



---

# Specification of the SEDUM model for modelling patterns of sediment transport based on unit stream power

NLWRA Sediment Project (CLW12)

C.J. Moran, I. Prosser and G. Cannon

---

CSIRO Land and Water, Canberra  
Technical Report 25/01, July 2001

---

# Specification of the SEDUM model for modelling patterns of sediment transport based on unit stream power

NLWRA Sediment Project (CLW12)

C.J. Moran, I. Prosser and G. Cannon

CSIRO Land and Water

Technical Report Number 25/01

This document details the intellectual property associated with the SEDUM erosion, deposition and sediment delivery model of CSIRO Land and Water, Australia. This is not a published document and therefore does not constitute a referable source. Those wishing to use material included in this specification should do so only having received written permission from the authors.

## Table of Contents

1. Background.....	3
2. Governing equations for sediment discharge.....	3
3. The one-dimensional transporting energy budget.....	8
4. The two-dimensional transporting energy budget .....	9
4.1. The flow routing algorithm.....	9
4.2. Definitions of flow path and critical variables .....	12
4.3. The transport algorithm .....	13
4.4. Sediment sorting and nutrient transport.....	15
4.5. The transport algorithm with sediment sorting.....	17

## 1. Background

Here we consider that the movement of soil through landscapes by overland flow is controlled by topography, soil properties, climate (frequency of large precipitation and drought events) and landuse. We attempt to model the behaviour of a segment of the landscape which, for convenience of internal flow properties, will often be the catchment basin of a river (or channel) branch point. The segment is represented as a 2-D matrix of elevations - a standard digital elevation model (DEM). Our interpretation of the DEM is that the elevation of each element (pixel) is considered to be constant across the pixel area. Therefore, in close-up, the DEM appears as a series of steps whose elevations change discontinuously at the edges.

Our interest is not in modelling the effect of individual flow events but rather to try to capture the behaviour over a large number of events by examining the accumulation of soil transport from the edges of our landscape segment to predefined delivery points. This will often be the catchment boundary to the points along the channel. The summation of delivery to the points along the channel network (or to other defined delivery points, e.g., dams, lakes, discharge areas) from the defined drainage point is used as a summary term of the *segment response*. The aim of this approach is to quantify the effect of changing the conditions within the segment on the segment response. Therefore, we are able to ascribe a hazard rating to each point within the segment based on its contribution to the segment response (this is the basis of a future paper).

This document specifies the approach to deriving the transported soil accumulation through the segment. This is performed by defining flow as movement from higher to connected lower pixels and supplying soil to the flow accumulation, from each unit, using the unit stream power. Variations in the surface friction are neglected; it will be considered at a later date when the use of the segment and the soil properties are examined. Therefore, the remaining terms driving stream power are the slope of each element of the segment and the area above it.

The surface soil across the segment is considered to be the potential supply of material for transport. The soil that can be exported from a given pixel is considered to be of uniform erodibility and texture.

The model will be called SEDUM. (other suggestions welcome I have a few more possibilities. e.g. SEDUMA, SEDAM, PASSED, SADEST, MADEST, DESPOT)

## 2. Governing equations for sediment discharge

Unit stream power,  $P$  is defined as the time rate of potential energy dissipation per unit weight of water.  $P$  is the dominant factor controlling sediment concentration when sediment availability does not limit the transport rate.

**Equation 1**

$$P = Vs$$

Where  $V$  is the velocity of flow in the downstream direction and  $s$  is the energy gradient which is approximated by the slope gradient,  $M = \sin \theta$ , for gradually varying flow.

Yang and others have found that the total sediment concentration,  $C_t$ , is predicted by:

**Equation 2**

$$C_t = k_1(P - P_{cr})^\zeta$$

where  $k_1$  is a constant dependant on particle size, particle fall velocity, kinematic viscosity of the fluid, and the average shear velocity of the flow;  $P_{cr}$  is the critical power required to initiate transport which can be neglected assuming  $C_t > 100 \text{ mg.l}^{-1}$ , also the concept of critical grain resistance has been questioned in recent years; the exponent  $\zeta$  has been found empirically to range from 0.82-1.12 and is usually approximated as 1.0. Therefore Equation 2 can be simplified to:

**Equation 3**

$$C_t = k_1P$$

Total sediment transport rate per unit width of slope,  $\sigma$ , is the product of discharge per unit width of slope and concentration:

**Equation 4**

$$\sigma = k_1qC_t.$$

Combining Equation 1 and Equation 3 and Equation 4:

**Equation 5**

$$\sigma = k_1qVM_t.$$

The velocity of flow is a function of  $q$  dependant on the slope and flow resistance characteristics. The function has been empirically defined in three commonly used equations: the Darcy-Weisbach Equation, the Chezy Equation, and the Manning Equation. All are merely rating equations which specify the relative response of  $V$  and flow depth,  $D$ , to increasing  $q$ . Moore and Burch (1986) use the Manning Equation. Here, the Darcy-Weisbach equation is preferred because it is dimensionally-consistent and the flow resistance term,  $f$ , varies predictably with Reynold's number ( $Re$ ).

The Darcy-Weisbach equation is:

**Equation 6**

$$V^2 = \frac{8gDM}{f}.$$

Where  $g$  is acceleration due to gravity. The flow depth is given by  $D = q / WV$ , where  $W$  is the width of overland flow. Therefore:

**Equation 7**

$$V^3 = \frac{8g\sigma M}{f}$$

The friction factor varies with  $Re$  by:

**Equation 8**

$$f = k_2 Re^c$$

where by theory and experiment  $c=-1$  for laminar flow and  $c=-0.25$  for fully-turbulent flow over a planar rough bed. Experimentally,  $c$  covers this full range, and beyond -1, for overland flow on natural surfaces. For Hereafter we assume that  $c=-1$ . Noting that  $Re = VD / \nu$ , where  $\nu$  is kinematic viscosity, and that  $VD = q$ , then Equation 7 becomes:

**Equation 9**

$$V = k_3 q^{2/3} M^{1/3}.$$

Substituting Equation 9 into Equation 5 gives:

$$\sigma = k_4 q^{5/3} M^{4/3}$$

which is merely one of a set of sediment transport equations of the form:

**Equation 10**

$$\sigma = kq^a M^b$$

where  $a$  and  $b$  are constants. Review of the literature and use of flow resistance relationships indicate that  $a$  is in the range 1.0-2.3 and  $b$  is in the range 0.7-2.0. There is some evidence that  $a$  and  $b$  are inversely related are not fully independent. To interpret such an equation in a DEM still requires some function to generate run-off,  $q$ . A spatially-uniform rainfall excess per unit area,  $r$ , is usually assumed:

**Equation 11**

$$q = Ar$$

where  $A$  is the upslope catchment area per unit width of slope. This assumption is not necessary and a more general sediment transport capacity equation is:

**Equation 12**

$$\sigma = kA^\alpha M^\beta$$

where  $\alpha$  is not necessarily equal to  $a$  and  $\beta$  is not necessarily equal to  $b$ . For example, an argument can be made that sediment travels only a short distance in a single event, independent of  $q$  so that  $\alpha=0$ .

Equation 12 predicts the transport capacity of sediment across a landscape element for a set of run-off, cover, land-use, sediment and soil properties incorporated into  $k$ . It does not predict erosion or deposition on the landscape element. Erosion occurs when the transport capacity increases with distance down slope, and deposition occurs when it decreases. Thus, in a finite difference approach, moving from one grid cell to another, mass balance states that the net sediment exchange rate,  $S$ , where erosion is positive and deposition is negative, is given by:

**Equation 13**

$$S_i = \sigma_{out} - \sigma_{in}$$

where  $\sigma_{in}$  is the sum of the transport capacities from cells flowing into a given pixel,  $i$ , and  $\sigma_{out}$  is the sum of the transport capacities to cells into which cell  $i$  flows.

Equation 13 can also be written in a differential form in which  $S$  is a function of slope and curvature attributes of the pixel,  $i$ . That is:

**Equation 14**

$$S_i = \frac{\partial \sigma}{\partial x}$$

**Equation 15**

$$M = \frac{\partial z}{\partial x}$$

Rewriting Equation 12, recognising that  $A$  is a function of  $x$ , where  $x$  is the direction of the flow path:

$$\sigma = kA_x^\alpha \left( \frac{\partial z}{\partial x} \right)^\beta$$

differentiating  $\sigma$  with respect to  $x$ :

**Equation 16**

$$S_i = kA_x^{\alpha-1} \left( \frac{\partial z}{\partial x} \right)^{\beta-1} \left( \beta A_x \left( \frac{\partial^2 x}{\partial z^2} \right) + \alpha \left( \frac{\partial z}{\partial x} \right) \left( \frac{\partial A_x}{\partial x} \right) \right).$$

Equation 16 is the subject of a future implementation not further described in tis document.

### 3. The one-dimensional transporting energy budget

For each pixel,  $i$ , along the flow path the net sediment exchange rate,  $S$ , (Equation 13) is computed from the incoming and outgoing total sediment transport rate per unit width of slope,  $\sigma$  (hereafter referred to as transport power). The first pixel in the path (A in Figure 1) has incoming transport power of zero. The outgoing transport power from A is given by:

**Equation 17**

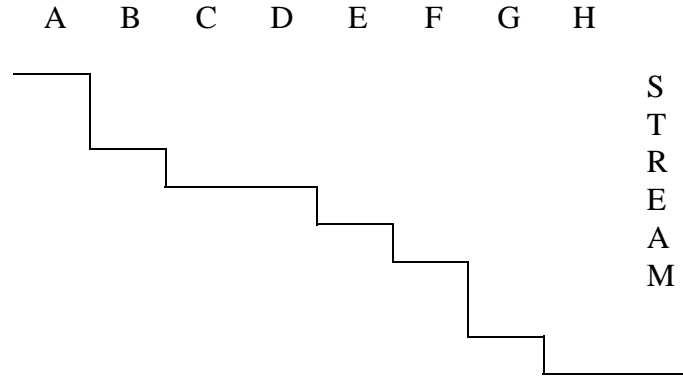
$$\sigma_{A,out} = A_A^\alpha M_{AB}^\beta \text{ with erosion (positive net sediment balance) from A of } S_A = 0 - A_A^\alpha M_{AB}^\beta$$

At pixel B the incoming transport power is equal to the outgoing transport power from the previous pixel, A. Subsequent pixels are dealt with reference to the pixels above and below in a similar fashion. The outgoing transport power is given by:

**Equation 18**

$$\sigma_{B,out} = (A_A + A_B)^\alpha M_{BC}^\beta$$

with net sediment balance from B of  $S_B = [(A_A + A_B)^\alpha M_{BC}^\beta] - (A_A^\alpha M_{AB}^\beta)$ .



**Figure 1. A one-dimensional hill slope example.**

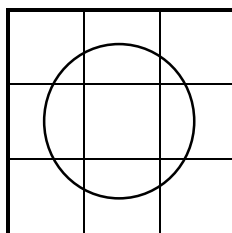
## 4. The two-dimensional transporting energy budget

### 4.1. The flow routing algorithm

In the two-dimensional (2D) case, the situation is more complex. A pixel can deliver sediment to one or more downhill neighbours.

A decision has been made that flow algorithms that consider only the vertical and horizontal neighbours and "fastest descent" algorithms (D8 and rho8) are not sufficiently realistic to provide acceptable flow approximation. Therefore, the neighbours of a pixel are the eight surrounding pixels; four touch at edges and four touch at vertices. Flow can potentially occur to any or all (assuming sink points can occur in the DEM which should be rare) of the neighbours. This presents a problem for flow because neighbours on the diagonal have flow through zero width, i.e., a point. Two approaches will be implemented to deal with this problem.

The first, and simplest, approach adopted is to define the width of flow,  $W$ , between pixels as shown in Figure 2. Flow between pixels is considered to occur across a width between them proportional to the circumference of a circle with radius equal to 1 pixel unit. Half the proportion of the circle circumference falls across each diagonal neighbour as across each horizontal and vertical neighbour. Whilst simple to implement, and therefore having some appeal, this approach is somewhat unrealistic.



**Figure 2. Estimation of width of flow**

The distance between the centre pixel and horizontal and vertical neighbours,  $d$ , is  $1.0L$ , where  $L$  is the side length of a pixel, whereas the distance between the pixel and diagonal neighbours is  $L\sqrt{2}$ .

The second approach is to transform the underlying grid so that zero flow width is avoided. A way to do this is to employ a common technique in image processing, particularly in mathematical morphology, i.e., use a hexagonal grid. The hexagonal grid has three advantages over the square grid:

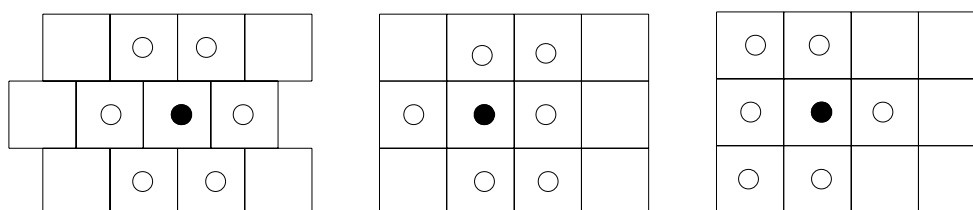
1. the flow width between elements is equal for all neighbours;
2. the distance between the centres of pixels is the same in all directions;

- fewer pixels are required to represent the neighbourhood and fewer pixels are needed to represent the image.

The disadvantages of the hexagonal grid are that:

- people are not accustomed to using it;
- the DEM has to be transformed by interpolation to produce a good approximation;
- some pixels at the edges of the catchment are lost in the interpolation procedure;
- the resolution of the DEM is slightly decreased (although this could be compensated for in generating the DEM especially if using something like ANUDEM).

It is considered here that the potential benefits are sufficient to test the hexagonal grid approach against the square implementation on well-established test surfaces, e.g., a planar slope and a cone.



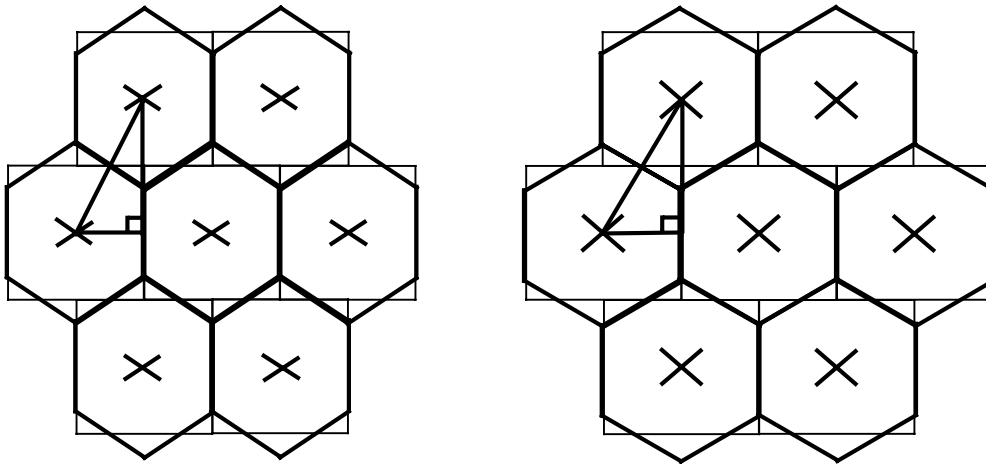
**Figure 3. Offset selection of neighbours to simulate a hexagonal grid. Left: the conceptual arrangement; middle: the neighbours selected when the pixel,  $i$ , is on an odd-numbered line; right: the neighbours selected when the pixel,  $i$ , is on an even-numbered line.**

Often the hexagonal grid is approximated by selecting a different set of neighbours from the square grid on alternating lines as shown in Figure 3.

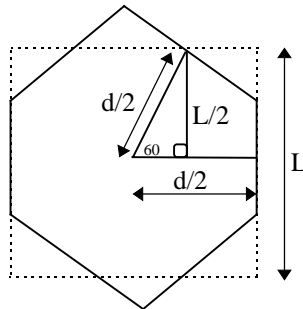
The approximation in Figure 3 is insufficient for flow routing. Previous testing using this procedure has demonstrated artefacts. Therefore, rather than using a conceptual shift of each second line we interpolate between adjacent pixels on every odd line. This provides for continuity of flow paths. However, a single pixel is sacrificed at the end of each row because there is no subsequent pixel to interpolate against. This interpolation provides a hexagonal grid from squares.

Figure 4 (left) shows the result of such an interpolation. The unit cell for flow consisting of the central pixel and its six neighbours are shown. It is evident that the underlying square grid produces a hexagonal flow grid that is distorted. It is necessary, therefore, to change the initial interpolation to account for this. Dealing with the hexagon regularity issue independently of the first interpolation to shift positions, solution of the triangle in Figure 5

shows that the interpolation in the x-dimension to provide regular hexagons is  $d = \frac{2}{\sqrt{3}}L$  where  $d$  is the distance between hexagon centres and  $L$  is the side length of the original square pixel. This grid provides the regular flow grid shown in figure 4 (right).



**Figure 4. . Left: following interpolation of the values on every other (odd) line hexagons are formed but are distorted by the underlying squares; right: following a second x-axis interpolation regular hexagons are formed.**



**Figure 5. . Interpolation to produce regular hexagons.**

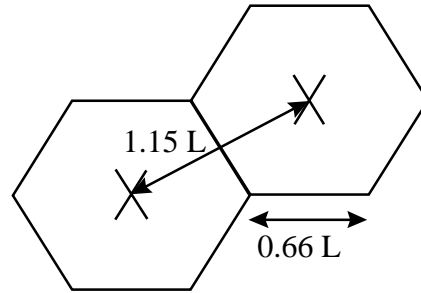
A local neighbourhood from the resultant flow field is shown in Figure 6. This figure illustrates the intuitive appeal of the hexagonal grid.

Two further attributes are required by SEDUM: *viz.* the flow width,  $W$ , and the area of one pixel,  $a_p$  (Figure 5).

**Equations 19**

$$W = 0.5d / \cos 30 = \frac{2L}{3} \approx 0.66L \quad \text{and}$$

$$a_p = \frac{W}{4} \times \frac{2}{2\sqrt{3}} L \times 12 = \frac{2}{\sqrt{3}} L^2 \approx 1.15L^2$$



**Figure 6. . The dimension of the hexagonal pairs in relation to the side length of the original square grid; L is the side length of one pixel in the square grid.**

*N.B. The initial versions of SEDUM will NOT implement hexagonal flow routines.*

#### **4.2. Definitions of flow path and critical variables**

The slope between two pixels is defined as:

**Equation 20**

$$M_{i,j} = \sin \left( \frac{|E_i - E_j|}{d} \right)$$

where  $E$  is elevation and  $d$  is the distance between pixel centres, and subscripts  $i$  and  $j$  refer to the pixel and the neighbour, respectively. The slope between pixels of the same elevation is given an arbitrarily small value of  $\sin \theta = 0.01$ .

Depending on the option used for flow width the appropriate distances are used in the slope calculations (Equation 20).

Two attributes are required to define flow paths. First, the distance from each pixel to the nearest downhill delivery point (channel pixel) is computed. This distance is derived from the uphill dilation distance transform. This provides a unique, one-to-one, association of distance and delivery. Thereby, the “catchment” areas of each delivery point can be identified.

Uphill neighbours are those whose elevation is greater than that of the pixel or whose elevation is the same but whose distance from the nearest channel delivery point is greater. Downhill neighbours are those whose elevation is less than the pixel or whose elevation is the same but whose distance to the channel delivery point is shorter. Pixels of the same elevation which are the same distance to the nearest delivery point are arbitrarily identified as downhill neighbours.

The uphill and downhill neighbours thus defined provide the local flow path members. A pixel can be a member of more than one flow path.

In the case of the square grid, a version of the FD8 flow algorithm is used where the allocation of values from  $i$  to the  $j$ th neighbour is weighted by  $M_{i,j}$  and the flow width between  $i$  and  $j$ . Equation 21 shows the derivation of a weighting factor for each neighbour.

**Equation 21**

$$\omega_j = \frac{(M_{i,j} \times W_{i,j})}{\sum_{j=1}^N (M_{i,j} \times W_{i,j})} \text{ and } \sum_{j=1}^N \omega_j = 1$$

Where  $N$  is the number of downhill neighbours and  $W$  is the flow width.

In the hexagonal case the  $W_{i,j}$  are constant and the weighting simplifies to:

**Equation 22**

$$\omega_j = \frac{M_{i,j}}{\sum_{j=1}^N M_{i,j}} \text{ and } \sum_{j=1}^N \omega_j = 1$$

with the additional advantage that arbitrary decisions regarding flow width are avoided.

### 4.3. The transport algorithm

The incoming transport power to a pixel,  $\sigma_{i,in}$ , is the sum of the outgoing transport power from each uphill neighbour to  $i$ . However, it is not necessary to explicitly compute this value; the algorithm for net sediment balance at each pixel follows. At first reading, the algorithm may seem counter-intuitive in that computation is focussed on distributing transport power downhill. The uphill direction is never explicitly examined.

1. The digital elevation model is sorted from highest to lowest and by distance to the nearest channel for adjacent pixels with the same elevation.
2. Commencing at the highest pixel, and visiting each pixel in sorted order...
  - i. increment the upslope area of pixel  $i$  by the area of a single pixel;

- ii. retrieve the downhill neighbours,  $j$ , of the current pixel,  $i$ ;
- iii. redistribute the transport power - for each downhill neighbour,  $j$ ;
  - a. compute the portion of the upslope area of  $i$  to be distributed to  $j$  and increment the upslope area of  $j$  by that quantity;
  - b. initialise the outgoing transport power of  $i$  to zero;
  - c. compute the transport power contribution to  $j$  from  $i$ ; and increment the outgoing transport power of  $i$  and the incoming transport power of  $j$  by that amount.
- iv. compute the net sediment budget for  $i$  and record it;
- v. propagate the deposition information.

Detail of each step of the algorithm follows.

1. The digital elevation model is sorted from highest to lowest using (not-so) quicksort.
2. Commencing at the highest pixel, and visiting each pixel,  $i$ , in elevation order...
  - i. increment the upslope area of pixel  $i$  by the area of a single pixel,
 
$$A_i = A_i + a_p$$
 where  $A$  is the upslope area and  $a_p$  is the area of one pixel (assumed constant for the DEM image);
  - ii. retrieve the downhill neighbours,  $j$ , of the current pixel,  $i$ ;
  - iii. for each downhill neighbour,  $j$ ,
    - a. compute the portion of the upslope area of  $i$  to be distributed to  $j$  and increment the upslope area of  $j$  by that quantity:

**Equation 23**

$$A_j = A_j + (A_i \omega_j)$$

- b. initialise the outgoing transport power of  $i$  to zero,  $\sigma_{out} = 0$ .  
(Note  $\sigma_{out}$  is not an image but a single variable re-used at each pixel);
- c. compute the transport power contribution to  $j$  from  $i$ ; and increment the outgoing transport power of  $i$  and the incoming transport power of  $j$  by that amount:

**Equation 24**

$$\sigma_{i,j} = (A_i \omega_j)^\alpha M_{i,j}^\beta$$

$$\sigma_{j,in} = \sigma_{j,in} + \sigma_{i,j}$$

$$\sigma_{out} = \sigma_{out} + \sigma_{i,j}$$

iv. compute the change in energy transport for  $i$  and record it:

**Equation 25**

$$S_i = \sigma_{out} - \sigma_{i,in}$$

- v. propagate the deposition path length information:  
 if deposition occurs at  $i$ , i.e., if  $S_i < 0$   
 if  $\delta_i = 0$  set  $\delta_i = 1$   
 Record the deposition that did occur at  $i$  and the value of  $\delta_i$  - then  
 for each downhill neighbour,  $j$   
 $\delta_j = \text{MAX}(\delta_j, \delta_i)$

**4.4. Sediment sorting and nutrient transport**

An important factor in distinguishing sediment transport from nutrient transport is the size distribution of material delivered to the channel compared to the size distribution at the point (pixel) of erosion. Adsorbed nutrients (particularly phosphorus) tend to be more concentrated on finer sediment because of the greater surface area per unit mass of soil and, in most soil material, the increased chance that fine material contains more reactive clay surfaces (here we distinguish that component of clay as defined by particle diameter from that which is mineralogically active). The transport of nutrients on large organic matter and in solution are not considered here.

The amount of nutrient associated with hillslope soil depends on its size distribution, its parent material and land use history. In this initial version of SEDUM, a single nutrient can be dealt with at one time. The nutrient concentration is supplied at each pixel. The nutrient concentration changes are tracked as sediment is mixed and sorted during transport using the same equations for weighting and mixing as used for sediment. In addition, the initial fineness of material is estimated for each pixel from a soil texture data layer. It is presumed that this layer will be appropriately scaled to represent a median particle diameter of the material that is sorted. There exists reasonable evidence in the (limited) literature that the aggregate size distribution remains effectively constant and that one can estimate the sorting effect by specifying a proportion of the material travelling as particles and apply sorting (and nutrient concentration magnification) to that fraction.. Future versions will consider the use of a simple parametric particle (aggregate?) size distribution function.

The effects of land use will be treated as part of risk assessment rather than hazard rating<sup>1</sup>.

The principal process involved in sorting eroded sediment is deposition. When net deposition occurs (Equation 13) we assume, albeit simplistically, that coarse material is preferentially deposited. Therefore, the sediment associated with  $\sigma_{in}$  is coarser than that which leaves the pixel associated with  $\sigma_{out}$ . We defined an index, the sediment coarseness index,  $\phi$  [0,1], which identifies the fineness of sediment in each pixel with respect to the source soil. For the underlying soil of caoirsest texture,  $\phi=1$ . As the sediment becomes finer through deposition  $\phi$  approaches 0.

Four processes are recognised:

i. Net deposition

In the case of net deposition the coarseness decreases in proportion to the amount of net deposition with respect to the incoming transport power, i.e.,

**Equation 26**

$$\phi_{out} = \phi_{in} \left( \frac{\sigma_{out}}{\sigma_{in}} \right)$$

ii. Net erosion

The coarseness index increases in proportion to the addition to the flow from erosion at pixel,  $i$ . The coarseness index is given by:

**Equation 27**

$$\phi_{out} = \phi_{in} \left( \frac{\sigma_{in}}{\sigma_{out}} \right) + \phi_e \left[ 1 - \left( \frac{\sigma_{in}}{\sigma_{out}} \right) \right]$$

where  $\phi_e$  is the coarseness index for the underlying soil and initially  $\phi_e=1$ .

iii. Sediment mixing.

At each pixel sediment can be sourced from any (and perhaps all but one) neighbours. It is not possible to compute the coarseness index until all neighbours have donated their sediment. Therefore, a small temporary list is kept for each pixel which is initialised when the pixel first receives sediment and which is freed once the transport power is resolved. Sediment is mixed in a similar fashion to the weighting used to partition area from a pixel to its down hill neighbours (refer Equation 21 and Equation 22). The sediment coarseness weighting factor for each neighbour,  $\gamma_n$ , is given by:

---

<sup>1</sup> Hazard is defined as the potential for occurrence of a defined event and risk as the likelihood that the hazard will be realised and to what degree.

**Equation 28**

$$\gamma_{i,n} = \frac{\sigma_{i,n}}{\sum_{n=1}^N \sigma_{i,n}} \text{ and } \sum \gamma_{i,n} = 1$$

where N is the number of up hill contributing neighbours. The incoming coarseness index is computed by adding the mixing components using the defined weighting factors according to:

**Equation 29**

$$\phi_m = \sum_{n=1}^N \phi_{i,n} \gamma_{i,n}$$

iv. Sediment fining through the transport process.

This process attempts to describe the decrease in coarseness that may occur as a result of the net erosion process and when  $\Delta\sigma = 0$ . Under both these resultant conditions, deposition may occur. Therefore, there will be a degree of fining because the erosion that replaces the deposited material will be coarser than the sediment added from the underlying soil. Even when net erosion occurs and the coarseness index is increased according to Equation 27 it should also be decreased to account for the deposition component. It is not clear whether this process is sufficiently great in magnitude to influence the result if it is ignored. However, it may play an important role in the trade-off between deposition and the consequent delivery of finer (albeit less) sediment with enriched nutrients and unabated erosion down slopes. A simple approach is to adopt a small constant decrease in coarseness index associated with pixels with net erosion or a balance between erosion and deposition. One pixel with net deposition, the same constant might be added to the coarseness index to represent that component of the net deposition that was erosion. Nothing will initially be implemented for this process.

#### **4.5. The transport algorithm with sediment sorting**

Given the complexity of order of events implied by taking account of sediment coarseness it is necessary to rewrite the sediment transport algorithm as given in section 1.4.3. A major change to the algorithm centres around the need to know some variables earlier than in the previous version of the algorithm. Also, the details of the transport power from uphill neighbours is required. One mechanism for dealing with both these requirements is to define a list for each pixel. We will hereafter call this list the *state* list. The state list maintains the information relating to the interaction between a pixel,  $i$ , and its  $n$  uphill neighbours. The state list contains the following information:

- a.  $A_i$  - the upslope area of the pixel;
- b.  $\sigma_m$  - the incoming transport power;
- c.  $\sigma_{i,j}$  - the incoming transport power inherited from each uphill neighbour,  $j$ ;

- d.  $\phi_{i,j}$  - the incoming sediment coarseness index associated with each uphill neighbour,  $j$ ;
- e.  $\delta_i$  - the deposition path length of the pixel;

In addition to the state list, there are a number of variables which exist only while each pixel is being computed. These variables are neither allocated nor de-allocated, but are merely reused for each pixel. They are:

- a.  $N$  - the number of up/downhill neighbours;
- b.  $O_j$  - the offset of each downhill neighbour,  $j$ ;
- c.  $A_{i,j}$  - the upslope area passed on to each downhill neighbour,  $j$ ;
- d.  $\sigma_{out}$  - the outgoing transport power;
- e.  $\sigma_{i,j}$  - the outgoing transport power passed on to each downhill neighbour,  $j$ ;
- f.  $S_i$  - the change in transport power ( $S_i = \Delta\sigma = \sigma_{out} - \sigma_{in}$ );
- g.  $\phi_{in}$  - the incoming sediment coarseness index;
- h.  $\phi_{out}$  - the outgoing sediment coarseness index;

It is important to be clear that the state list is filled gradually when each pixel is visited when the **downhill** computations are made as in the initial algorithm. It is not completed by examining the uphill neighbours. This avoids unnecessary double computation of variables.

In each case below where it is specified that a value be written this does not force the implementation to write the variable at that time. Rather it indicates that the variable should be written to an image file. This is made explicit because it should be possible to implement this algorithm in minimum memory by holding only the state lists, the DEM and the distance image in memory. All other variables can be held as needed in the state list for the pixel. The advantage of this approach is that the state list is only initialised when a pixel is first the downhill neighbour of another pixel. The state list is freed (no longer occupies memory) once the pixel has been visited.

1. The digital elevation model is sorted from highest to lowest (heapsort) and by distance to the nearest channel for adjacent pixels with the same elevation;
2. Commencing at the highest pixel, and visiting each pixel,  $i$ , in sorted order...
  - i. if the pixel's state list has not been allocated, allocate and initialise it;
  - ii. increment the upslope area,  $A_i$ , of the pixel by the area of a single pixel, and write  $A_i$ ;

- iii. compute the incoming sediment coarseness index,  $\phi_m$ , (Equation 28 and Equation 29); this computation requires the incoming coarseness index,  $\phi_{i,j}$ , and the incoming transport power,  $\sigma_{i,j}$  (state list), from each uphill neighbour,  $j$ ;
  - iv. retrieve the downhill neighbours,  $j$ , of the current pixel,  $i$ ; this resolves  $N$  and  $O_j$ ; all references to neighbours from this point on are to *downhill* neighbours;
  - v. for each downhill neighbour,  $j$ ,...
    - a. compute  $A_{i,j}$ , the portion of the upslope area of  $i$  to be distributed to each downhill neighbour,  $j$ ;
    - b. compute  $\sigma_{i,j}$ , (local variable) the transport power between  $i$  and each downhill neighbour,  $j$ ;
    - c) increment the outgoing transport power,  $\sigma_{out}$ , by  $\sigma_{i,j}$ ;
  - v. compute the change in transport power for  $i$ ,  $S_i$ , and write it;
  - vi. compute the outgoing sediment coarseness index,  $\phi_{out}$ , and write it;  $\phi_{out}$ , is computed by Equation 26 if there was net deposition on the pixel ( $S_i < 0$ ), or by Equation 27 if there was net erosion on the pixel ( $S_i > 0$ ); if  $S_i = 0$ , then  $\phi_{out} = \phi_m$ ;
  - vii. compute the deposition path length,  $\delta_i$ ; if deposition occurs at  $i$  ( $S_i < 0$ ), then  $\delta_i$  is incremented by 1; otherwise  $\delta_i$  becomes 0;
  - viii. for each downhill neighbour,  $j$ ,...
    - a. if neighbour  $j$ 's state list has not been allocated, allocate and initialise it;
    - b. increment neighbour  $j$ 's  $\sigma_{in}$  by  $\sigma_{i,j}$ ;
    - c) increment neighbour  $j$ 's upslope area,  $A_j$  by  $A_{i,j}$ ;
    - d) append  $\sigma_{i,j}$  (local variable) to neighbour  $j$ 's list of incoming transport energies,  $\sigma_{j,i}$  (state list);
    - e) compute the coarseness index passed to neighbour  $j$ , and append it to neighbour  $j$ 's list of incoming coarseness indices,  $\phi_{j,i}$ ;
    - f) set neighbour  $j$ 's deposition path length,  $\delta_j$ , to  $\text{MAX}(\delta_i, \delta_j)$ ;
  - v. free the state list for pixel  $i$ ;
3. Clean up;