# TD 603
## Water Resources

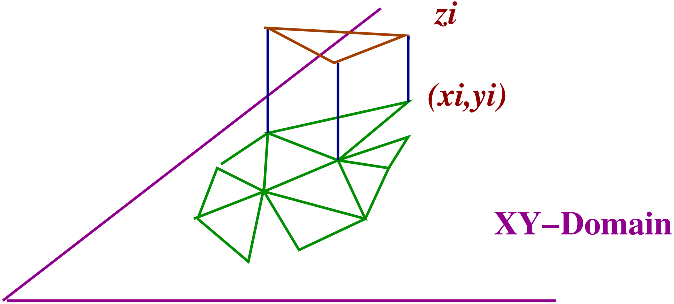
Milind Sohoni
www.cse.iitb.ac.in/∼sohoni/

## Building Numerical Models

# The entities and quantities

- The DTM or DEM: terrain model/elevation model is a representation of the terrain in a particular area.
- Attributes of the DTM-soil, geology, land-use, land-cover.
- Regions-subsets of the DTM, to represent watersheds, flows, etc.
- Functions and Computations: to simulate various physical quantities and their dynamics.
  - Infiltration : the process of rainwater moving down to groundwater.
  - Drainage lines : The development of streams due to rainfall.
  - Surface flow : a composite model of surface flow of water and infiltration which may include time to flow, movement of solids etc.

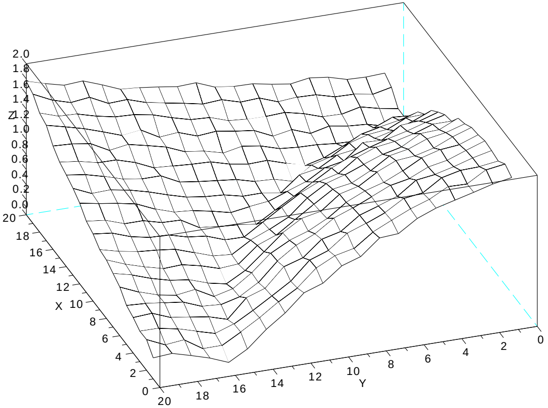
# The Digital Terrain/Elevation model




- Basic grid in the 2D-plane as an index set, *with adjacency*
- association of $z$-values for each point, to define 3D points, edges, triangles and adjacency.

# Grids

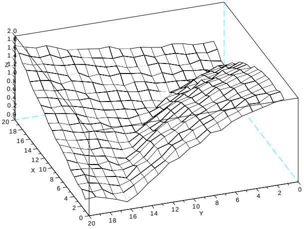
- Grids are where the base division in 2D is a grid of squares with a certain resolutions.
- These are the simplest of the meshes and will serve most purposes.



# Synthetic Grid

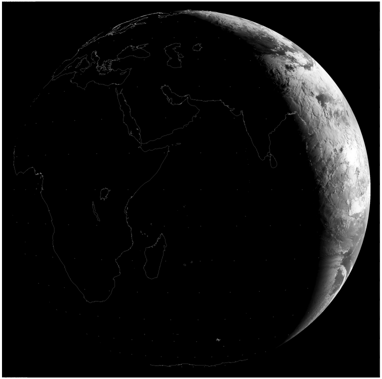
```
function [Z]=makedomain(n);
for i=1:1:n
    for j=1:1:n
     x=i/n; y=j/n;
     diff=1-(x+0.2)^2-(y+0.1)^2;
     Z(i,j)=2*abs(diff)+
           1/(9+abs(diff))*rand(1
            +0.9*y;
    end;
end;
endfunction;
```

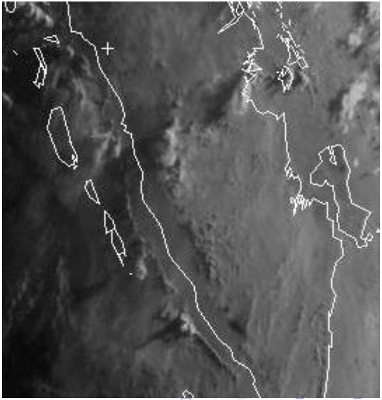
- a circular valley centered at $(-0.2, -0.1)$.
- randomization to simulate reality



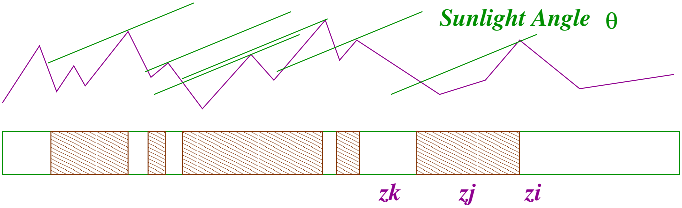
# Real DTMs

- Real DTMs generated mathematically through piecing together various inputs.
- For example, satellite images at different times of the day and observing shadows, gives a procedure for constructing heights.





# The Equations




- Locate the last lit point along a lit interval (in this case, $z_i$).
- For every dark point in the subsequent interval (here $z_j$), we have:

$$(z_i - z_j) \geq (i - j)\tan\theta$$

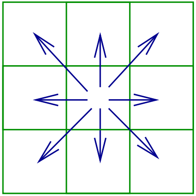
- For every lit point after that (here $z_k$), we have:

$$(z_i - z_k) \leq (i - k)\tan\theta$$

- For such equations for various $\theta$ and solve collectively.

# Lets Compute Drainage

- If a drop falls at point $(i, j)$ then how does it move?
- We consider the 8-neighbours of the point.



***8–way adjacency***

- The flow gets distributed relative to the height differences.

- Here is an example:

| | | |
|---|---|---|
| 1.1 | 1.5 | 1.6 |
| 1.3 | 1.4 | 1.7 |
| 1.3 | 1.3 | 1.5 |

- Hypothesis: Only lower, and in proportion to the height differences.

| | | |
|---|---|---|
| $3/6$ | 0 | 0 |
| $1/6$ | 0 | 0 |
| $1/6$ | $1/6$ | 0 |

- And this continues recursively downwards.

# The `flowout` function

```
function [vv]=flowout(Z,i,j)
// the flow function from a point (i,j) in a domain Z
// output is an array of 3-tuples, (i*,j*,f*), where f*
// is fraction flowing to (i*,j*)
// e.g., [1 1 3/6; 2 1 1/6; 3 1 1/6; 3 2 1/6]
xx=[]; ssum=0;
for ii=-1:2:1 // the X-neighbours
   if Z(i+ii,j)<Z(i,j);
     xx=[xx;(Z(i,j)-Z(i+ii,j)) i+ii j];
     ssum=ssum+(Z(i,j)-Z(i+ii,j));
   end;
end;
```

**and so on**

```
xx=gsort(xx,'lr');
// just so that neighbours in decreasing order
vv=[xx(:,2) xx(:,3) xx(:,1)/ssum];
```

# The drain function

```
function [ZZ,locn]=drain(Z,loc,i,j,f)
// finds the net flowout or local minimas where flow from the
// point (i,j) will go
// ZZ stores the above flowouts and minimas
// locn stores the paths
```

the main part

```
vv=flowout(Z,i,j); // compute the flowouts from (i,j)
for ii=1:1:k // for all the neighbours
   linn=zeros(m,n);
   linn(vv(ii,1),vv(ii,2))=vv(ii,3)*f; // distribute the flows
   [Zx,lx]=drain(Z,linn,vv(ii,1),vv(ii,2),vv(ii,3)*f); // recurse
   ZZ=ZZ+Zx; // do the tally
   loc=loc+lx;
end;
locn=loc;
```