U.S. Department of the Interior
U.S. Geological Survey

LAHARZ:
GIS programs for automated mapping of lahar-inundation hazard zones

By Steven P. Schilling

Open-File Report 98-638

Vancouver, Washington
1998

U.S. DEPARTMENT OF THE INTERIOR
Bruce Babbit, *Secretary*

U.S. DEPARTMENT OF THE INTERIOR
Gordon P. Eaton, *Director*

Copies of this report can be purchased from:

U.S. Geological Survey
Information Services
Federal Center, Box 25286
Denver, CO 80225

Copies of this report can be purchased from:

Willie Scott
U.S. Geological Survey
5400 MacArthur Blvd.,
Vancouver, WA 98661

CONTENTS

FIGURES

APPENDIX I - LAHARZ menus and AMLs

LAHARZ: GIS programs for automated mapping of lahar-inundation hazard zones

By Steven P. Schilling

ABSTRACT

Lahars are large debris flows that originate on volcano flanks and can surge tens or even hundreds of kilometers downstream from a volcano. Volcanologists help mitigate volcanic hazards by creating hazard-zonation maps that depict estimates of the location and extent of areas inundated by future lahars and other volcanic processes. Taking a new approach to the problem of estimating the extent of lahar inundation, Iverson and others (1998) used the results of scaling and statistical analyses of the geometry of 27 lahar paths at nine volcanoes to develop equations that predict inundated valley cross-sectional and planimetric areas as functions of lahar volume. LAHARZ, menu-driven software that runs within a Geographic Information System (GIS), uses these equations, a Digital Elevation Model (DEM) and user-specified lahar volumes to provide an automated method to map areas of potential lahar inundation. For user-selected drainages and user-specified lahar volumes, LAHARZ can delineate a set of nested lahar-inundation zones that depict gradations in hazard in a manner that is rapid, objective, and reproducible.

INTRODUCTION

Lahars are large debris flows that originate on volcano flanks and can surge tens or even hundreds of kilometers downstream from a volcano. Such remarkable flow mobility helped form Crandell's opinion that lahars are the "greatest hazard" posed by northwest volcanoes such as Mount Rainier, Washington, and that future lahars similar in size to those of the past "would doubtlessly be disastrous" (Crandell, 1971). Volcanologists help mitigate volcanic hazards by creating hazard-zonation maps that depict estimates of the areal extent inundated by future volcanic processes including lahars. Estimates of the area inundated by future lahars at a volcano are based on deposits left by lahars that have flowed in the past. Traditionally, volcanologists have identified the location and chronology of deposits left by lahars, estimated the extent of inundation from those lahar deposits, and extrapolated similar inundation extents along all drainages of a volcano (e.g., Scott et al., 1995). Although estimates of lahar inundation area derived from geologic studies of lahar deposits are a key element of maps that depict hazards that may arise from a volcano (e.g., Hoblitt et al., 1995), detailed records and observations of lahar deposits have not been made for all potentially hazardous volcanoes. Many remote volcanoes are difficult to access, geological and chronological data of lahar deposits for some volcanoes are sparse, and in volcanic crises, time for geologic field investigations may be non-existent. Earth scientists require another means of estimating areas of potential lahar inundation to produce volcano-hazard maps.

Iverson and others (1998) used scaling analysis, to predict mathematically the form of an equation, and statistical analyses of 27 lahar paths at nine volcanoes to develop equations that estimate cross-sectional and planimetric areas of valley inundation as functions of lahar volumes. LAHARZ, software (Appendix 1) that runs within a Geographic Information System (GIS), uses these equations to provide volcanologists with an automated method to estimate areas of potential lahar inundation (figure 1). Using a Digital Elevation Model (DEM) and several lahar volumes, LAHARZ employs Iverson and others' (1998) equations to delineate, for user-selected stream drainages, a set of nested, lahar-inundation hazard zones in a manner that is rapid, objective, and reproducible.

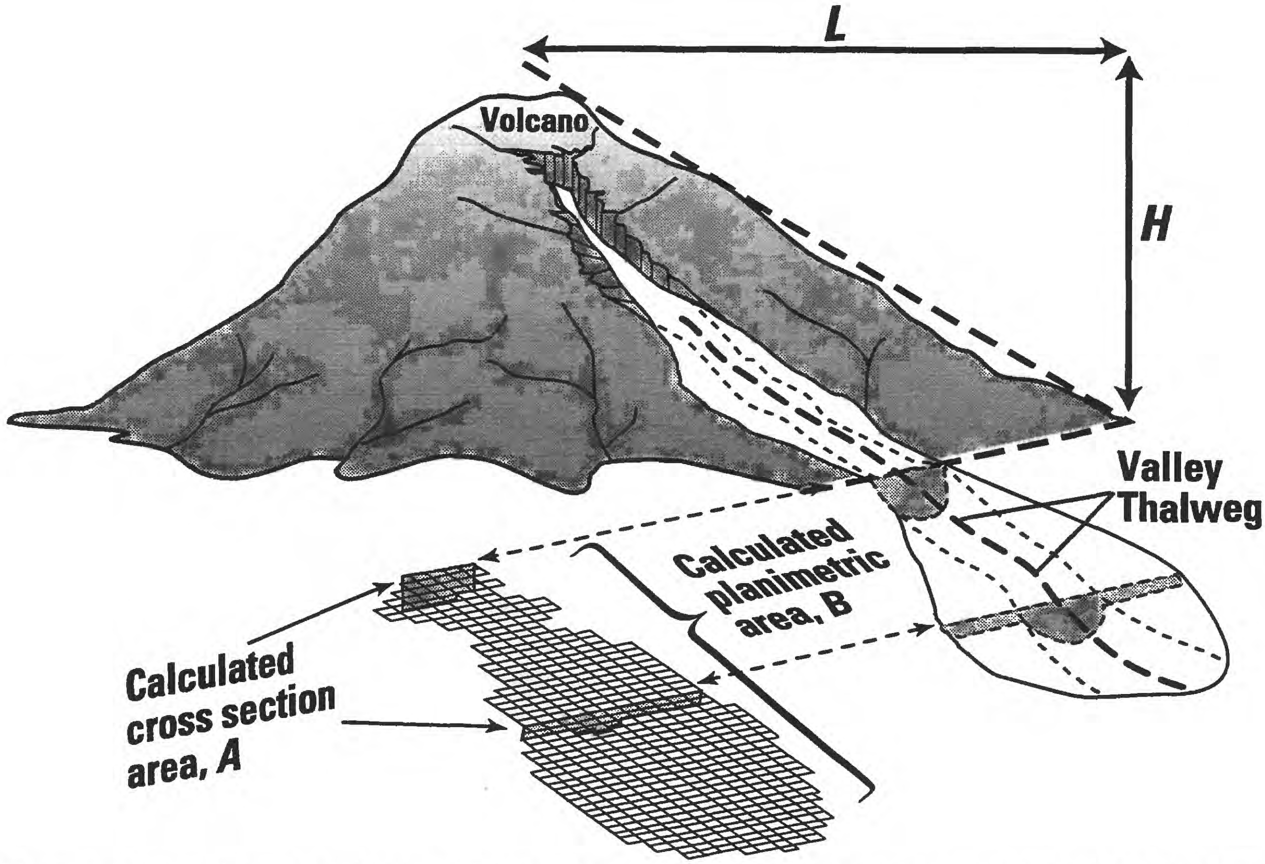


Figure 1. Diagram of association between dimensions of an idealized lahar and cross-sectional *(A)* and planimetric *(B)* areas calculated by LAHARZ for a hypothetical volcano. The ratio of vertical drop*(H)* to horizontal runout distance (*L*) describes the extent of proximal volcano hazard. LAHARZ begins calculations of lahar-inundation hazard zones where a user-specified stream and the proximal-hazard zone boundary intersect. Modifed from Iverson *et al.* (1998).

LAHARZ OVERVIEW

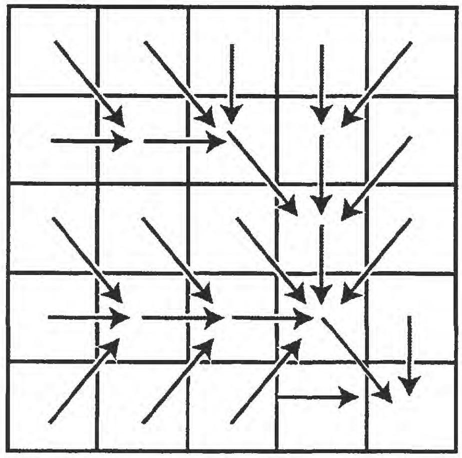
LAHARZ is software written in the ARCINFO Macro Language (AML) that runs within the cell-based GRID[1] portion of ARCINFO. It is essential that users of LAHARZ have some familiarity with DEMs and with ARCINFO, especially the GRID program.

[1] For purposes of this report, the word grid in upper case letters refers to the ARCINFO GRID module that manipulates raster data. The word grid in lower case letters refers to a data layer of square cells. In each grid, cells contain either integer, real number or NODATA values. The grids used in GRID are georeferenced; GRID creates and maintains a relationship between planar-map page coordinates and real world coordinates (ESRI, 1990) ensuring that grids in the same coordinate system and of the same extent will coincide. Text in italics are commands the user types at the ARC prompt.

LAHARZ was written to delimit areas of potential lahar inundation from one or more user-specified lahar volumes. If the user enters multiple lahar volumes, LAHARZ produces a lahar-inundation hazard zone for each volume. The planimetric area of lahar-inundation hazard zones often increase in width and length as lahar volume increases. The user can plot lahar-inundation hazard zones having smaller planimetric areas over those of larger area to produce a "nested" set of inundation hazard zones. These hazard zones can be plotted with other types of volcano-hazard information, such as a proximal-hazard zone boundary, infrastructure, hydrology, and topography to produce volcano-hazard maps. Prior to generating potential lahar-inundation areas, however, users must first run LAHARZ to remove any errors from the DEM that inhibit surface flow routing, create supplea)

64	60	59	59	60
63	59	58	57	58
62	60	59	56	59
61	57	56	54	55
60	60	59	57	52

b)



c)

380	**510**	440	420	410
380	410	**550**	440	420
420	440	**600**	480	460
380	460	480	**620**	480
410	300	460	490	**680**

d)

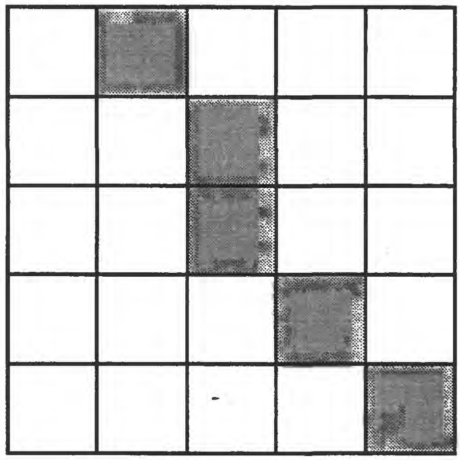


Figure 2. Diagram of part of a DEM and corresponding supplementary surface hydrology grids. a) Sketch of part of a DEM where cells contain values of elevation in meters. b) Schematic diagram of a flow direction grid derived from the DEM. The flow direction function determines the direction of flow from every cell in the DEM to the adjacent downstream cell and stores that flow direction in the corresponding cell in a flow direction grid. Arrows represent the direction of flow; tails of arrows identify each individual cell and arrow heads lie within the adjacent downstream cell. c) The flow accumulation function uses the flow direction grid to calculate, for each cell, the number of cells that flow into it and stores the results in a flow accumulation grid. LAHARZ identifies cells in the flow accumulation grid having values (values in bold text) above a user-specified stream-delineation threshold (in this example, a stream-delineation threshold of 500 cells). d) Cells in the flow accumulation grid that have values greater than the stream-delineation threshold are stored as stream cell (gray cells) in a stream grid.

mentary surface hydrology grids, and produce a proximal-hazard zone boundary, a boundary that encircles hazards affecting areas near a volcano.

PREPARING A DEM

A DEM is a raster representation of surface elevation. DEMs are available from many sources and vary in accuracy, resolution (size of grid cell), projection, and extent. Users should check their DEMs for quality and, if appropriate, join separate files into a single seamless DEM prior to running LAHARZ. The extent or area of a DEM used by LAHARZ to create lahar-inundation hazard zones must include the volcano as well as the downstream extent of major drainages that originate on the volcano. LAHARZ runs flow routing functions to create supplementary grids that describe hydrologic flow direction and flow accumulation over the DEM surface (figure 2). LAHARZ uses these surface hydrology grids to "move" down a user-specified stream and the DEM to create lahar-inundation hazard zones.

DEMs often contain erroneous elevations, referred to as "sinks", where flow direction is undefined. Sinks, often created during the generation of a DEM, inhibit surface flow routing. However, some sinks may represent real surface depressions such as quarries or natural erosional potholes (Jenson and Domingue, 1988). Thus, the user must be familiar with land features represented by the DEM to determine whether sinks in their DEM are errors. The user can choose to have LAHARZ run the ARCINFO GRID function "FILL" that automatically fills sinks that are shallower than a user-specified threshold value (see ESRI, 1994) in vertical units of elevation of the DEM (e.g., feet or meters). The resultant "depressionless" grid is then used for creating supplementary surface hydrology (flow routing) grids.

SURFACE HYDROLOGY GRIDS

ARCINFO GRID functions typically operate on all cells in a grid. GRID processes the top row of cells, moving cell by cell from the left edge to the right edge of a grid, then processes the next row of cells from the left edge to the right edge of the grid and so on, until all cells in the grid are processed. Processing, for example, might add a constant value to each cell in a grid or calculate a value for each cell based on values of adjacent, surrounding "neighborhood" cells. To create lahar-inundation hazard zones however, LAHARZ restricts processing to only those cells that form streams within drainages. In order to limit processing to stream cells, LAHARZ must create and use surface hydrology grids.

LAHARZ uses GRID surface hydrology functions to derive flow direction, flow accumulation, and stream delineation grids from a DEM (e.g., Jenson and Domingue, 1988; ESRI, 1994). The flow direction function calculates the direction of flow out of every cell in the DEM and stores those directions in a flow direction grid (figure 2). The flow accumulation function creates a flow accumulation grid and, using values in the flow direction grid, assigns each cell in the flow accumulation grid a value that is the sum of the number of cells that flow into it. LAHARZ identifies cells in the flow accumulation grid having values greater than a user-specified, stream-delineation threshold and stores those cell locations in a stream grid (ESRI, 1994) (figure 2). Even though LAHARZ has a default stream-delineation threshold value of 1000 cells, the user may have to try several threshold values to create an appropriate, reasonable stream pattern.

LAHARZ uses the surface hydrology grids while constructing inundation cross sections by repeatedly making cross section area calculations for a stream cell then moving to one of the eight adjacent cells that lies downstream. To "move" downstream, LAHARZ stores the X and Y coordinates of the center of the current cell, checks the flow direction grid for the next downstream direction, and moves to the next downstream cell by calculating the X and Y coordinates of the center of the downstream cell. LAHARZ begins calculations of lahar-inundation hazard zones at a cell where a user-specified stream and the proximal-hazard zone boundary intersect.

PROXIMAL-HAZARD ZONE BOUNDARY

Volcanologists commonly use energy-line cones (Sheridan, 1979) having unique, but characteristic slopes that describe the extent of deposits left around a volcano by various flowage phenomena. Volcanologists have used such energy-line cones to describe the distance traveled (runout) by debris avalanches that originate on volcano flanks (Siebert, 1984) or summits (e.g., Hsu, 1975) and by pyroclastic flows (Sheridan, 1979; Beget and Limke, 1988). These cones, often referred to as energy, mobility, or H/L cones, have an apex that usually coincides with a volcano summit and has a slope determined by a characteristic ratio of vertical drop (H) to horizontal runout distance (L) for different volcano processes (figure 1). Values of H/L ratios that define boundaries of "near volcano" or "proximal" hazard zones typically range from about 0.1 to 0.3, depending on the size and type of the proximal event (e.g., Hayashi and Self, 1992). LAHARZ calculates the line of intersection between an H/L cone and topography (e.g., Malin and Sheridan, 1982) that volcanologists refer to as the proximal-hazard zone boundary for a volcano (e.g., Scott et al., 1997). Areas on volcano flanks where lahars originate (lahar source areas) are difficult to predict. LAHARZ operates with the assumption that lahar inundation begins at the boundary of the proximal hazard zone and continues downstream, away from or distal to, the volcano.

CREATING CROSS SECTIONS

For each user-specified lahar volume, V, LAHARZ calculates a cross sectional area (A) and a planimetric area (B) from equations, $A = 0.05\ V^{2/3}$ and $B = 200\ V^{2/3}$ (Iverson and others, 1998). LAHARZ starts constructing lahar-inundation hazard zones by identifying a cell where a user-selected stream and the proximal-hazard zone boundary intersect. LAHARZ stores a single, calculated area that applies to all cross sections, but the shape of each cross section will vary. LAHARZ uses the elevation values of the DEM and the stored cross section area to determine the shape of each cross section.

If one imagines each cell is made of solid material extruded as columns from the X, Y grid plane in the Z direction to a height proportional to the elevation stored in the grid cell, a cross section would show a profile of the DEM across the selected stream valley formed by the tops of the cell columns (figure 3). Most often, the lowest cell elevation in the profile represents the thalweg for the stream in the drainage. In this hypothetical scenario, LAHARZ must "build" each cross section "upward" from the DEM surface and outward from, "outboard" of, a stream cell. As LAHARZ constructs a cross section it stores the locations and compares the elevations of grid cells on either side of and increasingly away from the stream cell.

Initially, LAHARZ stores the location and elevation of a stream cell as the "right" cell and an adjacent cell, arbitrarily the cell left of the stream cell when viewed downstream (figure 4), as the "left" cell. LAHARZ also stores the elevation of the "right" stream cell as the current top of the cross section or "fill level". LAHARZ compares the elevations of the "left" and "right" cells and the elevation of those cells to the "fill level". LAHARZ varies its action based on the results of this comparison of elevations. If either "left" or "right" cell is lower in elevation than the "fill level", LAHARZ calculates the area to "fill" the cell elevation to the "fill level", tallies the total area thus far accumulated, and changes cell location or "moves" away from the stream cell to an adjacent outboard cell. With each "move" to a new cell outboard of the

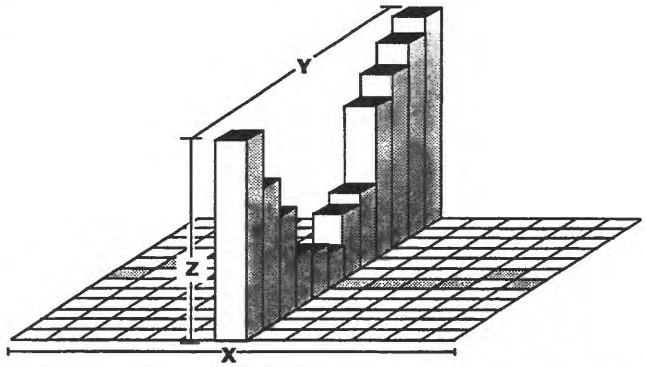


Figure 3. Illustration of a row of elevation values "extruded" from DEM along the Z axis as hypothetical solid columns to illustrate cross section construction. Columns in the Z,Y plane form a drainage profile. Gray cells in the X, Y plane are stream cells. After LAHARZ fills the cross section, it sorts the locations of cells occupied during cross section construction (black tops of extruded columns in Z, Y plane). These occupied cells form an increment in the total planimetric lahar-inundation area.

stream cell, LAHARZ adds one cell to the total number of cells occupied during construction of the cross section. If either "left" or "right" cell is equal in elevation to the "fill level", LAHARZ moves away from the stream cell to the adjacent outboard cell. If the elevations of the "left" and "right" cells are equal and greater than the "fill level", LAHARZ moves away from the stream cell to the adjacent "left" and "right" cells. Finally, if the "left" and "right" cell elevations are not equal and are both greater than the "fill level", LAHARZ calculates the difference in elevation between the cell having the lower elevation and the "fill level". LAHARZ also multiplies the total number of occupied cells by the cell width or diagonal length to calculate the current width of the cross section. LAHARZ then multiplies this current width by the difference in elevation of the lower cell and the "fill level" to calculate the current partial area, a "tier" of the cross section. The area of each successive tier is tallied and compared to (A). LAHARZ then stores the elevation of either the "left" or "right" cell, whichever has the lower elevation, as the new "fill level" and "moves" from the cell to the adjacent cell outboard of the stream cell (figure 4). LAHARZ continues this process of moving from "left" and (or) "right" cell locations to outboard cells, comparing elevations between "left" and "right" cells and between these cells and the "fill level", and calculating "tiers" of cross sectional area. After calculation of each tier of cross sectional area for the current cross section, LAHARZ compares the cross section area (A) to the

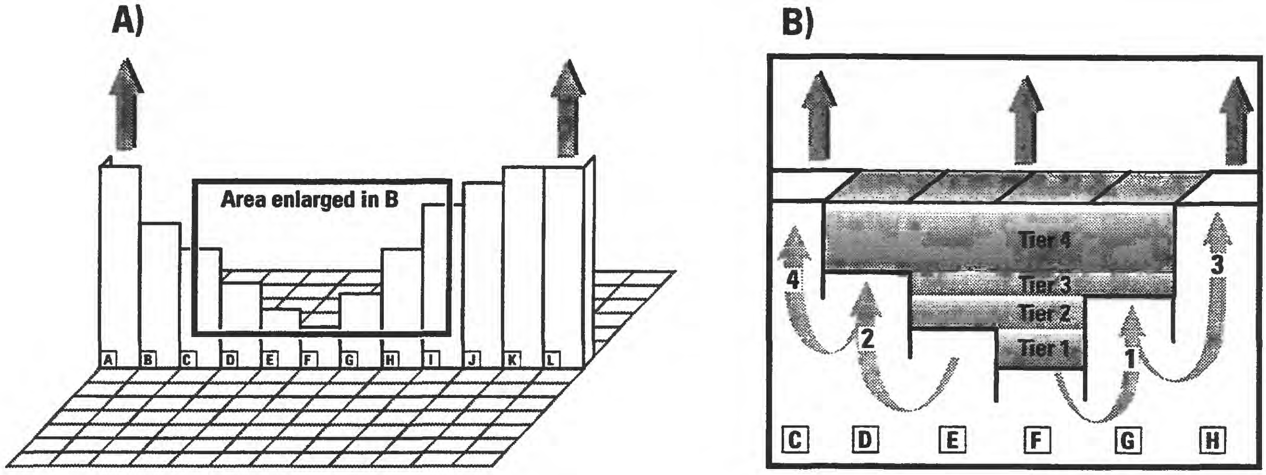


Figure 4. Sketch showing construction of hypothetical cross section. Tops of individual "imaginary" cell columns have heights proportional to cell elevation values in DEM. A) A cross-sectional profile of a DEM across a selected stream valley. The lowest cell elevation in the profile, column labeled "F" represents the thalweg for the drainage. B) Initially, LAHARZ stores the location and elevation of a stream cell as the "right" cell (cell F) and the adjacent left cell (cell E) as the "left" cell. LAHARZ also stores the elevation of the "right" cell as the top of the cross section or "fill level". LAHARZ compares the elevations of the "left" and "right" cells and the elevation of those cells to the "fill level". In this example, LAHARZ determines that the "right" cell is equal to the "fill level" and changes the "right"cell location to cell G (move shown by gray arrow labeled "1") and updates the cell count to 1 (cell F). LAHARZ compares elevation of cell E (the "left" cell) to cell G (the "right" cell), compares the elevations of both cells to the "fill level", and determines that both cells are greater than the "fill level" and that cell G is greater in elevation than cell E. LAHARZ calculates the difference in elevation between cell E and the "fill level" and multiplies that difference by the number of cells (1) multiplied by the cell width. The area calculated (labeled Tier 1) is subtracted from the total cross-sectional area *(A)*. LAHARZ moves the "left" cell location to cell D (move shown by gray arrow labeled "2"), stores the elevation of cell E as the new "fill level", and updates the cell count to 2 (cells E and F). LAHARZ continues this process of moving from "left" and (or) "right" cell locations to outboard cells, comparing elevations between "left" and "right" cells and between these cells and the "fill level", calculating "tiers" of cross-sectional area, and comparing the area accumulating from successive tiers to *(A)* until it is greater than or equal to *(A)*.

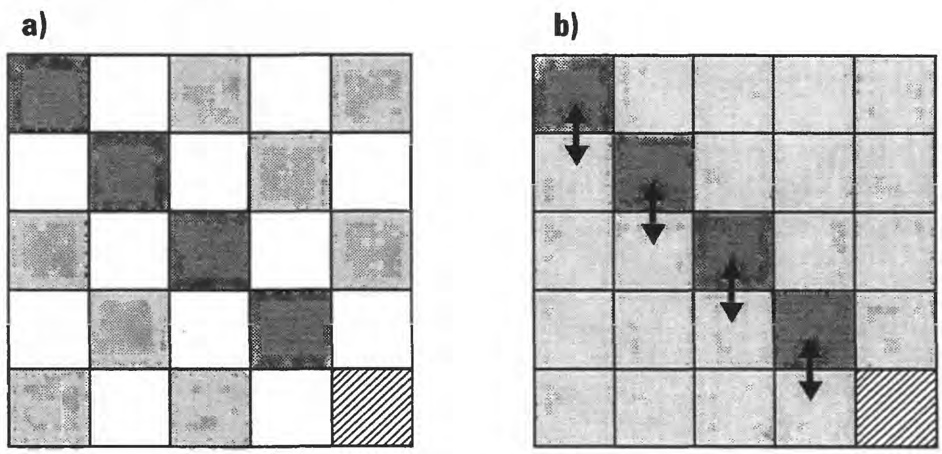


Figure 5. Sketch of section of planimetric area grid where stream cells intersect at their corners. Dark gray cells are locations of stream cells; flow direction is from northwest cell to southeast cell. Striped cell is "current" stream cell. a) Shows checkerboard pattern of cells occupied during cross section construction (light gray cells) and intervening 'voids" (white cells). b) LAHARZ stores the location of a stream cell, changes location (black arrows) to cell adjacent to the stream cell and the downstream cell to construct an additional cross section, then changes location back to original stream cell location and proceeds to the next downstream cell. Light gray cells show planimetric area now excludes "voids".

area of the current cross section. LAHARZ stops calculation of the current cross section when the area of the current cross section is equal to or greater than cross section area *(A)*.

For any cell there are four directions possible to construct cross sections; north-south, east-west, northwest-southeast, and southwest-northeast. The software checks the flow direction grid at the location of the stream cell currently being processed for the direction of the next downstream cell and eliminates that direction from the four possible choices. LAHARZ constructs cross sections in the three remaining directions for each of the user-specified volumes.

When a stream path is diagonal, stream cells intersect at their corners. Cross sections derived from successive stream cells along this diagonal path form a "checkerboard" pattern of cells (figure 5) with intervening cell "voids" that are not included in the cumulative planimetric area (T*i*), as a result of cell geometry. In these cases, LAHARZ records the location of the current stream cell, moves to the center of a cell adjacent to the current stream cell and the next downstream cell, constructs an additional cross section perpendicular to the flow direction, and moves back to the center of the stored location of the current stream cell. When all the cross sections for a stream cell are complete, the locations of cells occupied during cross section construction are added to the planimetric inundation area and LAHARZ continues to the next downstream cell.

In plan view, the length of the row, column, or diagonal of all cells occupied during construction of a cross section forms the maximum width of the inundation cross section. Each of these cells occupies an area and thereby a fraction of the predicted planimetric area *B*. The locations of cells of this row, column or diagonal of cells are added to the total cumulative planimetric area (T*i*). Although cell locations from construction of cross sections may coincide with cell locations stored in T*i* from previous cross sections, a cell location stored in the T*i* is used to calculate the planimetric area only once. LAHARZ continues to construct cross sections, adding the locations of cells from each cross section to the cumulative area of the current inundation zone Ti, and comparing the cumulative area of the current inundation zone (T*i*) to the planimetric area *(B)*. LAHARZ stops constructing the current inundation zone when the area of the current inundation zone T*i* is greater than or equal to the planimetric area *(B)*.

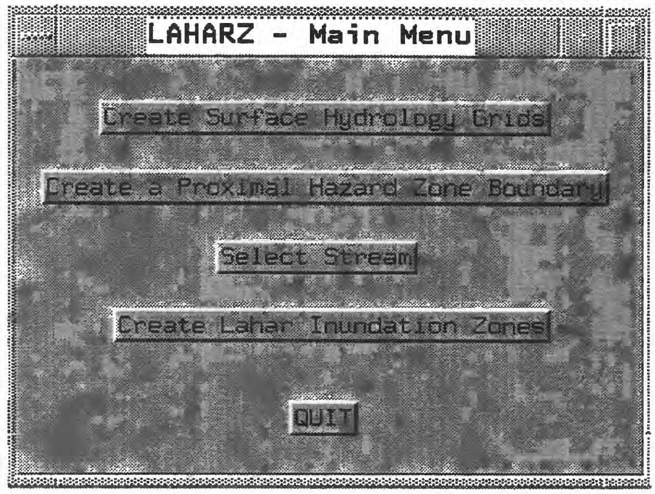


Figure 6. Main menu for LAHARZ program.

USING LAHARZ

The user controls LAHARZ through a series of simple graphical user interfaces (GUIs), consisting of cartographic displays and menus created in the AML (e.g., ESRI, 1991, 1993). With LAHARZ, the user creates surface hydrology grids, defines the slope for an energy cone to create a proximal-hazard zone boundary, selects a stream, and enters up to four lahar volumes to create four lahar-inundation hazard zones for the identified drainage. The user repeats the steps to select a stream and enter lahar volumes for each drainage of the volcano.

LAHARZ (Appendix 1) consists of 14 AMLs and 7 menus that should be stored in a single directory. If the LAHARZ directory is not the workspace where the user will process their data, the path to the LAHARZ directory should be added to the *&amlpath* and *&menupath* directives. LAHARZ is started at the ARC prompt by typing:

&r laharz <carriage return>.

LAHARZ first presents the "LAHARZ - Main Menu" menu (figure 6). The "LAHARZ - Main Menu" presents the user with five buttons labeled "Create Surface Hydrology Grids", "Create a Proximal Hazard Zone Boundary", "Select Stream", "Create Lahar Inundation Zones", and "QUIT".

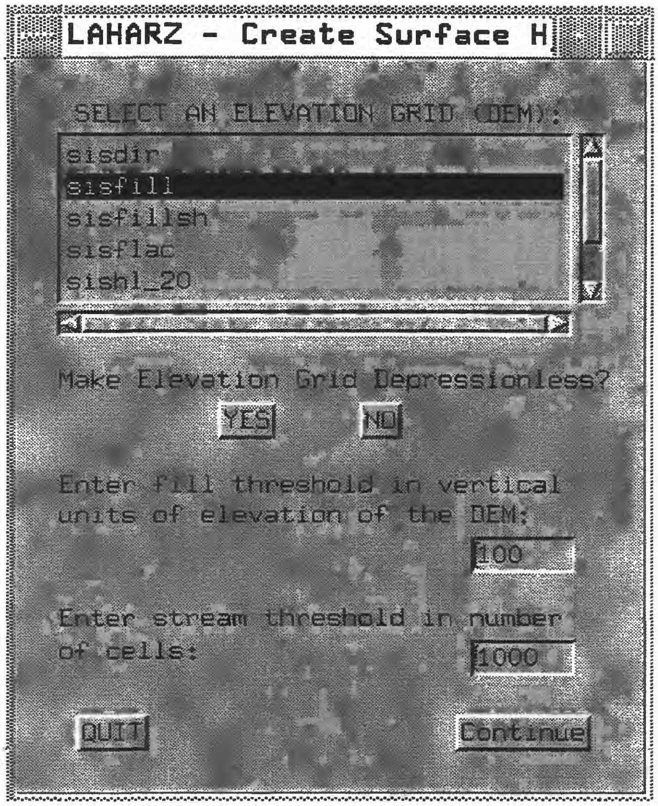


Figure 7. Menu for user to create supplementary surface hydrology grids. User specifies a DEM and indicates whether LAHARZ should run the "FILL" function to fill "sinks" (cell having undefined flow directions) in the DEM. If the user answers "yes" to the question "Make Elevation Grid Depressionless?", they must specify a fill threshold in units of elevation of the DEM (e.g., feet or meters). The "FILL" function will fill "sinks" having a depth less than or equal to the threshold. The user must also specify a stream threshold value. LAHARZ will store locations of cells in a stream grid where corresponding cells in the flow accumulation grid contains a value above the stream threshold.

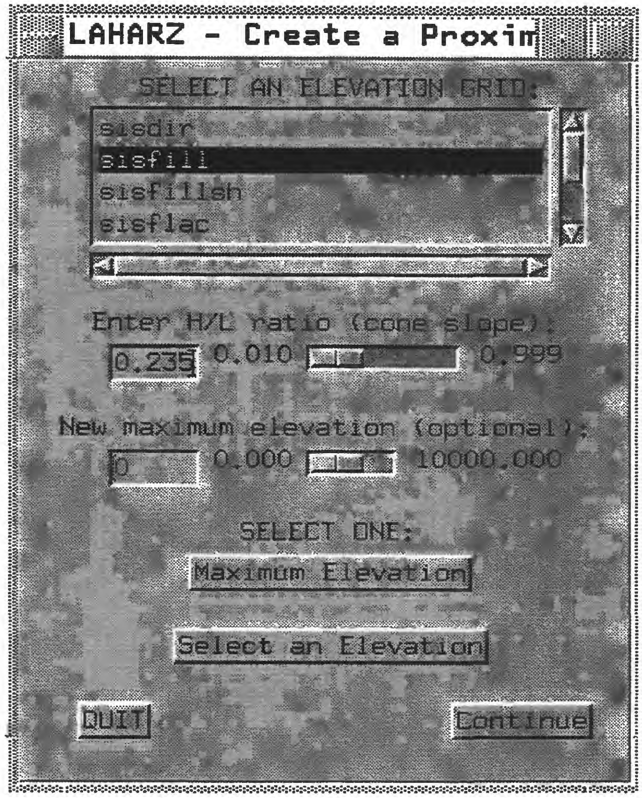


Figure 8. Menu for user to create a proximal-hazard zone boundary. User selects a DEM and specifies a ratio of *H* to *L* (slope of *H/L* cone). The user specifies whether LAHARZ automatically identifies the maximum elevation (pressing "Maximum Elevation" button) or the user manually selects a location (pressing the "Select an Elevation" button) from the DEM as the location of the *H/L* cone apex. As an option, the user can specify a new maximum elevation that will be added to the DEM elevation at the automatically or manually selected *H/L* cone apex location to represent a volcano plume height.

Creating Surface Hydrology Grids

When the user selects "Create surface hydrology grids" from the "LAHARZ - Main Menu" menu, LAHARZ presents the "Create Surface Hydrology Grids" menu (figure 7). Through this menu, the user selects a DEM from a list of grids in the current workspace. The user can choose to FILL sinks that are errors in the DEM by selecting YES in answer to the question "Make Elevation Grid Depressionless?" and entering a fill threshold value (in vertical elevation units of the DEM, e.g., feet or meters) the FILL function uses to determine how deep of a sink to fill. The user also specifies a stream threshold value. The stream function creates a grid identifying cell locations where the flow accumulation grid is greater than the stream threshold value.

LAHARZ uses the first three characters of the selected DEM name as the prefix for the name of grids created during the "Create Surface Hydrology Grids" process. The suffix "fill" is appended to the prefix to name the filled DEM, "dir" to name the flow direction grid, "flac" to name the flow accumulation grid, and "str" to name the stream delineation grid. For example, if the name of the input DEM is mtrainier.dem, LAHARZ will name the four new grids, mtrfill, mtrdir, mtrflac, and mtrstr and store the grids in the current workspace. If the user generates a "filled"

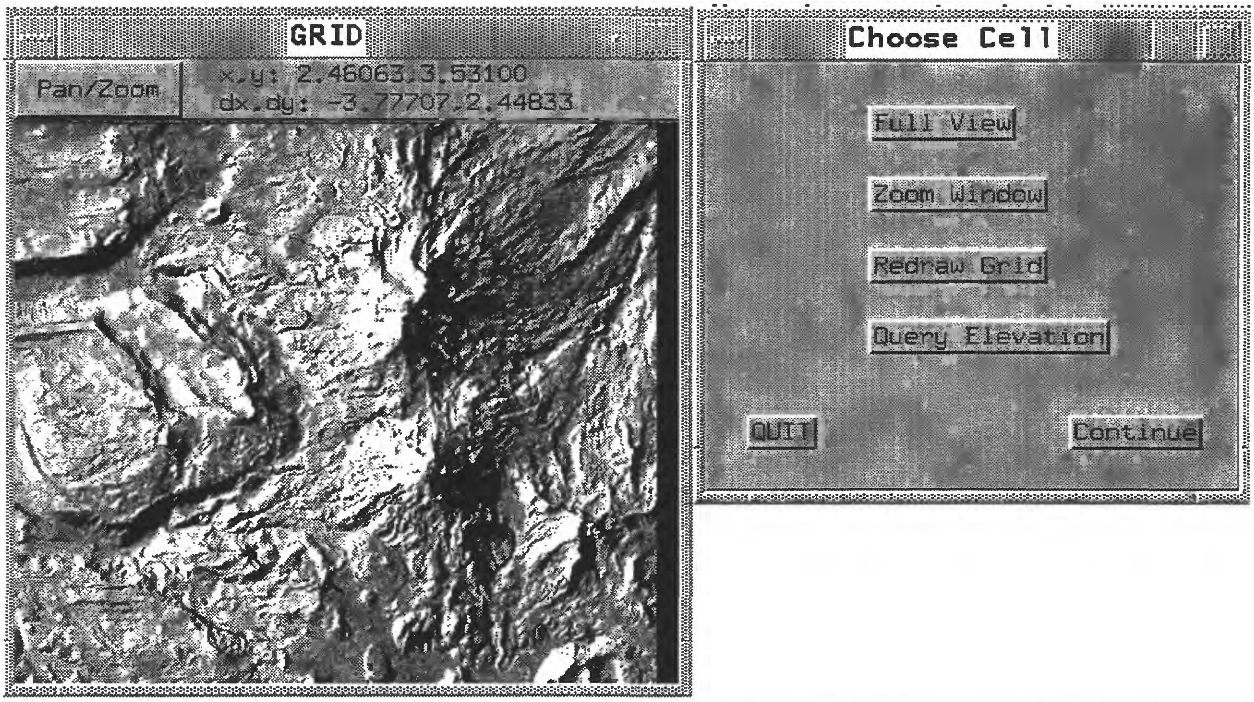


Figure 9. Graphics display and control panel for user to manually specify location of *H/L* cone apex. User changes the view in graphics display using buttons in the control panel. The query elevation button displays X and Y coordinates and elevations of cells the user selects with a mouse in the graphics window. Pressing the continue button selects the maximum elevation in a 50 cell x 50 cell box centered on the last queried cell as the location of the *H/L* cone apex.

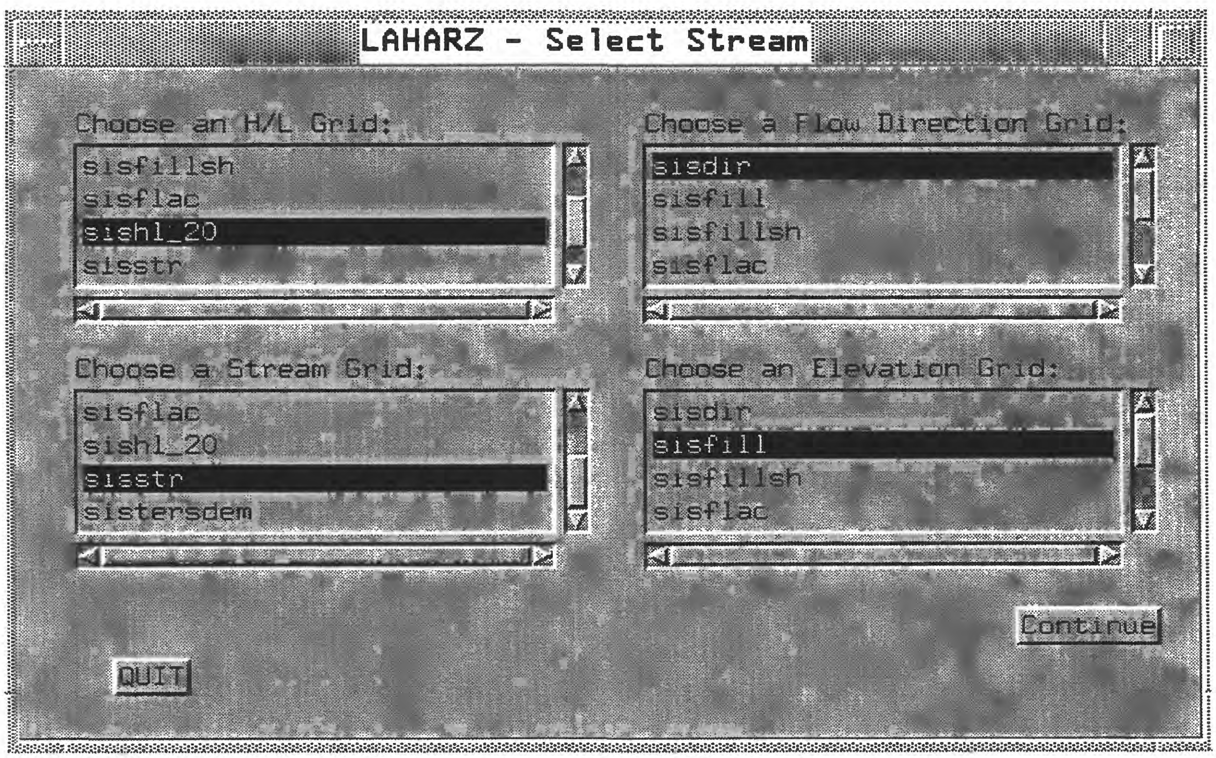


Figure 10. Menu for user to select stream. User specifies four grids: a proximal-hazard zone boundary (*H/L* grid), a stream grid, a flow direction grid, and an elevation grid.

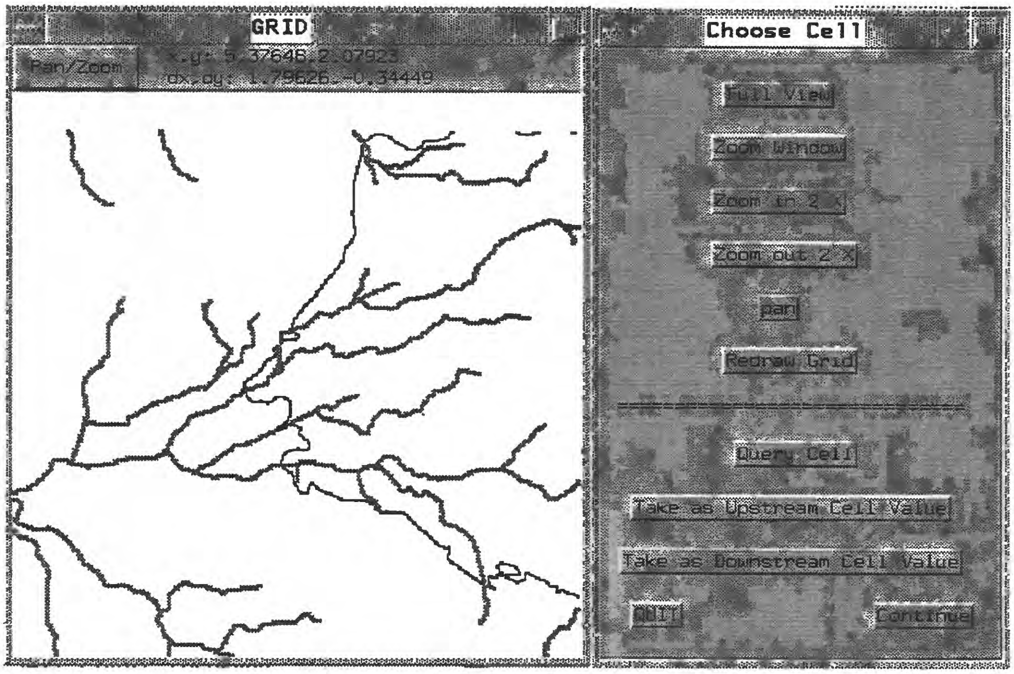


Figure 11. Graphics display and control panel for user to select a stream drainage for LAHARZ to create lahar-inundation hazard zones. Thick gray lines (drawn in cyan on screen) are streams and thin black line (drawn in magenta on screen) is the proximal-hazard zone boundary. Control panel buttons above double dashed line control view in graphics window. Buttons below the line inform the user whether they have a chosen a valid stream cell and stores coordinates for upstream and downstream cells. User must press the "Query Cell" button and specify a stream cell inside of proximal hazard zone boundary by using the mouse in the graphics window until the display lists the cell as a "valid" cell. The user then clicks the "Take as Upstream Cell Value" button storing the cell location as the upstream cell. The user must press the "Query Cell" button and specify a stream cell outside of proximal hazard zone boundary near edge of grid by using the mouse in the graphics window until the display lists the cell as a "valid" cell. The user then clicks the "Take as Downstream Cell Value" button storing the cell location as the downstream cell.

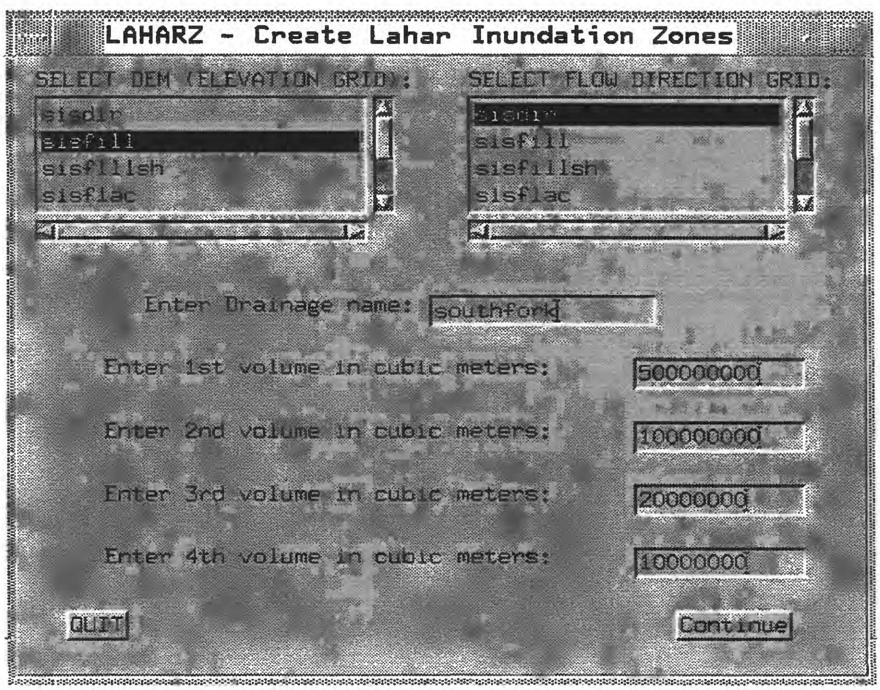


Figure 12. Menu for user to create lahar-inundation hazard zones. User selects a DEM and a flow direction grid. The user specifies a name for the selected stream drainage and up to four lahar volumes in cubic meters.

or "depressionless" DEM, they must remember to use it for the rest of the program, rather than the original DEM.

Create a Proximal-Hazard Zone Boundary

After the user presses the "Create a Proximal Hazard Zone Boundary" button from the "LAHARZ - Main Menu" menu, LAHARZ presents the "Create a Proximal Hazard Zone Boundary" menu (figure 8). Through this menu, the user specifies parameters for an H/L cone from which LAHARZ derives a proximal- hazard zone boundary. The user selects a DEM from a list of grids in the current workspace and enters an H/L ratio (slope of the cone). The user also specifies whether to automatically or manually select the location of the H/L cone apex. The simplest choice for the user is to press the "Maximum Elevation" button on the "Create a Proximal Hazard Zone Boundary" menu. LAHARZ searches the DEM to identify the location of the cell storing the maximum elevation (typically the volcano summit). If the user presses the "Select an Elevation" button LAHARZ presents the user with a shaded relief display of the DEM and a "Choose Cell" menu. The user can select a position for the cone apex from the shaded relief display of the DEM, using the "Choose Cell" menu (figure 9) to zoom in and out and store the selected cell location. After the user picks a location, LAHARZ searches a 50 cell by 50 cell area centered on the user-selected cell and stores the location of the cell within that area having the greatest elevation as the position for the cone apex. Moreover, the user can specify an optional "New Maximum Elevation" value that LAHARZ adds to the elevation of the automatically or manually selected cone apex cell to represent an eruption plume height of a volcano.

After the user has entered the parameter values for the cone, LAHARZ calculates a grid of H/L values where each cells' value is determined by the difference between the elevation of the cone apex and the product of that cell's Euclidean distance from the location of the cone apex multiplied by the constant, user-defined cone slope. LAHARZ compares the values of the H/L grid with the elevations of the DEM at each cell location. In a temporary grid, LAHARZ assigns cells an arbitrary value of 119 to cells where the H/L value is greater than the DEM elevation and a value of 71 to those cells where H/L values are less than the DEM elevation. LAHARZ creates a polygon that encircles the area defined by the boundary between cells having a value of 119 and those having a value of 71. LAHARZ then converts the polygon boundary back into a grid. LAHARZ names this grid by appending the prefix "hl" and the suffix "_g" to the user-specified H/L slope value. This grid contains a line of intersection between the H/L cone and the DEM, one cell in width, that is the proximal-hazard zone boundary. LAHARZ plots this boundary over a shaded relief version of the DEM on-screen.

Select Stream

LAHARZ calculates lahar-inundation hazard zones for a single stream drainage at a time. The user selects a stream by choosing any stream cell that lies within the proximal-hazard zone boundary (an "upstream" cell). The user chooses a second cell (a "downstream" cell) on the selected stream that lies at or near the edge of the grid to prevent overrunning the grid. Users must be aware that LAHARZ will stop calculations of lahar-inundation hazard zones when it encounters the downstream cell. If LAHARZ does stop calculations by encountering the downstream cell, the area of inundation is invalid. The user must re-run LAHARZ using a smaller volume or aquire a grid covering a larger area to calculate the inundation zone.

After the user presses the "Select Stream" button from the "LAHARZ - Main Menu" menu, LAHARZ presents the "LAHARZ - Select Stream" menu shown in (figure 10). This menu has four scrolling lists of grids stored in the current working directory. A label for each list indicates the type of grid the user should select. In the upper left corner of the menu the user selects an H/L grid, in the upper right corner a flow direction grid, in the lower right corner a stream grid, and the lower left corner an elevation grid (DEM). After selecting four grids and pressing the "Continue" button, LAHARZ generates a graphics display with the streams (drawn in cyan) and the proximal-hazard zone boundary (drawn in magenta). LAHARZ presents a graphics display of these features and a "Choose Cell" menu containing buttons to zoom, pan, and redraw the grid (figure 11). Pressing the "Query cell" button prints messages to the screen indicating whether the cell selected is a valid or invalid stream cell. Pressing the "Take as Upstream cell" or "Take as

Downstream cell" stores the location of the currently selected cell. When the user is satisfied with the two selected cells, they press the button labeled "Continue". LAHARZ begins with the location of the user-specified upstream cell and checks successive downstream cell locations until it encounters a cell location where the stream and the proximal-hazard zone boundary intersect. LAHARZ begins deriving lahar-inundation hazard zones for the drainage from the location of this intersection.

Create Lahar-Inundation Hazard Zones

After using LAHARZ to produce a depressionless DEM (if necessary), generate a flow direction grid, create a proximal-hazard zone boundary, and select a stream, the user must specify one or more lahar volumes. Users determine appropriate volumes on the basis of experience, judgment, and knowledge of the particular volcano (see Iverson and others, 1998). After the user presses the "Create Lahar Inundation Zones" button from the "LAHARZ-Main Menu" menu, LAHARZ presents the "LAHARZ - Create Lahar Inundation Zones" menu containing two scrolling lists of grids from which the user selects a DEM and a flow direction grid. The menu also contains data entry boxes where the user specifies a drainage name and up to four lahar volumes in cubic meters (figure 12). LAHARZ uses the equations $A = 0.05\ V^{2/3}$ and $B = 200\ V^{2/3}$ to calculate the cross section area (A) and total planimetric area (B) (Iverson and others, 1998) for each of the user-specified lahar volumes. At each successive downstream cell, LAHARZ constructs multiple cross sections for each volume specified by the user.

When processing of inundation zones is complete LAHARZ will have created a text file and four grids. The text file, named <drainage name>.pts, contains the user-specified volumes, the cross-sectional and planimetric area values calculated by LAHARZ from those user-specified volumes, and a list of decreasing planimetric areas, one row of values per stream cell. Each grid will be named with a prefix of the user-specified drainage name appended to a suffix of the numeral 1,2,3, or 4. The suffix 1 corresponds to the largest user-specified volume, a suffix of 2 for the second largest user-specified volume, and so on. The user can either plot these lahar-inundation hazard zone grids over a shaded relief version of the DEM using the IMAGE command with the TRANSPARENT option in GRID or convert the grids to polygons and plot with other themes of data to assess the extent of the hazard zones.

ACKNOWLEDGMENTS

I thank Richard Iverson for numerous conversations and guidance while completing this project. Roland Viger offered valuable advice and discussion on an early version of the software. William Scott and Tom Pierson suggested Mount Hood, Oregon as an area to test this software and potentially help their hazard assessment. John Ewert, Kevin Scott and Jim Vallance provided valuable discussion and feedback. Lisa Faust provided the illustrations and graphic design for this manuscript.

REFERENCES

Beget, J. E. and Limke, A. J., 1988, Two-dimensional kinematic and rheological modeling of the 1912 pyroclastic flow, Katmai, Alaska: Bulletin of Volcanology, 50, p. 148-160.

Crandell, D.R., 1971, Postglacial lahars from Mount Rainier volcano Washington: U.S. Geological Survey Professional Paper 677, 75 p.

ESRI (Environmental Systems Research Institute), 1990, Understanding GIS the ARCINFO method, Environmental Systems Research Institute Inc., Redlands, California, 416 p.

ESRI (Environmental Systems Research Institute), 1991, Customizing ARCINFO with AML, Vol I & II, Environmental Systems Research Institute Inc., Redlands, California, 1026 p.

ESRI (Environmental Systems Research Institute), 1993, Arc macro language developing ARCINFO menus and macros with AML, Environmental Systems Research Institute Inc., Redlands, California, 738 p.

ESRI (Environmental Systems Research Institute), 1994, Cell-based Modeling with GRID, Environmental Systems Research Institute Inc., Redlands, California, p. 309-327.

Hayashi, J.N. and Self, S., 1992, A comparison of pyroclastic flow and debris avalanche mobility: Journal of Geophysical Research, 97(B6), pp. 9063-9071.

Hoblitt, R.P., Walder, J.S., Driedger, C.L., Scott, K.M., Pringle, J.T., and Vallance, J.W., 1995, Volcano hazards from Mount Rainier, Washington: U.S. Geological Survey Open-File Report 95-273, 10 p.

Hsu, K.J., 1975, Catastrophic debris streams (sturzstroms) generated by rockfalls: Geological Society of America Bulletin, v. 86, p. 129-140.

Iverson, R.M., Schilling, S.P., and Vallance, J.W., 1998, Objective delineation of areas at risk from inundation by lahars: Geological Society of America Bulletin, V. 110, no.8, p. 972-984.

Jenson, S. K. and Domingue, J. O., 1988, Extracting topographic structure from digital elevation data for geographic information system analysis: Photogrammetric Engineering and Remote Sensing, 54, p. 1593-1600.

Malin, M.C. and Sheridan, M.F., 1982, Computer-assisted mapping of pyroclastic surges: Science, V(217), p. 637-639.

Scott, K.M., Vallance, J.W., and Pringle, P.T., 1995, Sedimentology, behavior, and hazards of debris flows at Mount Rainier, Washington: U.S. Geological Survey Professional Paper 1547, 56 p.

Scott, W. E., Pierson, T. C., Schilling, S. P., Costa, J. E., Gardner, C. A., Vallance, J. W., and Major, J. J., 1997, Volcano hazards in the Mount Hood Region, Oregon: U.S. Geological Survey Open-File Report 97-89, 14 p.

Sheridan, M. F., 1979, Emplacement of pyroclastic flows: A review, *in* Chapin, C. E. and Elston, W. E., editors, Ash-flow tuffs: Geological Society of America Special Paper 180, p. 125-136.

Siebert, L., 1984, Large volcanic debris avalanches: characteristics of source areas, deposits, and associated eruptions: Journal of Volcanology and Geothermal Research, 22, p. 163-197.

APPENDIX 1

LAHARZ menus and AML's

```
/* Laharz.aml
/*
/* Laharz software constructs cross section and planimetric areas to create
/* predicted inundation zones for large volcanic debris flows (lahars). Laharz
/* runs primarily in the GRID module of ArcInfo and requires input of a Digital
/* Elevation Model (DEM) and one or more initial lahar volumes. This aml calls
/* the main menu for the program, laharz.menu. The main menu consists of five
/* buttons:
/*
/*   Create Surface Hydrology Grids
/*   Create H/L Cone
/*   Select Up and Downstream Points
/*   Construct Stream Cross Sections
/*   Quit
/*
/* Each of these choices spawn menus and(or) graphics displays for the user to
/* create required hydrologic grids, create a proximal hazard zone (the
/* intersection of an H/L "mobility" cone surface and a topographic surface),
/* identify a stream drainage for the inundation calculations by picking an
/* upstream and downstream cell for a stream, and construct cross sections
/* that contribute areas to a cumulative planimetric area for each input volume.
/* Laharz output is a set of nested planimetric inundation areas, one area derived
/* from one input volume. The user selects a task by pressing one of the top
/* four buttons. If Laharz carries out the task successfully, processing control
/* returns to Laharz.menu.
/*
/*
/*
&severity &error &routine bailout
/*=================================
/* check which program is running
/*---------------------------------
&if [show program] ne ARC &then
  &return This program must run from ARC.

&terminal 9999
/*&echo &on
&sv .stopl = .FALSE.

&menu laharz.menu &position &UL &screen 200 200 &stripe 'LAHARZ - Main Menu'

/*======================
/* CLEANUP
/*======================
/*&echo &off
&call exit
&return
/*============= Exit Routine ===============
/*
&routine exit
```

```
/* RESTORE ENVIRONMENTS
&return
/*============= Bailout Routine =============
/*
&routine bailout
/* RESET SEVERITY
&severity &error &ignore
/* BAIL OUT
&call exit
&return; &return &error Bailing out of Laharz.aml....
/* ENDDO
```

7 Laharz.menu

```
/*
/* This is the main menu for the laharz program. The user selects
/* a task by choosing one of five buttons:
/* Create Surface Hydrology Grids - calls surfhydro.aml that will
/* fill sinks in a DEM, if necessary, and create supplementary surface hydrology
/* grids.
/* Create H/L cone - calls makecone.aml that creates a proximal
/* hazard boundary formed by the intersection of an H/L mobility cone
/* and a topographic surface.
/* Select Up and Downstream Points - calls setupdn_stream.aml that
/* provides a graphical interface for the user to pick up and downstream cells
/* that identify a stream.
/* Construct Stream Cross Sections - calls make_lahar.aml that
/* constructs cross section and planimetric areas from initial lahar
/* volumes provided by user. Also, creates predicted inundation zones for the
/* selected stream drainage.
/*
/*

   %button1

        %button2

   %button3

   %button4


               %button5
%button1 BUTTON KEEP 'Create Surface Hydrology Grids' &r surfhydro.aml
%button2 BUTTON KEEP 'Create H/L Cone' &r makecone.aml
%button3 BUTTON KEEP 'Select Up and Downstream Points' &r setupdn_stream.aml
%button4 BUTTON KEEP 'Construct Stream Cross Sections' &r make_lahar.aml
%button5 BUTTON KEEP 'QUIT' &dv .stopl; &return QUITTING LAHARZ...
```

```
/* Surfhydro.aml
/* This aml calls the setup_surfhydro.menu. This setup menu sets global
/*  variables to names of grids used in surfhydro.aml.  Surfhydro uses a local
/*  variable, areaname that stores the first three characters of the DEM name.
/*
/* If the user chooses to create a depressionless DEM, the program will run the
/* GRID fill function on the DEM creating an output grid <areaname>fill.  The
/* fill grid is input for the flow direction function that determines the direction
/* of flow out of every grid cell.  The directions of flow out of each cell are stored
/* in a flow direction grid named <areaname>dir.  The flow direction grid is input
/* for the flow accumulation function that counts the number of upslope cells that
/* flow into each cell in the grid.  These numbers are stored in a grid named
/* <areaname>flac.  Each cell in the flow accumulation grid is compared to a
/* threshold value.  If a cell's flow accumulation value is greater than the
/* threshold value, the output grid, <areaname>str, is given a value of 1,
/* otherwise the cell contains NODATA.
/*
/* Supplementary, surface hydrologic grids created by surfhydro.aml:
/* fill grid, flow direction grid, flow accumulation grid, stream grid
/*
/* Global Variables:
/*    .depless_g   - run fill function on DEM
/*    .elev_g      - elevation grid
/*    .threshval   -  thresh hold value defining stream
/*    .sink_thresh  -  thresh hold value for filling sinks
/*
/*
/*
&severity &error &routine bailout
/*================================
&terminal 9999   /* set for menu display
/*&echo &on          /* echo progress to the screen
display 1040 /* turn off display

/*======================================
/* initialize variables
/*======================================
&sv .depless_g = .FALSE.
&sv .elev_g = nullgrid
&sv .sink_thresh = 0
&sv .threshval = 0
&sv .stopl = .FALSE.

&if NOT [show program] = ARC &then      /* start GRID
  &do
    &type This program must be run from ARC
    &return
  &end
&else
   GRID
```

```
/*========================================
/*   call setup_surfhydro.menu
/*========================================
&menu setup_surfhydro.menu &position &UL &screen 100 100 ~
  &stripe 'LAHARZ - Create Surface Hydro Grids'

/*========================================
/* if user quits menu, call exit
/*========================================
&if %.stopl% &then
  &call exit
&else
  &do

/*========================================
/* store first three characters of elevation
/* grid in local variable
/*========================================
&sv areaname = [substr [entryname %.elev_g%] 1 3]


/*========================================
/* if user chooses, run fill algorithm on
/* input DEM to create fill grid
/*========================================
&if %.depless_g% &then
  fill %.elev_g% %areaname%fill sink %.sink_thresh%

/*========================================
/* create flow direction grid <areaname>dir
/* create flow accumulation grid <areaname>flac
/* create stream grid <areaname>str where
/* flow accumulation values are greater than
/* stream threshold value
/*========================================
%areaname%dir = flowdirection(%areaname%fill)
%areaname%flac = flowaccumulation(%areaname%dir)
%areaname%str = con(%areaname%flac > %.threshval%, 1)

&end
/*&echo &off
&call exit
&return
/*============== Exit Routine =================
/*
&routine exit
&if [show program] = GRID &then
  &do
    QUIT
    &dv .depless_g .threshval .elev_g .sink_thresh  /* release global variables
```

```
  &end
&return
/*============= Bailout Routine =============
/*
&routine bailout
/* RESET SEVERITY
&severity &error &ignore
/* BAIL OUT
&call exit
&return; &return &error Bailing out of Surfhydro.aml....
/* ENDDO
```

7 setup_surfhydro.menu

```
/* menu
/* This menu has the user select a Digital Elevation Model (DEM) and
/* select if they want to make the DEM depressionless (fill sinks).
/* The user can also enter a threshold value, where flow accumulation
/* values above the threshold defines a stream, or take the default
/* theshold value of 1000.
/* Global variables:
/*   .elev_g    : the name of a DEM
/*   .depless_g : flag whether to run fill function to create depressionless
/*              DEM
/*   .threshval : threshold value for flow accumulation grid that defines
/*              a stream


   SELECT AN ELEVATION GRID (DEM)
  OR DEPRESSIONALESS (FILLED) DEM:
  %datalist1






  Make Elevation Grid Depressionless?
        %button3  %button4


 Enter fill threshold:  %input3


 Enter stream threshold: %input1

   %button1              %button2
%datalist1 INPUT DATALIST1 36 TYPEIN NO SCROLL YES ROWS 5 ~
 KEEP ~
 RETURN '&sv .elev_g = %datalist1%;' ~
 GRID ~
 * -SORT
%button3 BUTTON KEEP 'NO' &sv .depless_g = .FALSE.;
%button4 BUTTON KEEP 'YES' &sv .depless_g = .TRUE.;
%input3 INPUT .sink_thresh 6 TYPEIN YES SCROLL NO SIZE 4 ~
 KEEP ~
 INITIAL '100' ~
 RANGE 1 1000 ~
 RETURN '&sv .sink_thresh = %.sink_thresh%;' ~
 INTEGER
%input1 INPUT .threshval 6 TYPEIN YES SCROLL NO SIZE 5 ~
 KEEP ~
 INITIAL '1000' ~
 RANGE 1 20000 ~
```

```
 RETURN '&sv .threshval = %.threshval%;' ~
 INTEGER
%button1 BUTTON KEEP 'QUIT' &sv .stopl = .TRUE.; &return STOPPING CREATION
OF HYDROLOGIC GRIDS...
%button2 BUTTON KEEP 'Continue' &return CREATING SUFACE HYDROLOGIC
GRIDS...
```

```
/* Makecone.aml
/*
/* This aml creates a grid that contains a proximal hazard zone boundary for a
/* volcano. The boundary marks the intersection between a hypothetical H/L
/* mobility cone and a topographic surface (DEM). After starting GRID, this
/* program calls setup_cone.menu. This menu gets parameters from the user to
/* make the H/L mobility cone and create a shaded relief grid from the user-
/* selected DEM. If the user chooses the maximum elevation of the grid as the
/* location and elevation of the cone apex, this program calls hipnt.aml, otherwise
/* it calls pikhipnt.aml.

/* Global Variables
/*   .shade_g    - stores the name of the shaded relief grid
/*   .peakchoice - flag for method of how to place cone apex
/*   .stop       - stop flag
/*   .hl         - value of cone slope
/*   .elevchange - optional value of elevation, represents plume height
/*   .grdx       - X coordinate of selected cell
/*   .grdy       - Y coordinate of selected cell
/*
/*
/*
&severity &error &routine bailout
display 1040

/*=====================================
/* check if running in arc, if so start GRID
/*=====================================
&if [show program] ne ARC &then
  &return This program must run from ARC.
&else
  GRID

/*=====================================
/* set terminal and initialize variables
/*=====================================
&terminal 9999
&sv .peakchoice = 0
&sv .stopl = .FALSE.
&sv .hl = 0
&sv .elev_g = nullgrid
&sv .shade_g = nullgrid
&sv .elevchange = 0
&sv .grdx = 0
&sv .grdy = 0

/*=====================================
/*  call setup_cone.menu
/*=====================================
&menu setup_cone.menu &position &UL &screen 100 100 &stripe 'LAHARZ - Create
```

```
H/L Cone Grid'

/*======================================
/* if user quits menu, call exit
/*======================================
&if %.stopl% &then
  &call exit
&else
  &do

/*======================================
/* if the shaded relief grid does not exist
/* create it from elevation grid
/*======================================
&if NOT [EXISTS %.elev_g%sh -GRID] &then
  &do
    &type CREATING SHADED RELIEF GRID...
    %.elev_g%sh = hillshade(%.elev_g%,290,40)
  &end

&sv .shade_g = %.elev_g%sh

&describe %.elev_g% /* load elevation attributes into GRID variables
&sv cellsize = %grd$dx%

/*======================================
/* use global variable .peakchoice to govern which cell
/* is isolated for use as apex of H/L mobility cone
/*
/* .peakchoice = 1 ==> create H/L cone with apex at
/*                     highest elevation in DEM
/* .peakchoice = 2 ==> select location of H/L cone apex.
/*                     call pikhipnt.aml for the user to
/*                     select the elevation for the cone
/*                     apex interactively. Stores the
/*                     X coordinate of the selected cell
/*                     in .grdx and the Y coordinate in
/*                     .grdy
/*======================================
&if %.peakchoice% = 2 &then
  &do
    &r pikhipnt.aml  /* select location of cone apex
  &end

display 1040
junk.gra


/*======================================
/* if user quits menu, call exit
```

```
/*========================================
&if %.stopl% &then
  &call exit
&else
  &do

&if [EXISTS xxhipntgrd -GRID] &then kill xxhipntgrd all  /* delete old xxhipntgrd

/*========================================
/* if .peakchoice = 2
/* creates a window 50 times the cellsize in width and
/* height that is centered on picked location (stored in
/* .grdx and .grdy). Creates a small grid the size of the
/* window. Values in the small grid are taken from the
/* elevation grid. Describe loads the grid variables from
/* the small grid. .elevchange is added to the maximum
/* elevation of the small grid. An intermediate grid stores
/* null values for all cells other than the cell having the
/* maximum elevation. The window is set to the .elev_g
/* (DEM) to create xxhipntgrd grid with all cells having
/* values of null except the user-selected cell.
/*
/* if .peakchoice = 1
/* all cells are changed to null except cell with maximum
/* elevation
/*========================================

 &type
 &type CREATING INTERMEDIATE GRIDS ...
 &type

/* All grid cells, except cone apex location, set to NODATA
&if %.peakchoice% = 2 &then
  &do
    &sv cursor = %cellsize% * 50.0
    setwindow [calc %.grdx% - %cursor%] [calc %.grdy% - %cursor%] ~
          [calc %.grdx% + %cursor%] [calc %.grdy% + %cursor%]
    xxsmpntgrd = %.elev_g%
    &describe xxsmpntgrd
    &sv .maxelev = [calc %grd$zmax% + %.elevchange%]
    xxintergrd = setnull(xxsmpntgrd < [calc %grd$zmax% - 1], %.maxelev%)
    kill xxsmpntgrd all
    setwindow %.elev_g%
    xxhipntgrd = xxintergrd
    kill xxintergrd all
  &end
&else   /* .peakchoice = 1
  &do
   &describe %.elev_g%
   &sv .maxelev = [calc %grd$zmax% + %.elevchange%]
```

```
  xxhipntgrd = setnull(%.elev_g% < [calc %grd$zmax% - 1], %.maxelev%)
 &end
/*===================================
/* call hipnt.aml to create H/L cone grid
/*===================================
&if NOT %.stopl% &then
 &r hipnt.aml
&end
&end
/*===================================
/*         CLEANUP
/*===================================
&call exit
&return
/*============= Exit Routine ===============
/*
&routine exit
&if [show program] = GRID &then
 &do
  QUIT
  &dv .peakchoice .hl .grdx .grdy .shade_g .elev_g .elevchange .maxelev
 &end

&return
/*============= Bailout Routine ============
/*
&routine bailout
/* RESET SEVERITY
&severity &error &ignore
/* BAIL OUT
&call exit
&return; &return &error Bailing out of makecone.aml....
/* ENDDO
```

7 setup_cone.menu

```
/*
/* The user selects an elevation grid (DEM), enters a value for the slope of the
/* H/L mobility cone, and chooses whether the apex of the H/L cone
/* is at the maximum elevation in the DEM or at a user-selected cell.
/* User can enter or pick on a slide bar to add elevation that represents an
/* eruption plume height.
/*
/* Global Variables:
/*   .elev_g      - stores the name of the elevation grid
/*   .hl          - stores the value of the slope of the H/L mobility cone
/*   .peakchoice  - stores whether the apex of the mobility cone is at the
/*                  maximum elevation or at a user-selected point
/*   .elevchange  - stores the elevation value that user wants to add to
/*                  the maximum elevation of the elevation grid
/*
/*
/*
  SELECT AN ELEVATION GRID:
  %datalist1




     Select H/L value:
   %slider1

     New maximum elevation:
   %slider2

       SELECT ONE:
      %button1

     %button2

  %button4          %button3
%datalist1 INPUT DATALIST1 32 TYPEIN NO SCROLL YES ROWS 4 ~
 KEEP ~
 RETURN '&sv .elev_g = [ENTRYNAME %datalist1%];' ~
 GRID ~
 * -SORT
%slider1 SLIDER SLIDER1 28 KEEP TYPEIN YES ~
 RETURN '&sv .hl = %slider1%;' ~
 STEP 0.001 ~
 REAL 0.010 0.999
%slider2 SLIDER SLIDER2 28 KEEP TYPEIN YES ~
 INITIAL 0.000 ~
 RETURN '&sv .elevchange = %slider2%;' ~
 STEP 10.000 ~
 REAL 0.000 10000.000
```

```
%button1 BUTTON KEEP 'Maximum Elevation' &sv .peakchoice = 1;
%button2 BUTTON KEEP 'Select an Elevation' &sv .peakchoice = 2;
%button4 BUTTON KEEP 'QUIT' &sv .stopl = .TRUE.; &return STOPPING H/L CONE
GENERATION...;
%button3 BUTTON KEEP 'Continue' &return CREATING H/L CONE...
```

```
/* Pikhipnt.aml
/*
/* Places a display window on the screen and draws the shaded relief grid
/* in the display. Calls pikhipnt.menu to place a control menu to the upper
/* right of the display window.
/*
/*
/*
&severity &error &routine bailout
/*==================================
&if NOT [show program] = GRID &then
  &return there is a problem in pikhipnt.aml

mape %.shade_g%
&sv sh_mape = [show mapextent]
&sv currwide = [extract 3 %sh_mape%] - [extract 1 %sh_mape%]
&sv currhigh = [extract 4 %sh_mape%] - [extract 2 %sh_mape%]
&if currwide > currhigh &then
  &do  /* landscape
    &sv framex = 1000
    &sv framey = 800
  &end
  &else
  &do
    &sv framex = 800
    &sv framey = 1000
  &end

/* set the size and position of the Xwindow
DISPLAY 9999 ~
  POSITION 10 10 ~
  SIZE FRAME %framex% %framey%

/* draw shaded relief grid in Xwindow
mape %.shade_g%
gridp %.shade_g% value linear # gray

/* place the pikhipnt.menu on the upper right side of the Xwindow display
&menu pikhipnt ~
  &position &ul &display &ur ~
  &stripe 'Choose Cell'

&echo &off
/*======================
/* CLEANUP
/*======================
&call exit
&return
/*============= Exit Routine ================
/*
```

```
&routine exit
/* RESTORE ENVIRONMENTS
&return
/*============= Bailout Routine =============
/*
&routine bailout
/* RESET SEVERITY
&severity &error &ignore
/* BAIL OUT
&call exit
&return; &return &error Bailing out of Pikhipnt.aml....
/* ENDDO
```

7 pikhipnt.menu

```
/*
/* This menu provides buttons for a user to zoom in to a user-defined box, zoom
/* to full extent of shaded relief grid, redraw the grid, or select points on the grid
/* with a mouse. The zoom window button calls pikcontrol.aml and passes a
/* parameter to control the type of zoom.  Redraw calls drwgrd.aml that refreshs
/* the display window.  Elevation query calls pikcontrol.aml and passes a
/* parameter that requesting storage of the x and y coordinates of the selected
/* cell location
/*
/*
/*

      %2

      %1

      %4

      %3


 %stop          %go
%2 BUTTON 'Full View' &r pikcontrol all
%1 BUTTON 'Zoom Window' &r pikcontrol box
%4 BUTTON 'Redraw Grid' &r drwgrd 1 %.shade_g%
%3 BUTTON 'Query Elevation' &r pikcontrol elev
%stop BUTTON 'QUIT' &sv .stopl = .TRUE.; &return STOPPING PICK OF CONE APEX
LOCATION...
%go BUTTON 'Continue' &return CREATING H/L GRID...
```

```
/* Pikcontrol.aml
/*
/* This aml gets passed a parameter from pikhipnt.menu
/* Parameter passed:
/*   box  - sets mapextent to a user-defined window, clears the display and
/*          calls drwgrd.aml with two parameters, a number controlling the
/*          type of draw and the name of the grid to be drawn
/*   all  - sets the mapextent to the elevation grid extent, clears the display
/*          and calls drwgrd.aml with two parameters, a number controlling the
/*          type of draw and the name of the grid to be drawn
/*   elev - flushes points from point buffer, takes x and y coordinates from
/*          mouse-picked location, stores x coordinate in .grdx and y
/*          coordinate in .grdy, gets the elevation of the location, and
/*          displays the coordinates and elevation for the user

/* Global Variables
/*    .grdx stores the x value of the last point selected with mouse
/*    .grdy stores the y value of the last point selected with mouse
/*
/*
/*
&severity &error &routine bailout
/*===================================
&args choice

&if [NULL %choice%] &then
  &return usage pikcontrol <choice>

&select %choice%
  &when box
  &do
    mapextent *
    clear
    &r drwgrd 1 %.shade_g%  /* call drwgrd.aml
  &end
  &when all
  &do
    mapextent %.elev_g%
    clear
    &r drwgrd 1 %.shade_g%  /* call drwgrd.aml
  &end
  &when elev
  &do
    &flushpoints
    &getpoint &map &push &mouse     /* get the X,Y coords in buffer
    &sv .grdx = [translate %PNT$X%]  /* store X coordinate
    &sv .grdy = [translate %PNT$Y%]  /* store Y coordinate
                                     /* store elevation
    &sv elev = [show cellvalue %.elev_g% %PNT$X% %PNT$Y%]
    &type X,Y (%.grdx% , %.grdy%)  Elevation: %elev%  /* write to screen
```

```
    &flushpoints
  &end
&end /* select when

/*======================
/* CLEANUP
/*======================
&call exit
&return
/*============= Exit Routine ================
/*
&routine exit
/* RESTORE ENVIRONMENTS
&return
/*============= Bailout Routine =============
/*
&routine bailout
/* RESET SEVERITY
&severity &error &ignore
/* BAIL OUT
&call exit
&return; &return &error Bailing out of pikcontrol.aml....
/* ENDDO
```

```
/* Drwgrd.aml
/*
/* This aml receives two parameters.  The first controls how a grid is drawn
/* to the screen and the second contains the name of the grid that will be
/* drawn to the screen
/*
/*
/*
&severity &error &routine bailout
/*=================================
&args draw shd_g

&if [NULL %shd_g%] &then
  &return usage drwgrd <draw> <shd_g>

&select %draw%
  &when 1 ; &do
    gridpaint [translate %shd_g%] value linear # gray
  &end
  &when 2 ; &do
    display 9999 3
    mape %shd_g%
    gridpaint [translate %shd_g%] value linear # gray
    &pause
  &end
  &when 3 ; &do
    grids [translate %shd_g%]
  &end
&end /* select

/*=====================
/* CLEANUP
/*=====================
&call exit
&return
/*============= Exit Routine ================
/*
&routine exit
/* RESTORE ENVIRONMENTS
&return
/*============= Bailout Routine =============
/*
&routine bailout
/* RESET SEVERITY
&severity &error &ignore
/* BAIL OUT
&call exit
&return; &return &error Bailing out of drwgrd.aml....
/* ENDDO
```

```
/* Hipnt.aml
/*
/* This aml creates a grid representation of a cone having a user-defined slope
/* and apex position. The aml compares elevations of the cone surface and the
/* DEM. Assigns integer values to a grid of 71 (green in shadeset colornames)
/* where DEM elevation is greater than the cone elevation and 119 (magenta in
/* shadeset colornames) where the cone elevations are greater than DEM
/* elevations. Creates a polygon coverage from the grid and converts the
/* polygon boundary back into a grid to produce a boundary one cell in width that
/* defines a proximal hazard zone for the volcano
/*
/*
/*
&severity &error &routine bailout
/*====================================
/* If in Arc then start Grid. Make sure elevation grid
/* exists, if so, run describe to load grid variables.
/* Assign width of cell in x direction to local variable
/* called cellsize. Assign digits to right of decimal
/* in .hl variable to local variable ext.' ext used to name
/* H/L cone grid.
/*====================================

&if NOT [show program] = GRID &then
  &return there is a problem in Hipnt.aml

&if NOT [EXISTS %.elev_g% -GRID] &then
  &return GRID %.elev_g% not found...

/*====================================
/*  set cellsize and ext (name extension)
/*====================================
&describe %.elev_g% /* load elevation attributes into GRID variables
&sv cellsize = %grd$dx%
&sv ext = [after %.hl% .]

/*====================================
/* load grid variables from xxhipntgrd grid
/*====================================
&describe xxhipntgrd

&if [EXISTS xxdistgrd -GRID] &then kill xxdistgrd all
&if [EXISTS xxhazgrd%ext% -GRID] &then kill xxhazgrd%ext% all

/*====================================
/* calculate euclidian distance from every cell to the single
/* source (non-null cell) in xxhipntgrd. The distance is multiplied
/* by the slope of the cone and the result subtracted from the
/* highest cell elevation in xxhipntgrd. The results are stored
/* in xxhazgrd to create the H/L cone.
```

```
/*====================================

xxhazgrd%ext% = %grd$zmax% - (eucdistance(xxhipntgrd) * %.hl%)
&type
&type Creating H/L cone ....
&type
&if [EXISTS hlgrd%ext% -GRID] &then kill hlgrd%ext% all

/*====================================
/* compare the elevations in the DEM grid and the cone
/* grid for each cell, if the H/L cone grid is less than
/* the value in the elevation grid, then store 71 (green in
/* shadeset colornames), if not, make the cell value
/* 119 (magenta in shadeset colornames)
/*====================================
hlgrd%ext% = con(xxhazgrd%ext% < %.elev_g%,71,119)

/*====================================
/* create polygon cover of the zone, make a grid of the
/* boundary of the polygon, make it the same size as the
/* original grid, set window smaller by three cell widths
/* around the perimeter of the grid, then make larger again
/* to get rid of the grid boundary
/*====================================

&type
&type FILE MANIPULATION IN PROGRESS...
&type
&if [EXISTS hlcovp%ext% -COVER] &then kill hlcovp%ext% all
hlcovp%ext% = gridpoly(hlgrd%ext%)

&if [EXISTS hlgrd_l%ext% -GRID] &then kill hlgrd_l%ext% all
hlgrd_l%ext% = linegrid(hlcovp%ext%,#,#,#,%cellsize%,NODATA)

setwindow hlgrd%ext%

&if [EXISTS hlgrd2%ext% -GRID] &then kill hlgrd2%ext% all
hlgrd2%ext% = hlgrd_l%ext%

&describe hlgrd%ext%
&sv cellsize = [calc 3 * %cellsize%]

setwindow [calc %grd$xmin% + %cellsize%] [calc %grd$ymin% + %cellsize%] ~
      [calc %grd$xmax% - %cellsize%] [calc %grd$ymax% - %cellsize%]

&if [EXISTS hlgrd3%ext% -GRID] &then kill hlgrd3%ext% all
hlgrd3%ext% = hlgrd2%ext%

setwindow hlgrd%ext%
&if [EXISTS hl%ext%_g -GRID] &then kill hl%ext%_g all
```

```
hl%ext%_g = hlgrd3%ext%

/*========================================
/* combine the H/L intersection of cone and DEM with
/* the shaded relief grid then draw the result to
/* the display
/*========================================

&if [EXISTS hlgrd5%ext% -GRID] &then kill hlgrd5%ext% all
hlgrd5%ext% = con(isnull(hl%ext%_g),%.shade_g%,254)

&type
&type DONE PROCESSING...DRAWING RESULT...
&TYPE
&r drwgrd 2 hlgrd5%ext%

/*========================================
/* kill intermediate grids and coverages
/*========================================

&type
&type DELETING INTERMEDIATE GRIDS...
&TYPE
&if [EXISTS hlcovp%ext% -COVER] &then kill hlcovp%ext% all
&do kill_g &list hlgrd_1%ext% hlgrd2%ext% hlgrd3%ext% xxhipntgrd ~
hlgrd%ext% xxhazgrd%ext% hlgrd5%ext%
  &if [EXISTS %kill_g% -GRID] &then kill %kill_g% all
&end

&delvar ext cellsize

/*======================
/* CLEANUP
/*======================
&echo &off
&call exit
&return
/*============= Exit Routine ================
/*
&routine exit
/* RESTORE ENVIRONMENTS
&return
/*============= Bailout Routine =============
/*
&routine bailout
/* RESET SEVERITY
&severity &error &ignore
/* BAIL OUT
&call exit
&return; &return &error Bailing out of Hipnt.aml....
```

```
/* Setupdn_stream.aml
/*
/* This aml spawns setupdn_stream.menu to get choices of four grids from user.
/* After returning from the menu, this aml merges H/L cone and stream grids to
/* create a temporary grid drawn to a graphics display.  A control menu,
/* pikstreampts.menu displays next to the graphics display.  After the user picks
/* up and downstream cells, the downstream is written to a temporary grid and
/* movedown.aml is called to traverse from the picked upstream cell to a cell
/* where the chosen stream and the proximal hazard zone boundary intersect.
/*
/* Global Variables:
/*   .hl_g      - proximal hazard zone boundary grid
/*   .dir_g     - flow direction grid
/*   .str_g     - stream grid
/*   .elev_g    - elevation grid
/*   .hlstr_g   - combined h/l and stream grids
/*   .currwide  - width of .hlstr_g map extent
/*   .currhigh  - height of .hlstr_g map extent
/*   .cellwidth - cellwidth of .hlstr_g
/*   .nextx     - x coordinate of next downstream cell
/*   .nexty     - y coordinate of next downstream cell
/*   .currdir   - current flow direction of .hlstr_g
/*   .currhl    - current value of .hlstr_g
/*   .count     - counter
/*   .found1stpt - flag set to true when program finds an h/l cell
/*
/*

&severity &error &routine bailout
/*================================
&terminal 9999
/*&echo &on
display 1040

/*=======================================
/* initialize variables
/*=======================================
&sv .hl_g = nullgrid
&sv .dir_g = nullgrid
&sv .str_g = nullgrid
&sv .hlstr_g = nullgrid
&sv .nextx = 0
&sv .nexty = 0
&sv .found1stpt = .FALSE.
&sv .currhl = 0
&sv .upstrx = 0
&sv .upstry = 0
&sv .cellwidth = 0
&sv .currdir = 0
&sv .count = 0
```

```
&sv .stopl = .FALSE.

/*=====================================
/* check if running in arc, if so start GRID
/*=====================================
&if NOT [show program] = ARC &then
  &return This program must run from ARC
&else
  GRID

/*=====================================
/*   call setupdn_stream.menu
/*=====================================
&menu setupdn_stream.menu &position &UL &screen 100 100 ~
  &stripe 'LAHARZ - Select Grids to Pick Stream Cells'

/*=====================================
/* if user quits menu, call exit
/*=====================================
&if %.stopl% &then
  &call exit
&else
  &do

/*=====================================
/* merge h/l grid and stream grid to create
/* xxdrwstr grid.  in xxdrwstr grid, stream
/* cells have value of 56 (cyan in shadeset
/* colornames), h/l cells have value of 119
/* (magenta in shadeset colornames.  These
/* values will be used later when drawing
/* xxdrwstr grid to the display.  set global
/* variable .hlstr_g to xxdrwstr.
/*=====================================
&if [EXISTS xxdrwstr -GRID] &then kill xxdrwstr all
  &type
  &type MERGING H/L CONE AND STREAM GRIDS...
  &type
  xxdrwstr = con(isnull(%.hl_g%),con(%.str_g% > 0,56),119)
  &sv .hlstr_g = xxdrwstr

/*=====================================
/* load .hlstr_g grid information into GRID
/* variables, store width and heighth of
/* map extent
/*=====================================
mape %.hlstr_g%
&sv str_mape = [show mapextent]
&sv .currwide = [extract 3 %str_mape%] - [extract 1 %str_mape%] /* xmax - xmin
&sv .currhigh = [extract 4 %str_mape%] - [extract 2 %str_mape%] /* ymax - ymin
```

```
&if %.currwide% > %.currhigh% &then
  &do  /* landscape
    &sv framex = 1000
    &sv framey = 800
  &end
  &else
  &do
    &sv framex = 800
    &sv framey = 1000
  &end

DISPLAY 9999 ~
  POSITION 10 10 ~
  SIZE FRAME %framex% %framey%

shadeset colornames
mape %.hlstr_g%
grids %.hlstr_g%
&describe %.hlstr_g%

/*====================================
/*  call pikstreampts.menu
/*====================================
&menu pikstreampts ~
  &position &ul &display &ur ~
  &stripe 'Choose Cell'

/*====================================
/* if user quits menu, call exit
/*====================================
&if %.stopl% &then
  &call exit
&else
  &do

display 1040
junk.gra
&delvar .currwide .currhigh .grdx .grdy

/*====================================
/* load .hlstr_g(xxdrwstr)information into GRID
/* variables using describe.  store cellwidth,
/* x, y, current flow direction and current
/* hlstr_g values
/*====================================
&describe %.hlstr_g%
&sv .cellwidth = %grd$dx%
&sv .nextx = %.upstrx%
&sv .nexty = %.upstry%
&sv .currdir = [show cellvalue %.dir_g% %.nextx% %.nexty%]
```

```
&setvar .currhl = [show cellvalue %.hlstr_g% %.nextx% %.nexty%]


/*=====================================
/* initialize variables, set map extent,
/* set window and cell size to hlstr_g grid
/*=====================================
&sv .count = 0
&sv .found1stpt = .FALSE.
mape %.hlstr_g%
setwindow %.hlstr_g%
setcell %.hlstr_g%

/*=====================================
/* create a grid the same dimensions as hlstr_g
/* and set all cells to value of 1
/*=====================================
&if [EXISTS xxallones_g -GRID] &then kill xxallones_g all
&type
&type CREATING INITIAL FOOTPRINT GRID...
&type
xxallones_g = 1

/*=====================================
/* create a grid with a single cell value of 1
/* at the user picked downstream cell with all
/* other values set to NODATA. used to stop
/* calculating inundation areas if about to
/* run into edge of grid
/*=====================================
&if [EXISTS xxstop -GRID] &then kill xxstop all
&type
&type CREATING STOP GRID - AVOID GRID EDGE COLLISION...
&type
xxstop = selectpoint(xxallones_g,%.dwnstrx%,%.dwnstry%)

&delvar .dwnstrx .dwnstry
&setvar .currstop = [show cellvalue xxstop %.nextx% %.nexty%]

/*=====================================
/* Loop to traverse downstream from upstream cell
/* picked by user until encounter either h/l cell
/* or downstream stop cell
/*=====================================
&type
&type SEARCHING FOR PROXIMAL HAZARD ZONE BOUNDARY...
&type

&do &until %.nextx% < [show $$WX0] or %.nextx% > [show $$WX1] ~
  or %.nexty% < [show $$WY0] or %.nexty% > [show $$WY1] or ~
```

```
  %.found1stpt%
  &r movednstream.aml
&end /* until

&type
&type UP AND DOWN STREAM CELL SELECTION COMPLETE...
&type

&end
&end
&echo &off
/*=======================
/* CLEANUP
/*=======================
&call exit
&return
/*============= Exit Routine ================
/*
&routine exit
&if [show program] = GRID &then
  &do
    QUIT
    &dv .hl_g .dir_g .str_g .hlstr_g .nextx .nexty ~  /* release global variables
       .found1stpt .currhl .upstrx .upstry ~
       .cellwidth .currdir .count
  &end
&return
/*============= Bailout Routine =============
/*
&routine bailout
/* RESET SEVERITY
&severity &error &ignore
/* BAIL OUT
&call exit
&return; &return &error Bailing out of setupdn_stream.aml....
/* ENDDO
```

7 Setupdn_stream.menu

```
/*
/* The user selects an H/L cone, flow direction, stream and elevation grid. These
/* grids will be used to select up and downstream points that identify a stream for
/* Laharz to calulate predicted lahar inundation zones.

/* Global Variables:
/*    .hl_g   - proximal hazard zone boundary, h/l grid
/*    .dir_g  - flow direction grid
/*    .str_g  - stream grid
/*    .elev_g - elevation grid DEM or filled
/*
/*
/*
      SELECT AN H/L GRID:          ^SELECT A FLOW DIRECTION GRID:
  %datalist1                  %datalist1_1

     SELECT A STREAM GRID:         ^SELECT AN ELEVATION GRID:
  %datalist2                  %datalist1_2

                                          %button2
   %button1
%datalist1 INPUT DATALIST1 32 TYPEIN NO SCROLL YES ROWS 4 ~
 KEEP ~
 RETURN '&sv .hl_g = %datalist1%;' ~
 GRID ~
 * -SORT
%datalist1_1 INPUT DATALIST1_1 32 TYPEIN NO SCROLL YES ROWS 4 ~
 KEEP ~
 RETURN '&sv .dir_g = %datalist1_1%;' ~
 GRID ~
 * -SORT
%datalist2 INPUT DATALIST2 32 TYPEIN NO SCROLL YES ROWS 4 ~
 KEEP ~
 RETURN '&sv .str_g = %datalist2%;' ~
 GRID ~
 * -SORT
%datalist1_2 INPUT datalist1_2 32 TYPEIN NO SCROLL YES ROWS 4 ~
 KEEP ~
 RETURN '&sv .elev_g = %datalist1_2%;' ~
 GRID ~
 * -SORT
```

%button2 BUTTON KEEP 'Continue' &return Setting up.. Please wait...
%button1 BUTTON KEEP 'QUIT' &sv .stopl = .TRUE.; &return STOPPING STREAM CELL SELECTION...

7 Pikstreampts.menu

```
/*
/* Displays buttons for user to zoom in or out by varuable user-defined window or
/* by factors of two, pan, redraw, query information about a cell, including
/* whether the cell is a valid or invalid cell, and select up and downstream cells.
/* The user selects an upstream cell that lies within the proximal hazard zone.
/* The up and downstream cell identify the stream laharz will use to create
/* potential lahar inundation zones. The user picks a downstream cell to ensure
/* processing does not go off the grid edge and to conduct trial runs.
/*
/*

      %2

      %1

      %button7

      %button8

        %3

      %4

 ================================

       %button9

 %button11

 %button10

 %stop            %go
%2 BUTTON 'Full View' &r pikstreampts all '&r drwgrd 3 %.hlstr_g%'
%1 BUTTON 'Zoom Window' &r pikstreampts box '&r drwgrd 3 %.hlstr_g%'
%button7 BUTTON KEEP 'Zoom in 2 X' &r pikstreampts half '&r drwgrd 3 %.hlstr_g%';
%button8 BUTTON KEEP 'Zoom out 2 X' &r pikstreampts double '&r drwgrd 3
%.hlstr_g%';
%3 BUTTON 'pan' &r pikstreampts pan '&r drwgrd 3 %.hlstr_g%'
%4 BUTTON 'Redraw Grid' &r drwgrd 3 %.hlstr_g%
%button9 BUTTON KEEP 'Query Cell' &r pikstreampts querycell '&r drwgrd 0 ~
%.hlstr_g';
%button11 BUTTON KEEP 'Take as Upstream Cell Value' ~
  &r pikstreampts takeupcell '&r drwgrd 0 %.hlstr_g%';
%button10 BUTTON KEEP 'Take as Downstream Cell Value' ~
  &r pikstreampts takedwncell '&r drwgrd 0 %.hlstr_g%';
%stop BUTTON 'QUIT' &sv .stopl = .TRUE.; &return STOPPING STREAM CELL
SELECTION...
%go BUTTON 'Continue' &return EXTRACTING STREAM...
```

```
/* Pikstreampts.aml
/* This aml takes one or two parameters.  The first parameter, choice,
/* is used to determine what action on the graphics screen takes place based
/* on a select statement.  Valid choices for choice are: box, all, half,
/* double, pan, querycell, takeupcell, takedowncell.  The second parameter,
/* drw:rest is a call to drwgrd.aml with desired parameters all in quotes.  The
/* parameter to run drwgrd.aml and its parameters are unquoted and
/* runs drwgrd.aml to redraw the screen.
/*
/* Global variables:
/*    .upstrx  - X coordinate of selected upstream cell
/*    .upstry  - Y coordinate of selected upstream cell
/*    .dwnstrx - X coordinate of selected downstream cell
/*    .dwnstry - Y coordinate of selected downstream cell
/*
/*

&severity &error &routine bailout
/*=================================
&args choice drw:rest

&if [NULL %drw%] &then
  &return usage stream <choice> <draw aml>

&select %choice%
&when box
&do
  mapextent *                 /* set mape to 2-point user box
  &call mapedraw
&end
&when all
&do
  mapextent %.hlstr_g%        /* set mape to merged hl and stream grid
  &call mapedraw
&end
&when half
&do
  &flushpoints
  &getpoint &map &push &mouse   /* get the X,Y coords in buffer
  /* Takes 1/2 of the width and height and puts 1/4 on each side
  /* of the point location
  &sv xmin = [calc %pnt$x% - ( %.currwide% / 4 ) ]
  &sv ymin = [calc %pnt$y% - ( %.currhigh% / 4 ) ]
  &sv xmax = [calc %pnt$x% + ( %.currwide% / 4 ) ]
  &sv ymax = [calc %pnt$y% + ( %.currhigh% / 4 ) ]
  &flushpoints
  mape %xmin% %ymin% %xmax% %ymax%
  &call mapedraw
&end
&when double
```

```
&do
  &flushpoints
  &getpoint &map &push &mouse  /* get the X,Y coords in buffer
  &sv xmin = [calc %pnt$x% - %.currwide% ]
  &sv ymin = [calc %pnt$y% - %.currhigh% ]
  &sv xmax = [calc %pnt$x% + %.currwide% ]
  &sv ymax = [calc %pnt$y% + %.currhigh% ]
  &flushpoints
  mape %xmin% %ymin% %xmax% %ymax%
  &call mapedraw
&end
&when pan
&do
  &flushpoints
  &getpoint &map &push &mouse  /* get the X,Y coords in buffer
  &sv xmin = [calc %pnt$x% - ( %.currwide% / 2 ) ]
  &sv ymin = [calc %pnt$y% - ( %.currhigh% / 2 ) ]
  &sv xmax = [calc %pnt$x% + ( %.currwide% / 2 ) ]
  &sv ymax = [calc %pnt$y% + ( %.currhigh% / 2 ) ]
  &flushpoints
  mape %xmin% %ymin% %xmax% %ymax%
  &call mapedraw
&end
&when querycell
&do
  &flushpoints
  &getpoint &map &push &mouse  /* get the X,Y coords in buffer
  &sv .grdx = [translate %pnt$x%]
  &sv .grdy = [translate %pnt$y%]
  &sv strval = [show cellvalue %.hlstr_g% %pnt$x% %pnt$y%]
  &type stream grid value = %strval%
  &select %strval%
  &when NODATA
  &do
    &type X,Y (%.grdx% , %.grdy%) INVALID CELL VALUE
  &end
  &when 56
  &do
    &type X,Y (%.grdx% , %.grdy%) VALID CELL VALUE
  &end
  &otherwise
    &type X,Y (%.grdx% , %.grdy%) INVALID CELL VALUE
  &end /* inside select
&end
&when takeupcell
&do
  &sv .upstrx = %.grdx%
  &sv .upstry = %.grdy%
&end
&when takedwncell
```

```
&do
  &sv .dwnstrx = %.grdx%
  &sv .dwnstry = %.grdy%
&end
&end /* outside select

/*&dv .grdx .grdy

/*======================
/* CLEANUP
/*======================
&call exit
&return
/*============= Map and Draw Routine ===============
&routine mapedraw
  &sv str_mape = [show mapextent]
  &sv .currwide = [extract 3 %str_mape%] - [extract 1 %str_mape%]
  &sv .currhigh = [extract 4 %str_mape%] - [extract 2 %str_mape%]
  clear
  [UNQUOTE %drw%]
  &flushpoints
&return

/*============= Exit Routine ================
/*
&routine exit
/* RESTORE ENVIRONMENTS
&return
/*============= Bailout Routine =============
/*
&routine bailout
/* RESET SEVERITY
&severity &error &ignore
/* BAIL OUT
&call exit
&return; &return &error Bailing out of pikstreampts.aml....
/* ENDDO
```

```
/* Movednstream.aml
/* This aml checks to see if .currhl grid value is a stream and a proximal hazard
/* zone boundary. If the current .currhl value is an h/l value, the flag .found1stpt
/* is set to true. Creates a grid called tempstart with an elevation value at the
/* current coordinates and a value of 99999 for all other cells. If the value of the
/* .currhl variable is a something other than a proximal hazard zone boundary
/* value, a select statement based on the current flow direction, updates the
/* coordinates to occupy the next down stream cell. Variables are updated for
/* current flow direction, current h/l-stream, and current stop values for the new
/* cell location.
/*
/*

&severity &error &routine bailout
/*=====================================
/* select statement, checks if current value
/* for .currhl is h/l value of 119. If it is
/* set .found1stpt to true to stop loop calling
/* this aml and creates tempstart grid having
/* current elevation at current cell location
/* and all other cell values of 99999
/*=====================================
&if NOT %.found1stpt% &then
&do
  &select %.currhl%
  &when 119
  &do
    &type
    &type FOUND PROXIMAL HAZARD ZONE BOUNDARY...
    &type
    &sv .found1stpt = .TRUE.
    &if [EXISTS tempstart -GRID] &then kill tempstart all
    &type
    &type CREATING GRID TO BEGIN MAKING CROSS SECTIONS...
    &type
    tempstart = con(isnull(selectpoint ~
              (%.elev_g%,%.nextx%,%.nexty%)),99999,%.elev_g%)
  &end
  &end /* select current hl and stream grid
&end /* if

/*=====================================
/* select statement updates "moves" x, y coordinates
/* based on flow direction. updates .count
/*=====================================
&if NOT %.found1stpt% &then
&do

&select %.currdir%
&when 4
```

```
&do
  &setvar .nexty = %.nexty% - %.cellwidth%
&end
&when 64
&do
  &setvar .nexty = %.nexty% + %.cellwidth%
&end
&when 1
&do
  &setvar .nextx = %.nextx% + %.cellwidth%
&end
&when 16
&do
  &setvar .nextx = %.nextx% - %.cellwidth%
&end
&when 2
&do
  &setvar .nexty = %.nexty% - %.cellwidth%
  &setvar .nextx = %.nextx% + %.cellwidth%
&end
&when 8
&do
  &setvar .nexty = %.nexty% - %.cellwidth%
  &setvar .nextx = %.nextx% - %.cellwidth%
&end
&when 32
&do
  &setvar .nexty = %.nexty% + %.cellwidth%
  &setvar .nextx = %.nextx% - %.cellwidth%
&end
&when 128
&do
  &setvar .nexty = %.nexty% + %.cellwidth%
  &setvar .nextx = %.nextx% + %.cellwidth%
&end
&end /* select when

&setvar .count = %.count% + 1
/*&end /* if

/*==========================================
/* update current flow direction, current
/* h/l-stream value, and current stop value
/* based on current x, y coordinates
/*==========================================
&setvar .currdir = [show cellvalue %.dir_g% %.nextx% %.nexty%]
&setvar .currhl = [show cellvalue %.hlstr_g% %.nextx% %.nexty%]
/*&setvar .currstop = [show cellvalue xxstop %.nextx% %.nexty%]

/*==========================================
```

```
/* show user count - number of cells traversed
/*=========================================
&type SEARCHING FOR PROXIMAL HAZARD BOUNDARY --- STREAM CELLS
TRAVERSED: %.count%

&end /* if

&call exit
&return
/*============= Exit Routine ================
/*
&routine exit
/* RESTORE ENVIRONMENTS
&return
/*============= Bailout Routine =============
/*
&routine bailout
/* RESET SEVERITY
&severity &error &ignore
/* BAIL OUT
&call exit
&return; &return &error Bailing out of Movednstream.aml....
/* ENDDO
```

```
/* Make_lahar.aml
/* This aml initializes variables and grids to create inundation zones for up to four
/* user-specified lahar volumes.  Calls setup_lahar.menu for user to select grids,
/* enter volumes, and enter a name for a drainage.  Checks grids, calculates
/* cross  section and planimetric areas foreach of the four  initial lahar volumes.
/* Gets the current X and Y coordinates ofthe starting cell, current flow direction,
/* creates four grids to track accumulation of planimetric area added by cells
/* occupied during cross section construction.  Opens a file, <drainage
/* name>.pts.  Enters loop that calls cellmove.aml until allstop flag is set to
/* true or moves outside grid.

/* Global Variables:
/*   .elev_g     - elevation grid
/*   .dir_g      - flow direction grid
/*   .drainname  - user entered name for current drainage
/*   .kmrs1 thru
/*   .kmrs4      - initial lahar volumes in cubic meters
/*   .celldiag   - length of cell diagonal
/*   .sectarea1 thru
/*   .sectarea4  - calculated cross section area
/*   .maparea1 thru
/*   .maparea4   - calculated planimetric area
/*   .plancount2 thru
/*   .plancount4 - planimetric area used to stop making
/*                 inundation areas
/*   .currdir    - flow direction at current coordinates
/*   .nextx      - x coordinate of cell location according to
/*                 flow direction
/*   .nexty      - y coordinate of cell location according to
/*                 flow direction
/*   .count      - counter
/*   .totdeficit - value of the total deficit
/*   .stopsect2 thru
/*   .stopsect4  - flags, when set to true stop calculation
/*                 of individual inundation zones
/*
/*
/*
&severity &error &routine bailout
/*====================================
/*&echo &on
&terminal 9999
display 1040
/*====================================
/* initialize variables
/*====================================
&sv .stop1 = .FALSE.
&sv .hl_g = nullgrid
&sv .dir_g = nullgrid
&sv .str_g = nullgrid
```

```
&sv .hlstr_g = nullgrid
&sv .elev_g = nullgrid
&sv .drainname = temp
&sv .nextx = 0
&sv .nexty = 0
&sv .found1stpt = .FALSE.
&sv .currhl = 0
&sv .upstrx = 0
&sv .upstry = 0
&sv .cellwidth = 0
&sv .celldiag = 0
&sv .currdir = 0
&sv .count = 0
&sv .totdeficit = 0
&do num = 1 &to 4 &by 1
  &sv .kmrs%num% = 0
&end

/*=====================================
/* call setup_lahar.menu
/*=====================================
&menu setup_lahar.menu &position &UL &screen 100 100 ~
  &stripe 'LAHARZ - Create Lahar Inundation Zones'

/*=====================================
/* if user quits menu, call exit
/*=====================================
&if %.stopl% &then
  &call exit
&else
  &do

/*=====================================
/* Sort the entered volumes to ensure that kmrs1 is the
/* largest volume down to kmrs4 being the smallest
/* volume
/*=====================================
&sv i = 3
&sv j = 4
&sv switched = .FALSE.
&do &until %i% = 0
  &if [value .kmrs%i%] < [value .kmrs%j%] &then
    &do
      &sv temp = [value .kmrs%i%]
      &sv .kmrs%i% = [value .kmrs%j%]
      &sv .kmrs%j% = %temp%
      &sv switched = .TRUE.
    &end
  &if NOT %switched% &then
    &do
```

```
    &sv i = %i% - 1
    &sv j = %j% - 1
  &end
  &else
  &do
    &sv i = 3
    &sv j = 4
    &sv switched = .FALSE.
  &end
&end /* do loop

/*======================================
/* Set .maxi variable to number of volumes entered > 0
/*======================================
&sv .maxi = 4
&do num = 4 &to 2 &by -1
&if [value .kmrs%num%] < 1 &then
  &sv .maxi = %num% - 1
&end

&do num = 1 &to %.maxi% &by 1
  &sv .sectarea%num% = 0
  &sv .maparea%num% = 0
  &sv .plancount%num% = 0
  &sv .stopsect%num% = .FALSE.
&end

/*======================================
/* check if running in arc, if so start GRID
/*======================================
&if [show program] ne ARC &then
  &return This program must run from ARC.
&else
  GRID

&setvar .allstop = .FALSE.
/*======================================
/* check to see if grids selected by user from
/* setup_lahar.menu exist
/*======================================
&if NOT [EXISTS %.elev_g% -GRID] &then /* elevation grid
  &return GRID %.elev_g% not found...
&if NOT [EXISTS %.dir_g% -GRID] &then /* direction grid
  &return GRID %.dir_g% not found...

/*======================================
/* make sure elevation and flow direction grids have
/* equal sided cells within and between grids
/*======================================
&sv i = 1
```

```
&do cellck &list %.elev_g% %.dir_g%
  &describe %cellck%
  &if %grd$dx% <> %grd$dy% &then &return Program assumes equal sided
cells
  &sv %i%cell = %grd$dx%
  &sv i = %i% + 1
&end
&if %1cell% <> %2cell% &then
   &return Cellsize for grids must be equal;
&sv .cellwidth = %grd$dx%
&delvar 1cell 2cell i cellck

/*====================================
/* calculate up to 4 cross section and 4 planimetric areas from
/* the 4 volumes input in setup_lahar.menu. store cross
/* section values in .sectarea1 - 4 store planimetric area
/* values in .maparea1 - 4
/*====================================
&do i = 1 &to %.maxi% &by 1
  /* calculation of maximum cross section area from volume
  &set vol%i% = [value .kmrs%i%]
  &set cnum%i% = [calc [value vol%i%] ** 0.666666666666666]
  &set .sectarea%i% = [round [calc [value cnum%i%] * 0.05]]
  &set .maparea%i% = [round [calc [value cnum%i%] * 200]]

  &delvar vol%i% cnum%i% c%i%num
&end /* do

/*====================================
/* calculate diagonal length of cell
/*====================================
&set .celldiag = [sqrt [calc %.cellwidth% ** 2 * 2]]

/*====================================
/* Set the cell size and the window
/*====================================
setcell %.elev_g%
setwindow %.elev_g%

/*====================================
/* for the window, creates a grid of cells, with values
/* of zero, that are copied at specific locations and stored
/* in planimetric inundation area grids
/*====================================
&if [EXISTS xxzerogrid -GRID] &then kill xxzerogrid all
&type
&type INITITALIZING FOOTPRINT GRID...
&type
xxzerogrid = 0
```

```
/*====================================
/* makes the single elevation value in tempstart an
/* integer value and stores in tempstart2. finds the
/* location of minimum z value of choices in tempstart2
/* to begin calculating cross sections. Also puts zero
/* in field of NODATA at that location in the footprint
/* grid. Gets the X,Y coordinates in DOCELL
/*====================================
tempstart2 = int(tempstart)
&describe tempstart2
&type
&type SEARCHING ELEVATION GRID FOR X, Y COORDINATES OF
STARTING CELL...
&type

DOCELL
if (tempstart2 == %grd$zmin%)
begin
 myx = scalar($$XMAP)
 myy = scalar($$YMAP)
end
END

&sv .nextx [show myx]
&sv .nexty [show myy]

/*====================================
/* store current flow direction
/*====================================
&sv .currdir = [show cellvalue %.dir_g% %.nextx% %.nexty%]

/*====================================
/* create up to 4 grids with all cells having value of 1 except
/* for single 0 value of starting point
/*====================================
&if [EXISTS xxone1 -GRID] &then kill xxone1 all
&type
&type CREATING FOOTPRINT GRIDS FOR EACH INPUT VOLUME...
&type

xxone1 = con(tempstart < 99999,0,1)

&if %.maxi% > 1 &then
&do
  &do i = 2 &to %.maxi% &by 1
    &if [EXISTS xxone%i% -GRID] &then kill xxone%i% all
    xxone%i% = xxone1
  &end
&end
```

```
/*==================================
/* open system file drainname.pts for write where
/* drainname is name entered in setup_lahar.menu
/*==================================

 &sv .strunit = [open %.drainname%.pts stropstat -write]
 &if %stropstat% ne 0 &then
   &return Unable to open file %.drainname%.pts, error code: %stropstat%

/*==================================
/* write header to drainname.pts file
/*==================================
&sv writestat = [write %.strunit% 'DRAINAGE NAME ENTERED: ~
',%.drainname%]
&sv writestat = [write %.strunit% 'VOLUMES ENTERED - SORTED LARGEST
TO SMALLEST : ']
&sv writestat = [write %.strunit% %.kmrs1%' : '%.kmrs2%' : '%.kmrs3%' ~
'%.kmrs4%]
&sv writestat = [write %.strunit% '-----------------------------------------------------']

&select %.maxi%
&when 1
&do
&sv writestat = [write %.strunit% 'CROSS SECTION AREA 1 ']
&sv writestat = [write %.strunit% %.sectarea1%]
&sv writestat = [write %.strunit% 'PLANIMETRIC MAP AREA 1 ']
&sv writestat = [write %.strunit% %.maparea1%]
&sv writestat = [write %.strunit%  '=====================']
&sv writestat = [write %.strunit% 'DECREASING PLANIMETRIC  AREA 1']
&sv writestat = [write %.strunit%  '=====================']
&end
&when 2
&do
&sv writestat = [write %.strunit% 'CROSS SECTION AREAS 1 2 ']
&sv writestat = [write %.strunit% %.sectarea1%,%.sectarea2%]
&sv writestat = [write %.strunit% 'PLANIMETRIC MAP AREAS 1 2 ']
&sv writestat = [write %.strunit% %.maparea1%,%.maparea2%]
&sv writestat = [write %.strunit%  '======================']
&sv writestat = [write %.strunit% 'DECREASING PLANIMETRIC AREAS 1 2 ']
&sv writestat = [write %.strunit%  '======================']
&end
&when 3
&do

&sv writestat = [write %.strunit% 'CROSS SECTION AREAS 1 2 3 ']
&sv writestat = [write %.strunit% %.sectarea1%,%.sectarea2%,%.sectarea3%]
&sv writestat = [write %.strunit% 'PLANIMETRIC MAP AREAS 1 2 3 ']
&sv writestat = [write %.strunit% %.maparea1%,%.maparea2%,%.maparea3%]
&sv writestat = [write %.strunit%  '===============================']
&sv writestat = [write %.strunit% 'DECREASING PLANIMETRIC AREAS 1 2 3 ']
```

```
&sv writestat = [write %.strunit% '=================================']
&end
&when 4
&do
&sv writestat = [write %.strunit% 'CROSS SECTION AREAS 1 2 3 4']
&sv writestat = [write %.strunit% ~
%.sectarea1%,%.sectarea2%,%.sectarea3%,%.sectarea4%]
&sv writestat = [write %.strunit% 'PLANIMETRIC MAP AREAS 1 2 3 4']
&sv writestat = [write %.strunit% ~
%.maparea1%,%.maparea2%,%.maparea3%,%.maparea4%]
&sv writestat = [write %.strunit% ~
'=====================================================']
&sv writestat = [write %.strunit% 'DECREASING PLANIMETRIC AREAS 1 2 3 4']
&sv writestat = [write %.strunit% ~
'=====================================================']
&end
&end /* select


/*======================================
/* Loop calls cellmove until allstop set to true or at
/* edge of grid
/*======================================
&do &until %.nextx% < [show $$WX0] or %.nextx% > [show $$WX1] ~
  or %.nexty% < [show $$WY0] or %.nexty% > [show $$WY1] or %.allstop%

  &r cellmove

&end /* UNTIL
/*======================================


/*======================================
/* close system file drainname.pts
/*======================================

 &sv strclstat = [close %.strunit%]
 &if %strclstat% = 0 &then
   &dv %.strunit%
 &else
   &type Error closing unit %.strunit% Error code: %strclstat%

/*======================================
/* create grids having name of drainage.  Values are
/* nodata equal 0 and data equal 1
/*======================================
&do i = 1 &to 4 &by 1
&if [EXISTS xxone%i% -GRID] &then
    %.drainname%%i% = con(xxone%i% == 0,1,0)
&end
```

```
&type
&type  CLEANING UP ...
&type

&call cleanup

&type
&type CREATION OF LAHAR INUNDATION ZONES COMPLETE...
&type
&end

/*&echo &off
&call exit
&return
/*============== Cleanup Routine ================
/*
&routine cleanup
/* kill temporary grids used in
/* creation of inundation zones

&if [EXISTS tempstart -GRID] &then
  kill tempstart all
&if [EXISTS tempstart2 -GRID] &then
  kill tempstart2 all
&if [EXISTS xxstop -GRID] &then
  kill xxstop all
&if [EXISTS xxzerogrid -GRID] &then
  kill xxzerogrid all
&if [EXISTS xxallones_g -GRID] &then
  kill xxallones_g all
&if [EXISTS xxdrwstr -GRID] &then
 kill xxdrwstr all
&do num = 1 &to 4 &by 1
&if [EXISTS xxarea%num%_g -GRID] &then
  kill xxarea%num%_g all
&if [EXISTS xxone%num% -GRID] &then
  kill xxone%num% all
&end
&return

/*============== Exit Routine ================
/*
&routine exit
&if [show program] = GRID &then
  &do
    QUIT
    &dv .elev_g .dir_g .hl_g .str_g .hlstr_g .drainname .celldiag ~
       .nextx .nexty .found1stpt .currhl .upstrx .upstry .count ~
       .cellwidth .currdir .totdeficit
```

```
&do num = 1 &to %.maxi% &by 1
  &dv .kmrs%num%
  &dv .sectarea%num%
  &dv .maparea%num%
  &dv .plancount%num%
  &dv .stopsect%num%
&end

  &end
&return
/*============= Bailout Routine =============
/*
&routine bailout
/* RESET SEVERITY
&severity &error &ignore
/* BAIL OUT
&call exit
&return; &return &error Bailing out of makelahar.aml....
/* ENDDO
```

7 setup_lahar.menu

```
/*
/* User selects an elevation grid and a flow direction grid.  The user enters a
/* name for selected stream and up to 4 initial lahar volumes in cubic meters.
/*
/* Global Variables set:
/*    .elev_g    - name of elevation grid
/*    .dir_g     - name of flow direction grid
/*    .drainname - name of current drainage
/*    .kmrs1 thru
/*    .kmrs4     - values of the 4 initial lahar volumes
/*
/*
/*
 SELECT DEM (ELEVATION GRID):   ^SELECT FLOW DIRECTION GRID:
 %datalist1                 %datalist3






      Enter Drainage name: %input7

    Enter 1st volume in cubic meters:     %input1

    Enter 2nd volume in cubic meters:     %input4

    Enter 3rd volume in cubic meters:     %input5

    Enter 4th volume in cubic meters:     %input6

  %button1                          %button2
%datalist1 INPUT DATALIST1 28 TYPEIN NO SCROLL YES ROWS 4 ~
 KEEP ~
 RETURN '&sv .elev_g = %datalist1%;' ~
 GRID ~
 * -SORT
%datalist3 INPUT DATALIST3 27 TYPEIN NO SCROLL YES ROWS 4 ~
 KEEP ~
 RETURN '&sv .dir_g = %datalist3%;' ~
 GRID ~
 * -SORT
%input7 INPUT input7 16 TYPEIN YES SCROLL NO SIZE 20 ~
 KEEP ~
 RETURN '&sv .drainname = %input7%;' ~
 CHARACTER
%input1 INPUT INPUT1 12 TYPEIN YES SCROLL NO SIZE 11 ~
 KEEP ~
 RETURN '&sv .kmrs1 = %input1%;' ~
 REAL
```

```
%input4 INPUT INPUT4 12 TYPEIN YES SCROLL NO SIZE 11 ~
 KEEP ~
 RETURN '&sv .kmrs2 = %input4%;' ~
 REAL
%input5 INPUT INPUT5 12 TYPEIN YES SCROLL NO SIZE 11 ~
 KEEP ~
 RETURN '&sv .kmrs3 = %input5%;' ~
 REAL
%input6 INPUT INPUT6 12 TYPEIN YES SCROLL NO SIZE 11 ~
 KEEP ~
 RETURN '&sv .kmrs4 = %input6%;' ~
 REAL
%button1 BUTTON KEEP 'QUIT' &sv .stopl = .TRUE.; &return QUITTING LAHAR
SETUP ...
%button2 BUTTON KEEP 'Continue' ~
 &delvar datalist1 datalist3 input1 input2 input4; &return;
```

```
/* Cellmove.aml
/*
/* Cellmove.aml opens up to four files to store X and Y coordinates of cells
/* occupied during cross section construction. Calls section.aml up to four times
/* per stream cell to construct cross sections for each user-specified volume.
/* For any cell there are four directions possible to construct sections;
/* north-south, east-west, northwest-southeast, and southwest-northeast. The
/* software checks the direction of the next downstream cell, eliminates that
/* direction from the four possible choices, and constructs cross sections in the
/* other three directions. When a drainage lies along a northwest-southeast or
/* southwest-northeast direction, adjacent stream cells meet at their corners.
/* In plan view, cell locations from these cross sections often form a
/* "checkerboard" pattern. In these cases, this aml moves to a cell location
/* adjacent to the two stream cells that meet at their corners and constructs a
/* fourth cross section perpendicular to the flow direction. Then cellmove.aml
/* closes the (up to) four open files. Cellmove.aml adds cell locations from the
/* four files to cell locations encountered from previous cross sections.
/* Cellmove.aml calls curs_area.aml to calculate total planimetric inundation
/* area. Finally, cellmove.aml "moves" by adding or subtracting cell width, or
/* diagonal length in the X and(or) Y direction according to the current flow
/* direction. Stores the new flow direction, checks if cell at current cell is the
/* stopping cell, and updates the count.
/*
/*
/*
/* Use bailout routine if an error
/*
&severity &error &routine bailout
/*==================================

/*==================================
/* open xxx.pts file for each of the four input
/* volumes. Write cell locations for all sections
/* for a current cell to the file
/*==================================
&do i = 1 &to %.maxi% &by 1
  &if NOT [value .stopsect%i%] &then
  &do
    &sv .areaunit%i% = [open xxx%i%.pts area%i%opstat -write]
    &if [value area%i%opstat] ne 0 &then
      &return Unable to open file xxx%i%.pts, error code: [value area%i%opstat]
  &end
&end

/*==================================
/* Call section.aml to create cross section
/* perpendicular to flow direction.
/*==================================
&type
&type CALCULATING 1ST CROSS SECTION...
```

```
&type
&r section.aml

/*================================
/* store current flow direction, set to construct
/* cross section in other two possible directions
/*================================
&sv .savdir = %.currdir%
&select %.savdir%
&when 8,32,128,2
&do
  &sv .currdir = 16
  &type
  &type CALCULATING 2ND CROSS SECTION - DIAGONAL...
  &type

  &r section.aml
  &sv .currdir = 4
  &type
  &type CALCULATING 3RD CROSS SECTION - DIAGONAL...
  &type

  &r section.aml
&end
&when 1,4,16,64
&do
  &sv .currdir = 32
  &type
  &type CALCULATING 2ND CROSS SECTION - ORDINAL...
  &type

  &r section.aml
  &sv .currdir = 128
  &type
  &type CALCULATING 3RD CROSS SECTION - ORDINAL...
  &type

  &r section.aml
&end
&end /* select

&sv .currdir = %.savdir% /* restore original flow direction
&delvar .savdir

/*================================
/* save old X,Y then move to an adjacent cell
/* and run a fourth section to fill in diagonal
/* "checkerboard" pattern.
/*================================
&if %.currdir% = 2 or %.currdir% = 8 or %.currdir% = 32 or %.currdir% = 128
```

```
&then
&do
  &sv .savx = %.nextx%
  &sv .savy = %.nexty%
  &type
  &type CALCULATING ADDITIONAL 4TH CROSS SECTION...
  &type

  &select %.currdir%
  &when 8 ; &do
    &sv .nextx = %.nextx% - %.cellwidth%
    &r section.aml
  &end
  &when 32 ; &do
    &sv .nexty = %.nexty% + %.cellwidth%
    &r section.aml
  &end
  &when 128 ; &do
    &sv .nextx = %.nextx% + %.cellwidth%
    &r section.aml
  &end
  &when 2 ; &do
    &sv .nexty = %.nexty% - %.cellwidth%
    &r section.aml
  &end
  &end /* select

  /*================================
  /* restore old X,Y values
  /*================================
  &sv .nextx = %.savx%
  &sv .nexty = %.savy%
  &delvar .savx .savy

&end /* check for 4th cross section

/*================================
/* close xxx.pts files, add to grids, calculate their
/* total area, rename grids, delete xxx.pts files
/*================================
&do i = 1 &to %.maxi% &by 1
  &if NOT [value .stopsect%i%] &then
  &do
  /*================================
  /* close four xxx.pts files that each contain one
  /* set of point locations for one section.
  /*================================
    &sv area%i%clstat = [close [value .areaunit%i%]]
    &if [value area%i%clstat] = 0 &then
      &dv .areaunit%i%
```

```
    &else
      &type Error closing unit [value .areaunit%i%] Error code: [value ~
area%i%clstat]

  /*================================
  /* delete grids if they exist.  adds cell locations stored
  /* in xxx1-4.pts file to cell locations stored from previous
  /* cross sections (xxone1-xxone4 grids) storing the
  /* results in xxarea1_g - xxarea4_g grids
  /*================================

    &if [EXISTS xxarea%i%_g -GRID] &then kill xxarea%i%_g all
 xxarea%i%_g = con(isnull(selectpoint(xxzerogrid,xxx%i%.pts,inside)) == ~
0,0,xxone%i%)

  /*================================
  /* call curs_area.aml to calculate planimetric area
  /*================================
    &r curs_area xxarea%i%_g %i%
  /*================================
  /* kill xxone1 - xxone4 grids then rename temporary
  /* xxarea1-4_g grids to xxone1-4 grids.
  /* cleanup by deleting grids and xxx.pts file.
  /*================================
    &if [EXISTS xxone%i% -GRID] &then kill xxone%i% all
    &if [EXISTS xxarea%i%_g -GRID] &then rename xxarea%i%_g xxone%i%
    &setvar delstat = [DELETE xxx%i%.pts]
  &end
&end

&type
&type CROSS SECTION CALCULATIONS FOR CELL COMPLETE...
PROCESSING...
&type

&select %.maxi%
&when 1
  &do
    &sv writestat = [write %.strunit% %.plancount1%]
  &end
&when 2
  &do
    &sv writestat = [write %.strunit% %.plancount1%,%.plancount2%]
  &end
&when 3
  &do
    &sv writestat = [write %.strunit%
%.plancount1%,%.plancount2%,%.plancount3%]
  &end
&when 4
```

```
  &do
    &sv writestat = [write %.strunit%
%.plancount1%,%.plancount2%,%.plancount3%,%.plancount4%]
  &end
&end /* select

/*==================================
/* if total planimetric area is greater than predicted
/* planimetric area, set flag to stop cross section
/* construction for the current grid
/*==================================
&do i = 1 &to %.maxi% &by 1
  &if [value .plancount%i%] < 0 &then
    &sv .stopsect%i% = .TRUE.
&end

/*==================================
/* reset maxi variable to reflect planimetric areas
/* that have more area to calculate
/*==================================
&do i = 4 &to 2 &by -1
  &if [value .stopsect%i%] &then
    &sv .maxi = %i% - 1
&end
&if %.stopsect1% &then
  &sv .allstop = .TRUE.

/*==================================
/* move to the next downstream cell depending
/* on current flow direction stored in .currdir
/* move by updating X,Y coordinates to next
/* downstream cell location.
/*==================================
&type
&type MOVING TO NEXT CELL...
&type

&select %.currdir%
&when 4
&do
  &setvar .nexty = %.nexty% - %.cellwidth%
&end
&when 64
&do
  &setvar .nexty = %.nexty% + %.cellwidth%
&end
&when 1
&do
  &setvar .nextx = %.nextx% + %.cellwidth%
&end
```

```
&when 16
&do
  &setvar .nextx = %.nextx% - %.cellwidth%
&end
&when 2
&do
  &setvar .nexty = %.nexty% - %.cellwidth%
  &setvar .nextx = %.nextx% + %.cellwidth%
&end
&when 8
&do
  &setvar .nexty = %.nexty% - %.cellwidth%
  &setvar .nextx = %.nextx% - %.cellwidth%
&end
&when 32
&do
  &setvar .nexty = %.nexty% + %.cellwidth%
  &setvar .nextx = %.nextx% - %.cellwidth%
&end
&when 128
&do
  &setvar .nexty = %.nexty% + %.cellwidth%
  &setvar .nextx = %.nextx% + %.cellwidth%
&end
&end /* select when

/*==================================
/* update stream cell count, current flow
/* direction, and if current location is same
/* as location of user-picked stopping cell
/*==================================
&setvar .count = %.count% + 1
&setvar .currdir = [show cellvalue %.dir_g% %.nextx% %.nexty%]
&setvar .currstop = [show cellvalue xxstop %.nextx% %.nexty%]

&select %.currstop%
  &when 1 ; &do
    &sv .allstop = .TRUE.
  &end
&end /* select currstop

/*==================================
/* show count on screen
/*==================================
&type              ------------------------------
&type              NUMBER OF STREAM CELLS TRAVERSED:
%.count%
&type              ------------------------------
```

```
&call exit
&return
/*============== Exit Routine ================
/*
&routine exit
/* RESTORE ENVIRONMENTS
&return
/*============== Bailout Routine ==============
/*
&routine bailout
/* RESET SEVERITY
&severity &error &ignore
/* BAIL OUT
&call exit
&return; &return &error Bailing out of cellmove.aml....
/* ENDDO
```

```
/* Section.aml
/*
/*
/* Section.aml initializes several variables, sets cell dimension variables, gets X
/* and Y coordinates of right (current stream cell) and adjacent left (facing
/* downstream) cells, gets elevation of the left and right cells, and writes the cell
/* locations to system files opened in cellmove.aml. Stores right cell elevation as
/* "fill level" variable. In main loop, section.aml determines if left and right
/* elevations are equal to or less than the stored fill level. Section.aml then
/* compares right and left elevations checking to see if they are equal or not. In
/* the case of left and right cell elevations greater fill level but unequal,
/* section.aml calculates a tier of area. Fill level is subtracted from either right or
/* left elevations whichever is less. The result is multiplied by the cell count
/* (number of cells in current cross section) that is in turn multiplied by the cell
/* width or diagonal length. The area thus calculated is subtracted from the total
/* predicted cross section area (A) and the result stored in variables maxarea1
/* through maxarea4. The fill level is updated to the new level (top of the tier,
/* lower elevation of left and right cells), writes the cell location of the lower
/* elevation cell to a system file and cell locations moved to adjacent outboard
/* cell (left, right, or both) as needed. If section.aml encounters the edge of the
/* grid, it stops processing for the cross section. The cell count is updated and if
/* the constructed cross section area is greater than the predicted cross section
/* area (maxarea1 - 4) a flag is set to stop cross section construction.
/* Routine writexxx determines whether a cell location is written to a file, and, if
/* so, whether to write the left or right cell to the file
/* Routine scalar sets variables to 0, 1, or -1 to control whether the X an Y
/* coordinates change sign or stay the same when moving to outboard cells,
/* stores the current (left or right) cell X and Y coordinate to a buffer, and moves
/* to next outboard left or right cell.
/* Routine window checks if next cell is outside processing window. If it is,
/* section.aml restores the current cell location to that stored in the buffer and
/* stops cross section processing. If not, it gets the next outboard cells'
/* elevation.
/*
/*
/*
/* Use bailout routine if an error
/*
&severity &error &routine bailout
/*==================================
/*          Variables
/*==================================
&sv count = 0                 /* Main loop counter
&sv winXmin = [show $$WX0]    /* load window coords
&sv winYmin = [show $$WY0]
&sv winXmax = [show $$WX1]
```

```
&sv winYmax = [show $$WY1]
&do num = 1 &to %.maxi% &by 1
  &sv maxarea%num% = [value .sectarea%num%]   /* store predicted cross
section areas
  &sv stoparea%num% = .FALSE.                 /* flags to stop single
sections
&end

/*=====================================
/* set cell dimension for area calculations
/* cell width is to move around grid
/*=====================================
&select %.currdir%
&when 8,128,2,32 ; &do
  &sv celldimen = %.celldiag%
&end
&when 1,16,4,64 ; &do
  &sv celldimen = %.cellwidth%
&end
&end /* select

/*=====================================
/* set X,Y coords of stream cell as right cell (facing
/* downstream)
/*=====================================
&sv cellrightx = %.nextx%
&sv cellrighty = %.nexty%

/*=====================================
/* set X,Y coords of left  cell (facing
/* downstream
/*=====================================
&select %.currdir%
&when 1 ; &do
  &sv cellleftx = %.nextx%
  &sv celllefty = %.nexty% + %.cellwidth%
&end
&when 16 ; &do
  &sv cellleftx = %.nextx%
  &sv celllefty = %.nexty% - %.cellwidth%
&end
&when 4 ; &do
  &sv cellleftx = %.nextx% + %.cellwidth%
  &sv celllefty = %.nexty%
&end
&when 64 ; &do
```

```
  &sv cellleftx = %.nextx% - %.cellwidth%
  &sv celllefty = %.nexty%
&end
&when 8 ; &do
  &sv cellleftx = %.nextx% + %.cellwidth%
  &sv celllefty = %.nexty% - %.cellwidth%
&end
&when 128 ; &do
  &sv cellleftx = %.nextx% - %.cellwidth%
  &sv celllefty = %.nexty% + %.cellwidth%
&end
&when 2 ; &do
  &sv cellleftx = %.nextx% + %.cellwidth%
  &sv celllefty = %.nexty% + %.cellwidth%
&end
&when 32 ; &do
  &sv cellleftx = %.nextx% - %.cellwidth%
  &sv celllefty = %.nexty% - %.cellwidth%
&end
&end /* select

/*====================================
/* get elevations of left and right cells
/*====================================
&sv cellrightelev =  [show cellvalue %.elev_g% %cellrightx% %cellrighty%]
&sv cellleftelev = [show cellvalue %.elev_g% %cellleftx% %celllefty%]

&sv filllevel = %cellrightelev%
&sv cellcount = 0
&sv casefound = .FALSE.

/*====================================
/*              main section loop
/*====================================
&do &until %maxarea1% < 0 or %count% > 1000

/*====================================
/* compare elevations equal to fill level
/*====================================
&if %cellleftelev% = %filllevel% or %cellrightelev% = %filllevel% &then
&do
  &if %cellleftelev% = %filllevel% &then
    &do
      &sv side = left
      &call writexxx
      &sv .cellwidth = [calc %.cellwidth% * -1] /* left is opposite
```

```
    &call scalar
    &sv .cellwidth = [calc %.cellwidth% * -1] /* change back to right
  &end
 &else     /* cellrightelev = filllevel
  &do
    &sv side = right
    &call writexxx
    &call scalar
  &end
&sv cellcount = %cellcount% + 1
&sv casefound = .TRUE.
&end

/*=====================================
/* compare elevations less than fill level
/*=====================================
&if %cellrightelev% < %filllevel% or %cellleftelev% < %filllevel% and ~
 NOT %casefound% &then
&do
 &if %cellrightelev% < %filllevel% &then
 &do
  &do i = 1 &to %.maxi% &by 1
   &if NOT [value stoparea%i%] &then
     &sv maxarea%i% = [value maxarea%i%] ~
      - ( %filllevel% - %cellrightelev% ) * %celldimen%
  &end
  &sv side = right
  &call writexxx
  &call scalar
 &end
&else    /* cellleftelev < filllevel
 &do
  &do i = 1 &to %.maxi% &by 1
    &if NOT [value stoparea%i%] &then
      &sv maxarea%i% = [value maxarea%i%] ~
       - ( %filllevel% - %cellleftelev% ) * %celldimen%
  &end
  &sv side = left
  &call writexxx
  &sv .cellwidth = [calc %.cellwidth% * -1] /* left is opposite
  &call scalar
  &sv .cellwidth = [calc %.cellwidth% * -1] /* change back to right
 &end
&sv cellcount = %cellcount% + 1
&sv casefound = .TRUE.
&end
```

```
/*========================================
/* compare equal elevations
/*========================================
&if %cellrightelev% = %cellleftelev%  AND NOT %casefound% &then
&do
  &do i = 1 &to %.maxi% &by 1
    &if NOT [value stoparea%i%] &then
  &sv maxarea%i% = [value maxarea%i%] - ( ( %cellrightelev% - %filllevel% ) * ~
      ( %celldimen% * %cellcount% ) )
  &end
  &sv filllevel = %cellrightelev%
  /*========================================
  /* move left and right
  /*========================================
  &sv side = left
  &call writexxx
  &sv .cellwidth = [calc %.cellwidth% * -1] /* left is opposite
  &call scalar
  &sv .cellwidth = [calc %.cellwidth% * -1] /* change back to right

  &sv side = right
  &call writexxx
  &call scalar
  &sv cellcount = %cellcount% + 2
  &sv casefound = .TRUE.
&end

/*========================================
/* compare unequal elevations
/*========================================
&if %cellrightelev% > %cellleftelev% or %cellrightelev% < %cellleftelev% and ~
  NOT %casefound% &then
&do
  &if %cellrightelev% > %cellleftelev%  &then
  &do
    &do i = 1 &to %.maxi% &by 1
      &if NOT [value stoparea%i%] &then
   &sv maxarea%i% = [value maxarea%i%] - ( ( %cellleftelev% - %filllevel% ) * ~
        ( %celldimen% * %cellcount% ) )
    &end
    &sv filllevel = %cellleftelev%
    &sv side = left
    &call writexxx
    &sv .cellwidth = [calc %.cellwidth% * -1] /* left is opposite
    &call scalar
```

```
    &sv .cellwidth = [calc %.cellwidth% * -1] /* change back to right
    &sv cellcount = %cellcount% + 1
  &end
  &else  /* cellleftelev > cellrightelev
  &do
    &do i = 1 &to %.maxi% &by 1
      &if NOT [value stoparea%i%] &then
  &sv maxarea%i% = [value maxarea%i%] - ( ( %cellrightelev% - %filllevel% ) * ~
        ( %celldimen% * %cellcount% ) )
    &end
    &sv filllevel = %cellrightelev%
    &sv side = right
    &call writexxx
    &call scalar
    &sv cellcount = %cellcount% + 1
  &end
&end

/*=====================================
/* hit an edge
/*=====================================
&if %cellleftelev% = 99999 or %cellrightelev% = 99999 &then
  &do
    &sv maxarea1 = -99999
    &sv maxarea2 = -99999
    &sv maxarea3 = -99999
    &sv maxarea4 = -99999
  &end

/*=====================================
/* update count of times through the MAIN
/* LOOP stops at 1000
/*=====================================
&sv count = %count% + 1

/*=====================================
/* check each of the section areas to see if negative
/* if so, set switch stoparea to TRUE
/*=====================================
&do i = 1 &to %.maxi% &by 1
  &if [value maxarea%i%] < 0 &then
    &sv stoparea%i% = .TRUE.
&end


/*=====================================
```

```
/* reset casefound
/*=========================================
&sv casefound = .FALSE.

&end /* MAIN LOOP

&sv .totdeficit = %.totdeficit% + %maxarea1%

&call exit
&return

/*============== Writexxx Routine ==============
&routine writexxx
&do i = 1 &to %.maxi% &by 1
    &if [value maxarea%i%] > 0 &then
    &do
      &select %side%
      &when left
        &do
          &sv area%i%stat = [write [value .areaunit%i%] %cellleftx%,%celllefty%]
        &end
      &when right
        &do
      &sv area%i%stat = [write [value .areaunit%i%] %cellrightx%,%cellrighty%]
        &end
     &end /* select
    &end
&end  /* do i
&return

/*============== Scalar Routine ==============
&routine scalar

&select %.currdir%
/* cardinal directions
&when 1 ; &do
  &sv xoper = 0
  &sv yoper = -1
&end
&when 16 ; &do
  &sv xoper = 0
  &sv yoper = 1
&end
&when 4 ; &do
  &sv xoper = -1
  &sv yoper = 0
```

```
&end
&when 64 ; &do
  &sv xoper = 1
  &sv yoper = 0
&end

/* diagonal directions
&when 8 ; &do
  &sv xoper = -1
  &sv yoper = 1
&end
&when 128 ; &do
  &sv xoper = 1
  &sv yoper = -1
&end
&when 2 ; &do
  &sv xoper = -1
  &sv yoper = -1
&end
&when 32 ; &do
  &sv xoper = 1
  &sv yoper = 1
&end
&end /* select

/*=======================================
/* store current location in buffer
/*=======================================
&sv cellbuffx = [value cell%side%x]
&sv cellbuffy = [value cell%side%y]


/*=======================================
/* calc next right or left location
/*=======================================
&sv cell%side%x = [value cell%side%x] + [calc %.cellwidth% * %xoper%]
&sv cell%side%y = [value cell%side%y] + [calc %.cellwidth% * %yoper%]

&call window
&return

/* =============== Routine Window ==============
&routine window
/*=======================================
/* check if outside window
/*=======================================
```

```
&if [value cell%side%x] < %winXmin% or [value cell%side%x] > %winXmax% ~
 or [value cell%side%y] < %winYmin% or [value cell%side%y] > %winYmax%
&then
&do
  &sv cell%side%x = %cellbuffx%
  &sv cell%side%y = %cellbuffy%
  &sv cell%side%elev = 99999
&end
&else     /* get next elevation
&do
  &sv cell%side%elev = [show cellvalue %.elev_g% ~
    [value cell%side%x] [value cell%side%y] ]
&end
&return

/*============= Exit Routine ================
/*
&routine exit
/* RESTORE ENVIRONMENTS
&return
/*============= Bailout Routine =============
/*
&routine bailout
/* RESET SEVERITY
&severity &error &ignore
/* BAIL OUT
&call exit
&return; &return &error Bailing out of section.aml....
/* ENDDO
```

```
/* Curs_area.aml
/*
/* Curs_area.aml is passed the name of a grid its number as parameters.
/* The grid contains 0's and 1's, where the 0's mark the cumulative cross
/* section cell locations. Curs_area.aml opens a cursor for the grid .vat
/* table until it finds the record that stores the number of 0's in the grid, and
/* stores that number in a variable. Curs_area.aml calculates the total area
/* from cells containing a 0 (cells that have been used during cross section
/* construction).
/*
/* Created by Steve Schilling
/*
&severity &error &routine bailout
/*==================================
&args xxx_grid  num  /* name and number of planimetric area grid passed from
cellmove.aml

/*==================================
/* Use cursor on xxx_grid.vat to find number of
/* cells containing zeros for calculation of total
/* planimetric area thus far
/*==================================
&sv numcells = 0
CURSOR totcells DECLARE %xxx_grid%.vat INFO RO
CURSOR totcells OPEN

&do &while %:totcells.AML$NEXT% and %:totcells.value% ne 0
  CURSOR totcells NEXT
&end /* until

&if %:totcells.value% = 0 &then
  &setvar numcells = %:totcells.count%
&else
  &Type Error could not find a value of 0

CURSOR totcells CLOSE
CURSOR totcells REMOVE

/*==================================
/* calculate total planimetric area for current
/* grid, subtract area from predicted area (calulated
/* from input volume) and store in variable .plancount
/*==================================

&setvar totarea = %numcells% * %.cellwidth% * %.cellwidth%
&setvar .plancount%num% = [value .maparea%num%] - %totarea%
```

```
&echo &off
&call exit
&return
/*============= Exit Routine ================
/*
&routine exit
/* RESTORE ENVIRONMENTS
&return
/*============= Bailout Routine =============
/*
&routine bailout
/* RESET SEVERITY
&severity &error &ignore
/* BAIL OUT
&call exit
&return; &return &error Bailing out of Curs_area.aml....
/* ENDDO
```

21