WORKING PAPER 474

MODELLING SOME INFLUENCES OF SOIL EROSION, LANDSLIDES AND VALLEY GRADIENT ON DRAINAGE DENSITY

M.J. KIRKBY

School of Geography
University of Leeds
Leeds LS2 9JT

October 1986

Modelling some influences of soil erosion, landslides and valley gradient on drainage density

Mike Kirkby: School of Geography, Leeds University, UK

Abstract

Landscape evolution has been simulated by a two dimensional model over a rhomboidal 32x32 grid. The overland flow contribution at any point is calculated in terms of the locally derived saturation deficit, which is summed along drainage paths to give total overland flow discharges. Sediment removal is modelled as being transport-limited; with a creep/splash term proportional to gradient added to a wash term which is proportional to overland flow power (i.e. flow x gradient) squared. The simulation results and the sediment equations have been analyzed in terms of the stability criterion for hollow growth. For the sediment transport law used, semi-arid climates with insignificant sub-surface flow show increasing drainage density with increasing valley gradients; whereas humid climates with substantial sub-surface flow show weak decreases in drainage density with gradient. A comparison is made with the stability criteria for hollow enlargement by landslides, for which drainage density always increases with valley gradient.

Introduction

This paper pursues the consequences of analysing whether hollows are unstable in the context of gully formation by either wash processes or landslides. Each of these groups of processes is asumed to act in combination with creep or splash processes which are dominant near divides. Instability is examined in the sense of the Smith and Bretherton (1972) criterion that a proto-hollow, considered as a small perturbation in the surface, will grow with an initially positive feed-back, into a macroscopic feature which would be identified on the ground as a valley head. The criterion for unstable hollow growth in this sense is that where flows of water and sediment converge, the combined water flow is able to carry away more than the combined sediment flow and so have excess capacity which is able to enlarge the hollow. In other words, instability occurs if sediment transport in the neighbourhood of the proto-hollow increases more than linearly with catchment area. This qualitiative argument may be formalized to give the mathematical condition that small perturbations will grow unstably if and only if:

$$a\ \partial S/\partial a > S \quad \text{.................................(1)}$$

where S is the sediment transport across unit contour length, and a is the area draining to the unit contour length. The differentiation should be performed holding gradient constant. This criterion was derived by Smith and Bretherton

(1972). They also showed that the same criterion is a necessary and sufficient condition for slope forms to be concave up in profile under conditions of equilibrium with a constant rate of uplift.

The stability criterion is applied below to two cases: first to potential hollow enlargement by soil erosion due to overland flow; and second to hollow enlargement by landslides along the valley axis. In each case, hollows tend to be infilled by the splash or creep processes. Possible conditions for hollow enlargement are obtained by analysis of simplified process rate equations. In the case of wash, the analytical forecasts are compared with a two dimensional simulation model which illustrates both the course of slope evolution and the strength of the tendency to hollow enlargement.

Two dimensional simulation model

The slope evolution model used here is a development of the one described in Kirkby (1986). Flow, sediment transport and erosion are simulated over a rhomboidal grid, in this case of 32x32 cells. The upper edge of the grid is treated as a divide, and the lower edge as a base level. The two sides are considered to roll onto one another in order to minimise edge effects. Flow of water and sediment from each point in the grid is assigned to the lowest of the six neighbouring cells, forcing the flow into a series of dendritic networks. Sediment removal is assumed to be transport limited, and erosion or deposition are derived from a mass continuity equation applied to each grid cell. Significant differences from the previous model formulation lie in the procedure for estimating overland flow, and in the sediment transport law used.

Overland flow is now formulated to take account of sub-surface saturation and thus partial contributing areas, whereas previously overland flow was assumed to be generated uniformly over the area. The formulation is similar to that used in Kirkby (1976, 1980). Saturation deficits at the start of a storm are estimated as in equilibrium with average net rainfall, on the assumption of an exponential soil store (Beven and Kirkby, 1979). This gives:

$$D = M \ln(\theta_c/\theta) \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (2)$$

when this expression is positive, and zero otherwise;
where D is the soil moisture deficit below saturation,
$\theta = a/g$ is the ratio of unit catchment area a (as defined above) to local slope gradient g,
θ_c is the value of θ at saturation ($D = 0$),
and M is the soil drainage parameter (with dimensions of length),

Overland flow production is estimated on the basis of daily rainfall totals. The production per unit area on a day with rainfall r is estimated as:

$$\Delta Q = p\ (r-D) \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (3)$$

when this expression is positive, and zero otherwise; where p is a fixed proportion related to soil properties. This expression assumes that all overland flow is generated by saturation excess or return flow mechanisms. Sediment transport by wash is estimated from annual totals of overland flow. These totals are obtained by summing local production rates over an assumed exponential distribution of daily rainfall amounts. Combining equations (1) and (2) above and summing over the distribution:

$$\Delta Q = p\ R \exp(-D/r\emptyset) = p\ (\theta/\theta_c)^{M/r\emptyset} \ \ldots\ldots\ldots\ldots (4)$$

where R is the mean annual rainfall,

$r\emptyset$ is the mean rainfall per rain day

and other symbols are as previously defined. This local mean annual production rate is then summed downslope along all flow lines. The case previously modelled was equivalent to an 'arid' situation with $M/r\emptyset = 0$.

Figure 1 shows a generalization of empirical relationships in the literature, in which rates of sediment transport have been expressed as proportional to distance, area or discharge to the power of m, multiplied by gradient to the power of n. This includes the classic research into soil erosion rates by Musgrave (1947) and others. The previous two dimensional model used exponents $m=0$, $n=1$ for creep and splash; and $m=2$, $n=1$ for wash. In the present version of the model, the gradient exponent for wash has been raised to $n=2$, which is more in keeping with previous empirical values. This exponent of two has been used to estimate wash sediment transport.

Figure 2 illustrates a realization of the revised model for conditions with appreciable sub-surface flow, starting from a fractally perturbed uniform initial slope. It can be seen that a well developed valley system has become established in the lower part of the area, while initial irregularities near the divide have been eliminated, leaving a smooth general convexity. The stable and unstable zones are thus clearly present. Figure 3 shows long profiles down the initial surface, and down the main divide and valley of the final form shown in figure 2. Although the condition of constant downcutting is not met, the zones of profile convexity and concavity can be seen to be approximately associated with the zones of stability and instability respectively.

The streams marked in figure 2 are not explicit in the model, since sediment transport rates are everywhere defined only in terms of overland flow discharge and gradient. Streams have instead been defined in a way which reflects the stability criterion as closely as possible. At each cross-profile of the modelled slope, measured parallel to the upper and lower boundaries of the grid, the initial relief due to the perturbations has been compared with the relief after an average lowering of approximately one metre. Figure 4 shows the ratios of final to initial relief at successive cross-profiles around the possible stream head positions. Points have been joined to show the downslope sequence. On the basis of the stability criterion, instability should be associated with a ratio greater than unity; and stability with lesser ratios. It may be seen that there is a zone where the criterion is in doubt, and the critical point has been taken from the heavy curve drawn through the lower values of the ratio at each unit area. Along each valley axis, this criterion has been used to define the stream head position in terms of demonstrable hollow growth, and the streams in figure 2 are drawn from this analysis. This method provides a refinement of the basic notion of a contour crenulation to distinguish those which are actively being enlarged.

Figure 5 shows the relationship obtained from a series of model runs, in the form of a relationship between critical unit catchment area, defined as just described, and the catchment gradient. Runs have been grouped by the exponent $M/r\varnothing$ which appears in equation (3) above, which is an indicator of the importance of sub-surface flow rising from zero for the arid case to a value of 2-3 for a humid climate like that of Britain. The slopes of the lines drawn through these points are not obtained as a best fit, but are derived from analysis of the stability criterion, which is described in the next section. It may be seen that critical unit area decreases as catchments steepen for the arid case, but where sub-surface flow is more important, the direction of this relationship is reversed.

Analysis of stability for wash process

The criterion of equation (1) may be applied to erosion under the influence of a particular process law to define the theoretical bounds of unstable hollow growth, for an assumed topography. To examine the initiation of channels under circumstancces comparable to the model runs, the topography has been assumed to be a uniformly sloping plane, of gradient g. For this surface $\theta = a/g$, and equation (4) may be integrated downslope to give the overland flow:

$$Q = pa\ (\theta/\theta_c)^{M/r\varnothing}\ /\ (M/r\varnothing+1) \quad \ldots\ldots\ldots\ldots\ldots (5)$$

This expression may be substituted into the sediment transport law, which is here taken in the slightly more general form:

$$S = k\ g + \alpha\ Q^2\ g^n \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (6)$$

where k, α are rate constants for creep/splash and wash and n is the gradient exponent which, in model runs, has had the value of 2.0.

Differentiating equation (6) with respect to a, keeping gradient constant, applying the stability criterion (equation 1), the critical area for instability is:

$$a_c = \left[\frac{k\ (1+\beta)^2}{\alpha\ (2\beta+1)}\ \theta_c^{2\beta}(\rho R)^{-2}\right]^{1/[2(1+\beta)]} g^{[(2\beta+1-n)/2(\beta+1)]} \quad \ldots\ldots (7)$$

where $\beta = M/r\varnothing$ and other terms are as previously defined.

The relationship between critical unit area and gradient is contained in the final term of this expression, and the gradient exponent is shown in figure 6 as $\beta=M/r0$ and the exponent n are varied. Where the gradient exponents for creep and wash are the same (*i.e.* $n=1$), the exponent is never negative, although in the arid case ($\beta=0$) the exponent falls to zero and critical area is completely independent of gradient (as in the former version of the two dimensional model). For the $n=2$ case now being modelled, the reversal of gradient dependence shown in figure 5 is clearly predicted. This change in direction may be best understood by considering the two components of gradient dependence involved. In the absence of sub-surface flow ($\beta=0$), higher gradients produce greater dominance by wash erosion over creep/splash for a given unit area. A smaller area is therefore required to sustain hollow growth from a steeper catchment. Where however, sub-surface flow is significant, soil water levels are closer to saturation on gentle than on steep slopes, so that overland flow is increased: this effect is able to outweigh the direct influence of gradient, and so produce the reversal in trend.

The gradient exponents predicted by equation (7) have been used to draw the lines in figure 5, but the absolute values for critical area are consistently underestimated by the equation. The reason probably lies in the strength of hollow enlargement near the theoretical threshold. At the threshold, the rate of hollow enlargement is zero, and rates of enlargement are directly related to differences from the threshold value. Thus observable enlargment can only occur some distance downslope from the threshold. To give best agreement with model runs, the lines shown in figure 5 are drawn for critical areas of 2.5x the predicted threshold value from equation (7).

The stability criterion for landslide hollows

There is considerable evidence that many valley head hollows are extended aand perhaps formed by landslides during heavy rainstorms and/or in association with earthquakes (*e.g.* Simonett 1967; Dietrich and Dunne, 1978; Crozier *et al*, 1980). There is evidence, for sites in California and Oregon, of an inverse relationship between the area needed to support a stream head and valley gradient (Dietrich *et al*, 1986; Dietrich *et al*, 1987: this volume). This relationship has been supported by an analysis of the dependence of critical slope angle on regolith thickness, which is partly derived from Iida and Okunishi, 1983. Here the landslide process is instead analysed in the context of the Smith and Bretherton criterion for unstable hollow growth, which refers to 'stability' in a different sense to that used in geotechnical stability analyses.

Removal by landslides in combination with creep or splash is modelled here in terms of the continuity equation:

$$-\partial z/\partial t = \partial S/\partial x \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (8)$$

where the rate of lowering is the sum of that from creep/splash and that from slides.
As above in equation (6), creep and/or splash removal is assumed to be transport limited, at a rate defined by the process law:

$$S = k\ g \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (9)$$

For landslides, removal is assumed to be erosion limited, that is to say that lowering is assumed to occur at a rate proportional to the surplus transporting capacity, or the excess of capacity over actual transport rate. This concept has been explored in Kirkby (1984, 1985). In general it may be expressed in the form:

$$-\partial z_L/\partial t = L - S/h \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (10)$$

where $-\partial z_L/\partial t$ is the rate of lowering by landslides,
L is the rate of potential lowering, unconstrained by deposition from above,
and h is the mean travel distance for landslide material.
In Kirkby (1984), the process laws for slides were taken as:

$$L = \lambda\ g(g-g_{\infty}) \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (11)$$

$$h = h_{\infty}/(g_{*}-g) \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (12)$$

where λ, h_{∞} are rate constants,
g_{∞} is the ultimate lower threshold for sliding
and g_{*} is the lowest gradient on which slide material will come to rest; *i.e.* the talus gradient.

The analysis presented by Iida and Okunishi (1983), or by Dietrich *et al* (1987) shows that landsliding should have a distance dependence, and that this should probably influence the threshold gradient g_{∞}, causing it to decrease downslope in response to wetter soils. Such a dependence has not been found to lead to hollow formation however, and equation (11) has here been replaced by a form with a power law increase with distance downslope:

$$L = \lambda\ g(g-g_{\infty})x^{\gamma} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (11a)$$

for an exponent γ greater than zero.

This process law for landslides is intended to represent long terms slide rates and not rates of movement during a slide event. It is used here to forecast slope profiles in equilibrium with a constant rate T of downcutting, to obtain the region of convexity. As has been seen above, this coincides with the region of hollow stability, even when the assumption of constant downcutting is not met. Equations (8) to (11a) above have been applied to the component of slide lowering, by subtracting the creep/splash component. For constant downcutting:

$$\partial S_L/\partial x = -\partial z_L/\partial t = T - k\ dg/dx \quad \ldots\ldots\ldots\ldots (13)$$

where the suffix L refers to landsliding.
Integrating, under the asumption of constant downcutting:

$$S_L = Tx - kg \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (14)$$

Substituting into equation (10), and inserting the process laws (11a) and (12):

$$-\partial z_L/\partial t = T-k\ dg/dx = L-S/h$$
$$= \lambda g(g-g_{\infty})x^{\gamma} - (Tx-kg)(g_{*}-g)/h_{\infty} \quad ...(15)$$

The landslide slopes illustrated below solve this equation for the profile in equilibrium with constant downcutting, ~~using an explicit numerical method.~~

Solutions to equation (15) generally show longer convex sections as gradient declines in response to reduced rates of lowering. Figure 7 illustrates the form of the profiles produced for $\gamma = 0.5$. It may be seen that all show a clear convexity and a gentle concavity. For any given value of γ, there is a well defined negative power law relationship between critical slope length to support a hollow, and mean gradient from divide to stream head (*i.e.* to the point of inflexion for this constant downcutting profile). The exponent in this relationship falls as γ is increased, as is shown in figure 8. A close match to the empirical exponents (0.7) found by Dietrich *et al* (1986) is obtained for $\gamma = 0.5$. It should be noted that hollow shape is implicit in this formulation, and increasing degrees of flow convergence can only be represented by higher γ values.

Discussion

The analysis of hollow formation by landsliding is thus able to give a better forecast of some observed rates of change of stream-head length or area with catchment gradient than the analysis of formation by wash processes. There are however two main difficulties with this interpretation of hollow formation. The more severe is that stream heads, although excavated in some cases by landslides, are neverthelesss the sites of streams. In the analysis of hollow formation by slides, there is no requirement that stream processes will be powerful enough to remove the material initially excavated by a slide. In principle at least, hollows might be formed by slides and never occupied by a functioning stream channel, a possibility which seems to be at variance with observation. For all but the most arid sites, the relationships between critical length or unit area and hollow gradient are as shown schematically in figure 9, with a positive relationship for wash enlargement, and a negative relationship for slide enlargement. In the presence of both processes, hollows might be presumed to form in response to whichever process has the lower threshold, so that hollows would be wash controlled on low gradients and slide controlled on high gradients. This interpretation does not however avoid the original difficulty, in that wash hollows can exist without slides; whereas slide hollows do not appear to exist without streams. There is also the difficulty that the negative relationship for the Californian data extends down to slopes on 10° or less: that is well below geotechnically derived thresholds for slope stability.

The second difficulty is that observed hollows, in California at least (Reneau *et al*, 1984), are occupied by a lens of colluvial fill which tends to taper out towards the stream head, forcing sub-surface flow up towards the surface. A possible method for analysing this problem is by disaggregating the overland flows forecast by equation (3) in the wash case, to give sediment transport and hence rates of hollow enlargement or infilling on a storm by storm basis. A stochastic process is defined in this way, giving distributions of hollow depth and fill depth over time. A preliminary analysis based on equations (3) to (6), with a gradient exponent of $n=1$ for wash, suggests that hollows begin to deepen appreciably for unit areas greater than about one tenth of the critical threshold value for hollow stability, but are typically full of colluvium until the unit area approaches the threshold. If sub-surface flow is largely confined to this colluvial layer, over a relatively impermeable bedrock, then flow will be brought up to the surface at the base of the colluvial lens, as observed by Dietrich *et al* (1987). The siting of the functional stream head near the point of critical threshold area will be further reinforced by this mechanism.

This paper raises a number of questions about the position of stream and valley heads. Two processes have been discussed as possible candidates for valley enlargement. Wash is thought to be capable of eroding a valley provided that the critical tractive stress for soil aggregates is exceeded, and that storms are of long duration relative to the time of overland flow travel down hillsides (*i.e.* not in arid climates under Horton overland flow). As the climate becomes more humid, and sub-surface flow more important, the area needed to support a stream head is forecast to increase as catchment gradients fall, suggesting an increase in drainage density as the landscape is lowered by erosion. Landslides have also been shown to be capable of producing valley heads, provided that long-term slide transport rates increase substantially with unit area or distance downslope, but it is not clear from the analysis that landslide hollows should necessarily contain a stream. For landslide hollows, drainage density should gradually decrease as gradients fall during catchment erosion, and hollows should be eliminated when gradients fall below an ultimate geotechnical threshold.

References

K.J. Beven and M.J Kirkby, 1979. A physically-based, variable contributing area model of basin hydrology. *Hydrological Sciences Bulletin* 24(1), 43-69.

M.J. Crozier, R.J. Eyles, S.L. Marx, J.A. McConchie and R.C. Owen, 1980. Distribution of landslips in the Wairarapa hill country. *New Zealand Journal of Geology and Geophysics* 23, 575-586.

W.E. Dietrich and T. Dunne, 1978. Sediment budget for a small catchment in mountainous terrain. *Zeitschrift fur*

Geomorphologie, Supp.Bd. 29, 191-206.

W.E. Dietrich, C.J. Wilson and S.L. Reneau, 1986. Hollows, colluvium and landslides in soil-mantled landscapes. in *Hillslope Processes* (Ed. A.D. Abrahams), Allen and Unwin, 361-388.

~~W.E. Dietrich, C.J. Wilson and S.L.~~ Reneau, (in press: this volume). Processes controlling the size of unchannelled valleys on soil-mantled slopes of humid regions. *Catena.*

T. Iida and K. Okunishi, 1983. Development of hillslopes due to landslides. *Zeitschrift fur Geomorphologie* 46, 67-77.

M.J. Kirkby, 1976. Implications for sediment transport. in *Hillslope hydrology* (ed. M.J. Kirkby), John Wiley, 325-363.

M.J. Kirkby, 1980. The stream head as a significant geomorphic threshold. in *Thresholds in geomorphology* (Ed. D.R. Coates and J.D. Vitek), Allen and Unwin, 53-73.

M.J. Kirkby, 1984. Modelling cliff development in South Wales: Savigear re-viewed. *Zeitschrift fur Geomorphologie* 28 (4), 405-426.

M.J. Kirkby, 1985. A model for the evolution of regolith-mantled slopes. in *Models in geomorphology* (Ed. M.J Woldenberg), Allen and Unwin, 213-237.

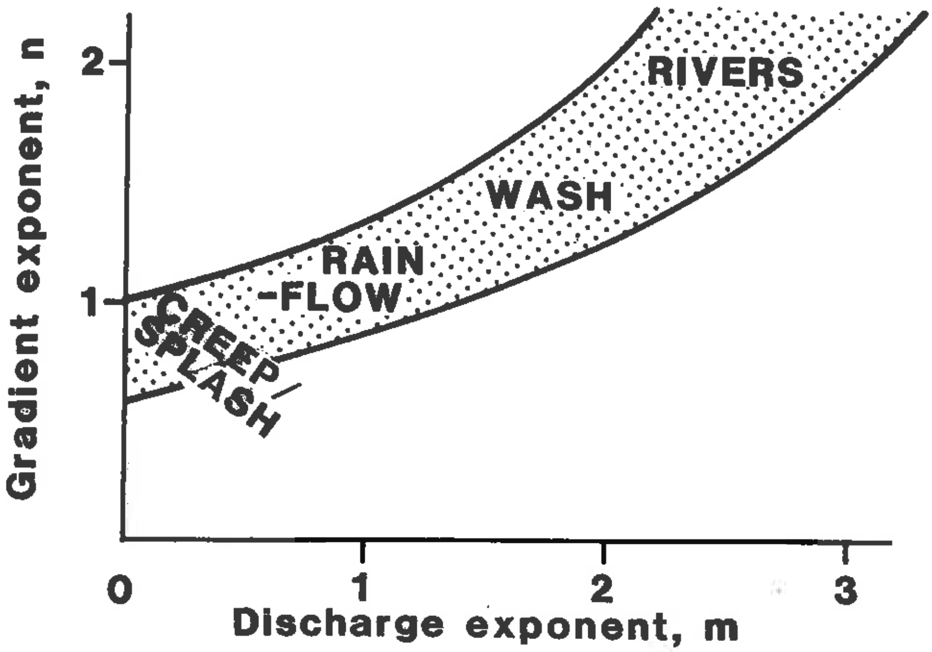
M.J. Kirkby, 1986. A two-dimensional simulation model for slope and stream evolution. in *Hillslope Processes* (Ed. A.D. Abrahams), Allen and Unwin, 203-222.

G.W. Musgrave, 1947. Quantitative evaluation of factors in water erosion - a first approximation. *Journal of Soil and Water conservation* 2, 133-138.

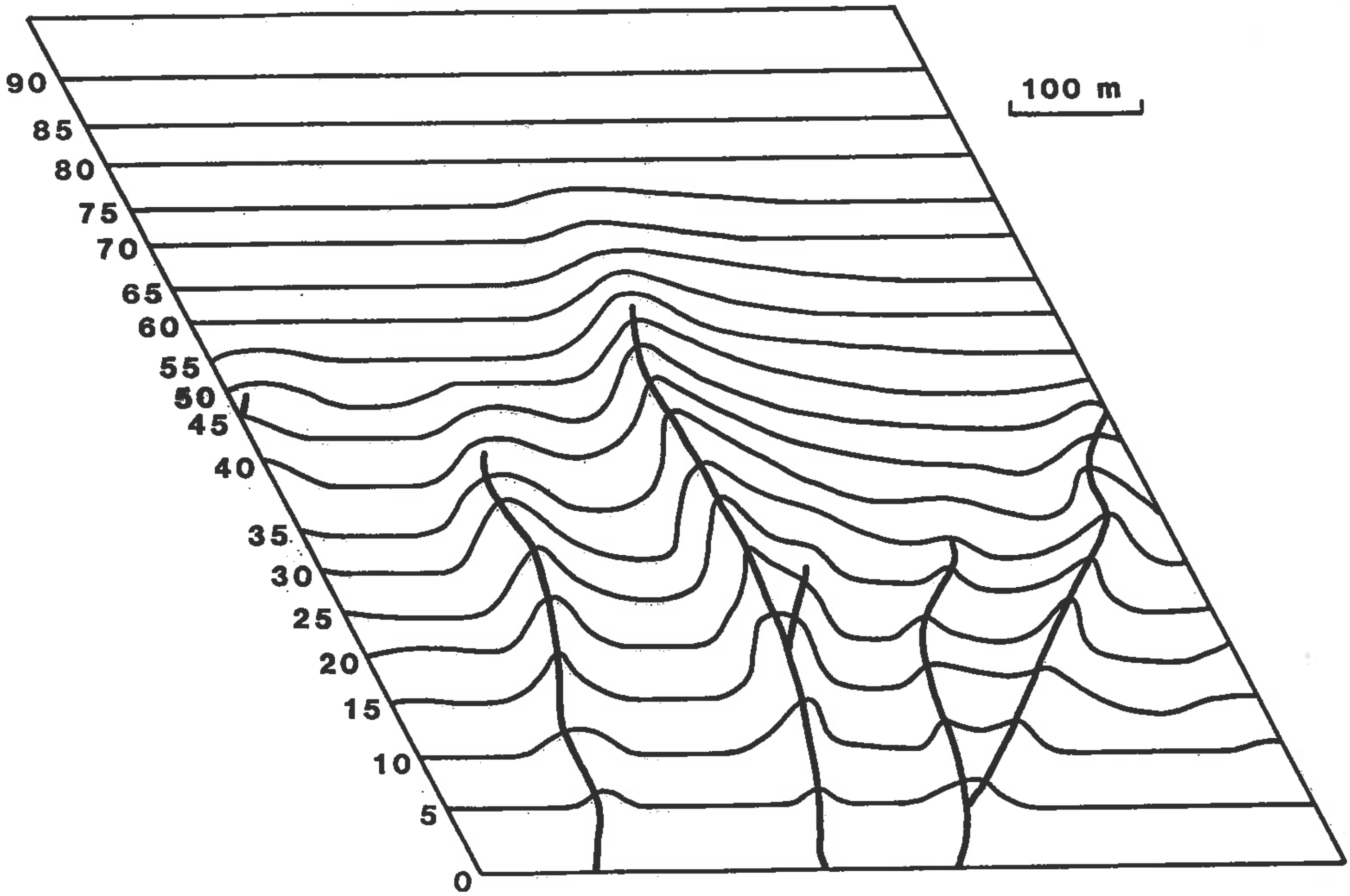
S.L. Reneau, W.E. Dietrich, C.J. Wilson and J.D. Rogers, 1984. Colluvial deposits and associated landslides in the Northern San Francisco Bay area, California, U.S.A. *Proceedings of the 4th International Symposium on landslides. International Society for Soil Mechanics and Foundation Engineering, Toronto, Ontario, Canada*, 425-430.

D.S. Simonett, 1967. Landslide distribution and earthquakes in the Dewani and Torricelli Mounntains, New Guinea. In *Landform studies from Australia and New Guinea* (ed. J.N. Jenninngs and J.A. Mabbutt). Methuen, London.

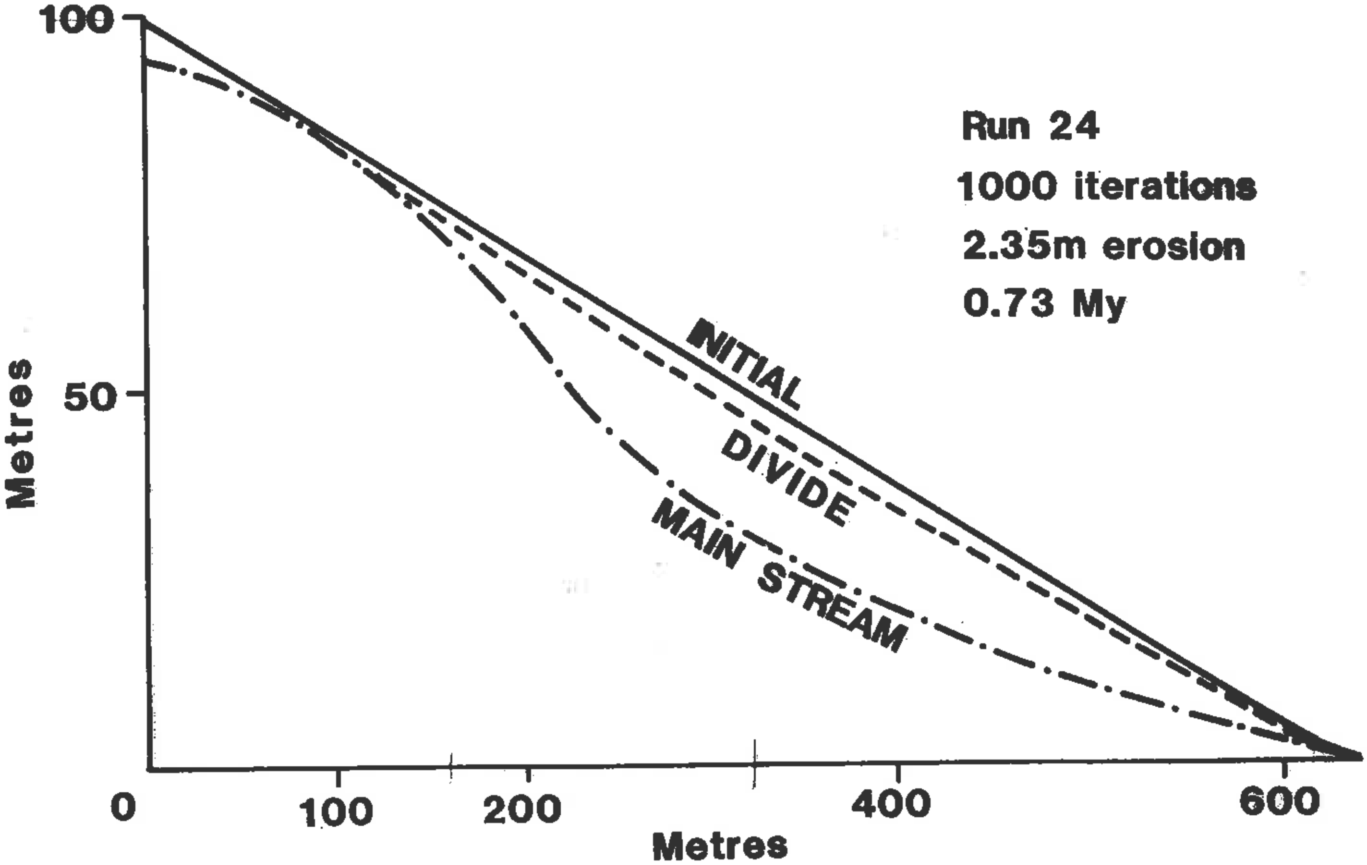
T.R. Smith and F.P. Bretherton, 1972. Stability and the conservation of mass in drainage basin evolution. *Water Resources Research* 8(6), 1506-1529.



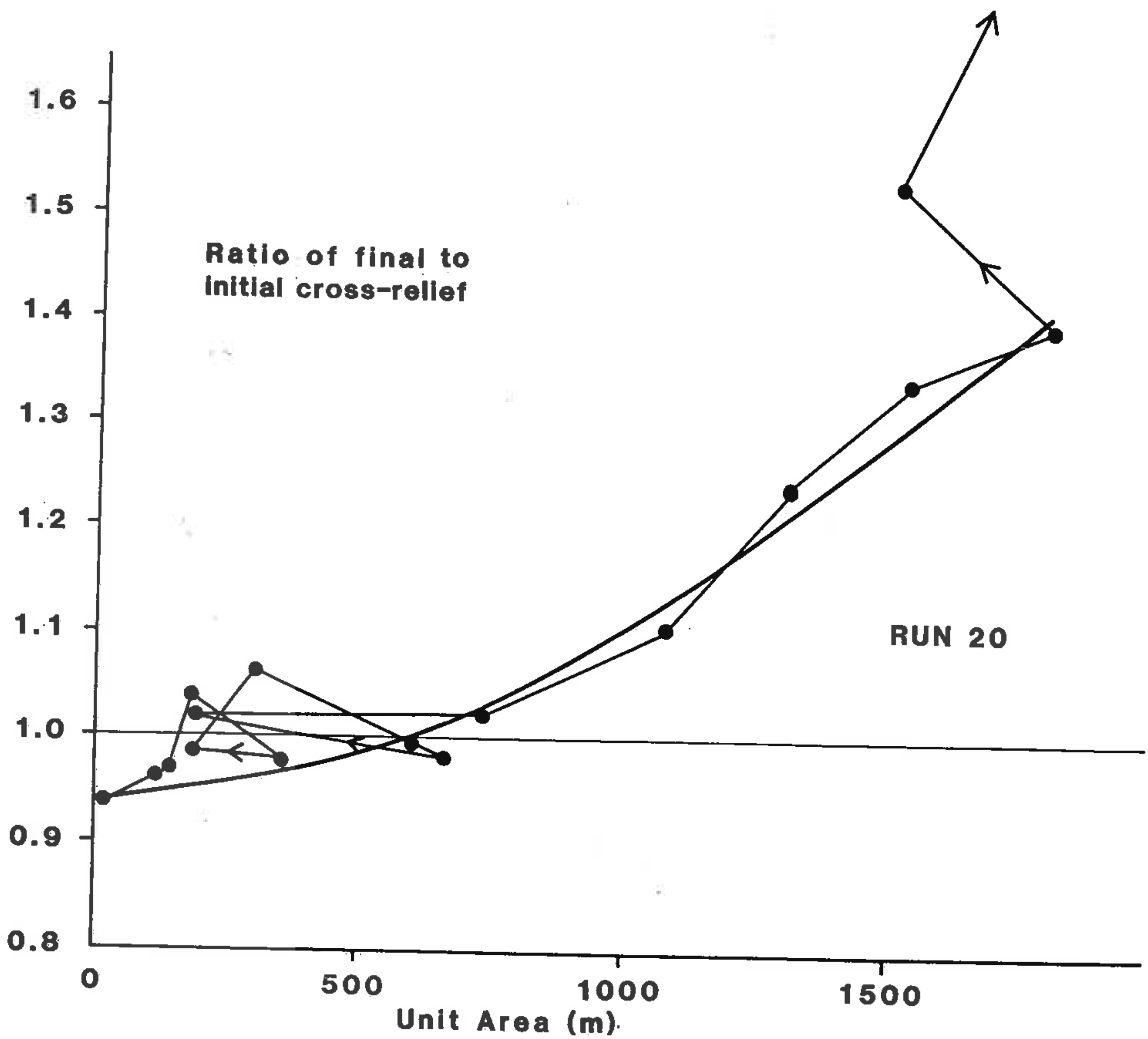
1. The trend of empirical expressions for sediment transport by creep, splash and wash; of the form Sediment Transport proportional to $(\text{Discharge})^m \times (\text{Gradient})^n$.



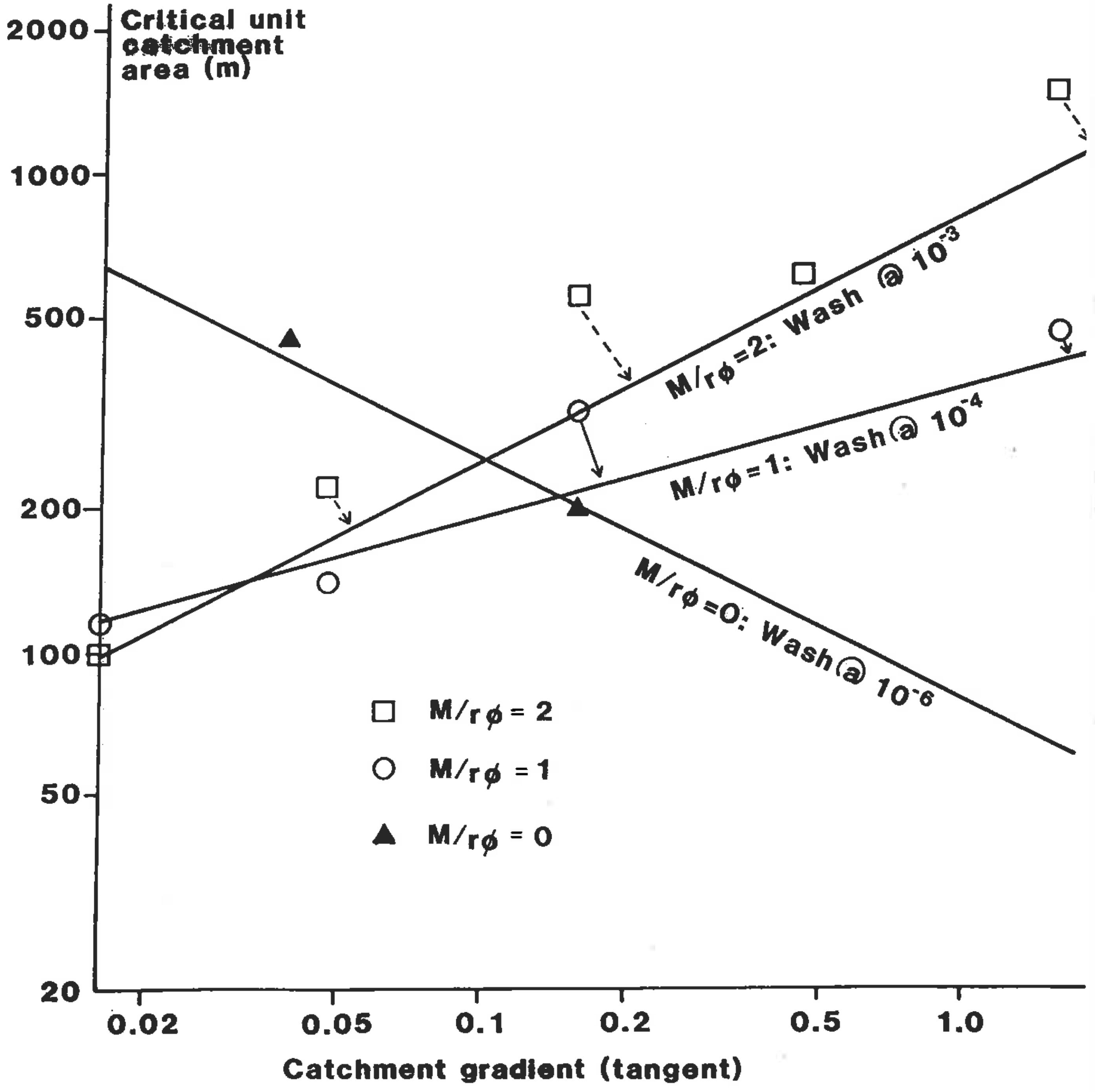
2. Example run from computer simulation of slope evolution under wash and splash/creep. Parameter values are: $\beta = H/r0 = 2$: Creep rate $k = 10^{-3}\ m^2y^{-1}$: Wash rate $\alpha = 10^{-5}\ m^2y^{-1}$: 2.35 m erosion after 1000 iterations, representing 0.73 My elapsed from initial perturbed uniform gradient.



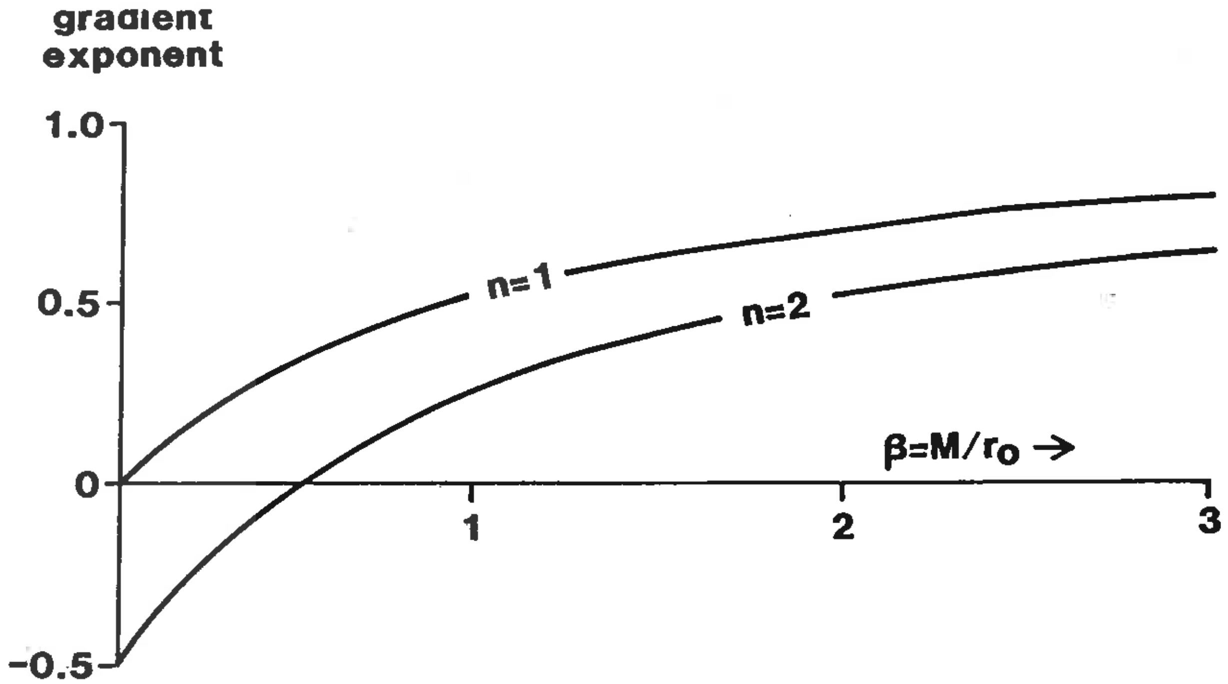
3. Long profiles for the simulation run shown in figure 2. Profiles shown for initial gradient, and final gradients along main divide and main valley axis.



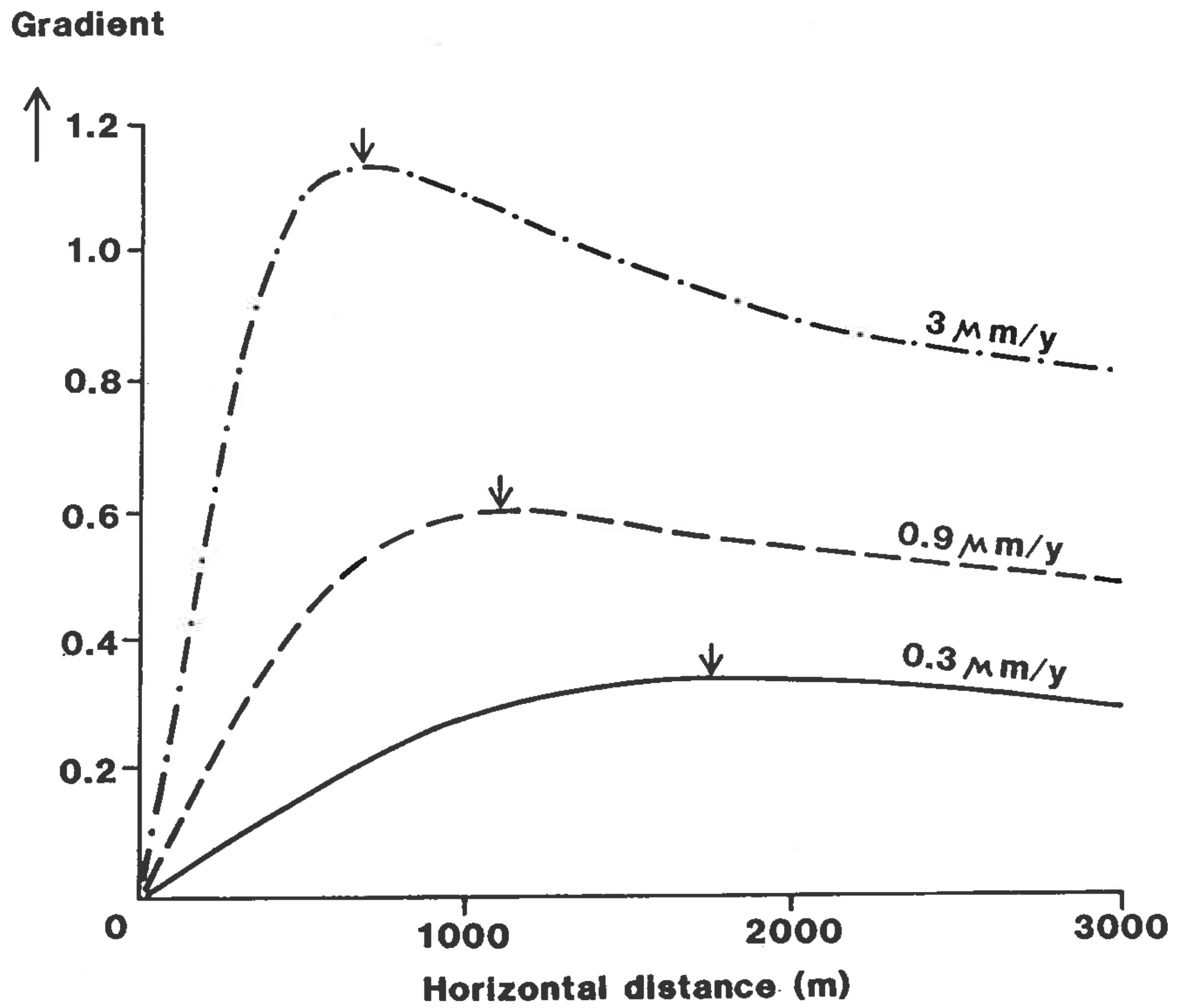
4. Example simulation result showing the ratio of final to initial cross-relief in terms of unit area at successive points downslope, joined by solid line. Heavy line shows best fit curve fitted to lower ratios, to establish effective detection threshold for hollow enlargement.



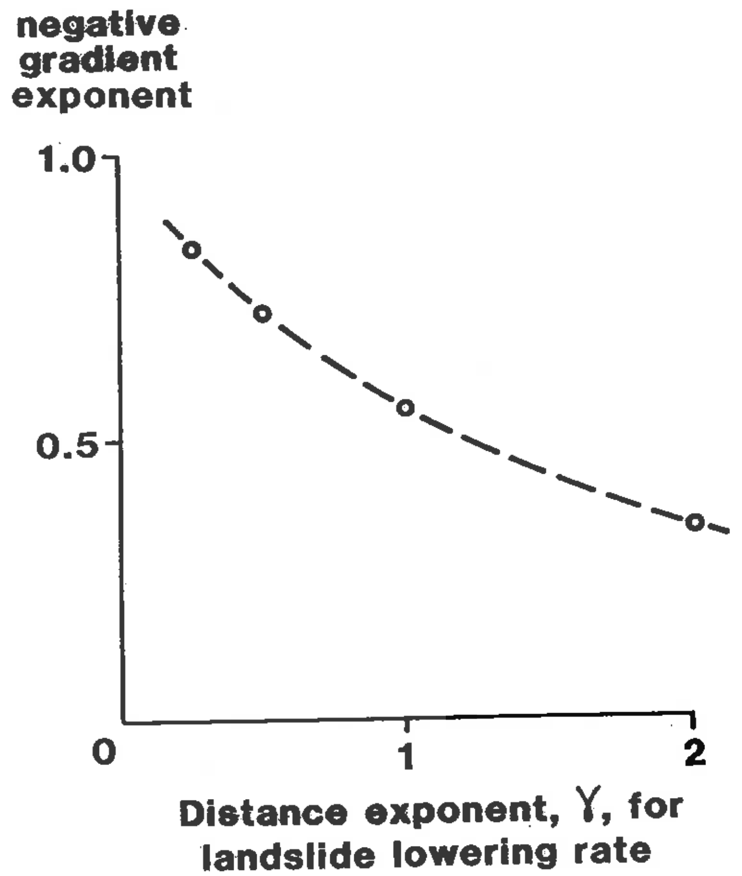
5. Critical unit area for hollow enlargement plotted against catchment gradient. Points are from computer simulations. Lines are calculated as 2.5x critical areas obtained from equation (7) with gradient exponent n=2.



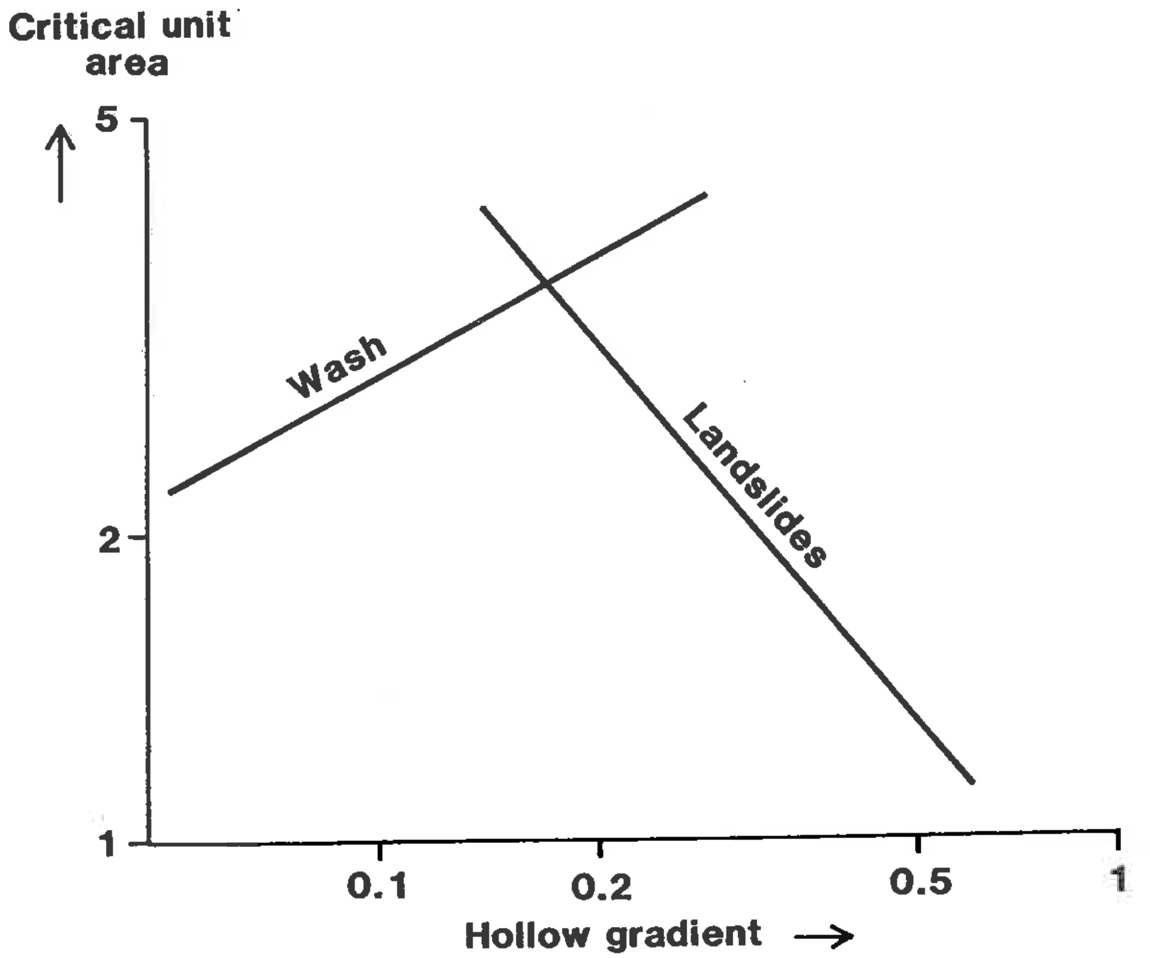
6. Gradient exponent from equation (7) for critical wash unit area as a power of catchment gradient; expressed in terms of wash gradient exponent n (in equation 6) and importance of sub-surface flow ($\beta = M/r_0$).



7. Example slope profiles in equilibrium with constant rates of downcutting, under erosion by landslides in combination with splash/creep. For this example $\gamma = 0.5$: Rates of lowering are shown on curves.



8. Gradient exponent for critical slide unit length as a power of catchment gradient; expressed in terms of distance exponent γ in equation (11a).



9. Hypothetical relationship between critical unit areas for wash and landslides, each in competitiion with splash/ creep. At any gradient, lower threshold should be the relevant one.

Appendix: Listing for PASCAL program slop3DB.

Program developed on Sinclair QL system using Metacomco ISO Pascal compiler. Screen and file handling routines are specific to this environment.

```
program slop3DB (output, input, filein, fileout, printer);
 {Uses an initial surface from flp file,}
 {Simulates slope evolution on surface using [s^m+(q^2.s^n] transport law}
 {Outputs elevations to printer and integer file, row by row}
include 'flp1_3dCom_inc' {contains global const, type, var declarations}
  {contains functions screen, window, ranorm, OK}
  {contains procedures textwin, graphwin, blob, readword, OpenFile, SaveData}
procedure Indata; {picks data from integer file of elevations}
  var inTitle: string; x,y: range; col, z: integer;
  begin {Indata}
    textwin(0); writeln('Enter filename for Initial form data'); writeln(' e.g.
     flp1_In1_dat');
    write('>'); readword(inTitle);
    reset(filein, inTitle);
    repeat
      read (filein, z, Iteration); Now:=z;
      textwin(4); writeln('Time =',Now:8);
      writeln('Iter',iteration:4); graphwin;
      for y:=0 to size-1  do
        for x:=0 to size-1  do begin
          read (filein, z); map[x,y].elev:=z;
          col:=z div 10000000; blob(x,y,col);
          if x=0 then begin blob(size,y,col); map[size,y].elev:=z end
          end; {x,y loops}
      for x:=0 to size do begin map[x,size].elev:= 0; blob (x,size,0) end
      until OK;
    writeln (printer, 'Starting at time ',Now:8, ' with Iteration',iteration:4)
  end; {indata}
procedure FindDirections; {assigns flow directions at each point in elev[x,y]}
  var x,y,tx,ty,index,min,current,colour: integer; pos: dirRange;
  begin
    for y:=0 to size-1 do
      for x:=0 to size-1 do begin
        min:=maxint;
        index:=0; if y=0 then index:=2;
        repeat
          index:=index+1;
          tx:=x+dx[index]; if tx<0 then tx:=size-1;
          ty:=y+dy[index];
          current:= map[tx,ty].elev;
          if current<min then begin min:=current; pos:=index end
        until index= dirmax;
      map[x,y].dir:=pos  end; {x,y loops}
    for y:=0 to size do begin
      map[y,size].dir:=4; map[size,y].dir:=map[0,y].dir end
  end; {FindDirections}
function Downslope (var x,y: integer):boolean; {finds next point downslope}
  var index: dirRange;
```

```
   begin
     index:=map[x,y].dir; x:=x+dx[index]; y:=y+dy[index];
     downslope:= (index<>4);
     if x>=size then x:=x-size;
     if x<0 then x:=x+size
    end; {downslope}
 procedure areas(flag: switch); {works out drainage areas}
   var x,y: range; nx,ny,h,increment,a, amax: integer; tt, AreaScale: real;
   begin
     textwin(17);
     writeln('Areas (',flag:2,' )'); amax:=0;
     for y:=0 to size do
       for x:=0 to size do
         map[x,y].area:=0;
     for y:= 0 to size do begin
       write(y:3);
       for x:=0 to size-1 do begin
         nx:=x; ny:=y;
         case flag of
           0: increment:=1;
           1: begin tt:=map[nx,ny].tpt; increment:=round(tt*1e5) end
         end;
         repeat
           a:=map[nx,ny].area+increment; map[nx,ny].area:=a;
           if a>amax then amax:= a
         until not downslope(nx,ny)
       end end; {for x,y}
     for y:=0 to size do
       map[size,y].area:=map[0,y].area;
     AreaScale:=7/ln(amax);  graphwin;
     for y:=0 to size do
       for x:=0 to size do begin
         a:=map[x,y].area;
         if a>1 then a:=trunc(ln(a)*AreaScale) else a:=0;
         blob(x,y,a) end; {x,y loops}
     textwin (17)
   end; {areas}
 procedure flow; {works out overland flow contribution at each point}
   var x,y: range; grad,tx,ty: integer; temp, OvFlow: real ; flag: boolean;
   begin  temp:=hstep*hstep/vstep/AScrit*oblique*1e6;
     textwin(17); writeln('doing OF');
     for x:=0 to size-1 do
       for y:=0 to size do begin
         tx:=x; ty:=y; flag:=downslope(tx,ty);
      grad:=map[x,y].elev-map[tx,ty].elev;
         if grad=0 then OvFlow:=1
         else begin
           OvFlow:=map[x,y].area*temp/grad; if OvFlow>=1 then OvFlow:=1
             else OvFlow:=exp(OFexp*ln(OvFlow)) end; {if}
         map[x,y].tpt:=OvFlow end; {for x,y loops}
     for y:=0 to size do begin temp:=map[0,y].tpt; map[size,y].tpt:=temp end;
     areas(1) {accumulates OvFlow downslope in area[x,y]}
   end; {flow}
 procedure SedTpt; {works out gains and losses of sediment}
```

```
   var x, y: range  ; tx, ty, OvFlow, u, v: integer;
       z, grad, const1, const2, const3, tt: real;
   begin
     textwin(17); writeln('Erosion rates (m/y)');
     const1:=rainfall*hstep*oblique; const2:=vstep/hstep/1e6;
     const3:= 1/oblique/hstep;
     for x:=0 to size do
       for y:=0 to size do
         map[x,y].tpt:=0.0;
     for x:=0 to size-1 do
       for y:=0 to size-1 do begin
         tx:=x; ty:= y;
         if downslope(tx,ty) then begin
           u:=map[x,y].elev; v:=map[tx,ty].elev; grad:=ln((u-v)* const2);
           case state of
             1: z:= exp(creepexp*grad)*creep*const3;
             2: z:=map[x,y].area*const1;
             3: z:=map[x,y].area*const1/1e5
             end; {case}
           if state>1 then begin
             z:=z*z; {assumes OF exponent of 2}
             z:=(exp(creepexp*grad)*creep+z*exp(washexp*grad)*wash)*const3
             end; {if}
           tt:=map[x,y].tpt; map[x,y].tpt:=tt-z; tt:=map[tx,ty].tpt;
     map[tx,ty].tpt:=tt+z
           end {if downslope}
         end; {for x,y}
    for y:=0 to size do begin tt:=map[0,y].tpt; map[size,y].tpt:= tt;
     map[y,size].tpt:=0.0 end; {y loop}
    erosion:= 0.0;
    for y:=0 to size do
      for x:=0 to size-1 do begin
      z:= map[x,y].tpt; erosion:=erosion-z end {x,y loops}
  end;  {SedTpt}
procedure Update; {Increments by dt or up to TimeStep: Redraws elevation map}
  var x, y, tx, ty: range; z,cc, nx, ny: integer; grad, max, test, dt, u, v:
     real;
  begin
    max:=-maxint;
    for y:=0 to size-1 do
      for x:=0 to size-1 do
        begin
          nx:=x; ny:=y;
          if downslope (nx,ny) then begin
            u:=map[x,y].elev; v:=map[nx,ny].elev; grad:=(u-v)/1E6;
            if (grad>MinElevDiff) then begin
              u:=map[x,y].tpt; v:= map[nx,ny].tpt; test:=abs((u-v)/grad);
              if test>max then begin max:=test; tx:=x; ty:=y end
              end end {both if's}
        end; {x,y loops}
    dt:=Threshold/max; if now+dt > NextPrint then dt:=NextPrint-now;
    textwin(0); writeln('Critical at'); writeln(tx:4,', ',ty:4);
    now:=now+dt; writeln('Iter ',Iteration:4);
    writeln('of ',dt:10); writeln('to ',now:10);
```

```
erosion:=0.0; graphwin;
    for z:=0 to 200 do begin blob(tx,ty,0); blob (tx,ty,7) end;
    for y:=0 to size do
      for x:=0 to size do
        begin
          test:=map[x,y].tpt; test:= test*1e6; if (x>0) then
     erosion:=erosion-test;
          z:=map[x,y].elev; z:=z+round(test*dt); cc:=z div 10000000;
          map[x,y].elev:=z; blob(x,y,cc)
          end; {x,y loops}
    textwin(8); erosion:=erosion/size/size; writeln('Erosion rate');
     writeln(erosion:12:2,' um/yr');
    total:=total+erosion*dt/1e6; writeln('Total =',total:10:4,'m.');
    writeln( printer,'Critical at ',tx:4,' , ',ty:4);
    writeln (printer,'Iter ',Iteration:4,' of ',dt:12,' to ',now:12);
    writeln (printer,'Erosion @',erosion:12:2,'um/y; and Total so far
     =',total:13:7,'m'); writeln (printer)
  end; {UpDate}
begin {MainProgram}
  rewrite (printer,'ser');
  yunit:=128 div size; xunit:= 192 div size;
  graphwin; err:=screen2(screenpaper,2); err:=screen2(screenink,7);
  err:=screen1(screenclear);
  writeln; writeln('   Enter grid TOTAL diameter and height unit (m)');
  write('   >'); readln(hstep,vstep);
  writeln('   Enter parameters as follows:');
  writeln('      1: Total annual rainfall in mm');
  writeln('      2: Exponent M/ro (dimensionless: 0-3)');
  writeln('      3: a/s in m.(>500) at saturation under mean Net Rf');
  writeln('         =365 * Surface Ksat(m/day) * M(mm) / Net Rf(mm/yr)');
  write('   >'); readln (Rainfall, OFexp, AScrit);
  writeln; writeln('   Enter process rates in msq/year for:');
  writeln ('      1: Creep, splash (0.001) or solifluction (0.01)');
  writeln ('      2: Gradient exponent for creep etc (1)');
  writeln ('      3: Wash rate constant (0.02)');
  writeln ('      4: Gradient exponent for wash(1 to 2)');
  write('   >');readln(creep, creepexp, wash, washexp);
  InData;
  writeln; writeln('Data file gives Start time =',Now:8,' and
     Iter',iteration:4);
  writeln; writeln('   Enter Times (yrs) to start, finish, and save data');
  write('   >'); readln(StartTime, EndTime, TimeStep);
  write('Enter frequency at which to save data and quit>'); readln(freq,
     EndFreq);
  writeln (printer,'Rf =',Rainfall:6:1,': M/ro =',OFexp:5:3,': A/S for OF
     =',AScrit:5:3);
  writeln(printer, 'Creep rate =',creep:6:4,'msq/y: Gradient exponent
     =',creepexp:4:1);
  writeln(printer, 'Wash rate  =',wash:6:4,'msq/y : Gradient exponent
     =',washexp:4:1);
  writeln(printer,'Times: from',StartTime:10:1,' to',EndTime:10:1,' Years
     or',EndFreq:5,' Iterations');
  writeln (printer,'Printing every',Timestep:10:1,' yr or',freq:4,'
```

```
       Iterations');
    Rainfall:=Rainfall/1000; hstep:=hstep/size;
    writeln (printer,'Grid horiz unit =',hstep:6:2,' and Vert unit
       =',vstep:6:2,'m.');
    if wash=0 then state:=1
      else if OFexp=0 then state:=2 else state:=3;
    for mx:=1 to dirmax do begin
      dx[mx]:=mx mod 3 -1; dy[mx]:=mx div 3 -1 end;
    Now:= StartTime; NextPrint:=trunc(Now/TimeStep)*TimeStep;
    if NextPrint<Now then NextPrint:=Nextprint+TimeStep;
    Total:=0.0; Iteration:=0; OpenFile;
    while (Now<EndTime) and (Iteration<EndFreq) do
      begin
        FindDirections;
        if state>1 then Areas(0);
        if ((now=NextPrint) or (Iteration/freq=Iteration div freq)) then
          begin
            SaveData(now);
            if now= NextPrint then NextPrint:=now + TimeStep
          end; {if}
        if state=3 then Flow;
        SedTpt;
        Iteration:= Iteration +1; Update;
      end; {while}
   saveData (now)
  end.{MainProgram}
```

Listing for '3Dc-m_inc' include file used by main listing above.

```
const
  size                 = 32;  {size of fractal map: must be a power of 2<=64}
  dirmax               = 7;   {hexagonal case: 6 directions + centre}
  screenclear          = 32; {constants for QL graphics routines}
  screenclearEOL       = 36;
  screenink            = 41;
  screenpaper          = 39;
  screenat             = 16;
  screenmode           = 44;
  screenstrip          = 40;
  windowdefine         = 13;
  windowfillblock      = 46;
  oblique              = 0.8660254;
  ranmult              = 25173; {parameters     of}
  ranadd               = 13849;      {random number}
  ranbase              = 65536;          {generator}
  blank                ='                   ';
  maxword              =19;
  ARstring             ='aArR';
  rowmax               =24;
  MinElevDiff          =0.001;  {To allow divide migration}
  Threshold            =0.7; {Maximum prop change in gradient away from divide}
type afile       = file of integer;
```

```
        range        = 0..size;
        dirRange     = 1..dirmax;
        switch       = 0..1;
        deltarange   = -1..1;
        delta        = array[dirRange] of deltarange;
        string       = packed array[1..maxword] of char;
        landscape    = record
                          tpt: real;
                          elev: integer;
                          area: integer;
                          dir: dirRange;
                        end;

  var  title: string;
       teststring: packed array[1..4] of char;
       filein, fileout: afile;
       printer: text;
       ed,sf,ch: real;
       state: 1..3;
       hstep, vstep, Rainfall, OFexp, AScrit, creep, wash, creepexp, washexp:
        real;
       status,row,err,xunit,yunit,ranval,ox,oy,w,d,mx,my, freq, EndFreq,
        Iteration: integer;
       StartTime, TimeStep, EndTime, Now, NextPrint, erosion, total: real;
       dx,dy: delta;
       minx: array[range] of range;
       map: array[range, range] of landscape;
  {external functions for QL graphics from include files}
 function screen1(code: integer): integer; external 22;
 function screen2(code,arg1: integer): integer; external 22;
 function screen3(code, arg1, arg2: integer): integer; external 22;
 function window6(code: integer; var d1,d2,d3,d4: integer; colour: integer):
       integer; external 21;
 function window7(code: integer; var d1,d2,d3,d4: integer; colour, width:
       integer): integer; external 21;
 function ranorm(var seed: integer):integer;
   {random number normally distributed, mean = 0, SD = ranbase}
   var sum,index: integer;
   begin
     sum:=0;
     for index:= 1 to 12 do
       begin seed:= (ranmult*seed+ranadd) mod ranbase;
         sum:= sum+seed
       end;
     ranorm:=sum-ranbase*6
   end;  {ranorm}
 procedure textwin(row: integer); {opens text window on right of screen}
   begin
     w:=row; w:=w+w; oy:=w+w; oy:=oy+oy+w;
     w:=128; d:=256-oy; ox:=384;
     err:=window7(windowdefine,w,d,ox,oy,7,2);
     err:=screen1(screenclear); writeln(output);
   end; {textwin}
 procedure graphwin; {map window on left of screen}
```

```
    begin
      w:=384; d:= 256; ox:=0; oy:=0;
      err:=window7(windowdefine,w,d,ox,oy,0,0); if err<>0 then
        writeln(output,'Err',err:4,'/graphwin')
    end;  {graphwin}
 function OK: boolean;
   var tch: char; pos, index: integer;
   begin
     teststring:=ARstring;
     writeln('Accept/ Revise ?'); pos:=0;
     while pos = 0 do begin
       write('>'); readln (tch);
       for index:= 1 to 4 do
         if tch=teststring[index] then pos:= index
     end; {while}
     OK:=(pos<3)
   end;  {OK}
 procedure readword(var word: string);
   var ch: char; pos: integer;
   begin
     word:=blank; pos:=0; read (ch);
     while (ch<>' ') and (pos<maxword) do begin
       pos:=pos+1; word[pos]:=ch; read (ch)
     end; {while}
   writeln(output, word); end; {readword}
 procedure blob(x,y,colour: integer); {draws a unit square centred on x,y}
   var bw,bd,bx,by,bc,btemp: integer;
   begin
     bc:=colour mod 8; err:=0;
     bd:=yunit+yunit;bw:=xunit+xunit;
     bx:=(x+x+y)*xunit-96; if bx<0 then bx:=bx+384;if bx>=384 then bx:=bx-384;
     by:=(y+y-1)*yunit; if by<0 then begin bd:=bd+by;by:=0 end;
     if (by+bd)>256 then bd:=256-by;
     if bx+bw>384 then
       begin btemp:=bw; bw:=384-bx;
       err:=err+window6(windowfillblock,bw,bd,bx,by,bc);
         bx:=0; bw:= btemp-bw; err:= err+window6(windowfillblock,bw,bd,bx,by,bc)
       end
     else
       begin
         err:= err+window6(windowfillblock,bw,bd,bx,by,bc)
       end; if err<>0 then writeln(output,'Err',err:4,'/blob')
   end; {blob}
 procedure openfile; {opens a file for output of integer data values}
   begin
     writeln (output); writeln ('Enter filename for Data output eg.');
     writeln('flp1_3D_dat');
       write('>');readword(title);
     rewrite (fileout,title); writeln(output,title,' open');
     writeln (printer,title,' open to save data for elevations and areas')
   end; {openfile}
 procedure SaveData(time: real); {Saves elevations and areas to disc}
   {Trunc(time), Iteration No, All Elev*1e6, All Areas (4-byte integers)}
   var x, y, xmin: range; zmin, zmax, z: integer;
```

```
begin
  write(fileout,trunc(time),Iteration);
          writeln(printer); writeln(printer);
          writeln(printer,'Time', time:12:3);
          writeln (printer,'Elevations in m.');
    for y:=0 to size-1 do begin
      writeln (printer,'y =',y:3,' *************');
      zmin:=maxint; zmax:=-zmin;
      for x:=0 to size-1 do begin
        z:=map[x,y].elev;  write (printer,z/1e6:10:6);
        if z>zmax then zmax:=z;
        if z<zmin then begin zmin:=z; xmin:=x end;
        write(fileout,z) end; {x loop}
      minx[y]:=xmin; z:= map[xmin,y].area;
      writeln (printer,'Range =',(zmax-zmin)/1e6:9:5,' at x =',xmin:3,' with
    Area of',z:5)
      end {y loop}
end;   {SaveData}
```