regarding the AML code, executables, and similar code developed for the IDRISI GIS software.[2] In

---

[2] Bob's Slope Page. http://www.cwu.edu/~rhickey/slope/slope.html

the new array-based framework that has been adopted, the overall flowpath-based iterative slope-length cumulation and LS factor computation steps are now performed within the *lsfac_c.exe* C++ program, outside ArcInfo. The file will execute independently at the command-line prompt, either with an argument string or with a series of interactive prompt responses, to perform an LS run for a single spatial reporting unit (e.g., one watershed at a time). Users also have a third option of running the *lsfac.aml* AML shell program in ArcInfo Workstation which automatically calls upon the executable as necessary to perform the LS factor processing for single or multiple spatial reporting units (e.g., a large number of study area watersheds in a single run).

Irrespective of whether or not a user chooses to employ the optional AML shell program for a particular analytical run, all users will benefit from the fact that the new *lsfac_c.exe* executable program utilizes a C++ two-dimensional array structure to improve the efficiency of the flowpath processing algorithms over that of the original AML program from which it was derived. The new executable uses computer memory dynamically to perform work *across* an array matrix instead of simply acting upon individual rows of data within an ArcInfo grid as performed by the original AMLs. In this way, dynamic memory can utilize a computer's random-access memory (RAM) as well as virtual memory allocated on a hard drive to accomplish routine writing-to-disk and paging operations.

3.1. Overview of the executable command-line arguments

Each argument enumerated below is followed by a brief description of the kind of information that must be supplied in order to be accepted by the program. (*Note: A tenth command line argument to allow the user to continue calculations with interior* nodata *clusters has been commented-out and that option is no longer available*):

1. Name of executable program—The executable is initialized by typing *lsfac_c* as the first argument at a command-line prompt.
2. Path and name of DEM file—The second argument is the path and name of the input DEM data set (an ASCII file with header information).
3. Path of output directory—The third argument is the path to the directory (or ArcInfo workspace) that will hold the output files.
4. Watershed or study area acronym—The fourth argument is a short prefix (e.g., study area acronym or watershed ID), generally no more than four characters or digits in length, which is appended to the output file names.
5. Whether intermediate files are to be saved—The fifth argument is an affirmative or negative response to the question of whether all intermediate (temporary) array matrices from the LS factor processing should be written to output files.
6. Slope cutoff factor for slope gradients less than 5% —The sixth argument is the cutoff factor for slope gradients less than 5% that will terminate slope length cumulation along a flowpath.
7. Slope cutoff factor for slope gradients equal to or greater than 5%—The seventh argument is the cutoff factor for slope gradients equal to or greater than 5% that will terminate slope length cumulation along a flowpath.
8. Horizontal/vertical DEM reporting units—The eighth argument is a response to the question of whether the DEM's horizontal/vertical reporting units are meters or feet.
9. Whether isolated DEM *nodata* cells should be filled—The ninth argument is an affirmative or negative response to the question of whether the program should attempt to replace isolated *nodata* cells with interpolated average values from surrounding cells.

3.2. Execution with a continuous command-line argument string

A successive string of as few as four (4) or as many as nine (9) of the command-line arguments must be provided by the user to initiate this type of run of the *lsfac_c.exe* program. The resources of ArcInfo Workstation are not required here, although the input DEM data set must already be in ASCII format with supporting header information. The first four arguments in the string require specific user-supplied entries, while the remaining five arguments have built-in default assignments that can be accepted as is or be overridden by successive user-supplied entries. The executable is initialized by entering the following specific usage in a continuous argument string at the command-line prompt:

1. Type *lsfac_c*; then:
2. [required] the path and name of the input DEM file; the program assumes a default path to the directory (or ArcInfo workspace) holding the executable program if only the DEM file name is supplied; then:
3. [required] the path of the output directory (or ArcInfo workspace); entering a pound sign (#) will activate a default path to the directory (or ArcInfo workspace) holding the executable program; then:
4. [required] the watershed or study area prefix; entering a pound sign (#) will activate a default blank whereby no prefix is appended and the output files will have simple generic file names; then:
5. [optional] whether intermediate files are to be saved; a negative response is highly recommended unless

there is a considerable amount of available hard-drive space; typing a value of *n* or *N* or a pound sign (#) will activate a default of *no* for the run, while a value of *y* or *Y* instructs the program to save as many as thirteen (13) intermediate processing floating point *.dat files per spatial reporting unit (e.g., watershed) within the output file directory; then:

(6) [optional] the slope cutoff factor for slope gradients less than 5%; the default value is recommended unless there is specific justification to support an alternate value; typing a pound sign (#) will activate a default value of *0.7* for the run; then:

(7) [optional] the slope cutoff factor for slope gradients equal to or greater than 5%; the default value is recommended unless there is specific justification to support an alternate value; typing a pound sign (#) will activate a default value of 0.5 for the run; then:

(8) [optional] the horizontal/vertical DEM reporting units; typing *meters* or a pound sign (#) will activate a default value of *meters* for the run; a value of *feet* should be used if units are *feet*; then:

(9) [optional] whether isolated DEM *nodata* cells should be filled; an affirmative entry is recommended in almost every instance, otherwise the program will terminate prematurely if isolated *nodata* cells are encountered; typing a value of *y* or *Y* or a pound sign (#) will activate a default affirmative assignment for the run, while a value of *n* or *N* provides a negative assignment.

3.3. Execution with a series of command-line interactive prompt responses

A series of reponses to interactive prompts following the initial command-line argument(s) must be provided by the user to initiate this type of run of the *lsfac_c.exe* program. The resources of ArcInfo Workstation are not required here, although the input DEM data set must already be in ASCII format with supporting header information. All arguments require user-supplied responses to command-line interactive prompts, and have built-in default assignments that can either be accepted as is with a carriage return or be overridden by a specific user-supplied response. This manner of argument processing is very similar to that described in 3.2 above, with a few exceptions. The executable is initialized by entering the following specific usage of the arguments at the successive prompts:

(1) Enter *lsfac_c,* then:

(2) [required] enter the path and name of the input DEM file; the program assumes a default path to the directory (or ArcInfo workspace) holding the executable program if only the DEM file name is supplied; then:

(3) [required] enter the path of the output directory (or ArcInfo workspace); simply entering a carriage return will activate a default path to the directory (or ArcInfo workspace) holding the executable program; then:

(4) [required] enter the watershed or study area prefix; simply entering a carriage return will activate a default blank whereby no prefix is appended and the output files will have simple generic file names; then:

(5) [required] enter whether intermediate files are to be saved; a negative response is highly recommended unless there is a considerable amount of available hard-drive space; entering a value of *n* or *N* or a simple carriage return will activate a default of *no* for the run, while a value of *y* or *Y* instructs the program to save as many as thirteen (13) intermediate processing floating point *.dat files per spatial reporting unit (e.g., watershed) within the output file directory; then:

(6) [transparent] the slope cutoff factor for slope gradients less than 5% is activated with a default value of *0.7* for the run; if an alternate value is desired, the command-line argument string approach described in *3.2* must be used; then:

(7) [transparent] the slope cutoff factor for slope gradients equal to or greater than 5% is activated with a default value of *0.5* for the run; if an alternate value is desired, the command-line argument string approach described in 3.2 must be used; then:

(8) [transparent] the horizontal/vertical DEM reporting units are activated with a default value of *meters* for the run; if the alternate value of *feet* is desired, the command-line argument string approach described in 3.2 must be used; then:

(9) [required] enter whether isolated DEM *nodata* cells should be filled; an affirmative entry is recommended in almost every instance, otherwise the program will terminate prematurely if isolated *nodata* cells are encountered; entering a value of *y* or *Y* or a carriage return will activate a default affirmative assignment for the run, while a value of *n* or *N* provides a negative assignment.

3.4. Execution with an AML shell which automatically calls the executable

The principal benefits of using the *lsfac.aml* AML shell for processing are that it initializes a *&do &until* iterative loop for 1…*n* user-defined watersheds (or equivalent spatial reporting unit) and automatically calls the *lsfac_c.exe* executable program as necessary to compute the LS factor. It also accomplishes several pre- and post-processing tasks in the ArcInfo software and produces ArcInfo grids from the resulting array matrices. All command-line arguments needed to execute the executable program are assigned from user-provided responses to queries during the initialization of the AML run.

Only a few additional ArcInfo input files are required to successfully run the AML. First, a study area Arc workspace must be created and named by a brief acronym (e.g., *mt3* for a third set of Montana watersheds); it will hold the *lsfac.aml* and *lsfac_c.exe* programs. Two more workspaces are created immediately under this study area workspace: *gds* to hold the study area geodatasets, and *wsh* to hold the individual watershed workspaces and associated geodatasets as they are produced. A study area watersheds coverage (e.g., *mt3wsh*) and a study area DEM grid with integer values of elevation*100 (e.g., *mt3demx100*) are placed in the *gds* workspace. The watershed cover-IDs (subclass-ID if it has regions features) of the watersheds are assigned 1…*n* continuous unique values so that looping can be accomplished and each watershed can be linked by a unique–ID number that is traceable throughout the analysis. Ideally, the watersheds should be hydrologically defined *catchment* watersheds to ensure accurate and complete runs of the programs. Using a spatial reporting unit other than this may be inappropriate and cause the run to terminate prematurely or the results to be flawed or incomplete.

The *lsfac.aml* AML program is initialized by starting ArcInfo Workstation, migrating to the study area workspace, entering *&run lsfac* at the *Arc:* prompt, and entering appropriate responses to the ensuing interactive prompts. When dealing with multiple spatial reporting units (e.g., watersheds), a particular file-naming convention is adopted to enable looping through the set of watersheds. An ArcInfo workspace (e.g., *w1*…*wn*) is automatically created for each study area watershed being processed. Under each of these workspaces, a watershed boundary coverage (e.g., *w1bnd*) is extracted from the overall study area watersheds coverage and used to clip the study area DEM grid to produce DEM data for the watershed (e.g., *w1dem*). This watershed DEM grid then undergoes an Arc-to-ASCII file conversion and the *lsfac_c.exe* program is called upon to perform the slope processing steps. A final LS factor grid is subsequently produced from the resulting LS factor matrix.

3.5. Description of the executable run processing steps

During any given run, a series of DEM-derived intermediate matrices are produced and subsequently used in the final calculation of the LS-factor values. Fig. 1 contains a flowchart that shows an overall view of the process.

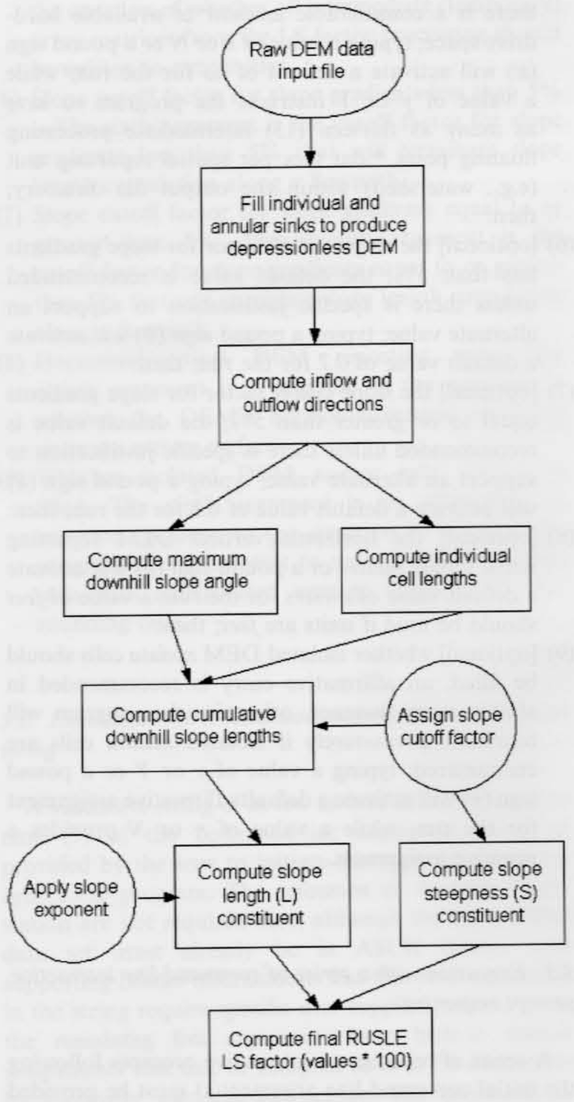

Fig. 1. Flowchart illustrating process of calculating cumulative downhill slope length and final LS factor values using C++ executable program, for applications of RUSLE erosion model.

Upon program execution, startup information is written to a log file and a series of pathing and file-prefix strings is created. The program reads the DEM file header data and the relevant information (e.g., number of rows and columns assigned to fields in the data structure) is passed to subsequent routines. Dynamic memory is requested from the operating system for a floating-point matrix to hold the DEM elevation data and a Boolean matrix to hold *nodata* cell locations. If system resources are inadequate to allocate this or any subsequent request for dynamic memory, program execution is halted and a message to that effect is written to the log file and screen. Otherwise, the DEM elevation and nodata values are read into their allocated matrices and passed along to subsequent routines to calculate additional downslope parameters.

The DEM data are analyzed to determine if any interior *nodata* cells exist, and the results of this check are written to the log file and screen. If there is one or more interior *nodata* cell(s), the program identifies whether the cells are isolated or occur in clusters. Isolated cells or linear stringers of seven or fewer *nodata* cells can be automatically repaired and populated with values within the program, but if larger clusters of such cells exist the user is informed of this condition and the program execution is halted. If the user has selected the option to not attempt the repair of isolated cells, a message is written to the screen and log file, and the program execution is halted. Otherwise, the interior *nodata* cells are repaired and the DEM and *nodata* matrices are updated. If intermediate files have been requested, the original DEM data file is saved as well as all subsequent matrices.

The program then begins a *fill* sequence on the DEM data to perform either individual-cell *sink* filling or cluster-cell *annulus* filling as necessary until no additional filling can be accomplished. In this process, elevations of the affected cells are raised (i.e., flattened) to correspond with surrounding cells. The filled (i.e., depressionless) DEM data set is written to an output and is ready for subsequent LS factor computation. All newly filled cells are identified in the log file by their row/column location, initial and filled elevation values, and how filling was accomplished (i.e., sink or annulus). This rapid filling procedure replaces the previously used iterative AML procedure from Hickey et al. (1994) that had been demonstrated to produce better results than using the standard *fill* command in the ArcInfo Workstation Grid module.

To conserve available memory, two unsigned-character flow-type matrices are allocated for the assignment of cell outflow and inflow directions which are based on a widely used approximation of flow direction for a square-celled elevation surface grid, known as the Deterministic 8 (D8) algorithm (O'Callaghan and Mark, 1984). For instance, outflow direction is a numerical representation in which each cell takes on one of eight D8 values depending on which of its neighboring cells (four on the orthogonal cardinal axes and four on the diagonal half-cardinal axes) is the direction of steepest descent to which each cell is outwardly flowing, as determined by the highest elevational gradient among the *in* and *out* cells. A cell or group of cells with outflow values other than cardinal or half-cardinal is considered to be a *flat* area (e.g., bench, terrace) lacking a single defined outflow direction. In like manner, an inflow cell adopts a coded value identifying the D8 direction(s) of all cells that could possibly be flowing into that cell. This is done by examining all surrounding cells of higher elevation and selecting only those whose outflow direction indicates flow into the selected cell. This flow direction assignment procedure replaces (but is functionally the same as) the previously used AML procedure which utilized the ArcInfo Workstation Grid commands *focalflow* and *flowdirection*, respectively. A slightly modified approach is used to reconcile a situation where two or more potential outflow directions exist for a given cell; the program assigns a single primary outflow direction based on the following top-to-bottom and left-to right sequential order: northwest, north, northeast, west, east, southwest, south and southeast.

Dynamic memory is requested from the operating system to allow the allocation of a Boolean matrix to accommodate user-defined slope cutoff factors. When considering sediment-laden water flowing across the earth's surface, at various points the flow velocity may decrease enough that the sediment carried will tend to deposit rather than erode more sediment. Defining the parts of the landscape in which this will happen is not a straightforward task, as deposition is a dynamic function of slope gradient (which largely determines the velocity) and the sediment concentration within the flow. If the flow is fully saturated with sediment, any decrease in velocity will result in deposition rather than erosion. Conversely, if the flow is relatively unsaturated, it will take a very significant decrease in slope (possibly to near zero) to result in deposition. For our purposes, the slope cutoff factor is defined as the ratio of change in slope angle from one cell to the next along a flow direction pathway, where only the nearest upslope cell is considered for the cutoff calculations. This factor ranges in value from 0 to 1 and is applied wherever the slope angle decreases from one cell to the next. A cutoff value of 0.0 will cause the slope length to reset with any decrease in slope angle, whereas a value of 1.0 will prevent the slope length from ever resetting. A value closer to 0.5 (slope decreasing by 50% or greater) may be appropriate based on assumptions made in other studies (Griffin et al., 1988; Wilson, 1986). User-defined factors of 0.7 and 0.5 are recommended for use with slope gradients of less than and greater than 5%, respectively, because deposition is generally easier to initiate on lesser gradient slopes. Ideally, the appropriate values for slope cutoff factors are set by an expert having knowledge of the particular area in question.

Dynamic memory is then requested from the operating system to allow the allocation of floating point matrices for slope angle and cell length. Initial *target cells* are defined which represent the beginning points for every flowpath; these include cells having outflow direction but no inflow cells (e.g., ridgelines, benches, mesas, etc.) as well as cells associated with flat areas (undefined outflow direction), and are assigned slope lengths of one-half of their cell slope length values. The maximum downhill slope angle of each cell is then calculated from the filled DEM on an individual neighboring-cell basis (Dunn and Hickey, 1998) according to each cell's outflow direction, allowing for both cardinal and half-cardinal directional flow. A matrix containing the cell slope length, or *non-cumulative* slope length of each cell, is calculated from the slope angle and flow direction matrices as either the cardinal or half-cardinal length of that cell (i.e., the center-to-center cell length of each *from/to* cell combination) according to its outflow direction. In order to correspond with USLE and RUSLE guidelines, the value is calculated in x,y space (i.e., the horizontal projection of a gridded surface) instead of x,y,z space (i.e., the surface of a natural landscape).

Dynamic memory is then requested to allow the allocation of floating point matrices for the slope length cumulation, slope exponent, and final slope length (L constituent) values. A routine is called to sum the cell slope length values along flow direction pathways initiated from the target cells, and the resulting cumulative slope length values (typically in meters) are subsequently written to a matrix. The C++ cumulation algorithm operates iteratively as a series of passes over the array matrix, using a forward-and-reverse 'sweeping' motion (i.e., from the upper left to the lower right of an array, then back again) to accomplish the greatest amount of cumulation possible within a pass before starting another iteration. In this way, the initial sweep of a pass handles all possible flowpath cells flowing to the east, southeast, south, and southwest directions, plus one cell away flowing to the west, northwest, north, and northeast directions. Then, before proceeding with the next iteration, the return sweep of the pass handles all possible flowpath cells flowing to the west, northwest, north, and northeast directions, plus one cell away flowing to the east, southeast, south, and southwest directions. This modification and the efficiency of the C programming language have combined to dramatically improve the efficiency of the RUSLE LS computation process, as the original AML routine could process only a single adjacent cell for each flowpath during an iterative pass. Each cell associated with a particular flowpath direction is assigned a cumulative slope length only if the possible flow source cells (as indicated by the inflow direction matrix) also have a cumulative slope length assigned, an outflow direction into the cell of interest, and a slope angle to the cell of interest that exceeds the relevant slope cutoff factor. In areas of converging flows, the longest cumulative slope length takes precedence (Hickey et al., 1994; Hickey, 2000; Van Remortel et al., 2001).

If the DEM data has the default reporting units of meters, a floating-point matrix to hold cumulative slope length in feet (i.e., the units required by the L-constituent algorithm) is allocated and assigned the conversion of the cumulative length matrix from meters to feet. A routine to compute L is called which raises the measured-to-reference slope length ratio to the power of the slope exponent, after which the DEM and cell length matrices are no longer needed and their respective allocated memory is returned to the operating system. Dynamic memory is then requested to allow the allocation of floating-point matrices for slope steepness (the S constituent) and the definitive RUSLE LS factor. A routine to compute S is called and the slope steepness values are written to file. The dynamic memory allocated to hold the slope angle, slope cutoff factor, and inflow/outflow direction matrices is then returned to the operating system. A routine to compute the RUSLE LS factor is called and the LS values are written to the output file directory as integers that are scaled to 100 times the actual RUSLE LS factor. This is done for the sake of file storage efficiency and to allow users to maintain two decimal places of accuracy (i.e., 0.01) within the reported LS factor. Appropriate caveats concerning the scaling of LS-factor values are written to the screen and log file, and the executable program completes its run.

4. Results of test runs

Informal time-trial runs comparing results from the original RUSLE-based AML program (Van Remortel et al., 2001) with results from the new C++ executable program were conducted for two distinct study areas. The first study area selected was the 230-km^2 Rincon Creek watershed catchment in southeastern Arizona, which utilized 30-m horizontal and 0.01-m vertical resolution derivatives from the US Geological Survey's National Elevation Dataset DEM data. Figs. 2–5 show

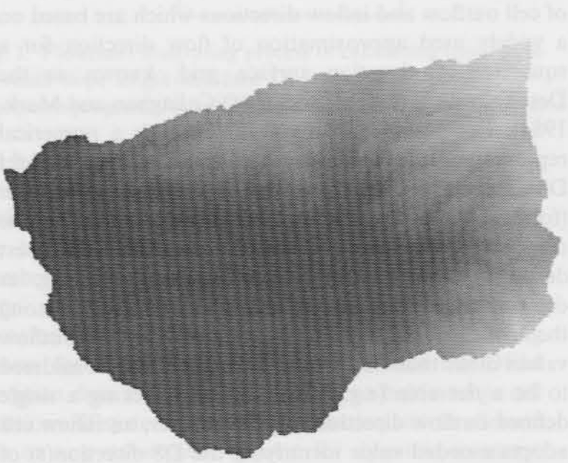

Fig. 2. Representation of 30-m DEM elevation input matrix for Rincon Creek test run of executable program (min = 856.70 m, max = 2639.76 m, mean = 1326.87 m, std = 425.24 m).

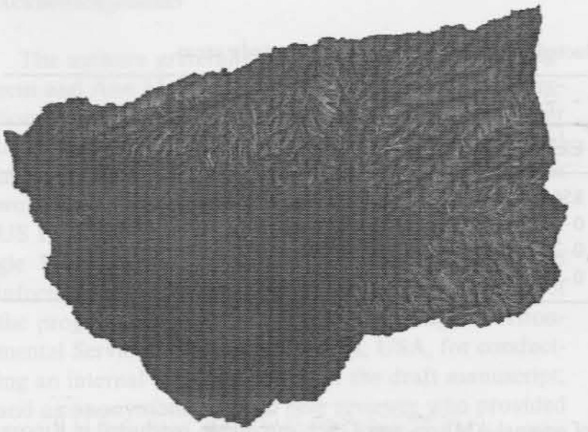

Fig. 3. Representation of 30-m slope length (*L* constituent) output matrix from Rincon Creek test run of executable program (min = 0.01, max = 14.49, mean = 1.92, std = 1.13).

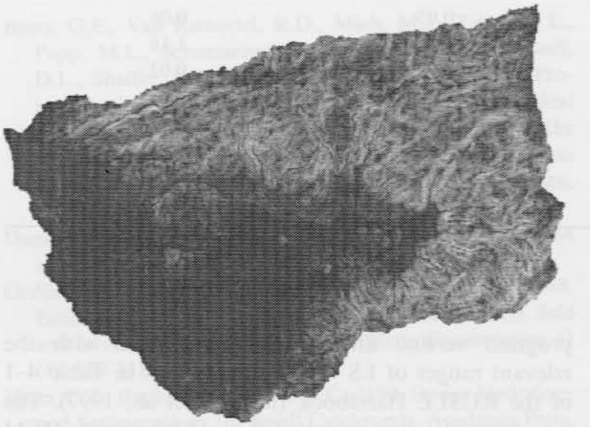

Fig. 4. Representation of 30-m slope steepness (*S* constituent) output matrix from Rincon Creek test run of executable program (min = 0.01, max = 13.73, mean = 3.08, std = 2.64).

graphical images of the DEM elevation, slope length, slope steepness, and LS factor matrices, respectively, from the executable run performed on the Rincon Creek watershed. The second study area selected was a portion of the Upper Yakima River subbasin in central Washington, which utilized 10-m horizontal and 1-m vertical resolution DEM source data. Elevation files for three overlapping Yakima subunits, comprising approximately 900, 2125, and 3750-km^2 rectangular blocks, were extracted for the test run processing. It is important to note here that this latter block-processing approach is appropriate only for users who ultimately intend to clip the LS factor results from the run to yield a smaller, specifically defined zone of interest (i.e., study area) within the block, after the LS factor processing has completed.

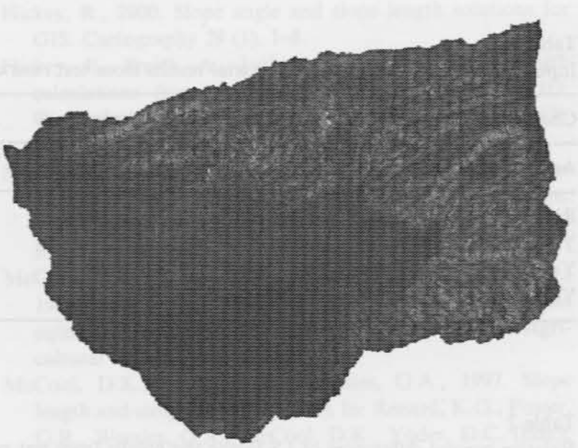

Fig. 5. Representation of 30-m RUSLE LS factor output matrix from Rincon Creek test run of executable program (min = 0.01, max = 104.88, mean = 7.21, std = 8.72).

Table 1 shows the input DEM specifications and timetrial results from the test runs conducted for the two study areas, both of which were selected because they exhibit wide ranges of elevations and slope gradients suitable for general testing of the programs' efficacy. Clearly, an enormous relative increase in processing speed has been realized with the utilization of the new C++ executable. This enhancement is proving to be particularly useful in landscape ecology studies, which commonly derive their analytical data sets from large regional study areas with large numbers of watersheds that require processing.

Table 2 shows some of the relative percent differences in the LS factor test results from test runs of the original RUSLE-based AML with respect to the new C++ executable. The relative percent differences for all key matrices examined, with the exception of the *L* constituent, were very small (i.e., less than one-third of 1%) among the range, mean, and standard deviation values for the matrices. Differences of as much as 15% or more in the *L* constutuent values can be expected when examining results from various slope length runs, in part because of a tendency for seemingly minuscule differences among small absolute values to become artificially inflated when expressed on a relative percentage basis, a characteristic also referred to as *concentration dependency* (Byers et al., 1990). In the examples cited, average absolute values of *L* are less than 2.0 and their differences for a given run are in fact seemingly minor when viewed in an absolute context. Variations in *L* also may occur during the programs' flowpath assignment process due to the possible selection among any number of alternate flowpaths within the matrix, owing to subtle differences in the way that competing identical slope angles are addressed with respect to selecting a dominant flow direction. The authors

Table 1
Input DEM specifications and time trial results from test runs conducted in the Rincon and Yakima study areas

Characteristics of raw input DEM data for the study areas					Run minutes elapsed	
Area	File size (Mb)	Cell size (m)	Units (m)	Elevation range (m)	Original AML	C++ executable
Rincon	3	30	0.01	856.70–2639.76	18	<1
Yakima1	9	10	1	0–1731	855	4
Yakima2	21	10	1	0–2015	2550	9
Yakima3	37	10	1	0–2079	5760	18

Table 2
Relative percent differences of LS factor test results from test runs of original AML vs. new C++ executable, conducted in Rincon and Yakima study areas

Area	Grid or matrix	Relative percent differences between old and new runs		
		Range (%)	Mean (%)	Std. dev. (%)
Rincon	Slope angle	0.00	0.05	0.06
Rincon	Slope length (*L*)	15.43	0.78	1.16
Rincon	Slope steepness (*S*)	0.00	0.03	0.02
Rincon	LS factor	0.26	0.25	0.23
Yakima2	Slope angle	0.00	0.01	0.00
Yakima2	Slope length (*L*)	3.76	9.14	9.29
Yakima2	Slope steepness (*S*)	0.01	0.00	0.00
Yakima2	LS factor	0.15	0.03	0.01

consider these or other such differences in L to constitute only a small portion of the overall measurement error and essentially unimportant because, as shown in the table, the differences are ultimately muted within the final cell-by-cell computation of the LS factor.

5. Conclusions and recommendations

A significant drawback of the original AML-driven runs was that the ArcInfo processing was not able to make efficient use of available computer resources. This problem has now been effectively alleviated with the advent of the executable-driven array processing. Much faster run times can be expected as the DEM is loaded into a matrix and run in memory, assuming the dynamic memory limits of the computer are not exceeded. Performance can be expected to diminish somewhat when the computer must rely primarily on virtual memory allocated on the hard drive.

Results of test runs using high-quality 10 and 30-m DEMs indicate that RUSLE-based LS factor estimates generated by the new C++ executable program are equivalent to those generated by the previous AML program version and are closely aligned with the relevant ranges of LS values summarized in Table 4–1 of the RUSLE Handbook (McCool et al., 1997). The runs have also demonstrated that a high-quality DEM input data set is the key element for ensuring reliable LS-factor output data. Errors within the input DEM may result in erratic and discontinuous slope length features and will likely produce low slope length values. Conversely, efforts to smooth such errors in the input DEM may result in gross over-extensions of slope lengths and should be avoided where possible.

Interested users are directed to *Bob's Slope Page* on the Internet for more information regarding the AML code, executables, and similar code developed for the IDRISI GIS software.[2] The RUSLE-based programs described in this paper are also available for FTP download through the journal's website.[3] All users of these programs should be advised that the programs represent a prototype implementation of a method for calculating slope length and steepness from DEM data and should not be construed as the definitive solution for calculating the LS factor within a GIS framework.

---

[3] Computers & Geosciences FTP site access. http://www.iamg.org/

Acknowledgements

The authors gratefully acknowledge Daniel T. Heggem and Ann M. Pitchford, US Environmental Protection Agency, Las Vegas, Nevada, USA, for their unwavering support in the development and technical transfer of the computer programs, without which this work would not have been possible; Timothy G. Wade, US Environmental Protection Agency, Research Triangle Park, North Carolina, USA, for conducting an informal review and computational accuracy checks of the programs; Lee A. Bice, Lockheed Martin Environmental Services, Las Vegas, Nevada, USA, for conducting an internal technical review of the draft manuscript; and an anonymous external peer reviewer who provided several good suggestions for clarifying the concepts introduced within the manuscript.

References

Byers, G.E., Van Remortel, R.D., Miah, M.J., Teberg, J.E., Papp, M.L., Schumacher, B.A., Conkling, B.L., Cassell, D.L., Shaffer, P.W., 1990. Direct/Delayed Response Project: Quality Assurance Report for Physical and Chemical Analyses of Soils from the Mid-Appalachian Region of the US EPA/600/4-90/001. US Environmental Protection Agency, Office of Research and Development, Washington, DC, USA, 337pp.

Dunn, M., Hickey, R., 1998. The effect of slope algorithms on slope estimates within a GIS. Cartography 27 (1), 9–15.

Griffin, M., Beasley, D., Fletcher, J., Foster, G., 1988. Estimating soil loss on topographically nonuniform field and farm units. Journal of Soil and Water Conservation 43 (4), 326–331.

Haan, C.T., Barfield, B.J., Hayes, J.C., 1994. Design Hydrology and Sedimentology for Small Catchments. Academic Press, San Diego, CA, USA 588pp.

Hickey, R., 2000. Slope angle and slope length solutions for GIS. Cartography 29 (1), 1–8.

Hickey, R., Smith, A., Jankowski, P., 1994. Slope length calculations from a DEM within ARC/INFO GRID. Computers, Environment, and Urban Systems 18 (5), 365–380.

McCool, D.K., Brown, L.C., Foster, G.R., 1987. Revised slope steepness factor for the universal soil loss equation. Transactions of the American Society of Agricultural Engineers 30, 1387–1396.

McCool, D.K., Foster, G.R., Mutchler, C.K., Meyer, L.D., 1989. Revised slope length factor for the universal soil loss equation. Transactions of the American Society of Agricultural Engineers 32, 1571–1576.

McCool, D.K., Foster, G.R., Weesies, G.A., 1997. Slope length and steepness factors (LS). In: Renard, K.G., Foster, G.R., Weesies, G.A., McCool, D.K., Yoder, D.C. (Eds.), Predicting Soil Erosion by Water: A Guide to Conservation Planning with the Revised Universal Soil Loss Equation (RUSLE). Agriculture Handbook No. 703. US Department of Agriculture, Agricultural Research Service, Washington, DC, USA, pp. 101–141 (404pp.) (Chapter 4).

O'Callaghan, J.F., Mark, D.M., 1984. The extraction of drainage networks from digital elevation data. Computer Vision, Graphics, and Image Processing 28, 323–344.

Van Remortel, R.D., Hamilton, M.E., Hickey, R.J., 2001. Estimating the LS factor for RUSLE through iterative slope length processing of digital elevation data within ArcInfo Grid. Cartography 30 (1), 27–35.

Wilson, J.P., 1986. Estimating the topographic factor in the universal soil loss equation for watersheds. Journal of Soil and Water Conservation 41, 179–184.

Wischmeier, W.H., Smith, D.D., 1978. Predicting Rainfall Erosion Losses—A Guide to Conservation Planning. Agriculture Handbook No. 537. US Department of Agriculture Science and Education Administration, Washington, DC, USA, 163pp.