

Generalized Terrain Based Flow Analysis of Digital Elevation Models

Kimberly A. T. Schreuders

David G. Tarboton

Daniel W. Watson

Matthew E. Baker

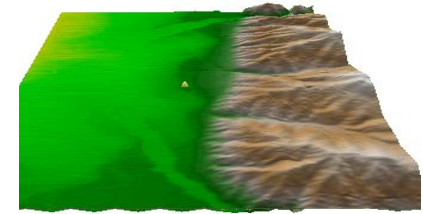
kim.schreuders@usu.edu

<http://www.engineering.usu.edu/dtarb>

UtahState
UNIVERSITY

Hydrologic Terrain Analysis Information Model

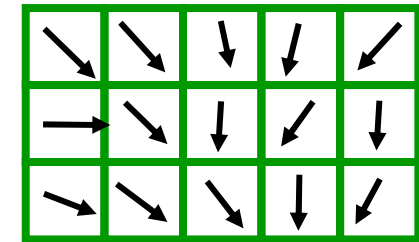
- **DEM**



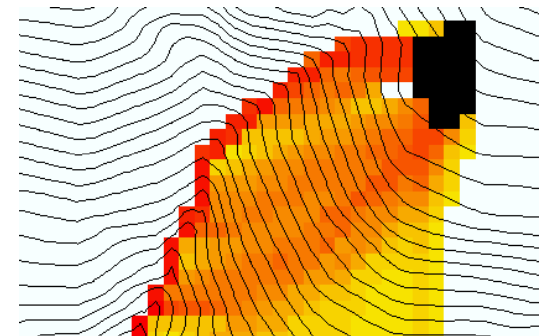
- **Sink Removal**



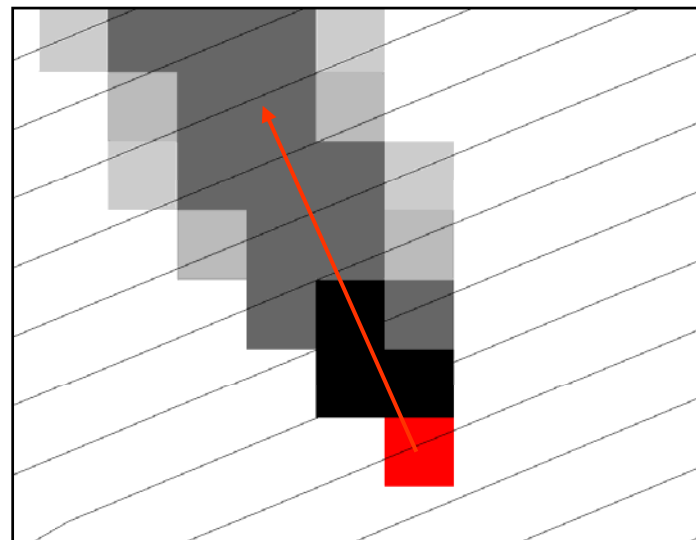
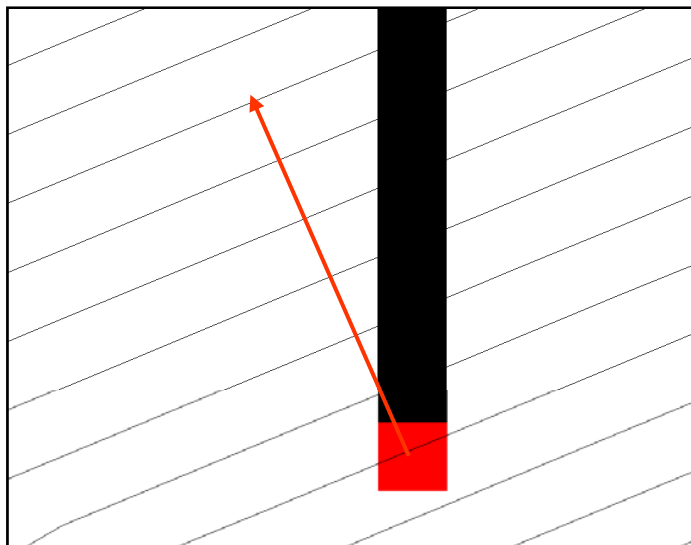
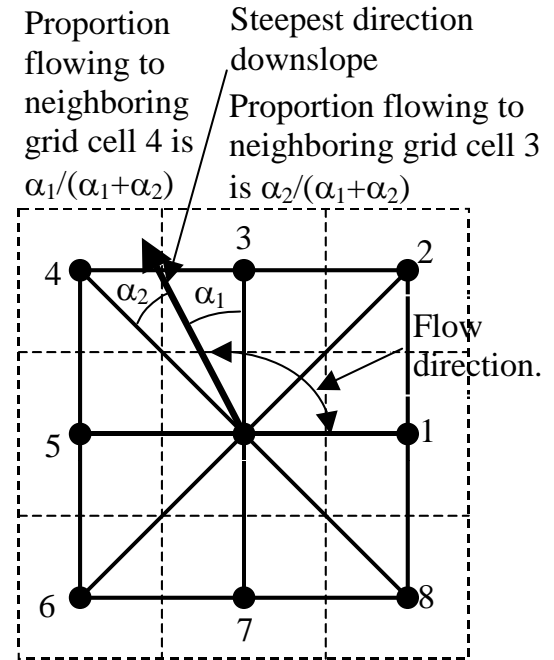
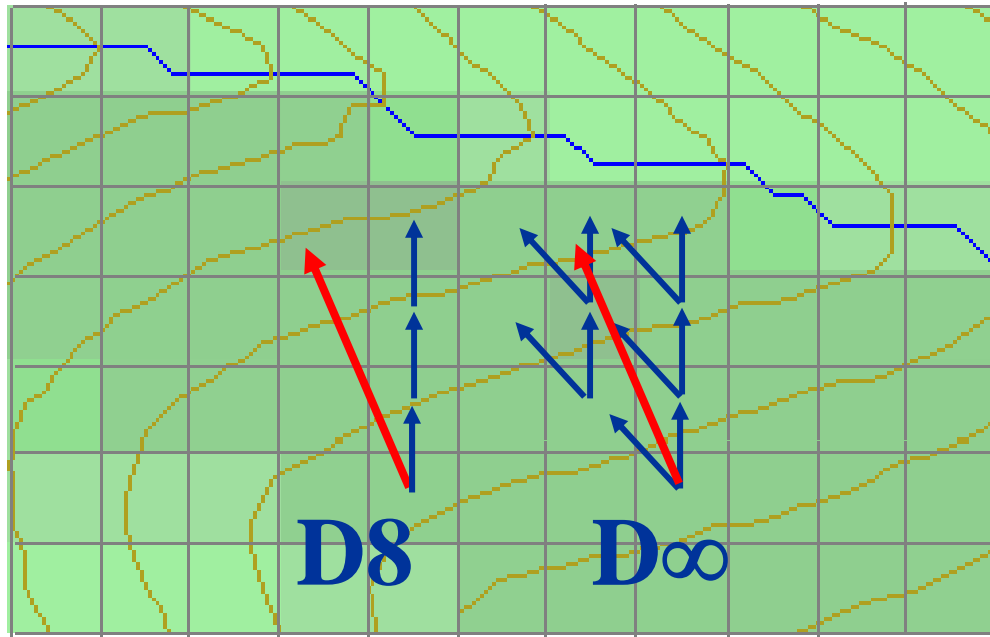
- **Flow Field**



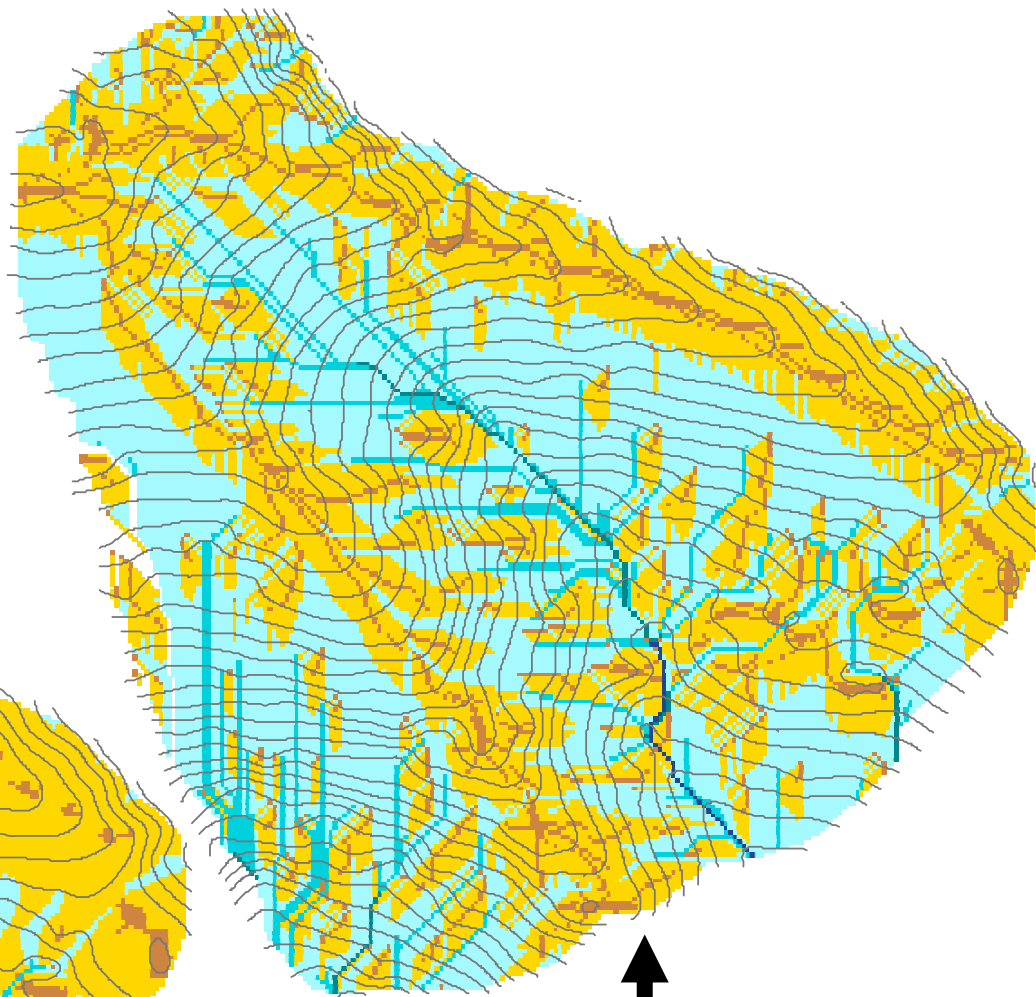
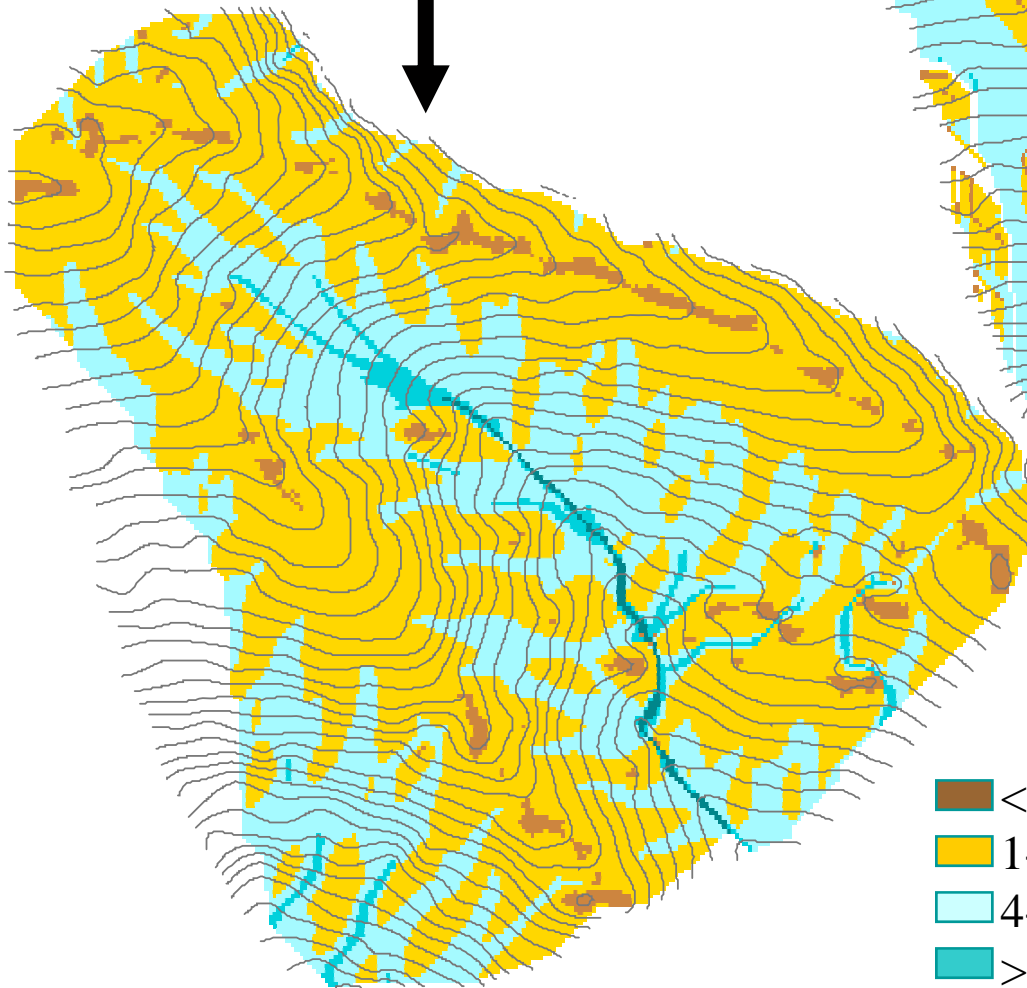
- **Flow Related Terrain Information**







Representation of Flow Field



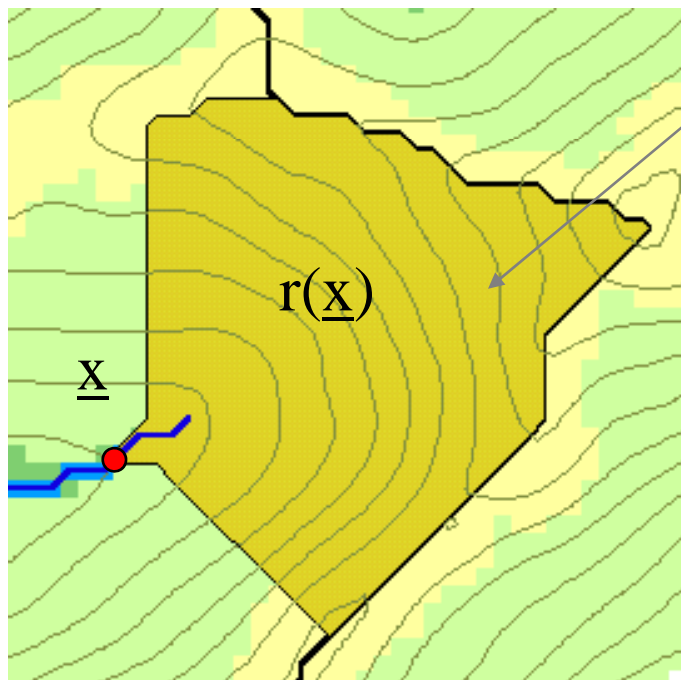
Contributing Area using
 D_{∞}



Contributing Area using
D8

-  <1 ha
-  1-4 ha
-  4-8 ha
-  >8 ha

Flow Algebra



$$A(\underline{x}) = A[r(\underline{x})] = \int_{CA} r(\underline{x}) d\underline{x}$$

$A[.]$ is a functional operator that takes as input a spatial field $r(x)$, and the topographic flow direction field and produces a field $A(x)$ representing the accumulation of $r(x)$ up to each point x .

Numerical evaluation

$$A[r(x)] = A(i,j) = r(i,j)\Delta + \sum_{\text{k contributing neighbors}} p_k A(i_k, j_k)$$

p_k is the proportion of flow from neighbor k contributing to the grid cell (i,j) .

$\sum p_k = 1$ is required to ensure 'conservation'.

Flow directions must not have loops.

General Pseudocode for Upstream Flow Algebra Evaluation

Global variables $\underline{\gamma}$, $\underline{\theta}$, \underline{P}_{ij}

Function **FlowAlgebraUpstream**(\underline{x}_i)

if $\underline{\theta}(\underline{x}_i)$ is known

then

no action

else

for each neighbor location \underline{x}_k indexed by k

if($P_{ki} > 0$)then

call FlowAlgebraUpstream(\underline{x}_k)

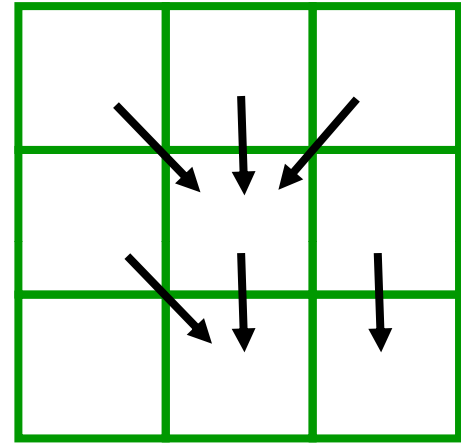
//This is the recursive call to traverse to an upslope neighbor

Next k

// At this point all the necessary inputs are available

Evaluate Algebraic expression $\underline{\theta}(\underline{x}_i) = f(\underline{\gamma}(\underline{x}_i), \underline{P}_{ki}, \underline{\theta}(\underline{x}_k), \underline{\gamma}(\underline{x}_k))$

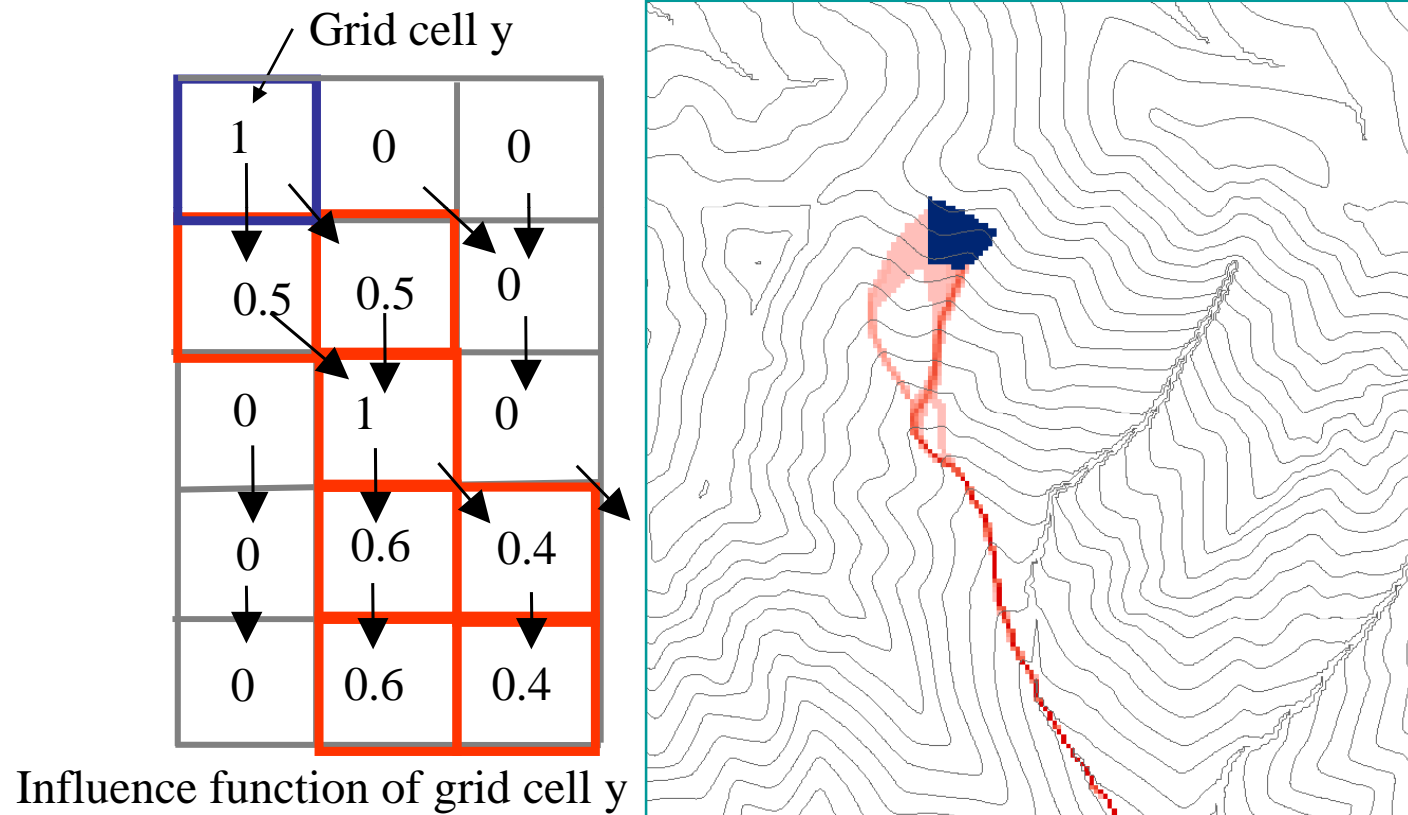
return



Influence function (or influence zone) of y is

$$I(x;y) = A[i(x;y)]$$

The contributing area only of points in the set y . $I(x;y)$ says what the contribution from point y is at point x .



Useful for example to track where sediment or contaminant moves

General Pseudocode for **Downstream** Flow Algebra Evaluation

Global variables γ , θ , P_{ij}

Function **FlowAlgebraDownstream**(\underline{x}_i)

if $\theta(\underline{x}_i)$ is known

then

no action

else

for each neighbor location \underline{x}_k indexed by k

if($P_{ik} > 0$)then

call **FlowAlgebraDownstream**(\underline{x}_k)

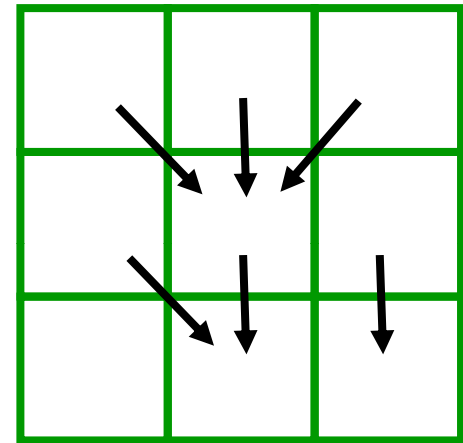
//This is the recursive call to a downstream neighbor

Next k

// At this point all the necessary inputs are available

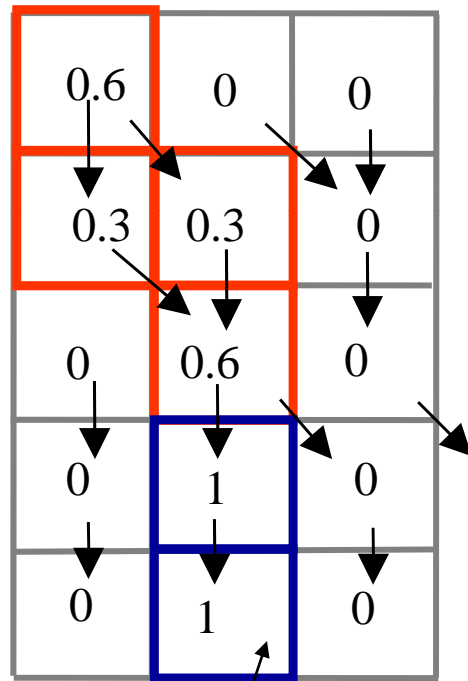
Evaluate Algebraic expression $\theta(\underline{x}_i) = f(\gamma(\underline{x}_i), P_{ik}, \theta(\underline{x}_k), \gamma(\underline{x}_k))$

return



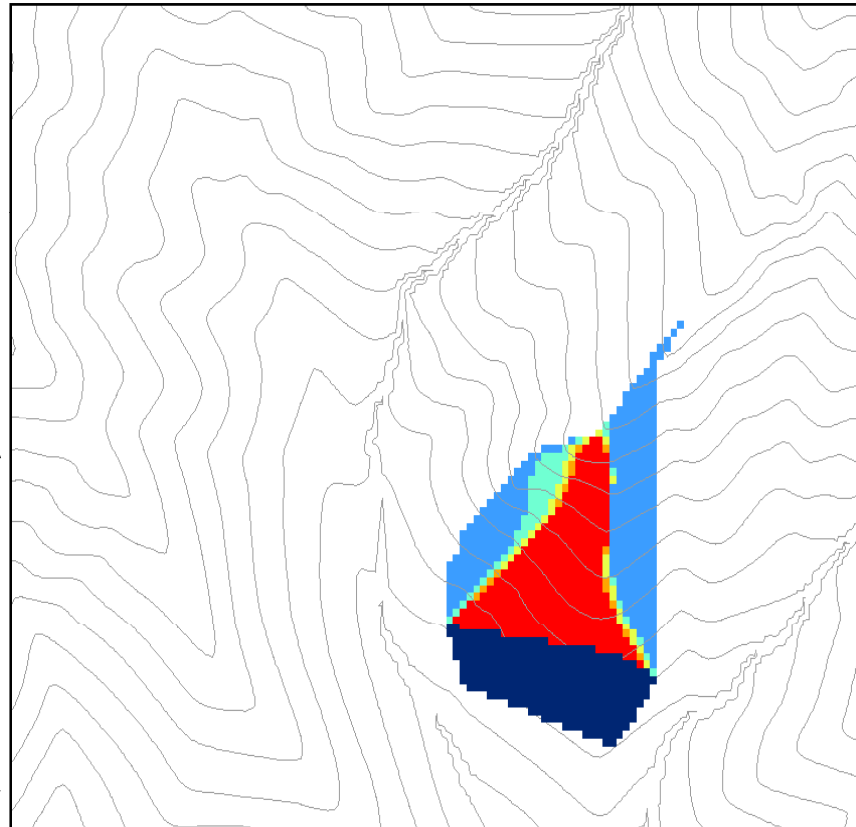
Dependence function. Quantifies the amount a point x contributes to the point or zone y . The inverse of the influence function

$$D(x;y) = I(y;x)$$



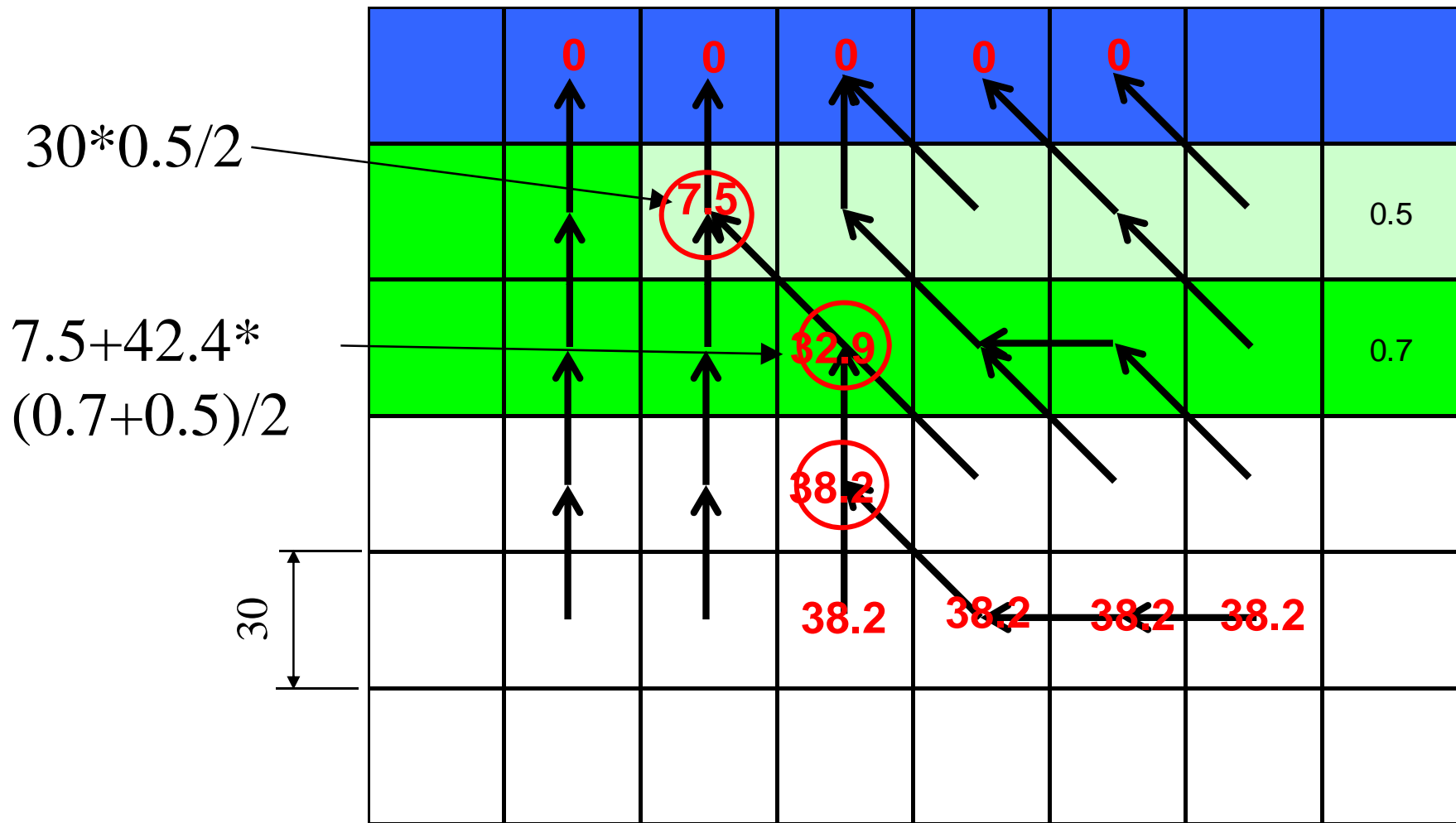
Grid cells y

Dependence function of grid cells y



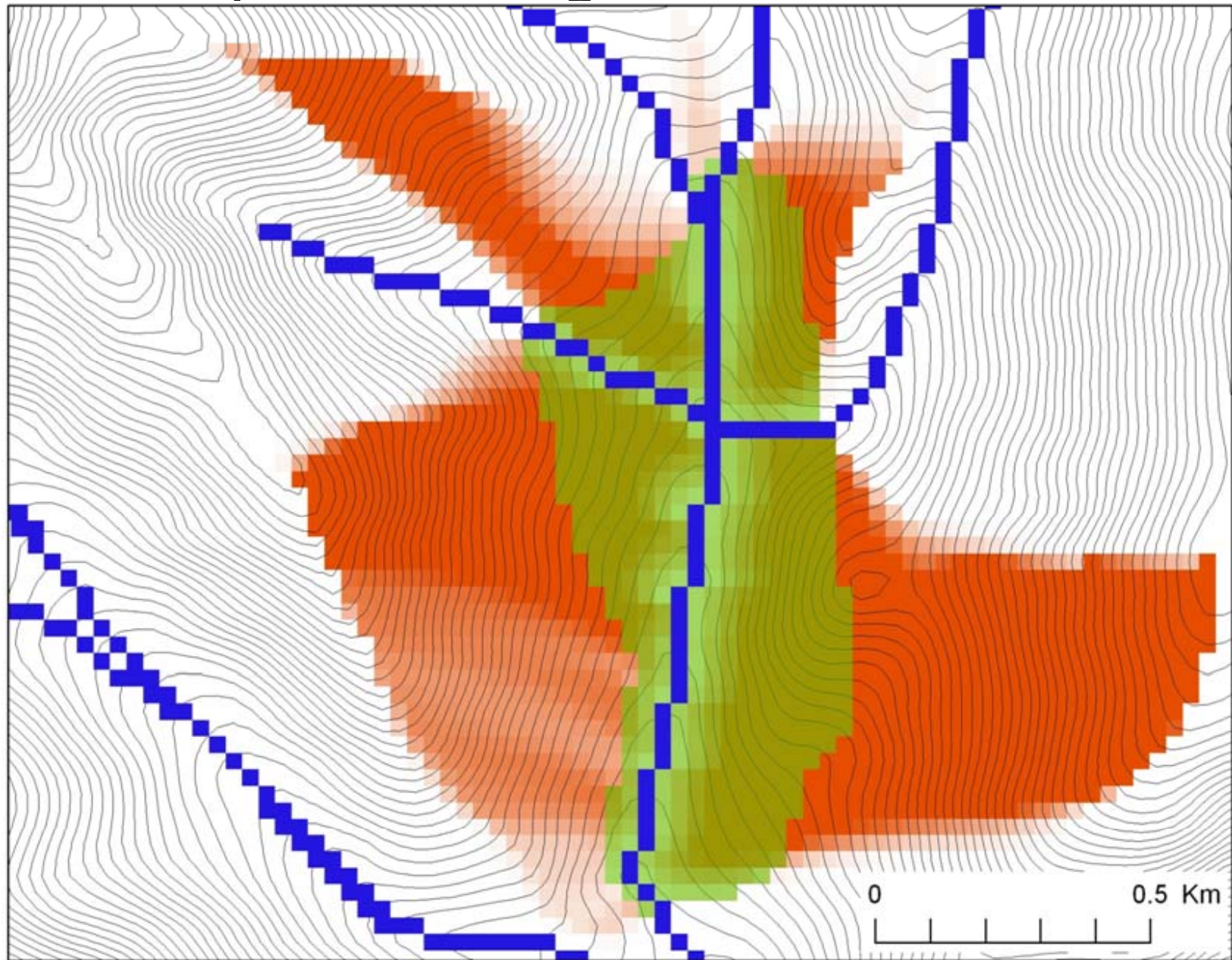
Useful for example to track where a contaminant may come from

Buffer potential weighted distance to stream downslope recursion

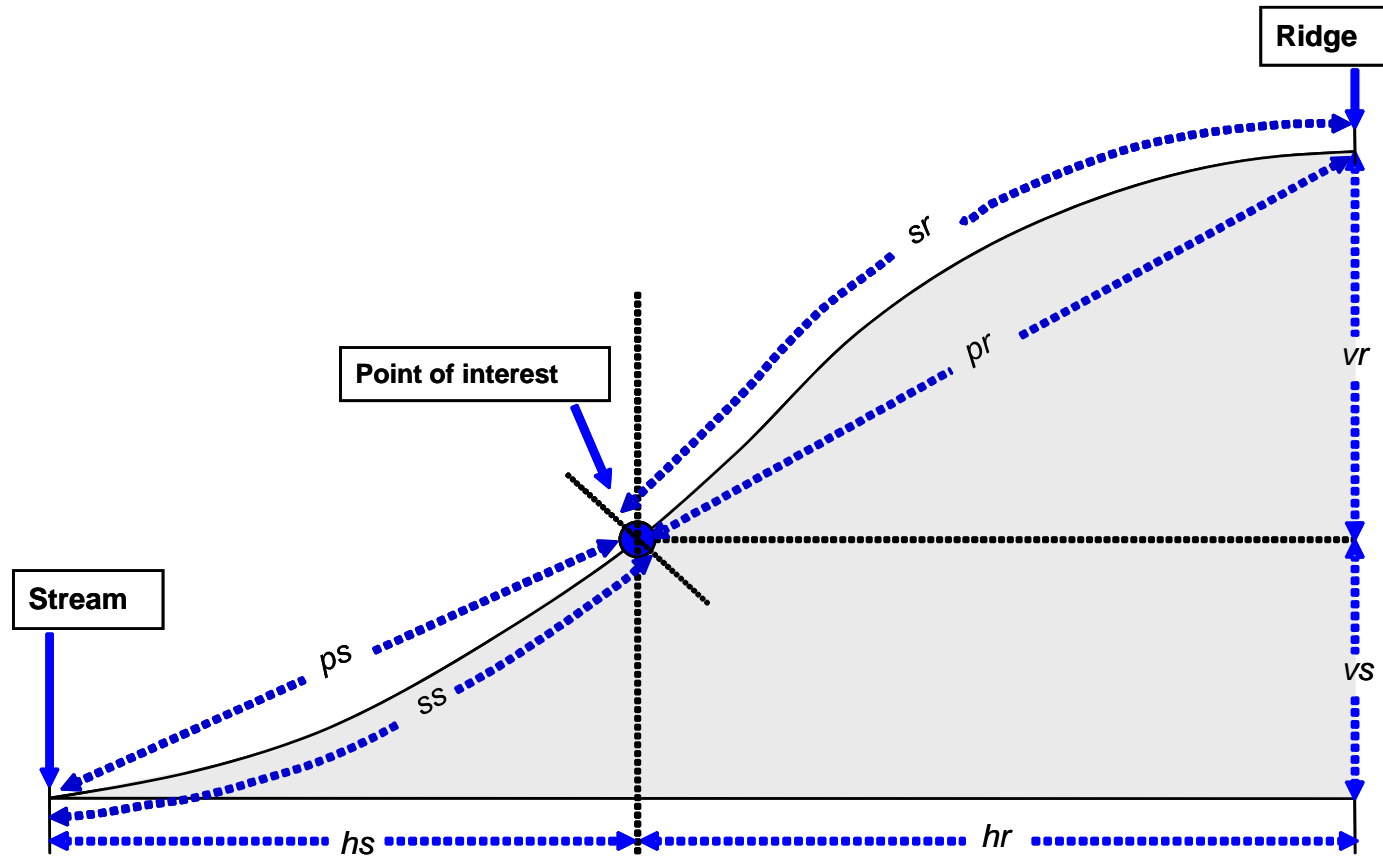


Example to quantify effectiveness of riparian zone sediment capture based on buffer potential

Buffer potential weighted distance to stream

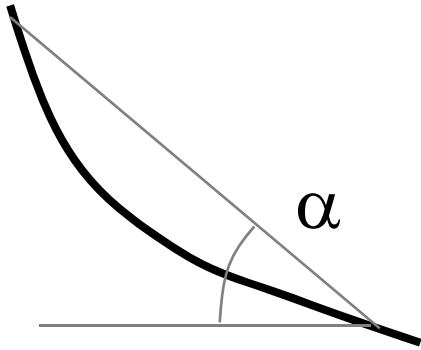


Distance Down and Distance Up

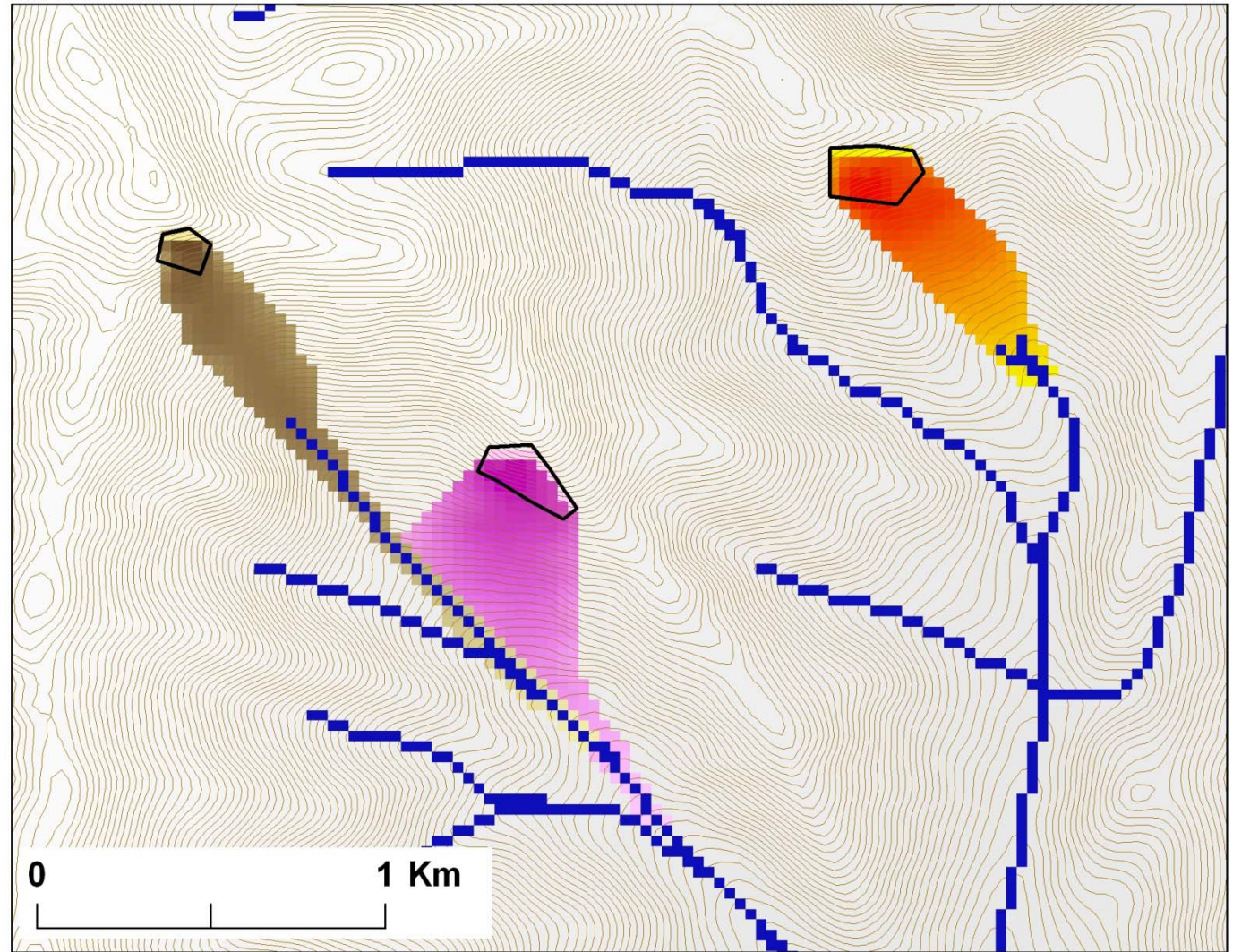


Types of distance measurements possible in distance down and distance up functions.

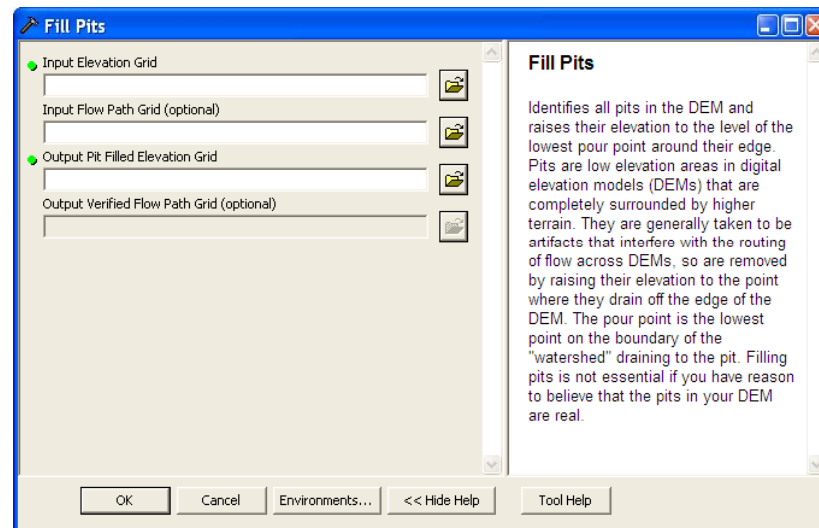
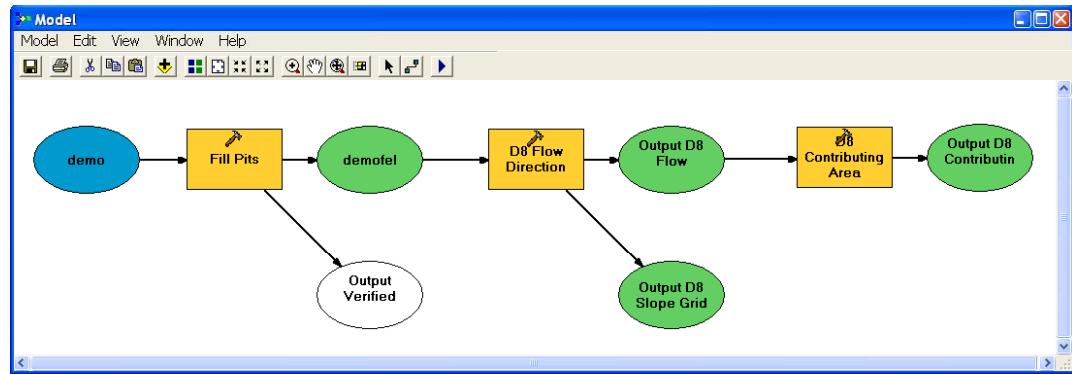
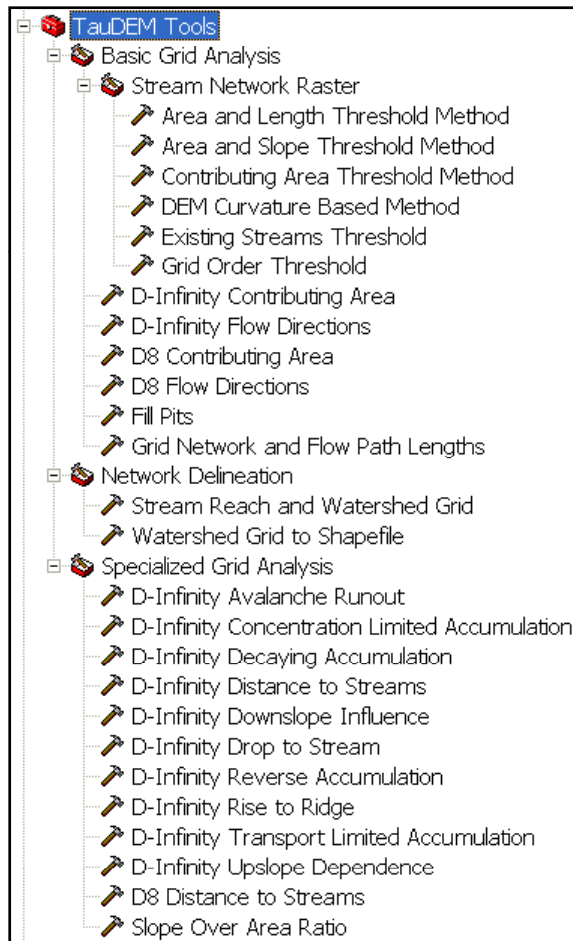
Avalanche Runout



Upslope recursion to determine elevation and distance to point in trigger zone that has the highest alpha angle



TauDEM Toolbox



Serial and Parallel Processing

Serial—General recursive flow algebra algorithm

```
Global  $\underline{P}, \gamma, \underline{\theta}$   
FlowAlgebra(i)  
for all k neighbors of i  
  if  $P_{ki} > 0$   
    FlowAlgebra(k)  
next k  
 $\underline{\theta}_i = \text{FA}(\gamma_i, \underline{P}_{ki}, \underline{\theta}_k, \gamma_k)$   
return
```

Executed by every process with grid flow field \underline{P} , grid dependencies D initialized to 0 and an empty queue Q .

```
FindDependencies(P,Q,D)  
for all i  
  for all k neighbors of i  
    if  $P_{ki} > 0$   
       $D(i) = D(i) + 1$   
  next k  
  if  $D(i) = 0$   
    add i to Q  
next
```

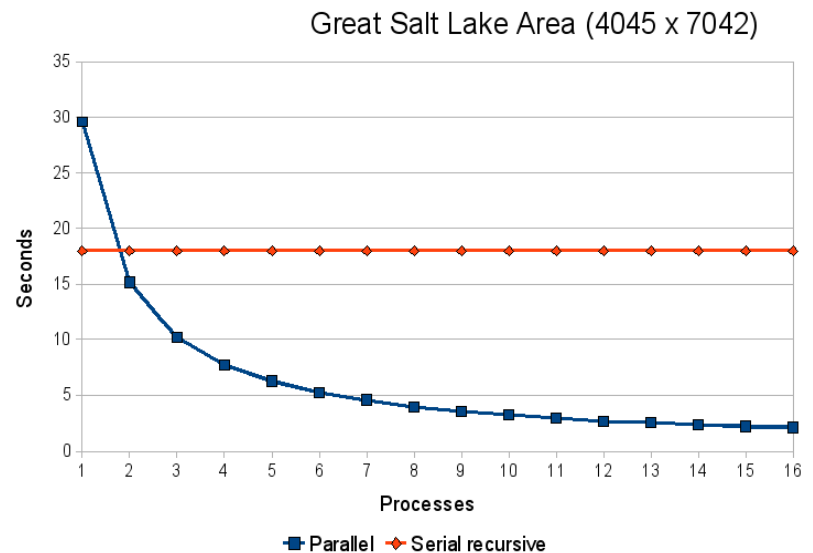
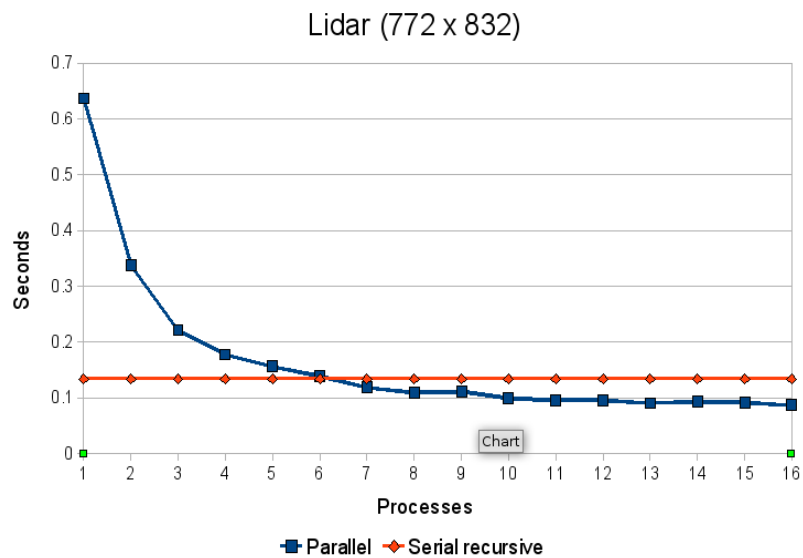
Parallel—Building the dependency grid

Executed by every process with D and Q initialized from FindDependencies.

```
FlowAlgebra(P,Q,D, $\underline{\theta}$ , $\gamma$ )  
while Q isn't empty  
  get i from Q  
   $\underline{\theta}_i = \text{FA}(\gamma_i, \underline{P}_{ki}, \underline{\theta}_k, \gamma_k)$   
  for each downslope neighbor n of i  
    if  $P_{in} > 0$   
       $D(n) = D(n) - 1$   
      if  $D(n) = 0$   
        add n to Q  
  next n  
end while  
swap process buffers and repeat
```

Parallel—Flow algebra function

Serial vs. Parallel Timing



D8 Contributing Area Function

Conclusions

- Terrain based flow data model enriches the information content of digital elevation data
- Flow algebra generalizes the recursive flow accumulation methodology
- Several new flow algebra functions
- New toolbox allows use within model builder
- Parallelization speeds up processing and partitioned processing reduces size limitations
- Methods and software are available in TauDEM at: <http://www.engineering.usu.edu/dtarb>