**REPORT DOCUMENTATION PAGE**

Public Reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

<table>
<thead>
<tr>
<th>1. AGENCY USE ONLY (Leave Blank)</th>
<th>2. REPORT DATE</th>
<th>3. REPORT TYPE AND DATES COVERED</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1/25/99</td>
<td>Final Rept; 6/1/94-12/31/98</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>4. TITLE AND SUBTITLE</th>
<th>5. FUNDING NUMBERS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Discrete Models in Sediment Transport</td>
<td>DAAH04-94-G-0067</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>6. AUTHOR(S)</th>
<th>8. PERFORMING ORGANIZATION REPORT NUMBER</th>
</tr>
</thead>
<tbody>
<tr>
<td>P. K. Haff</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)</th>
<th>10. SPONSORING / MONITORING AGENCY REPORT NUMBER</th>
</tr>
</thead>
<tbody>
<tr>
<td>Duke University</td>
<td>ARO 30685.11-EV</td>
</tr>
<tr>
<td>Durham, NC</td>
<td></td>
</tr>
</tbody>
</table>

| U. S. Army Research Office                        |                                               |
| P.O. Box 12211                                    |                                               |
| Research Triangle Park, NC 27709-2211             |                                               |

**SUPPLEMENTARY NOTES**

The views, opinions and/or findings contained in this report are those of the author(s) and should not be construed as an official Department of the Army position, policy or decision, unless so designated by other documentation.

**DISTRIBUTION / AVAILABILITY STATEMENT**

Approved for public release; distribution unlimited.

**ABSTRACT (Maximum 200 words)**

The principal goal of the present study has been to construct and test computer algorithms for fluvial sediment erosion and deposition processes. Real-life topographic features in arid terrain have been used as a source of groundtruth information. Much use is being made today of generic landscape evolution models. But little effort has gone into testing such models against actual landscape evolution as measured in the field. In particular, modern landscape models are seldom used for site-specific studies. This work has attempted to bridge that gap. The near-term objective has been to test some commonly used constitutive rules for sediment transport against geomorphic evidence as observed in the field. The present work has focussed on defining broad scale erosion/deposition patterns to fluvial erosion patterns. These comparative studies are be critical for transferring generic sediment transport rules used by most landscape modelers into actual hands-on algorithms that can be used in real life situations of interest to the Army. We have also been interested in applying our studies of desert pavement to problems of Army interest, in particular, to possible ways to restore or stabilize these ancient surfaces. An ancillary goal has been to understand the problems involved in scaling-up fundamental, small-scale sediment transport physics to large-scale engineering and environmental applications involving erosion and landscape change with time, and to develop computational tools appropriate for such large-scale applications.

**SUBJECT TERMS**

erosion, landscape evolution, desert pavement

**NUMBER OF PAGES**

75

**PRICE CODE**

UL

**SECURITY CLASSIFICATION OR REPORT**

UNCLASSIFIED

**SECURITY CLASSIFICATION ON THIS PAGE**

UNCLASSIFIED

**SECURITY CLASSIFICATION OF ABSTRACT**

UNCLASSIFIED

**LIMITATION OF ABSTRACT**

UL

NSN 7540-01-280-5500

DTIC QUALITY INSPECTED 4

Standard Form 298 (Rev.2-89)
Prescribed by ANSI Std. 239-18
298-102
Final Progress Report
1998

Discrete Methods for Sediment Transport Modeling (30685-EV)

View Toward Ft. Irwin, CA, from Soda Mountains

P. K. Haff
Division of Earth and Ocean Sciences
Nicholas School of the Environment
Duke University, Box 90230
Durham, NC 27708-0230

January 1999

19990616 068
# Table of Contents

Memorandum of Transmittal  
Title Page 1  
Table of Contents 2  
Research Objectives 3  
Approach to Problem 3  
Significance Of Research Results For US Army 4  
Accomplishments 6  
Technology Transfer 9  
Figure Captions 11  
List of Publications and Abstracts 13  
Invited Talks 16  
Scientific Personnel Supported During Grant Period 17  
Degrees Awarded 17  
Figures 18  
Appendix: Fortran program WATERBOT 33
Research Objectives:

The principal goal of the present study has been to construct and test computer algorithms for fluvial sediment erosion and deposition processes. Real-life topographic features in arid terrain have been used as a source of groundtruth information. Much use is being made today of generic landscape evolution models. But little effort has gone into testing such models against actual landscape evolution as measured in the field. In particular, landscape models are seldom used for site-specific studies. This work has attempted to bridge that gap. The near-term objective has been to test some commonly used constitutive rules for sediment transport against geomorphic evidence as observed in the field. The present work has focussed on defining broad scale erosion/deposition patterns to fluvial erosion/deposition processes. These comparative studies are critical for transferring generic sediment transport rules into actual hands-on algorithms that can be used in real life situations of interest to the Army.

We have also been interested in applying our studies of desert pavement to problems of Army interest, in particular, to possible ways to restore or stabilize these ancient surfaces.

Approach to Problem:

1. Erosion studies were carried out through a combination of computer simulation and field studies. Computer simulations of hillslope transport were performed for sites where fluvial erosion and deposition are important, ongoing, and where field observation could provide suitable feedback for improving and refining the model. We modeled specific geographic sites, not generic landscapes. Field sites chosen were in the Mojave Desert on terrain units of the type that commonly occur in military training areas. Thus most of the work was focussed on the granitic dome/pediment surfaces in the Cima Dome, CA, area that resembles in many respects granitic terrains found extensively at Fort Irwin.

2. Field work was carried out to provide feedback to model development. Field investigations pinpointed particular location-specific processes indicated as a result of modeling, and identified characteristic regions of erosion and deposition that could be checked against model output. Using electronic survey equipment acquired through ARO resources, detailed hillslope transects were made in the Mojave Desert. Computer simulations of hillslope transport was performed for a variety of different sediment transport rules. These comparative studies were used to test sediment transport rules for use in specific terrain situations.
3. Studies of processes on, and stability of, desert pavement surfaces have been pursued via long-time-based observational studies. Desert pavement is a commonly occurring desert surface that is widely used (e.g., at Yuma Proving Ground) for military vehicle traffic. Field experimentation involving disturbance of pavement surfaces has been continued in order to understand recovery processes where the disturbances heal themselves, or to understand why surface disruption provides a positive feedback loop that leads to further unraveling of pavement surfaces. Pavement studies have been mainly field based so far, but it is anticipated that modeling studies will be initiated as the result of the studies performed here.

Significance Of Research Results For US Army:

i) Large-Scale Erosion Studies: These methods have been developed for use on a PC, and can be applied to large geographic areas of interest to both civilian and military land-use managers. The simulations have been run on areas of over 100 square kilometers, and are applicable to NTC, YPG and other regions of comparable size, Figs. 1 and 2. Runoff is tracked over the entire geographical region of interest, Fig. 3. Regions of erosion and deposition can be determined quickly, Fig. 4, and the magnitude (thickness) of eroded and deposited sediment computed. The model can run in an asynchronous mode whereby individual storms can activate flow in a subset of available drainage channels in the study area. The model is thus applicable to sedimentation resulting from a single localized storm, Fig. 5. Because slope is a central factor in all sediment transport rules, slope maps are useful for an intuitive assessment of regions likely to show serious erosion or deposition, Fig. 6. On a smaller scale, the model will be useful for assessing the effects of fluvial sedimentation and erosion resulting from land-use practices such as construction of roads, berms, soil compaction and so on. It is clear that digital data must be available that is sufficiently accurate in terms of both vertical and lateral resolution to enable an accurate determination of hydrologic flow directions to be obtained. A present limitation in implementing the model is the lack of adequate digital topographic data in regions of low slope typical of areas where vehicle-based training and testing is performed.

Accurate assessment of erosion rates is a critical national need. For example, the Yucca Mountain Nuclear Waste Disposal Site in Nevada has been chosen partly on the basis of assumed low erosion rates. Proposed expansion of the US Army's Fort Irwin training facility in California requires knowledge about the response of landscape, including erosion, to new types of land use. Modern attempts to simulate the evolution of landscape in response to erosion by running water have so far mostly been aimed at geological reconstruction of existing landforms. But the technical capability to apply such methods to prediction of future changes in landscape is rapidly maturing. The present study has constructed a
basis for understanding what limitations exist in scaling basic physical measurements to field scale problems.

ii) *Hillslope Studies*: It is of importance to all land managers, including the Army, to be able to assess the future behavior of specific terrain units to human impact of various kinds. Detailed field studies of representative terrain units, such as the small Mojave Desert hillslopes studied here, are necessary to identify and quantify the geomorphic processes that may affect terrain response to disturbance. Our studies have helped to distinguish between those processes that are likely active under today's climate regime from those that have been operative under past climate regimes. This kind of investigation is critical for assessment of landscape response to human activity.

iii) *Advective And Diffusive Models*: Landscape evolution models are an essential part of the tool box of landscape management. Most such models (except for some specialize models used in agricultural studies) are generic, in the sense that they purport to indicate the general kind of change that can be expected in landscape over some period of time (usually a time of geologic interest). Our studies of advective and diffusive sediment transport models are aimed at calibrating such models against field conditions for specific arid terrain sites of the type of interest to the US Army. Our studies at Cima Dome represent an exercise of the model in a setting similar to that found at locations in nearby Fort Irwin.

iv) *Long-Time-Based Studies of the Stability of Gravel Surfaces (Desert Pavement)*: Armored gravel surfaces are ubiquitous across much of the southwest US, including Army reservations at Fort Irwin and Yuma Proving Grounds. Disturbance and destruction of these surfaces, mainly by vehicle traffic, is widespread. Our long-time-based studies of controlled disturbance of desert pavement surfaces in the Mojave Desert have identified some of the mechanisms responsible for pavement development and stability, and hence suggest strategies for restoring disturbed surfaces to an approximation of their original form. This should be of considerable interest to the Army, both for reasons of environmental stewardship, as well as for the purpose of maintaining training surfaces in something like their original condition.

v) *The Dynamics Of Cryptogamic Crust on Gravel Surfaces*: Cryptogamic soil is a biologic crust that forms over large areas in the Mojave Desert and Basin and range area, and consequently is found commonly on arid military lands. The occurrence and motion of cryptogamic soil islands on otherwise bare gravel patches appears to be a result of orientation and edge effects associate with soil-island geometry. The interaction of such islands with vehicular or foot traffic has not been studied, but it is clear that disturbance of the island perimeter by human activity is likely to have a large effect on the stability of the soil surface.
Army efforts to maintain existing surface conditions on land under Army control depends on the ability to identify and understand the nature of such surface features and processes.

vi) **Predictability in Geomorphology:** Assume that a rigorous, physically-based model has been developed for use in landscape evolution studies. The Army wishes to use this model for purposes of land management. Can management decisions be based upon the outcome of predictions based upon this model? Our studies suggest that the answer is probably "no", if one is talking about site-specific predictions where the model is used without strong attention being paid to the geologic record and to previous experience either at the site in question or at similar (analogous) sites. Also, even with an otherwise good physical model, uncertainties in the actual configuration of soils, bedrock exposure, particle size, vegetation distributions, etc., can render the model ineffective. Our studies have suggested practical strategies for overcoming such inadequacies in model application – such as the incorporation of feedback loops in predictive schemes. These results could have a potentially significant impact on Army land management practices.

**Accomplishments:**

*Development of Physically Based Sediment Transport Model:* The discrete computer model WATERBOT has been developed. A simplified flow-diagram is shown in Fig. 14. The model tracks hydrologic “marker particles”, Fig. 7, as they move downslope, employing an accompanying sediment transport rule to drive erosion and deposition processes on the chosen surface. The topographic data sets typically used to run the model are USGS 30 meter DEMs. The model correlates rainfall patterns to patterns of erosion and deposition. Adjacent 7.5' quadrangles can be joined into a single map for use in WATERBOT; thus the model is suitable for large scale applications. It is feasible to simulate sediment transport over the area covered by as many as six or more 7.5' quadrangles. WATERBOT is a PC-based model written in FORTRAN.

*Model Indicators of Areas of Erosion and Deposition:* WATERBOT can be used to study erosion and deposition of sediment across large geographical areas. Fig. 4 shows computed areas of erosion (blue) and deposition (green and yellow) on an area measuring about 10 km on a side. The location is in the Cima Dome area, Mojave Desert, California. On pediment surfaces such as Cima Dome or geomorphically equivalent surfaces in areas such as NTC, zones of erosion generally indicate that bedrock is at or near the surface, while zones of significant deposition generally indicate that the surface is lose alluvium. These correlations are prima facie indicators of both surface characteristics and load bearing ability. Thus yellow zones in Fig. 4 can be expected to be areas in which
surface mobility for wheeled vehicles will be degraded, while blue zones generally indicate a firm surface.

Model Indicators of Zones of Sediment Accumulation: The model shows how modifications of surface topography can induce local zones of erosion and deposition. Topographic barriers to downstream runoff are shown to produce upstream zones of sediment accumulation, Fig. 8. These sediment accumulation zones are seen in the field where natural flow deflectors such as cinder cones have interfered with surface runoff from higher elevations. Modern artificial barriers also exhibit this behavior, as seen in the sediment trapping upstream of freeway flood control berms, Figs. 9 and 10.

Model Indicators of Zones of Enhanced Erosion: Surfaces that have low rates of intrinsic erosion lead to runoff of “clear” water that produces enhanced erosion downslope as soon as it reaches a more erodible substrate. Model studies of natural low-erodibility surfaces such as some lava flows show striking erosion features downslope, Fig. 11. Similar behavior occurs whenever terrain surface properties are modified to increase run-off (such as by compaction) or to decrease erosion (as occurs when a surface is covered with concrete or other erosion resistant material, Fig. 8). The WATERBOT model provides a way to envision the possible erosion and deposition side effects that may accompany artificial landscape modification.

Hillslope Diffusion: Hillslope diffusion represents that set of natural surface processes such as soil creep and rainbeat that delivers sediment from unchanneled hillslopes to local drainages. Diffusion tends to smooth surfaces. Diffusion represents the main set of natural processes that will over time eliminate the presence of man-made surface disturbances that are not destroyed by channelized flow. Our studies have looked at the rate at which diffusive processes must operate to remove irregularities on the surface imposed at a given rate. By considering the simultaneous function of two otherwise independent diffusion processes, the smoothing effect can be quantified in terms of observable density of disturbances, such as impact crater or road berms.

Presently Inactive Hillslope Processes: Fine-scale geomorphic mapping of a hill in the Mojave Desert has identified at least half a dozen transport mechanisms that are, or have been, important in hillslope evolution – earth flows, slumps, animal burrowing, dry ravel, boulder role, and overland and channel flow. Age information regarding timing of these processes is inferred from desert varnish characteristics. Some of the identified processes, such as earth flows, are probably early Holocene or late Pleistocene. This implies that the modern hillslope configuration – its slopes, soil thickness (which was measured by seismic transects), clast distribution and so forth – is at least partly a product of processes that are no longer operating today. This suggests that “recovery” of a
surface from anthropogenic disturbance may not converge toward existing undisturbed surfaces, since those surfaces may not be a product of presently occurring processes.

**Advection Processes:** Advection sediment transport processes are those mediated directly by running water. Comparison of advective sediment transport with field surveys that determined regions of erosion and deposition showed that in some cases good agreement could be obtained between theory and observation, but that in other cases, agreement between surveyed surfaces and modeled regions of erosion and deposition was poor. Gullies developed on the artificial embankment of Fig. 12 could be modeled with standard sediment transport laws, but some natural gullies, Fig. 13, could not be modeled with standard sediment transport power-laws. Further work is needed in this area, but our results suggest the limitations of some commonly used transport laws.

**Predictability in Geomorphology:** Studies in uncertainty of prediction in geomorphology and sediment transport have been developed. This work provides guidance for organizations, including the Army, who need to make specific recommendations for land use management. The results of the study identify a number of factors that contribute to errors and uncertainty in predicting the future behavior of large natural systems such as landscapes. The role of uncertainty in geomorphic systems is a tricky one, and its study is potentially controversial. The results of our studies show that prediction is not possible, but that attempts at site-specific prediction — a prediction mode of substantial interest to organizations like the Army — is not likely to be possible on the basis of mathematical modeling alone. Rather, use of analogy, and reliance on the historical and geological record, is likely to be at least as important as the use of quantitative mathematical models. Further, prediction may be limited by our lack of knowledge of the present state of the system — an observation which suggests that resources might be more effectively applied to instrumentation and data gathering than to improvement of computational models.

**Long-Time-Based Studies of the Stability of Gravel Surfaces:** A long-time-based study of diffusion on desert pavement has been continued, detailing the dynamics of the these important desert surfaces. This worked focused particularly on the response of surfaces to controlled human disturbance. The results of long-time-based studies of changes on desert pavement surfaces show clearly that these surfaces, although stable over millennia, are not static, but rather exist in a state of dynamic stability. Repeat photography shows how animal activity and other agencies is effective at creating a continuing dislodgment and transport of small surface stones, even on flat surfaces. This dynamical background of activity is an essential ingredient in the ability of pavements to repair disruptions of their surface. As discussed below, this observation, and the measurement of clast
size, areal densities, and other surface parameters, provides a scientific basis for approaching the problem of stabilizing or rehabilitating artificially disturbed desert surfaces. This work is of potential value for Army efforts to remediate and restore desert landscapes that have been disturbed by vehicle traffic or ordnance impact. This work is preparatory to planned future controlled studies at Yuma Proving Ground of the response of natural landscape to anthropogenic disturbances.

The Dynamics Of Cryptogamic Crust on Gravel Surfaces: The occurrence of natural dynamical changes in arid terrain surfaces needs to be understood as part of a larger program to assess the role of human disturbance in landscape behavior over time. These studies of the influence of cryptogamic crusts on the form of desert pavement surfaces in the northern Great Basin indicates that significant biological activity is associated with pavement surface in this climatic regime, which may be contrasted with the modern pavements in more arid regions such as YPG and NTC. In climates where cryptogam is an important component of the soil ecology studies of cryptogam-associated surface stability represent an important baseline conservation or restoration studies. It is important to understand the nature of the cryptogamic crust since pavements further south may have formed under climatic regimes that resembled those found now only at more northerly latitudes, where cryptogamic soils are well-developed.

Technology Transfer:

*Yuma Proving Ground (YPG):* Communications between the PI and YPG (Ms. Valerie Morrill) have been established regarding application to problems at YPG of some of the ideas on landscape processes developed under the present proposal. Two trips to YPG by the PI and graduate student Lonny Boring have given us an introduction to the local terrain and to some of the problems facing environmental managers there. Talks were presented by the PI and Boring to base personnel. Time was also spent in the field with David Lashley of WES. It seems clear that our analysis of pavement surface processes can be of use to the general problem of the origin, nature and age of the various pavement and fan units being studied by Lashley. We have also made a formal presentation of some of our work to YPG personnel. We have submitted a proposal for further investigation of disturbance by Army traffic of desert pavement surfaces at YPG. We will assess the construction of experimental plots on disturbed areas of desert pavement in an effort to better understand the problem of pavement degradation and destruction. Experimental plots that were graded, raked or otherwise smoothed, and which were seeded with appropriate populations of stone sizes would provide, over a few years, important information on stability and potential for restoration for desert pavement surfaces.
Construction Engineering Research Laboratory (CERL): The PI presented a talk at a CERL workshop in Urbana. Conversations were held with Bill Goran about interest of CERL in landscape modeling and surface process studies and restoration and maintenance of disturbed lands. The work on gravel surface dynamics described above is a point of common interest.

Waterways Experiment Station (WES): The PI attended two workshops on vehicle terrain interaction, and presented talks on how the particle dynamics method could be applied to traction problems. The PDM method seems optimized to treating that difficult zone at the boundary of tread or wheel and soil where the engineered precision of the vehicle meets the undesigned complexity of the soil.

Zzyzx Workshop: The PI and graduate student Lonny Boring attended week-long workshop at Zzyzx CA on “New Research Directions in Desert Surficial Processes and Landscape Dynamics on Military Lands”. The PI made a presentation on modeling work carried out under the present project. At this meeting previous phone interactions with Dr. Fred Brieuer of WES were further developed. Dr. Brieuer is interested in the archaeological implications of various surface features found on the YPG pavement surfaces, while our expertise lies in a knowledge of natural surface processes on pavements. Important synergies are anticipated in combing our expertise with that of Dr. Brieuer. We anticipate collaborating on future work at YPG. Similar discussions were had at Zzyzx with David Lashley, also of WES, with the idea of correlating our ground based analysis of surface processes with Lashley’s spectrographic work regarding classification of distinct pavement units.

Computer Program: The WATERBOT program is being actively used to study erosion/deposition processes on arid land surfaces. It is anticipated that this program will become available to Army personnel. The program is written in FORTRAN and runs under Windows NT on a PC. A copy of WATERBOT appears in the Appendix.
Figure Captions

Fig. 1. Photograph of pediment area at Cima Dome similar to areas at Fort Irwin used for training purposes.

Fig. 2. Shaded relief map of Cima Dome area. "A" is summit of Cima Dome.

Fig. 3. Colored lines represent the channel pattern as determined by waterbots as they move downslope. "A" is summit of Cima Dome. The channels are colored by magnitude of discharge (or contributing area). Blue represents a small discharge, with greens, yellows and reds representing higher discharges. The high discharge channels correspond to mapped ("blue line") channels found on the USGS 7.5 minute topographic quadrangles. Channels are dynamic, and small channels especially can change with time as sedimentation causes avulsion. This figure corresponds to flow from uniform rainfall over the entire area, but localized precipitation can also be modeled, leading to localized (asynchronous) flow in a subset of available channels, see Fig. 5.

Fig. 4. Erosion and deposition patterns in the Cima Dome area, Mojave Desert, California. Topography for maps here and below (except for Fig. 5) is from USGS 1/24,000 or 1/250,000 (Fig. 4) DEM data. Spot marked "A" corresponds to local topographic high (Cima Dome), and is marked on the maps shown in figures below as well. Waterbots dropped on each 30m X 30m pixel move downhill, entraining and detraining sediment according to changes in local slope. The erosion/deposition pattern calculated here matches approximately that seen in the field. Pink shading on insert shows area of near-surface bedrock, which approximately reflects region of net long-term erosion.

Fig. 5. Same as Fig. 3, except a localized flow has been initiated by precipitation near the area marked "A". This map is derived from USGS 1/250,000 quadrangles.

Fig. 6. Distribution of slope in the same area and at the same scale as that shown in Fig. 1. Lighter colors correspond to higher slopes. Deposition occurs where high gradients change to lower gradients along the path followed by individual waterbots. In the present version of the waterbot model, slope is the main controlling variable on waterbot dynamics and sediment capacity. However, the influence of changes in infiltration rate, exposure of bedrock, and similar features that can affect sediment transport can be included in the model in a straightforward way where field data is available.

Fig. 7: Schematic picture of waterbot model, in which discrete "water particles" move downslope, picking up and depositing sediment in accordance with a chosen sediment transport rule. Generally, as the slope steepens, waterbots tend
to pick up more sediment, and as slope flattens, they tend to drop some of the sediment they are carrying.

Fig. 8. Topographic map of a simulated uniformly sloping surface (downslope is to the left) upon which sits a nonerodible feature. The surface has been subject to erosion and deposition under the influence of a uniform rainfall. In nature, this feature might be a lava flow, or it might represent a man-made feature constructed, for example, of concrete. The deflection of contour lines in the vicinity of the obstacle indicates the growth of an upslope sediment stagnation zone, and an increase in downslope erosion. The region of orange coloration indicates the area in from which upslope flows are deflected. Both of these effects are observable in the field. This example suggests schematically some ways in which the waterbot model might be applied.

Fig. 9: Man-made obstruction to flow – a freeway flood control berm is a construct in which a berm is created out of alluvial fan material that is typically bulldozed up from the fan surface, leaving a channel or ditch just upslope of the berm. Water running downslope is deflected into the berm. Lessening of the flow angle by deflection leads to enhanced sedimentation in the ditch.

Fig. 10: Schematic illustration of flow and enhanced sedimentation for a simulated berm similar to that in Fig. 9. Deposition upstream of the berm is accompanied by incision of upslope channels due to increased slope as the channels attempt to grade themselves to the bottom of the ditch. Enhanced erosion, on the other hand, is expected whenever a relatively unerodible surface sheds water discharge onto a more erodible surface, as shown in Fig. 8.

Fig. 11: Photograph of erosion features in lava flows in the Cima Dome area. Lack of sediment loading of water discharge running off the downstream end of the flows leads to enhanced incision into the pediment, and ultimately to gully (canyon) cutting into the lava itself. Similar erosion processes are expected to occur downslope of artificially constructed non-erodible surfaces.

Fig. 12: Gullies on artificial embankment could be modeled with standard sediment transport algorithms, but see Fig. 13.

Fig. 13: Characteristics of naturally occurring gullies in alluvial material (below the survey tripod) could not be matched by standard power-law sediment transport algorithms, indicating that these sediment transport rules may not always be adequate to explain details of erosion patterns.

Fig. 14: (a) Schematic flow diagram of program WATERBOT; (b) flow diagram indicating mapping of topographic information.
List of Publications and Abstracts

The following work was supported wholly or in part by ARO Grant Nos. : 30685-EV and 34207-EV-AAS:

Constitutive Laws and Prediction in Granular Systems
P. K. Haff

Vertical Mixing of Grains During Bedload Transport
P. K. Haff

Transport of Solids by Flowing Surface Water: Constitutive Rules for Simulation of Large-Scale Erosion of Sediments
J. Raghuraman and P. K. Haff

Test of Scale Invariance of Hydrological Constitutive Laws
Allen G. Hunt

Clast Diffusion and Storm History of Desert Pavement
P. K. Haff

Gully-Head Dynamics on Desert Pavement, Mojave Desert
P. K. Haff

Overturned Stones, Vegetation, and Stability of Desert Pavement Surfaces
P. K. Haff

Limitations on Predictive Modeling in Geomorphology
P. K. Haff
Dynamical Processes on Desert Pavement and the Healing of Surficial Disturbances
P. K. Haff and B. T. Werner

Microtopography as an Indicator of Modern Hillslope Diffusivity in Arid Terrain
R. Jyotsna and P. K. Haff
Geology, 25, 695-698 (1997)

Why Prediction of Grain Behavior is Difficult in Geological Granular Systems
P. K. Haff
Powders and Grains 97, Proc. of 3rd Intl. Conf. on Powders and Grains,
Durham, NC, ed. R. P. Behringer and J. T. Jenkins, pp 61-64, Balkema,
Rotterdam 1997

The Relation of Surface Characteristics to Landscape Evolution Processes in Arid Terrain
P. K. Haff
Eos Trans. AGU, Spring Meet. Suppl., 1997

Landscape Evolution Using Digital Elevation Models of the Cima Dome Area, Mojave Desert, California
L. R. Boring and P. K. Haff

An Empirical Model of Large Scale Sediment Transport in Arid Terrain: Application to Basalt Flow Erosion and Pediment Evolution near the Cima Volcanic Field, Mojave Desert, California

Sediment Dynamics of Canyons and Fans of the Black Mountains, Death Valley, California.
P. K. Haff

In preparation:

Climatic Dependence of Soil Erosion on a Small Hill in the Mojave Desert.
A. G. Hunt and Q. Joan Wu

A Probabilistic Treatment of Fluvial Entrainment of Cohesionless Particles.
A. G. Hunt
Models of Erosion and Deposition on Cima Dome, California
L. R. Boring
Invited Talks

"Localized Energy Dissipation in Strained Granular Material", 1st North American Workshop on Modeling the Mechanics of Off-Road Mobility, at US Army Engineer Waterways Experiment Station, Vicksburg, MS, 1995.


"Desert Storms and Desert Pavement", at Duke University Quaternary Seminar, 1995


Scientific Personnel Supported During Grant Period

P. K. Haff, Principal Investigator (30685-EV)

Allen G. Hunt, Graduate Student (34207-EV-AAS)

Lonny Boring, Graduate Student (34207-EV-AAS)

Brian Smith, Graduate Student (30685-EV)

Jyotsna Raghuraman. Postdoctoral Fellow (30685-EV)

Degrees Awarded

Allen G. Hunt
MS degree

Lonny R. Boring
MS Degree to be awarded may 1999
Figure 7.
Figure 8
Figure 9
Figure 10
WATERBOT Model Flow Chart
Mapping of Topographic Information

Legend
- Index Files
- Main Program
- 1st Order Subroutine
- 2nd Order Subroutine
- 3rd Order Subroutine

Figure 14b
Appendix: FORTRAN program WATERBOT
PROGRAM EROSION8
*
$noex

implicit none

c Declaration of world variables
include 'E8variables.fi'

c Assemble the native formate DEMs
  call assemble(Z,space,south,north,west,east)
  call alluvium(Z,cover,geology,south,north,west,east)

c Read geology TIF file to delineate areas of basalt
  ! call lava(Z,cover,geology,basaltk,basaltbot,
  ! > south,north,west,east)

c Fill pits in DEM data
  call fill(Z,step1,step2,stream,south,north,west,east,
     > Wro,Wco)

c Map change in gradient
  ! call gradient(Z,DelG,south,north,west,east)

c Call Sediment Transport subroutine
  call transport5(Z,cover,modify,geology,basaltk,basaltbot,
     > step1,step2,stream,n,south,north,west,east,Wro,Wco,counter2)

c Delineate stream channels
  ! call channel1(Z,stream,counter1,step1,step2,n,south,north,
  ! > west,east,Wro,Wco)

c Call subroutine to generate shaded relief map
  ! call shade(Z,shadow,space,south,north,west,east)

end PROGRAM EROSION8
subroutine assemble(Z, space, south, north, west, east)

* $noex
  implicit none

c Declaration of variables
  include 'E8ArraySize.fi'
  integer :: I       ! Row index
  integer :: J       ! Column index
  integer :: M       ! Index to real in all elevation inputs
  integer :: N       ! Index to read in each input file
  integer :: unit1   ! Number of lines for DEM south-to-north array
  integer :: unit2   ! Column number of DEM
  integer :: unit3
  integer :: number  ! Number of elevation in south-to-north array
  integer :: numfiles ! Number of input DEMs to assemble
  integer :: Rows, Columns !
  integer :: Xposition, Yposition
  integer :: K
  integer :: G
  integer :: south, north, west, east ! Values of map boundaries
  integer :: flag    ! Flag to write output files
  integer :: Seed    ! Random seed variable for random number

  real :: Xcoor      ! UTM X coordinate
  real :: Ycoor      ! UTM Y coordinate
  real :: Xdatum     ! Value to transform UTM to working array
  real :: Ydatum     ! Value to transform UTM to working array
  real :: Xmax, Ymax ! Value to transform UTM to working array
  real :: Ymaxtest
  real :: sealevel
  real :: minelev
  real :: maxelev
  real :: ran2
  real :: random

  real, dimension(MaxR,MaxC) :: Z ! Elevation array for study area
  real, dimension(MaxR) :: elev ! Holds elevation data
  real :: space ! Horizontal resolution of USGS DEMs (m)

  parameter (numfiles = 6)

  call system_clock( Seed )

c Input file names and units
  open (unit=1, file='C:\MSDEV\Projects\Inputs\S_DEM.txt',
  > status='unknown')

2
open (unit=2, file='C:\MSDEV\Projects\Inputs\MM DEM.txt',
> status='unknown')
open (unit=3, file='C:\MSDEV\Projects\Inputs\C DEM.txt',
> status='unknown')
open (unit=4, file='C:\MSDEV\Projects\Inputs\GS DEM.txt',
> status='unknown')
open (unit=5, file='C:\MSDEV\Projects\Inputs\CC DEM.txt',
> status='unknown')
open (unit=6, file='C:\MSDEV\Projects\Inputs\CD DEM.txt',
> status='unknown')
open (unit=1, file='C:\MSDEV\Projects\Inputs\BB DEM.txt',
> status='unknown')
open (unit=1, file='C:\MSDEV\Projects\Inputs\HC DEM.txt',
> status='unknown')
open (unit=1, file='C:\MSDEV\Projects\Inputs\GP DEM.txt',
> status='unknown')
open (unit=2, file='C:\MSDEV\Projects\Inputs\IM DEM.txt',
> status='unknown')
open (unit=1,
> file='C:\MSDEV\Projects\Inputs\Bill DEMs\pintwells_CA.txt',
> status='unknown')
open (unit=1,
> file='C:\MSDEV\Projects\Inputs\Bill DEMs\summerford_NM.txt',
> status='unknown')

write (*,*) 'Assembling Information...

Xdatum = 1000000000.0
Ydatum = 1000000000.0
Xmax = 0.0
Ymax = 0.0

c Scan data set to determine UTM location
do N = 1, numfiles
  do M = 1, 500
    read(N, *, end = 10) unit1,unit2,number,unit3,Xcoor,Ycoor
    if (Xcoor .LE. Xdatum) then
      Xdatum = Xcoor
    endif
    if (Xcoor .GE. Xmax) then
      Xmax = Xcoor
    endif

if (Ycoor .LE. Ydatum) then
    Ydatum = Ycoor
endif

Ymaxtest = Ycoor + (float(number)-1.0)*space
if (Ymaxtest .GE. Ymax) then
    Ymax = Ymaxtest
endif

enddo

10 continue
    rewind N
enddo

c         Determine number of array rows and columns
Columns = INT(Xmax-Xdatum)/space
Rows = INT(Ymax-Ydatum)/space

c         Establish working array of elevation data
do N = 1, numfiles
    do M = 1, Columns
        read(N, *, end = 20) unit1,unit2,number,unit3,Xcoor,
        Ycoor,sealevel,minelev,maxelev,(elev(K), K=1, number)
        Xposition = INT((Xcoor-Xdatum)/space) + 1
        Yposition = INT((Ycoor-Ydatum)/space) + 1
        do G = 1, number
            random = ran2( Seed )*0.5 - 0.25
            Z(Yposition,Xposition) = elev(G) + random
            if (Yposition .EQ. 252 .and. Xposition .GE. 469)
                if (Xposition .EQ. 469) then
                    Z(Yposition,Xposition) = 1201.25
                else if (Xposition .EQ. 470) then
                    Z(Yposition,Xposition) = 1201.40
                else
                    Z(Yposition,Xposition) = 1201.50
                endif
            endif
            Yposition = Yposition + 1
        enddo
    enddo
enddo
enddo
20 continue
   rewind N
   enddo

! The follow extracts workable section for Cima Volcanic Field
! south = 120
! north = 350
! west = 10
! east = 620

! Subroutine finds south, north, west, east values of working array
! call boundary(Z,Rows,Columns,south,north,west,east)

  print*, 'Write OriginalZ.out and ERMapper.out Elevation files?'
  print*, '1 == Yes, 0 == No'
  read*, flag

  if (flag == 1) then
    Output file names and units
    open (unit=30, file='OriginalZ.out', status='unknown')
    open (unit=50, file='ERMapperZ.out', status='unknown')

    Write working array of elevation data to file.out
    do I = north, south, -1
      write(30,40) (Z(I,J), J = west, east)
    enddo

    write XYZ ASCII file for import into ERMapper
    do I = south, north
      do J = west, east
        write(50,60) float(J*30), float(I*30), Z(I,J)
      enddo
    enddo

    Close output files
    close (unit=30)
    close (unit=50)
  endif

! Format statements
40 format (1X, 5000E15.8E2)
60 format (1X, 5000F12.4)
80 format (1X, 2E15.8E2, 1F12.4)

! Close files
do l = 1, numfiles
    close (unit=l)
  enddo

  c ending the subroutine
  return
end
subroutine boundary(Z,Rows,Columns,south,north,west,east)

*noex
implicit none

include 'E8ArraySize.fii'

! Declaration of variables
integer i, south, north, west, east, Rows, Columns, No, Ea
real Z
dimension Z(MaxR,MaxC)

! Find working boundaries to elevation matrix
do i = 1, 15
  if (Z(i,Columns) .LT. 1.0) then
    south = south + 1
  endif
  if (Z(1,i) .LT. 1.0) then
    west = west + 1
  endif
  if (Z(Rows-i,1) .LT. 1.0) then
    No = No + 1
  endif
  if (Z(Rows,Columns-i) .LT. 1.0) then
    Ea = Ea + 1
  endif
enddo

south = south + 1
north = Rows - (No+1)
west = west + 1
east = Columns - (Ea+1)

return
end
subroutine alluvium(Z,cover,geology,south,north,west,east)

*\$noex
    implicit none
    Declare variables
include 'E8ArraySize.fi'

integer :: south, north, west, east  !Location of map boundaries
integer :: I, J  !Row (I) and Column (J) indices
real, dimension(MaxR,MaxC) :: Z  !Elevation array
real, dimension(MaxR,MaxC) :: cover  !Array of thickness of alluvial cover

integer, dimension(MaxR,MaxC) :: geology  !Array of geology type
real :: dep  !Depth of alluvial cover

write (*,*) 'Initializing Alluvial Cover...'

Initialize cover to depth "dep"
do I = south, north  
do J = west, east  
    dep = Z(I,J)  
    cover(I,J) = dep  
    geology(I,J) = 0  
enddo  
enddo

return
end
subroutine lava(Z,cover,geology,basalthk,basaltbot,
>       south,north,west,east)

*noex
  implicit none

character(20) :: E8ArraySize

integer :: south, north, west, east      ! Integer corners of Study area
integer :: I, J                           ! Row (I) and Column (J) indices
integer, dimension(MaxR,MaxC) :: geology
real, dimension(MaxR,MaxC) :: Z            :: Z
real, dimension(MaxR,MaxC) :: cover        :: cover
real, dimension(MaxR,MaxC) :: basalthk      :: basalthk
real, dimension(MaxR,MaxC) :: basaltbot     :: basaltbot
real, parameter           :: thickness = 3.0

c Open Geology TXT file
open(unit=10, file='C:\MSDEV\Projects\Inputs\CVFGeology.txt',
>       status='old')

c Initialize cover depth
!call alluvium(Z,cover,geology,south,north,west,east)

write (*,*) 'Placing Lava Flows/Resistant Bedrock...

    do I = north, south, -1
       read(10, *, end=100) (geology(I,J), J = west, east)
    enddo
100 continue

    do I = south, north
       do J = west, east

           if (geology(I,J) .EQ. 176) then  ! Read GEOLOGY from TIFF
              basalthk(I,J) = thickness
              basaltbot(I,J) = Z(I,J) - thickness
              geology(I,J) = 1
              cover(I,J) = 0.0
           else
              geology(I,J) = 0
           endif

           ! geology(I,J) = 0       ! Make all material PEDIMENT Type
enddo
enddo

Write output of DelG to output file
open(unit=10, file='Geology.out', status='unknown')
open(unit=30, file='BasThk.out', status='unknown')
open(unit=50, file='BasBot.out', status='unknown')
open(unit=70, file='Cover.out', status='unknown')

d o I = north, south, -1
   write(10,20) (geology(I,J), J = west, east)
   write(30,40) (basalthk(I,J), J = west, east)
   write(50,40) (basaltbot(I,J), J = west, east)
   write(70,40) (cover(I,J), J = west, east)
endo d

do I = 10, 70, 10
   close (unit=I)
endo d

20 format (1X, 5000I12)
40 format (1X, 5000E15.8E2)

return
end
subroutine fill(Z,step1,step2,stream,south,north,west,east, Wro,Wco)

**$noex**

implicit none

c Declaration of variables
include 'E8ArraySize.fi'

real :: modify(MaxR,MaxC) !2D array of amount of elevation change (+/-)

!MODIFY is calculated/updated in the TRANSPORT subroutines
integer :: stream(NumCells,3) !Contains (Column,Row,Integer Location of next step position)

!STREAM is created in the RANORDER subroutine
integer :: depress(NumCells,3) !Contains (Column,Row,Integer Location of next step position)
integer :: step2(NumCells) !Contains integer position (Translatable to Row,Column)
integer :: south, north, west, east !Location of map boundaries
integer :: Wro !Number of working Rows
integer :: Wco !Number of working Columns
integer :: n !Number of topographic cells
integer :: A, B, F, S !Array indices
integer :: X, Y !Position in map
!(Row == Y, Column == X)
integer :: count !Counter index to assign values
!within Depress array

c Array indices
integer :: I !Row index
integer :: J !Column index

real, dimension(MaxR,MaxC) :: Z !Elevation matrix
real :: diffelev !Difference in Elev of current cell and neighbor
real :: mindiff !Holds value of Minimum difference in elevation
real :: step1(NumCells) !Random numbers to sort and generate random waterbot drop

n = Wro*Wco

c Call subroutine spatial to generate 1D array of elevations

c Subroutine 'ranorder' is required for both Subroutines 'transport'
call ranorder(step1,step2,stream,n,south,north,west,east,Wro,Wco)
c Call subroutine network to calculate and store position of maximum gradient
call network(Z, stream, south, north, west, east, Wro, Wco)

do F = 1, 200

write (*, *) 'Filling Depressions...', F
count = 0

c Scan all positions in stream looking for depressions
do S = 1, n

   if (stream(S,3) .GT. n) then
      count = count + 1
      depress(count,3) = stream(S,3)
      depress(count,1) = stream(S,1)
      depress(count,2) = stream(S,2)
      X = stream(S,1)
      Y = stream(S,2)
      mindiff = 100.0

      do A = -1, 1
         do B = -1, 1

            difflev = Z(Y+A,X+B) - Z(Y,X)
            if (difflev .LT. mindiff .AND. difflev .NE. 0.0) then
               mindiff = difflev
            else
               endif

         enddo
      enddo
   endif
   else
      endif
   endif

Re-establish the drainage network
call netfill(Z, stream, depress, count, south, north, west, east,
>       Wro, Wco)
c  Write output to output files
open(unit=30, file='FilledZ.out', status='unknown')
open(unit=50, file='FilMod.out', status='unknown')

do l = north, south, -1
  *  write(50,400) (modify(l,J), J = west, east)
  *  write(30,400) (Z(l,J), J = west, east)
endo

close (unit=30)
close (unit=50)

400  format (1X, 5000E15.8E2)

return
end
 subroutine ranorder(step1, step2, stream, n, south, north, west, east, 
>       Wro, Wco)

*$noex
 implicit none

 c Declaration of variables
 include 'E8ArraySize.fi'

 integer :: stream(NumCells, 3) ! Contains (Column, Row, Integer Location
 of next step position)

 ! STREAM is created in the RANORDER subroutine
 integer :: step2(NumCells) ! Contains integer position
 ! Translatable to Row, Column
 integer :: south, north, west, east ! Location of map boundaries
 integer :: Wro ! Number of working Rows in map.
 ! Created in RANORDER
 integer :: Wco ! Number of working Columns in map.
 ! Created in RANORDER
 integer :: n ! Number of topographic cells in map
 integer :: Seed ! Random seed for number generator
 integer :: I, J ! Row (I) and Column (J) indices
 integer :: position ! Counter index for position pointer in
 ! Step arrays

 real :: step1(NumCells) ! Random numbers to sort and generate
 ! Random waterbot drop
 real :: ran3 ! Random number function (-ve input)
 real :: jack ! Random number b/t 0.0 and 1.0E06

 write (*, *) 'Ordering Information...'

 call system_clock( Seed ) ! Call clock to generate Seed Variable
 Seed = -Seed ! Function ran3 uses (-ve) Seed

 Wro = north-south+1
 Wco = east-west+1
 position = 0

 do J = west, east
    do I = south, north

       position = position + 1

       jack = ran3( Seed )*1000000.0

       step1(position) = jack

    end do

 end do
step2(position) = position
stream(position,1) = J
stream(position,2) = I

enddo
enddo

n = position

Call sort2 to organize order of random particle drops
call sort2(n,step1,step2)

return
end
subroutine network(Z,stream,south,north,west,east,
>       Wro,Wco)

*noex
implicit none

! Declaration of variables
include 'E8ArraySize.fi'

integer :: stream(NumCells,3) !Contains (Column,Row,Integer Location
   !of next step position)

!STREAM is created in the RANORDER subroutine
integer :: south, north, west, east  !Location of map boundaries
integer :: Wro                !Number of working Rows in map.
   !Created in RANORDER
integer :: Wco                !Number of working Columns in map.
   !Created in RANORDER
integer :: I,J                !Row (I) and Column (J) indices
integer :: position          !Counter index for position pointer in
   !step arrays

! Position Indices
integer :: A                   !Scan Row
integer :: B                   !Scan Column
integer, parameter :: scan = 1 !Defines size of box to scan gradients
integer :: AmaxG               !Holds Scan Row of maximum gradient
integer :: BmaxG               !Holds Scan Column of maximum gradient

! Topography variables, Directionality variables
real :: DeltaZ                 !Elev. difference b/t current cell and neighbor
real, dimension(MaxR,MaxC) :: Z !Elevation matrix
real :: max                    !Holds value of maximum gradient (using RHO8)
real :: rho8                   !Fairfields (1991) directionality variable
integer :: Seed                !Random seed variable for random number
real :: ran2                   !Random number generator 2 (+ve input)
real :: random                 !Random number b/t 0.0 and 1.0

! Variables for file output
integer :: S
integer :: P
integer :: flag

write (*,*) 'Creating Network Information...'
call system_clock( Seed ) !Call clock to generate Seed Variable

c ! Scan each elevation bin in working elevation matrix w/in edges
do I = south+1, north-1
   do J = west+1, east-1

16
AmaxG = 0  
BmaxG = 0  
max = 0.0

do A = -scan, scan, 1
  do B = -scan, scan, 1
    c Delineate stream network with Rho8
    c Calculate grad for scan .EQ. 1
    c Cardinal directions
    if ( (B .EQ. 0 .AND. (A .EQ. +scan .OR. A .EQ. -scan)) .OR. (A .EQ. 0 .AND. (B .EQ. -scan .OR. B .EQ. +scan))) then
      DeltaZ = (Z(I,J)-Z(I+A,J+B))
      DeltaZ = (Z(I,J)-Z(I+A,J+B))
    !
    c Diagonal directions
    else
      random = ran2(Seed)
      rho8 = 1.0 / (2.0-random)
      DeltaZ = rho8 * (Z(I,J)-Z(I+A,J+B))
      DeltaZ = (Z(I,J)-Z(I+A,J+B))  
    endif
    c Check for maximum DeltaZ
    if(DeltaZ .GE. max) then
      max = DeltaZ
      AmaxG = A
      BmaxG = B
    endif
    !
    c End adjacent bin scanning do loops
  enddo
enddo

c Determine position in step array from to I,J location
position = I-south + (J-west)*Wro + 1
Record the step location in network matrix

if (AmaxG .EQ. 0 .and. BmaxG .EQ. 0) then
  stream(position,3) = Wro*Wco+10
else
  stream(position,3) = position + AmaxG +
                  (BmaxG*(Wro))
endif

End do loops to check each elevation bin enddo
do I = 1, Wro, 1
  stream(I,3) = 0
  stream(I,3) = 0
endo

do I = (Wro*Wco), (Wro*Wco-Wro+1), -1
  stream(I,3) = 0
  stream(I,3) = 0
endo

do J = 1, (Wro*Wco), Wro
  stream(J,3) = 0
  stream(J,3) = 0
endo

do J = Wro, (Wro*Wco), Wro
  stream(J,3) = 0
  stream(J,3) = 0
endo

print*, 'Write Stream.out output file?'
print*, '1 == Yes, 0 == No'
read*, flag

if (flag == 1) then
  open(unit=10, file='Stream.out', status='unknown')
  do S = 1, (Wro*Wco)
      write(10,200) (stream(S,P), P = 1,3)
  enddo
  close(unit=10)
endif

200 format(1X, 3I10)
return
end
subroutine netfill(Z,stream,depress,count,south,north,west,east,
> Wro,Wco)

"$noex

implicit none

C Declaration of variables
include 'E8ArraySize.fi'

integer :: stream(NumCells,3) !Contains (Column,Row,Integer Location
of next step position)

STREAM is created in the RANORDER subroutine
integer :: south, north, west, east !Location of map boundaries
integer :: Wro !Number of working Rows in map.
!Created in RANORDER
integer :: Wco !Number of working Columns in map.
!Created in RANORDER
integer :: I,J !Row (I) and Column (J) indices
integer :: position !Counter index for position pointer in step arrays
integer :: count !Number of depressions in map
integer :: N !Do loop index
integer :: depress(NumCells,3) !Array containing Column,Row,Location
of depression

C Position Indices
integer :: A !Scan Row
integer :: B !Scan Column
integer, parameter :: scan = 1 !Defines size of box to scan gradients
integer :: AmaxG !Holds Scan Row of maximum gradient
integer :: BmaxG !Holds Scan Column of maximum gradient

C Topography variables, Directionality variables
real :: DeltaZ !Elev. difference b/t current cell and neighbor cells
real, dimension(MaxR,MaxC) :: Z !Elevation matrix
real :: max !Holds value of maximum
!gradient (using RHO8)
real :: rho8 !Fairfields (1991) directionality variable
integer :: Seed !Random seed variable for random number
real :: ran2 !Random number generator 2 (+ve input)
real :: random !Random number b/t 0.0 and 1.0

write (*,*) 'Reestablising Network Information...',
> 'Number of Depressions = ', count

call system_clock( Seed ) !Generate random seed variable for ran2

C Scan each elevation bin in working elevation matrix w/ edges
do N = 1, count
do I = (depress(N,2)-scan), (depress(N,2)+scan)
do J = (depress(N,1)-scan), (depress(N,1)+scan)
   if (I .GT. south .and. I .LT. north .and.
       J .GT. west .and. J .LT. east) then
      AmaxG = 0
      BmaxG = 0
      max = 0.0
      c
      Delineate stream network with Rho8
      do A = -scan, scan, 1
        do B = - scan, scan, 1
          c
          Calculate grad for scan .EQ. 1
          c
          Cardinal directions
          if (( B .EQ. 0 .AND. (A .EQ. +scan .OR. A .EQ.
             -scan)).OR. (A .EQ. 0 .AND. (B .EQ. -scan
             .OR. B .EQ. +scan))) then
            DeltaZ = (Z(I,J)-Z(I+A,J+B))
            DeltaZ = (Z(I,J)-Z(I+A,J+B))/30.0
          c
          Diagonal directions
          else
            Random = ran2( Seed )
            rho8 = 1.0 / (2.0-random)
            DeltaZ = rho8 * (Z(I,J)-Z(I+A,J+B))
            DeltaZ = (Z(I,J)-Z(I+A,J+B))/(42.4264)
          endif
          c
          Check for maximum DeltaZ
          if (DeltaZ .GE. max) then
            max = DeltaZ
            AmaxG = A
            BmaxG = B
          endif
        c
        End adjacent bin scanning do loops
      enddo
    endd
c Determine position in step array from to I,J location
  position = I-south + (J-west)*Wro + 1

c Record the step location in network matrix
  if (AmaxG .EQ. 0 .and. BmaxG .EQ. 0) then
    stream(position,3) = Wro*Wco+10
  else
    stream(position,3) = position + AmaxG +
      (BmaxG*(Wro))
  endif
  endif

  End do loops to check each elevation bin
  enddo
  enddo

endo
doo

  Put 0 also in edge bins
  do I = 1, Wro, 1
    stream(I,3) = 0
    stream(I,3) = 0
  enddo
  do I = (Wro*Wco), (Wro*Wco-Wro+1), -1
    stream(I,3) = 0
    stream(I,3) = 0
  enddo

  do J = 1, (Wro*Wco), Wro
    stream(J,3) = 0
    stream(J,3) = 0
  enddo
  do J = Wro, (Wro*Wco), Wro
    stream(J,3) = 0
    stream(J,3) = 0
  enddo

return
end

22
subroutine transport5(Z,cover,modify,geology,basalthk,basaltbot, > step1,step2,stream,n,south,north,west,east,Wro,Wco,counter2)  
*noex
implicit none

Declaration of variables
Defining common ARRAY dimensions
include 'E8ArraySize.f'

Two-D ARRAYs (Matrix of Data)
real :: Z(MaxR,MaxC)  !2D array of elevation data
IZ is created in the ASSEMBLE subroutine
real :: cover(MaxR,MaxC)  !2D array of thickness of alluvial cover
ICOVER is created in the ALLUVIUM
real :: modify(MaxR,MaxC)  !2D array of amount of elevation change

!MODIFY is calculated/updated in the TRANSPORT subroutines
integer :: counter2(MaxR,MaxC)  !2D array containing number of
!WATERBOTS passing thru each cell

!COUNTER2 is calculated/updated in the TRANSPORT subroutines
integer :: geology(MaxR,MaxC)  !GEOLGY contains information on the
!type of material present in cell

!GEOLGY is created in the LAVA subroutine
real :: basalthk(MaxR,MaxC)  !BASALTHK contains the thickness of
!the Basalt in the Cell

!BASALTHK is created in the LAVA subroutine
real :: basaltbot(MaxR,MaxC)  !BASALTBOT contains the elevation of
!the basalt bottom

!BASALTBOT is created in the LAVA subroutine
real :: BotCount(MaxR,MaxC)  !BOTCOUNT contains the Count of
!waterbot when basalt was breached

!BOTCOUNT is output to files after each iteration
character*16 :: filename(40,6)  !FILENAME holds the output file names

One-D ARRAYs
real :: step1(NumCells)  !1D array of random numbers b/t 1, NumCells
!used to sort random waterbot drop sequence

!STEP1 is created in the RANORDER subroutine
integer :: step2(NumCells)  !1D array of unique integer location
!((translatable to Row,Column) of each
!elevation

23
ISTEP2 is created in the RANORDER subroutine
integer :: stream(NumCells,3)  !Contains (Column,Row,Integer Location of next step position)

ISTREAM is created in the RANORDER subroutine

DEM variables Size, Dimensions, Etc.
integer :: south  !Value of Row (+ve from south) of Southern Boundary
integer :: north  !Value of Row (+ve from south) of Northern Boundary
integer :: west   !Value of Row (+ve from west) of Western Boundary
integer :: east   !Value of Row (+ve from west) of Eastern Boundary
integer :: n      !Number of elevation cells in the Z array

Calculated in RANORDER subroutine
integer :: Wro   !Number of Rows in working Z array, in RANORDER subroutine
integer :: Wco   !Number of Columns in working Z array, in RANORDER subroutine
real :: space = 30.0  !Horizontal resolution of USGS DEMs (m)
integer :: count   !Counts number of times Subroutine is called

real :: outH       !Thickness of material leaving topo cell (-ve)
real :: fraction   !Fraction of total material to erode that is contained w/n alluvium
real :: degree = 20.0*(0.01745)  !Threshold slope to move basalt material
                                 !((Degree*Radian Conversion)
real :: Bthreshk = 0.05     !Threshold thickness of basalt to treat as basalt material
real :: Bthresconcen = 250   !Threshold concentration to move basalt counter 2 counts waterbots previously moved thru current cell

Declaration of local variables
include 'E8TransVar.fi'
include 'E8SubTrans.fi'

integer :: file   !Index used to close output files

call system_clock( Seed )  !Call clock for seed

Initialize FILENAME array for output file names.
open(unit=150, file='FileName.txt', status='old')
open(unit=250, file='HeadChan.out', status='unknown')

do l = 1, INT(Totyears/Time)
   read(150,*) (filename(l,J), J = 1, 6)
   write(*,*) (filename(l,J), J = 1, 6)
endo
close(unit=150)

c  Initialize position locations in DeltaZ array (Row,Column,(Elev. Diff))
      C = 0
      do B = -scan, scan, 1
          do A = -scan, scan, 1

              C = C + 1
              DeltaZ(C,1) = A
              DeltaZ(C,2) = B

          enddo
      enddo

c  call ranorder(step1,step2,stream,n,south,north,west,east,
                     >   Wro,Wco)

c  DO LOOP to evolve landscape thru time
      do T = 1, INT(Totyears/Time)

          write('**') 'Calculating Sediment Transport...', T, ' of',
                     >   INT(Totyears/Time)

          count = 0

      c  Reset counter2 array to zero
          do I = south, north
              do J = west, east

                  counter2(I,J) = 0

              enddo
          enddo

c  DO LOOP to hit each Row,Column position in Z array
      do S = 1, n

          I = stream(step2(S),2) !Extract Row from STREAM array
          J = stream(step2(S),1) !Extract Column from STREAM array

          inH = 0.0 !Initialize inH to Zero
          Rolls = 0.0 !Initialize Rolls to Zero

c  DO LOOP to follow Waterbot position and calculate to boundary
      do while (I >= south+1 .AND. I <= north-1
               >   .AND. J >= west+1 .AND. J <= east-1)

          ! Code...

      enddo

25
AmxG = 0  // Initialize Row to ZERO
BmaxG = 0  // Initialize Column to ZERO
max = 0.0  // Initialize MaxGrad to ZERO
DeltaH = 0.0  // Initialize Elevation diff to ZERO
Rolls = Rolls + 1.0  // Increment the Rolls
C = 0  // Set DeltaZ array to ZERO
counter2(I,J) = counter2(I,J) + 1
  // Increment COUNTER2 by one

Scan Eight Neighbors and locate maximum gradient
  do B = -scan, scan, 1  // B to Columns
    do A = -scan, scan, 1  // A to Rows

      C = C + 1
      // Cardinal Directions
      if ((B==0.AND. (A==+scan.OR.
        A==-scan)).OR. (A == 0 .AND.
        (B == +scan .OR. B == -scan))) then

        DeltaZ(C,3) = (Z(I,J)-Z(I+A,J+B))
        Delta = DeltaZ(C,3)

        // Corners
        else

        random = ran2( Seed )
        // Generate random number
        rho8 = 1.0/(2.0-random)
        // Calculate RHO8

        DeltaZ(C,3) = (Z(I,J)-Z(I+A,J+B))
        Delta = rho8*DeltaZ(C,3)

      endif

      // Check for maximum gradient to neighbors
      if (Delta >= max) then
        max = Delta
        maxdelZ = DeltaZ(C,3)
        AmxG = A
        BmaxG = B
      endif

    enddo  // END A (Row) scan of neighbors

    enddo  // END B (Column) scan of neighbors

26
if (AmaxG == 0 .AND. BmaxG == 0) then
  !Waterbot is in a Hole

  min = -100.0
  do C = 1, 9
    if (DeltaZ(C,3) > min .AND. C /= 5) then

      min = DeltaZ(C,3)
      AmaxG = DeltaZ(C,1)
      BmaxG = DeltaZ(C,2)

    endif
  enddo

  min = min-0.0001

  c
  If sed. load is enough to fill depression
  if (inH > -min) then
    DeltaH = -min
    cover(I,J) = cover(I,J) + DeltaH
    Z(I,J) = Z(I,J) + DeltaH
    modify(I,J) = modify(I,J) + DeltaH
    inH = inH - DeltaH
    I = I + AmaxG
    J = J + BmaxG
    counter2(I,J) = counter2(I,J) + 1
    goto 20
  endc

  c
  If sed. load is not enough to fill depression
  else
    DeltaH = inH
    cover(I,J) = cover(I,J) + DeltaH
    Z(I,J) = Z(I,J) + DeltaH
    modify(I,J) = modify(I,J) + DeltaH
    counter2(I,J) = counter2(I,J) + 1
    goto 30
  endif

else
  !Waterbot is not in a Hole
  length = (SQRT((float(AmaxG))**2 +
  !Calculate distance next step
  (float(BmaxG))**2))*space
  grad = maxdelZ / length
  !Calculate gradient to next step
  grad = 0.0
endif
Find the desired change in cell height (outH)

\[
\text{outH} = \text{Const} \times \text{Rate} \times \text{Time} \times \text{grad}
\]

if (Rolls < step) then

\[
\text{outH} = \text{Const} \times \text{Rate} \times \text{Time} \times \text{grad} \times (\text{Rolls}/\text{step})
\]

else

\[
\text{outH} = \text{Const} \times \text{Rate} \times \text{Time} \times \text{grad}
\]

endif

\[
\Delta \text{H} = \text{iH} - \text{outH}
\]

Adjust Z to Erode or Deposit difference between outH and inH

If Alluvium is Greater Than Erosion amount

if (\Delta \text{H} < 0.0 \text{ AND} \ (-\Delta \text{H}) <= \text{cover}(1,J))

then

\[
\text{cover}(1,J) = \text{cover}(1,J) + \Delta \text{H}
\]

If Alluvium is Less Than (<) Erosion amount

else if (\Delta \text{H} < 0.0 \text{ AND} \ (-\Delta \text{H}) >

\[
\text{cover}(1,J)
\]

then

\[
\text{fraction} = \text{cover}(1,J) / \Delta \text{H}
\]

if (\text{geology}(1,J) == 1 \text{ AND} \ (\text{grad} >=

\[
\text{TAN} \times \text{(degree)} \text{ OR} \ \text{counter2}(1,J) \geq \text{Bthreshconc})
\]

then

\[
\text{weight} = \text{float} \times \text{counter2}(1,J)^{\text{Bpower}-1.0}
\]

\[
\text{outH} = \text{Const} \times \text{Rate} \times \text{Time} \times (\text{grad} \times 0.001) \times \text{weight} \times \text{(1.0-fraction)}
\]

\[
\Delta \text{H} = (-\text{cover}(1,J) + \text{outH})
\]

basalthk(1,J) = basalthk(1,J) - outH

if (basalthk(1,J) <= \text{Bthreshk}) then

basalthk(1,J) = 0.0

\text{geology}(1,J) = 0

\text{cover}(1,J) = \text{Z}(1,J) + \Delta \text{H}

\text{BotCount}(1,J) = \text{counter2}(1,J)

\text{write}(250,800) \ J, I, \text{counter2}(1,J), \ T

endif

else

endif
DeltaH = -cover(I,J)

endif

If Material is being Deposited NOT Eroded
else

cover(I,J) = cover(I,J) + DeltaH
endif

Z(I,J) = Z(I,J) + DeltaH
modify(I,J) = modify(I,J) + DeltaH
inH = inH - DeltaH

I = I + AmaxG
J = J + BmaxG

ENDDO of follow Waterbot calculate Sed. Trans.
continue
endo

ENDDO to hit each Row,Column position in Z array
continue
endo

Write output to output files
open(unit=T*1, file=filename(T,5), status='unknown')
open(unit=T*3, file=filename(T,2), status='unknown')
open(unit=T*5, file=filename(T,1), status='unknown')
open(unit=T*7, file=filename(T,4), status='unknown')
open(unit=T*9, file=filename(T,3), status='unknown')
open(unit=T*11, file=filename(T,6), status='unknown')

do I = north, south, -1
  write(T*1,200) (counter2(I,J), J = west, east)
  write(T*3,400) (Z(I,J), J = west, east)
  write(T*5,400) (modify(I,J), J = west, east)
  write(T*7,400) (basaltthk(I,J), J = west, east)
  write(T*9,400) (cover(I,J), J = west, east)
  write(T*11,400) (BotCount(I,J), J = west, east)
endo

do file = T*1, T*11, T*2
  close(unit=file)
endo
Lower the Western boundary by 2cm/1000 yrs.

```
  do I = south, north
      Z(I,west) = Z(I,west) - ((0.013/1000.0)*Time)
      modify(I,west) = modify(I,west) - ((0.013/1000.0)*Time)
  enddo

ENDDO to evolve landscape through time
```

```
write XYX ASCII file for import into ERMapper
  open(unit=110, file='ERMapperZ.out', status='unknown')
  do I = south, north
      do J = west, east
          write(110,600) float(J*30), float(*30), Z(I,J)
      enddo
  enddo
  close (unit=110)

close(250)
```

```
200 format (1X, 5000I12)
400 format (1X, 5000E15.8E2)
600 format (1X, 5000F12.4)
800 format (1X, 2I6, 1I12, 1I6)

return
end
```
subroutine gradient(Z, DelG, south, north, west, east)

*$noex

implicit none

c Declaration of variables
include 'E8ArraySize.fi'

integer :: south, north, west, east !!! Location of map boundaries
integer :: I,J !!! Row (I) and Column (J) indices
c Position Indices
integer :: A !!! Scan Row
integer :: B !!! Scan Column
integer, parameter :: scan = 1 !!! Defines size of box to scan gradients
integer :: AmaxG !!! Holds Scan Row of maximum gradient
integer :: BmaxG !!! Holds Scan Column of maximum gradient
c Topography variables, Directionality variables
real :: DeltaZ !!! Elev. difference b/t current cell and neighbor cells
real, dimension(MaxR,MaxC) :: Z !!! Elevation matrix
real, dimension(MaxR,MaxC) :: DelG !!! Gradient matrix
real :: max !!! Holds value of maximum gradient
!!! (using RHO8) to neighbor cells
real :: length !!! Distance to lowest neighbor cell
real :: grad !!! Gradient to lowest neighbor cell
real :: space = 30.0 !!! Horizontal resolution of USGS DEMs (m)

write (*,*) 'Calculating Gradients...

do I = south+1, north-1
do J = west+1, east-1

    AmaxG = 0
    BmaxG = 0
    max = 0.0

c Find Maximum gradient
doi A = -scan, scan, 1
do B = -scan, scan, 1

c Calculate grad for scan .EQ. 1
if ( (B .EQ. 0 .AND. (A .EQ. +scan .OR.
> A .EQ. -scan)) .OR. (A .EQ. 0 .AND. (B .EQ. ->
scan .OR. B .EQ. +scan))) then

    DeltaZ = (Z(I,J)-Z(I+A,J+B))/30.0

else

31
DeltaZ = (Z(I,J)-Z(I+A,J+B))/42.42641

endif

c Check for maximum DeltaZ
if(DeltaZ .GT. max) then
  max = DeltaZ
  AmaxG = A
  BmaxG = B
endif

c End neighbor DO LOOPS

enddo

enddo

c Calculate Gradient to lowest elevation
if (AmaxG .EQ. 0 .and. BmaxG .EQ. 0) then
  length = 1.0
else
  length =
  > (SQRT((float(AmaxG))**2+(float(BmaxG))**2))
  > *(space)
  endif

grad = (Z(I,J)-Z(I+AmaxG,J+BmaxG)) / length

DelG(I,J) = grad

endo

enddo

c Write output of DelG to output file
open(unit=10, file='MaxG.out', status='unknown')

do l = north, south, -1
  write(10,20) (DelG(I,J), J = west, east)
endo

close (unit=10)

20 format (1X, 5000E15.8E2)

return

end
subroutine channel1(Z,stream,counter1,step1,step2,n,south,north,
  west,east,Wro,Wco)

*noex
  implicit none

  Declaration of variables
  include 'E8ArraySize.f9'

  Two-D ARRAYs (Matrix of Data)
  real :: Z(MaxR,MaxC) !2D array of elevation data
  !Z is created in the ASSEMBLE subroutine
  integer :: counter1(MaxR,MaxC) !2D array containing number of
  !WATERBOTS thru each cell

  !COUNTER1 is calculated/updated in the CHANNEL1 subroutine

  One-D ARRAYs
  real :: step1(NumCells) !1D array of random numbers b/t 1, NumCells
  !used to sort random waterbot drop sequence
  !STEP1 is created in the RANORDER subroutine
  integer :: step2(NumCells) !1D array of unique integer location
  !((translatable to Row,Column) elevation
  !STEP2 is created in the RANORDER subroutine
  integer :: stream(NumCells,3) !Contains (Column,Row,Integer Location
  !of next step position)

  !STREAM is created in the RANORDER subroutine

  Array indices
  integer :: I !Row index
  integer :: J !Column index

  DEM variables Size, Dimensions, Etc.
  integer :: south !Value of Row (+ve from south) of Southern Boundary
  integer :: north !Value of Row (+ve from south) of Northern Boundary
  integer :: west !Value of Row (+ve from west) of Western Boundary
  integer :: east !Value of Row (+ve from west) of Eastern Boundary
  integer :: n !Number of elevation cells in the Z array

  !Calculated in RANORDER subroutine
  integer :: Wro !Number of Rows in working Z array, RANORDER
  integer :: Wco !Number of Columns in working Z array, RANORDER

  integer :: M !Index to drop waterbot on each cell
  integer :: next !Counter/placehold in stream array

  Call subroutine spatial to generate 1D array of elevations

  Subroutine 'ranorder' is required for both Subroutines 'transport'
call ranorder(step1,step2,stream,n,south,north,west,east,
Call subroutine network to calculate and store position of maximum gradient

write (*,*) 'Delineating Stream Network (Channel1)...

initialize counter array to zero
do I = south, north
do J = west, east
    counter1(I,J) = 0
endo
do
endo

Create array with appropriate fluvion count
do M = 1, (Wro*Wco)
    next = M
    do while (next .NE. 0)
        J = stream(next,1)
        I = stream(next,2)
        counter1(I,J) = counter1(I,J) + 1
        next = stream(next,3)
    enddo
endo
endo

Write output of DelG to output file
open(unit=10, file='Channel1.out', status='unknown')
do I = north, south, -1
    write(10,20) (counter1(I,J), J = west, east)
endo
close (unit=10)

return
end
subroutine shade(Z,shadow,space,south,north,west,east)

implicit none

c Declaration of variables
c Defining common ARRAY dimensions
include 'E8ArraySize.fi'

c Two-D ARRAYs (Matrix of Data)
real :: Z(MaxR,MaxC) !2D array of elevation data
IZ is created in the ASSEMBLE subroutine
real :: Shadow(MaxR,MaxC) !2D array containing the Relief info

!SHADOW is created/calculated in the SHADE subroutine

c DEM variables Size, Dimensions, Etc.
integer :: south !Value of Row (+ve from south) of Southern Boundary
integer :: north !Value of Row (+ve from south) of Northern Boundary
integer :: west !Value of Row (+ve from west) of Western Boundary
integer :: east !Value of Row (+ve from west) of Eastern Boundary

c Array indices
integer :: I !Row index
integer :: J !Column index
real :: space

real :: grad !elevation difference in direction of sun
real :: slangle !Slope angle in direction of sun
real :: snormal !Slope normal angle
real :: epsilon !Slope angle in direction of sun
real :: incidence !Angle of incidence
real :: sunangle !Sun angle (0 Degrees is Eastern Horizon)
real :: pi = 3.142 !Value of PI
real :: conv !Conversion from degrees to radians
real :: exaggerate !Amount of verticle exaggeration to add to elev.

write (*,*) 'Creating Shaded Relief Image Data...'

conv = pi/180.0
sunangle = (11/12.0)*pi
exaggerate = 1.0

c West/East Do Loops
do I = south, north
   do J = west, (east-1)
      grad = (Z(I,J+1)-Z(I,J))*exaggerate
      slangle = ATAN(grad/space)
      snormal = slangle + (pi/2.0)

   end do
end do

36
epsilon = slangle
incidence = sunangle - snormal

shadow(I,J) = 1.0 / ( 1.0 + (COS(epsilon)/COS(incidence)) )

if (shadow(I,J) .LT. 0.0) then
    shadow(I,J) = 0.0
endif
if (shadow(I,J) .GT. 0.7) then
    shadow(I,J) = 0.7
endif
enddo
enddo

c  Write output of DelG to output file
open(unit=10, file='Shaded.out', status='unknown')

do l = north, south, -1
    write(10,20) (shadow(I,J), J = west, east)
enddo

close (unit=10)

20  format (1X, 5000E15.8E2)

return
end
E8ArraySize.fi

c Declaration of variables
c Defining common ARRAY dimensions
  integer, parameter :: MaxC = 1200  !Dimension of elev array in cols.
  integer, parameter :: MaxR = 1000  !Dimension of elev array in rows
  integer, parameter :: NumCells = MaxR*MaxC  !Number of elevations
  !cells expected in DEM
Waterbot characteristics/information
real :: Rolls !Number os step the waterbot has already taken
real :: step = 2.0 !Number os steps to build to full Sediment
                  !Capacity
real :: Gpower = 1.0 !Power of discharge for pediment surface
real :: Bpower = 1.1 !Power of discharge for basalt surface
real :: weight

Array indices
integer :: I !Row index
integer :: J !Column index

Variables used in calculation of Waterbot Sediment Capacity
real, parameter :: Totyears = 5.0E04 !Total number of years
                             !to evolve the landscape
real, parameter :: Time  = 5.0E04 !Time (yrs) of each successive
                             !time step
real, parameter :: Rate  = 0.1  !Time (yrs) of each successive
                             !time step
real, parameter :: Const = 0.003 !Time (yrs) of each successive
                             !time step
Declaration of local variables in Subroutine TRANSPORT

Do Loop Indices

integer :: T  !Used in Do Loop to evolve landscape thru time
integer :: S  !Used in Do Loop to hit each Row,Column with Waterbot

Position Indices

integer :: A  !Scan Row
integer :: B  !Scan Column
integer :: C  !Index for DeltaZ array
integer, parameter :: scan = 1  !Defines size of box to scan gradients
integer :: AmaxG  !Holds Scan Row of maximum gradient
integer :: BmaxG  !Holds Scan Column of maximum gradient

Sediment Capacity Variables

real :: lnH  !Thickness of material coming into cell (+ve)
real :: DeltaH  !Thickness to erode/deposit from current cell (-ve for erosion)
real :: length  !Distance b/t current cell and next step
real :: grad  !Gradient b/t current cell and next step

Topography variables, Directionality variables

real, dimension(9,3) :: DeltaZ  !Elev. difference b/t current cell and neighbor cells
real :: Delta  !Calculated difference in elevation of current cell
             !and neighbor
real :: max  !Holds value of maximum gradient (using RHO8)
            !for neighbor cells
real :: min  !Holds value of lowest neighbor
real :: maxdelZ  !Holds value of true maximum difference in elevation b/t cells
real :: rho8  !Fairfields (1991) directionality variable
integer :: Seed  !Random seed variable for random number generation
real :: ran2  !Random number generator 2 (+ve input)
real :: random  !Random number b/t 0.0 and 1.0
c Declaration of variables

c Defining common ARRAY dimensions
  include 'E8ArraySize.fi'

c Two-D ARRAYs (Matrix of Data)
  real :: Z(MaxR,MaxC)  I2D array of elevation data
  IZ is created in the ASSEMBLE subroutine
  real :: cover(MaxR,MaxC)  I2D array of thickness of alluvial cover

  ICOVER is created in the ALLUVIUM subroutine
  real :: modify(MaxR,MaxC)  I2D array of amount of elev change (+/-)

  !MODIFY is calculated/updated in the TRANSPORT subroutines
  integer :: counter1(MaxR,MaxC)  I2D array containing number of
  !WATERBOTS passing thru each cell

  !COUNTER1 is calculated/updated in the CHANNEL1 subroutine
  integer :: counter2(MaxR,MaxC)  I2D array containing number of
  !WATERBOTS passing thru each cell

  !COUNTER2 is calculated/updated in the TRANSPORT subroutines
  real :: DelG(MaxR,MaxC)  I2D array containing the maximum
  !gradient value of 8 neighbors

  !DELG is created/calculated in the GRADIENT subroutine
  real :: Shadow(MaxR,MaxC)  I2D array containing the Shaded Relief

  !SHADOW is created/calculated in the SHADE subroutine
  integer :: geology(MaxR,MaxC)  !GEOLOGY contains information on the
  !type of material present in cell

  !GEOLOGY is created in the LAVA subroutine
  real :: basalthk(MaxR,MaxC)  !BASALTHK contains the thickness of
  !the Basalt in the Cell

  !BASALTHK is created in the LAVA subroutine
  real :: basaltbot(MaxR,MaxC)  !BASALTBOT contains the elevation to
  !the bottom of the basalt

  !BASALTBOT is created in the LAVA subroutine

  c One-D ARRAYs
  real :: step1(NumCells)  !1D array of random numbers b/t 1, NumCells
  !used to sort and order random waterbot drop

  !STEP1 is created in the RANORDER subroutine
integer :: step2(NumCells) ! 1D array of unique integer location
                   !(translatable to Row,Column) of each
elevation

!STEP2 is created in the RANORDER subroutine
integer :: stream(NumCells,3) ! Contains (Column,Row,Integer Location
                               ! of next step position)

!STREAM is created in the RANORDER subroutine

c DEM variables Size, Dimensions, Etc.
integer :: south  ! Value of Row (+ve from south) of Southern Boundary
integer :: north  ! Value of Row (+ve from south) of Northern Boundary
integer :: west   ! Value of Row (+ve from west) of Western Boundary
integer :: east   ! Value of Row (+ve from west) of Eastern Boundary
integer :: n      ! Number of elevation cells in the Z array

! Calculated in RANORDER subroutine
integer :: Wro    ! Number of Rows in working Z array, RANORDER
integer :: Wco    ! Number of Columns in working Z array, RANORDER
real :: space = 30.0 ! Horizontal resolution of USGS DEMs (m)