A new method for computing the drainage network based on raising the level of an ocean surrounding the terrain

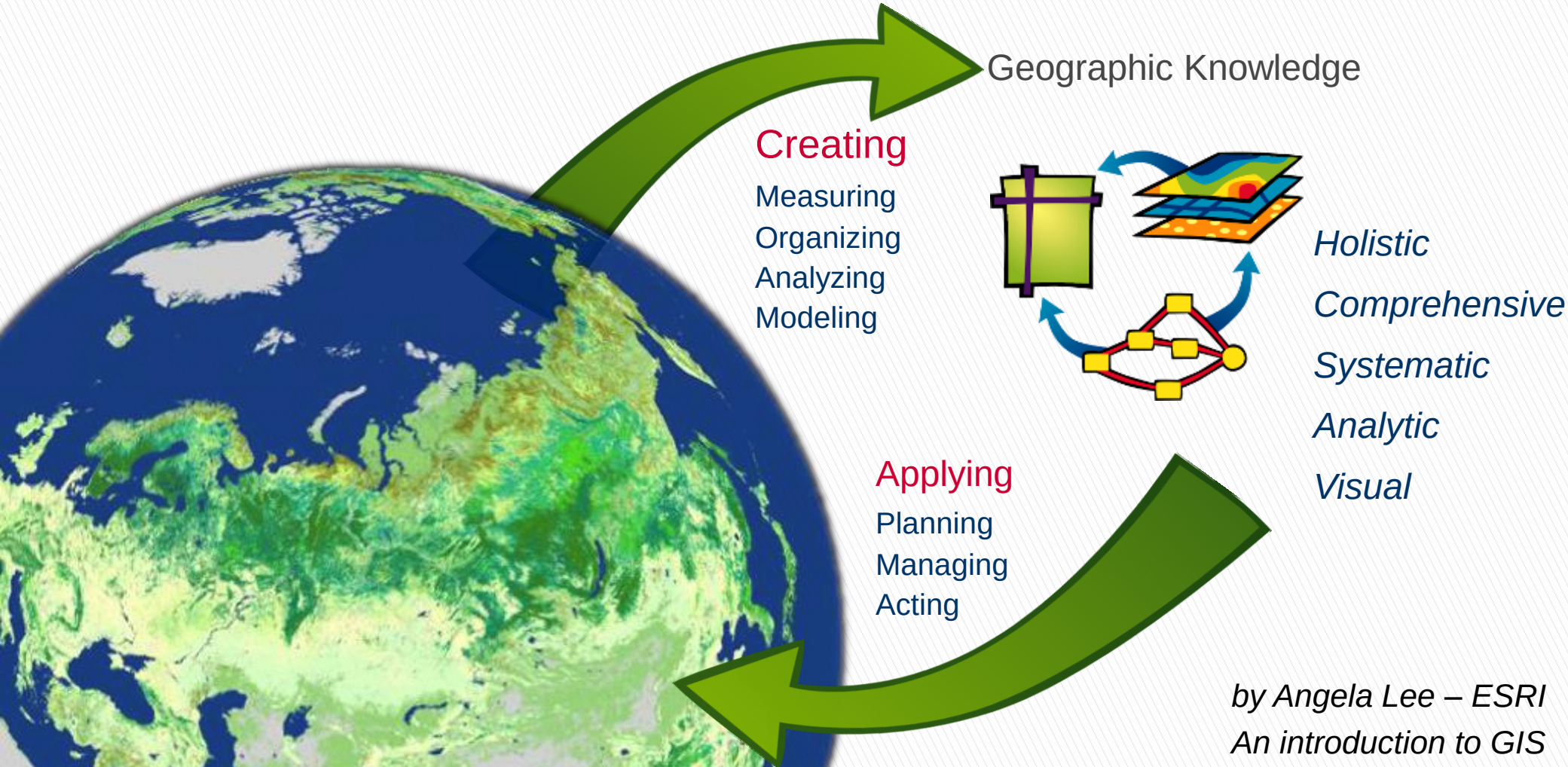
Salles V. G. Magalhães
Marcus V. A. Andrade
W. Randolph Franklin
Guilherme C. Pena

Universidade Federal de Viçosa (UFV) – Brazil
Rensselaer Polytechnic Institute (RPI) – USA

Summary

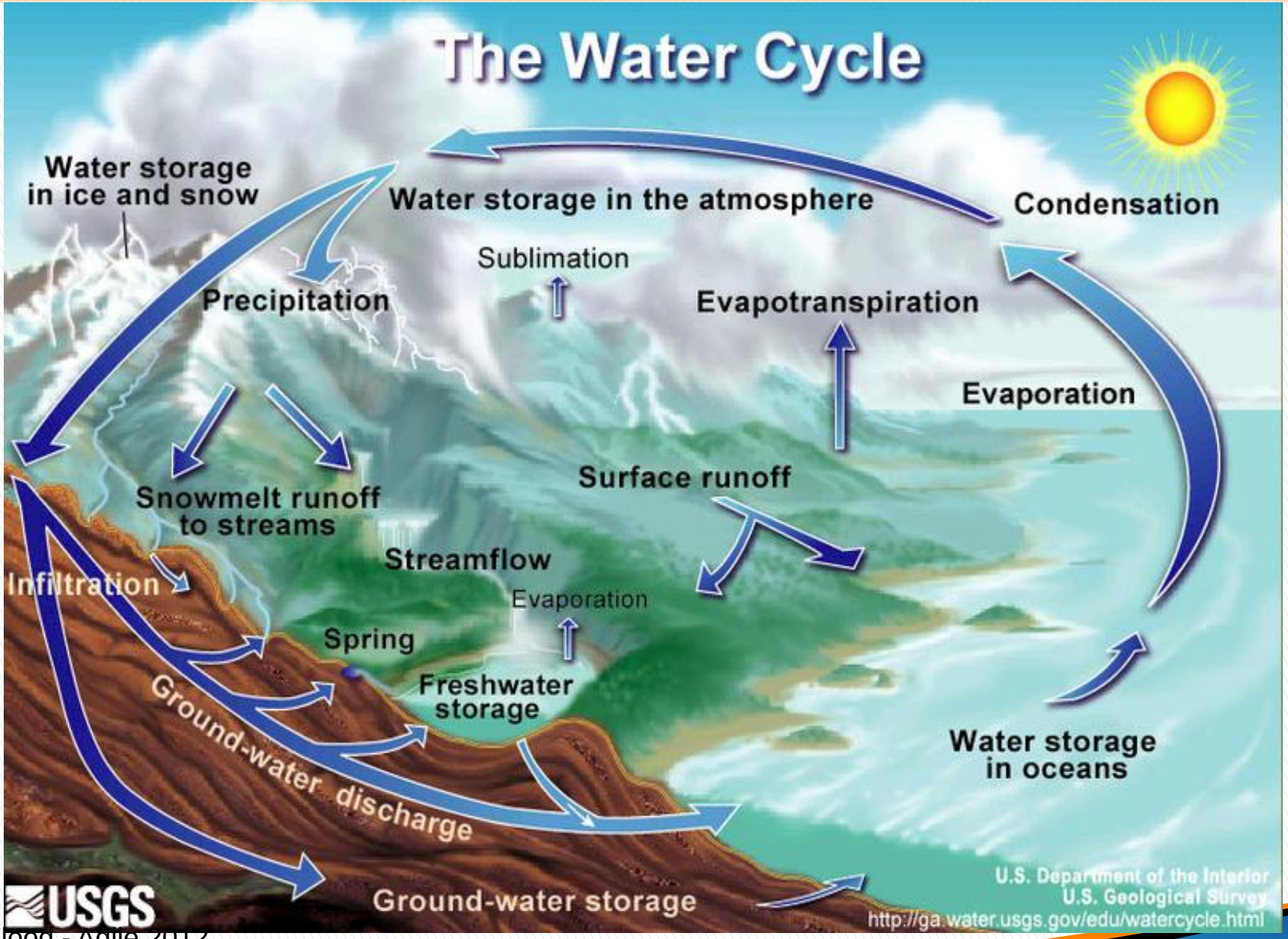
- Introduction
- Drainage network computation
- *RWFlood* algorithm description
- Implementation details
- Results

GIS: A Framework for Understanding and Managing Our Earth



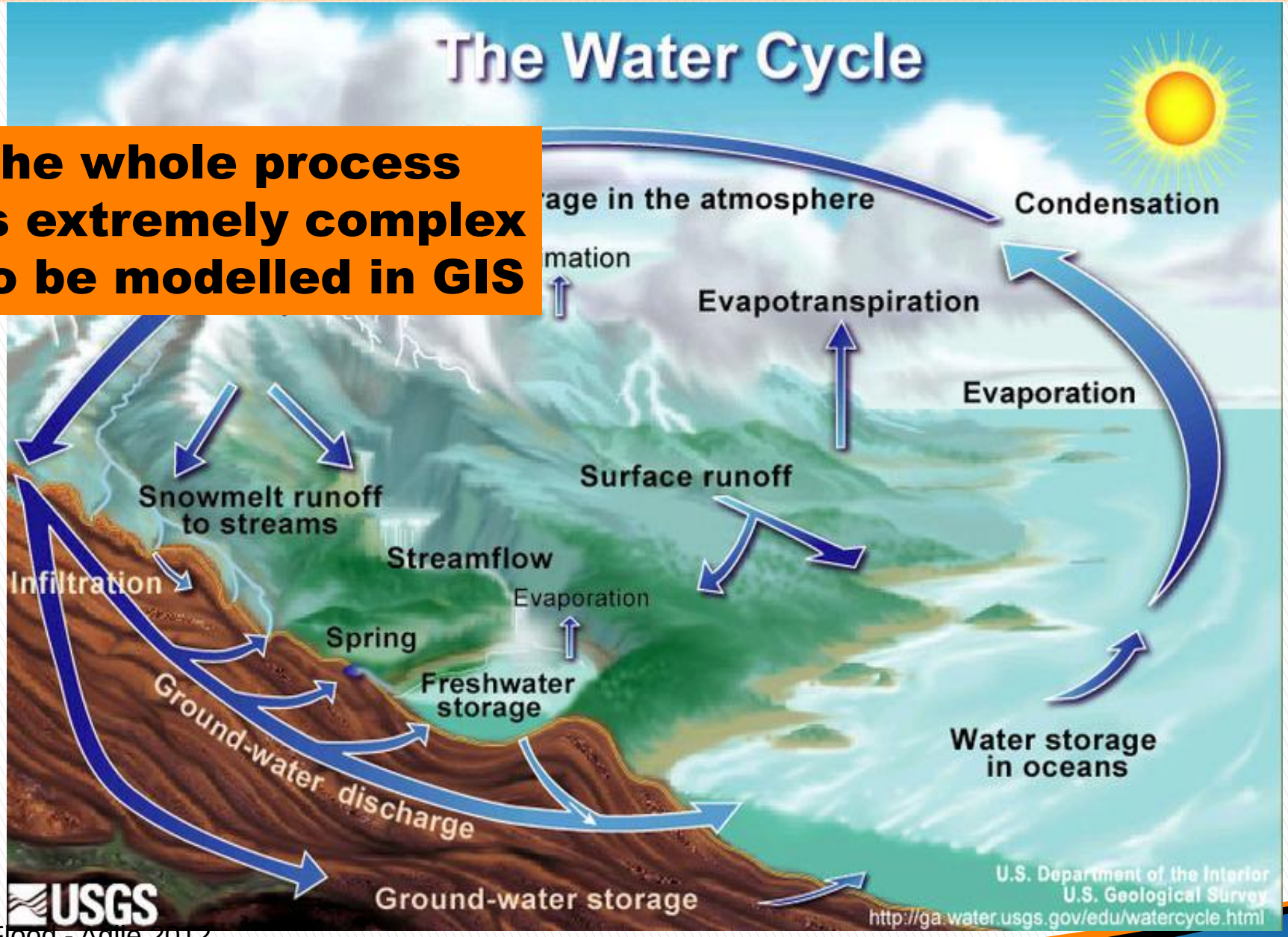
by Angela Lee – ESRI
An introduction to GIS

Hydrologic modelling

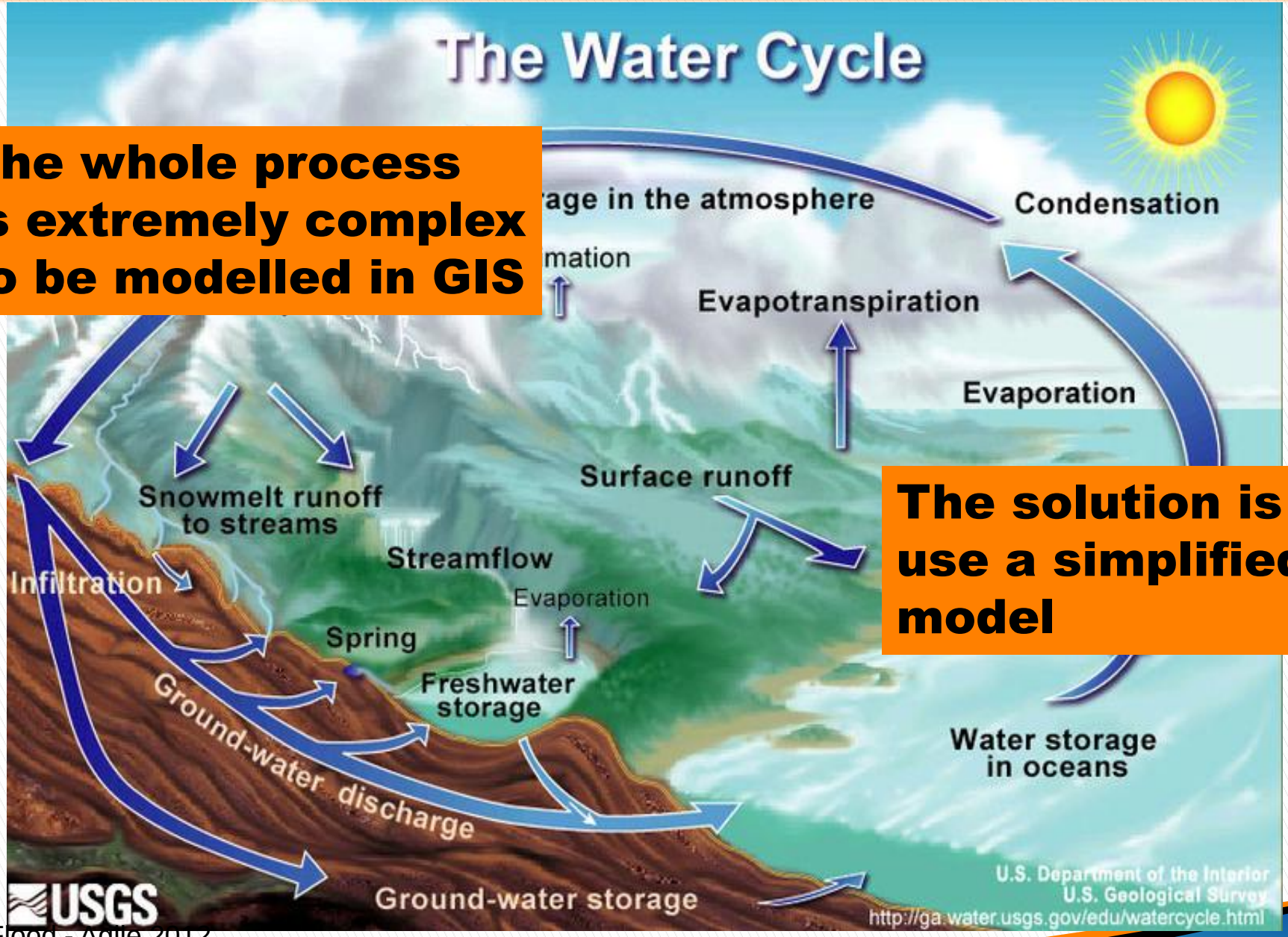


Hydrologic modelling

The whole process is extremely complex to be modelled in GIS



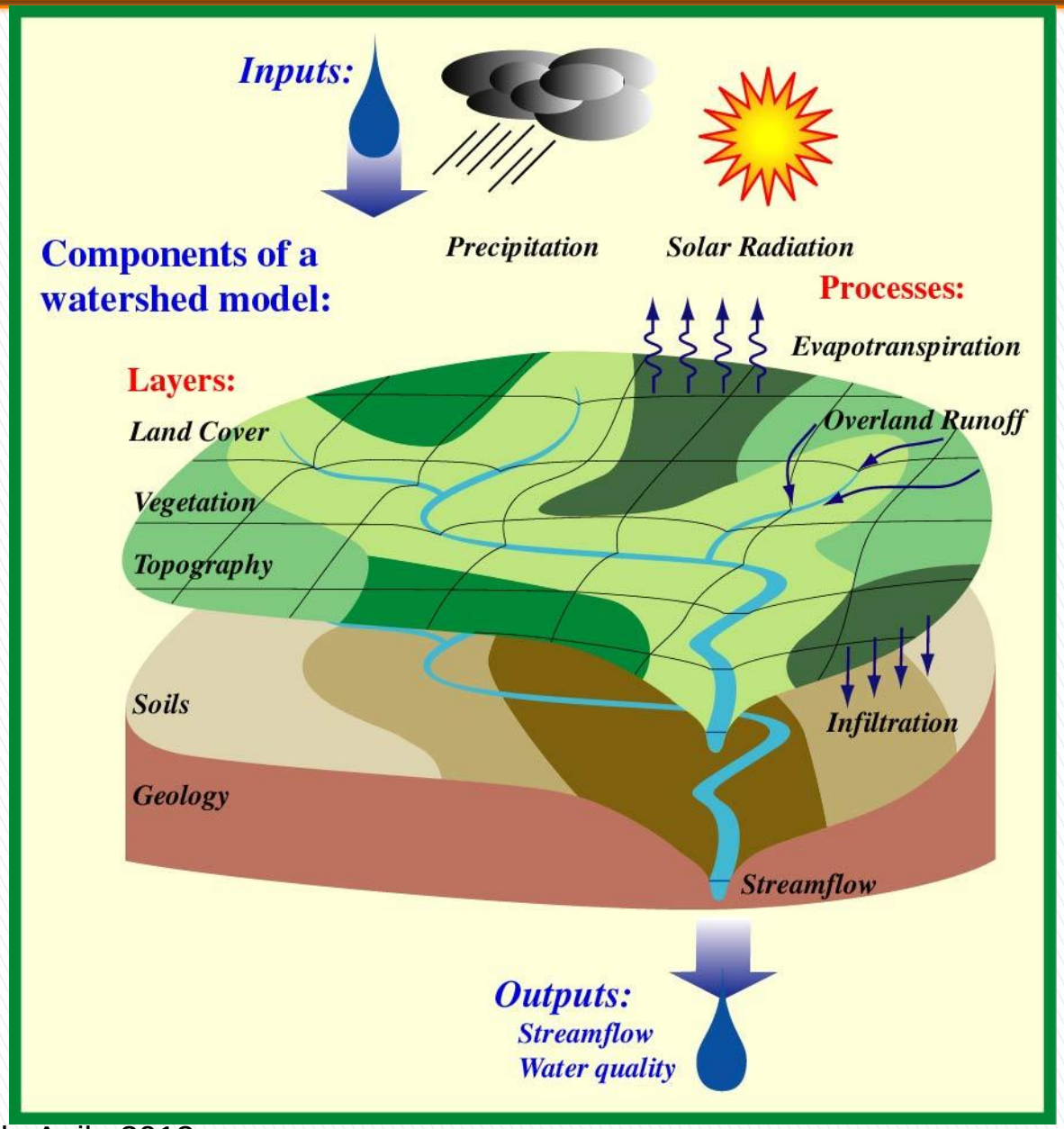
Hydrologic modelling



The whole process is extremely complex to be modelled in GIS

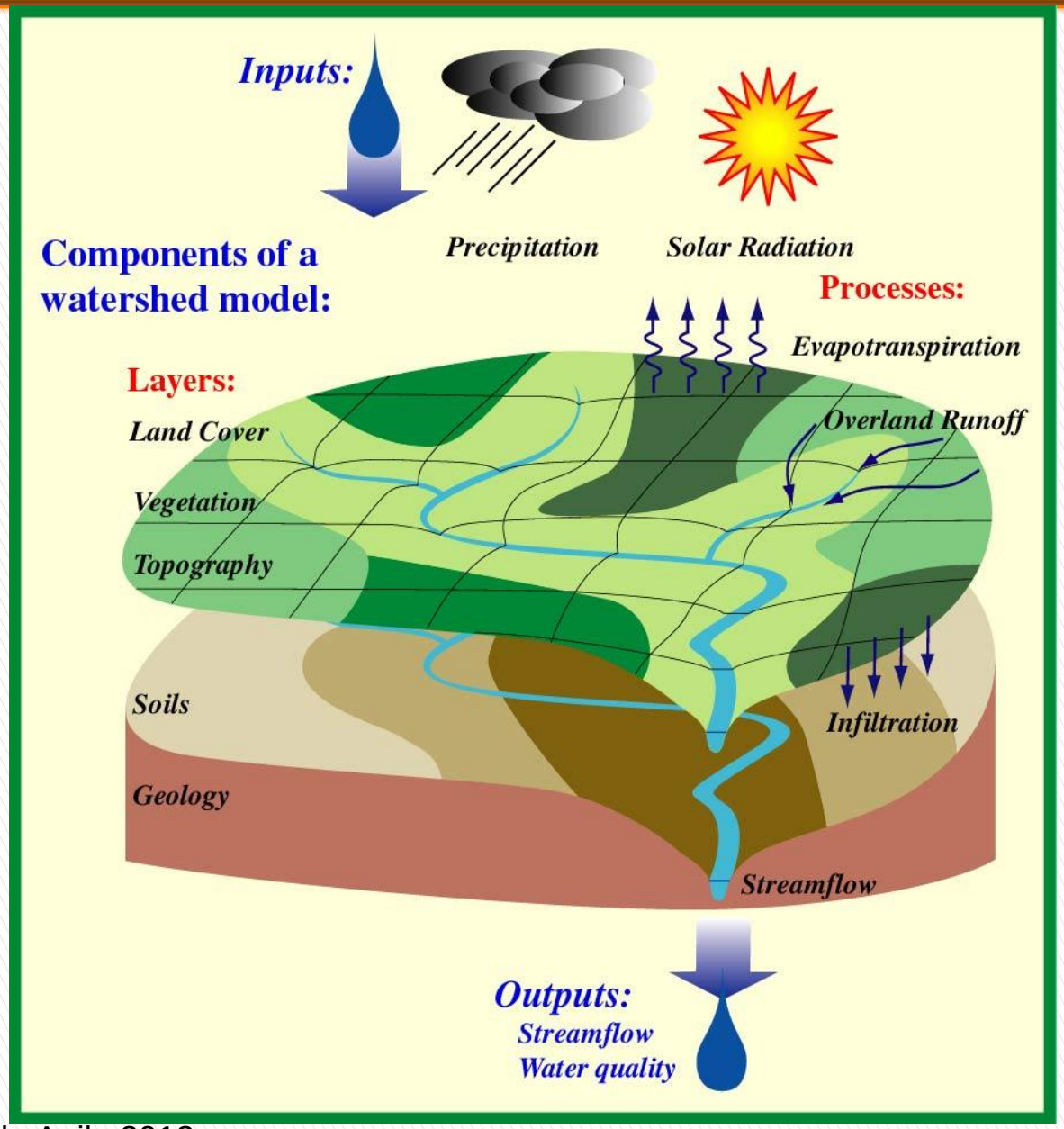
The solution is to use a simplified model

Hydrologic model and GIS



http:// swatmodel.tamu.edu

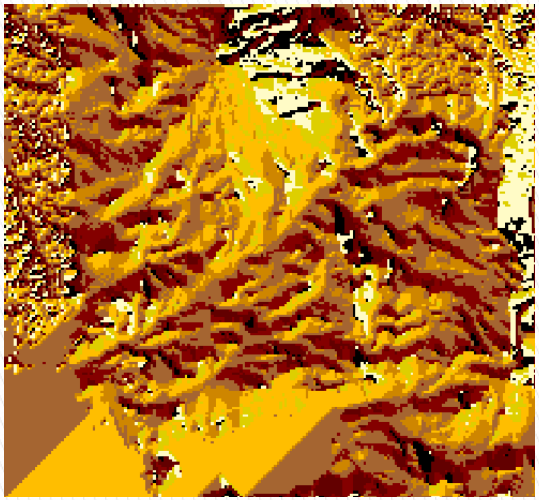
Hydrologic model and GIS



An important component of hydrologic models is the watershed delineation

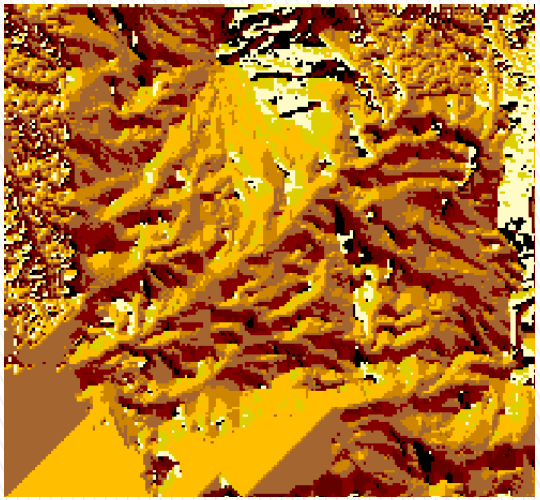
http:// swatmodel.tamu.edu

Watershed computation

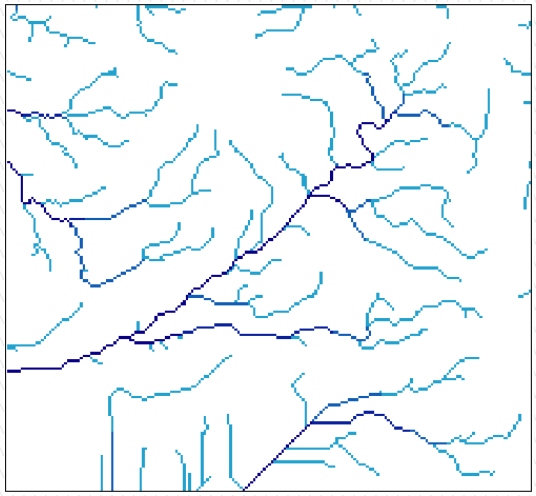


DEM

Watershed computation

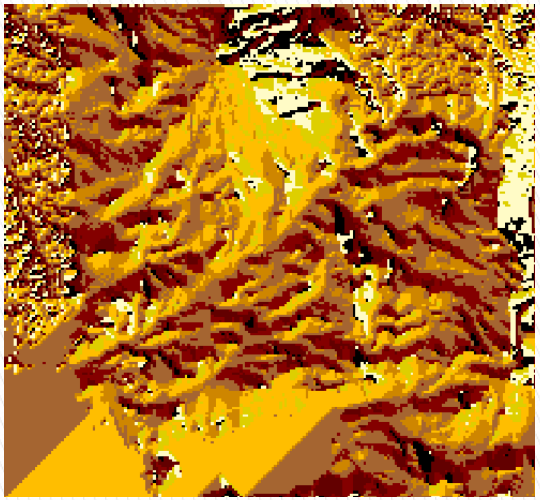


DEM

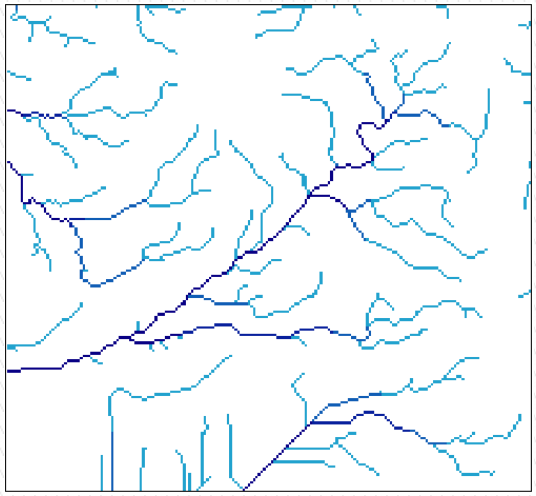


Flow
Directon

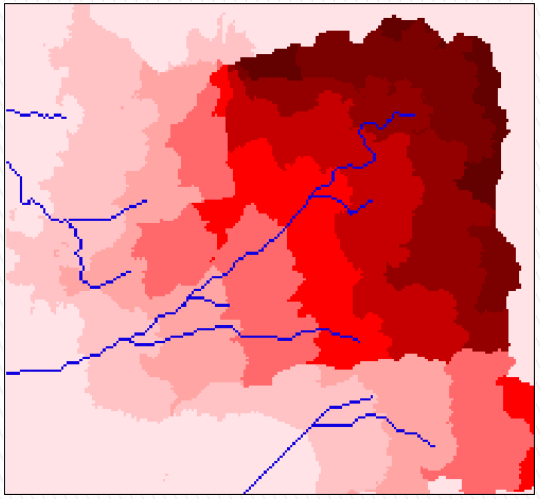
Watershed computation



DEM

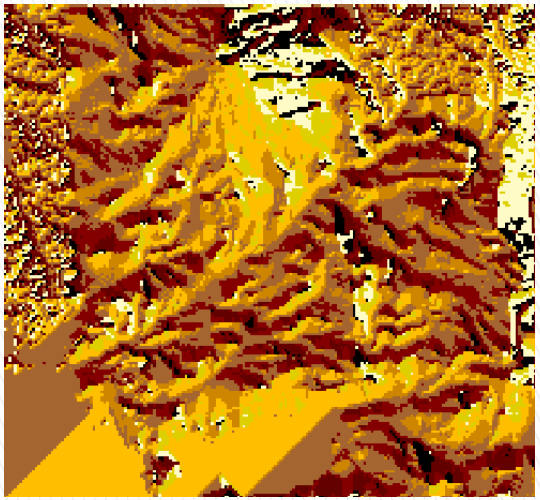


Flow
Directon

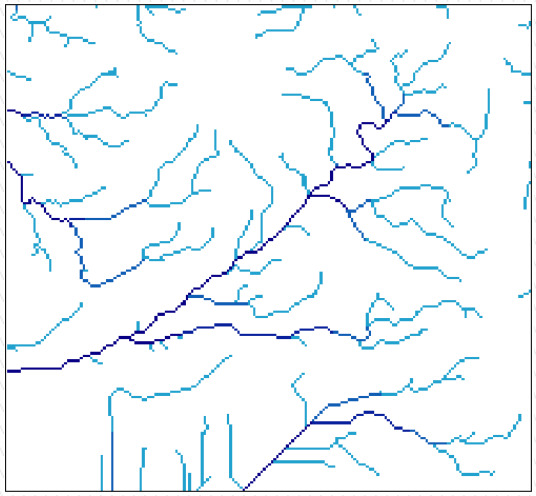


Flow
Accumulation

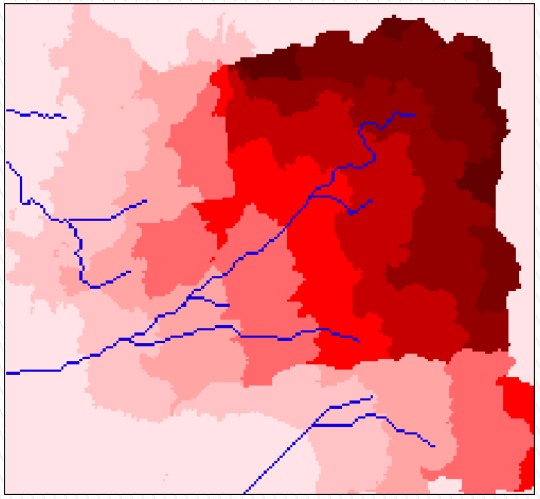
Watershed computation



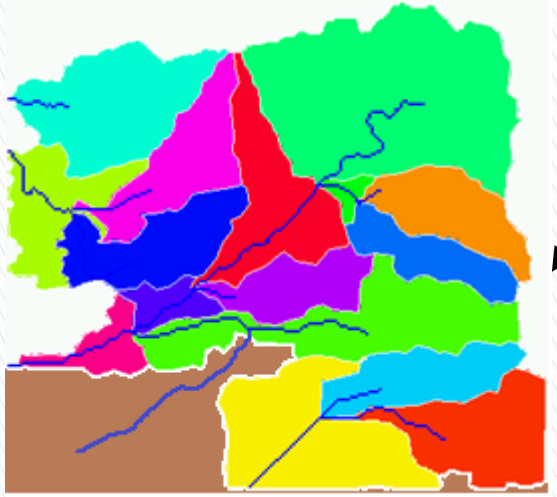
DEM



Flow
Directon



Flow
Accumulation

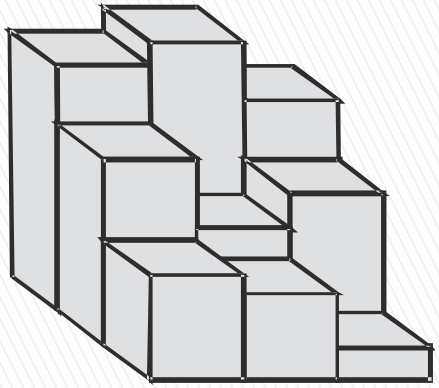


Watershed boundaries

Drainage network computation

71	72	67
68	62	65
63	61	78

DEM

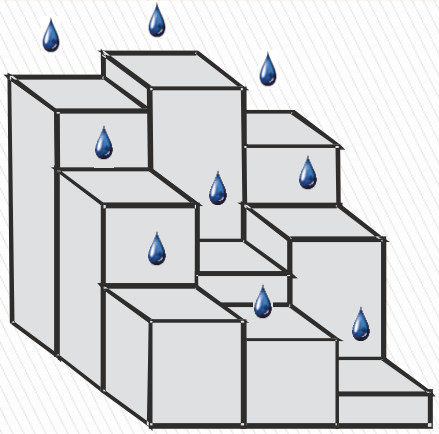


3D Viewing

Drainage network computation

71	72	67
68	62	65
63	61	58

DEM

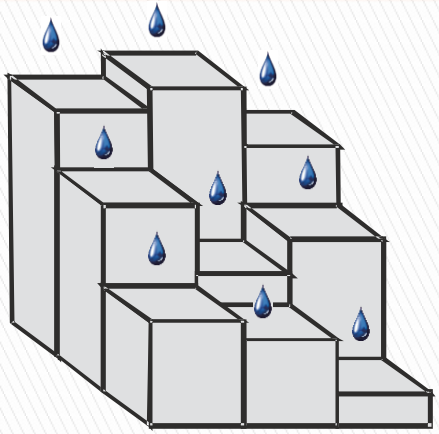


3D Viewing

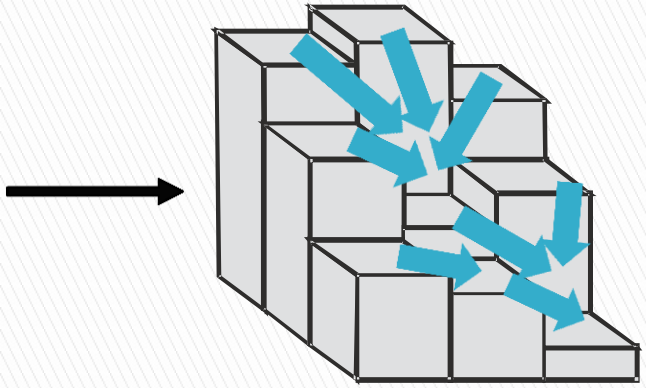
Drainage network computation

71	72	67
68	62	65
63	61	58

DEM



3D Viewing



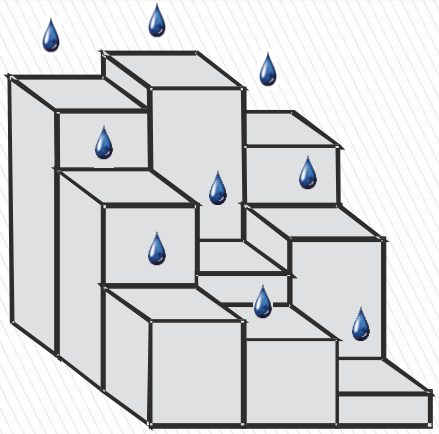
71	72	67
68	62	65
63	61	58

Flow direction

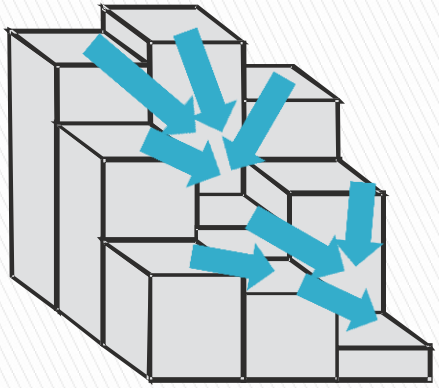
Drainage network computation

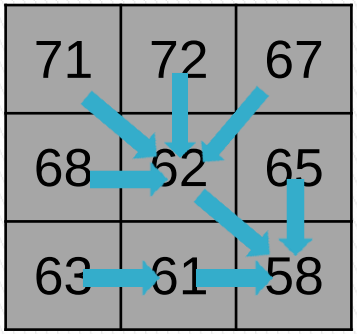
71	72	67
68	62	65
63	61	58

DEM



3D Viewing





Flow direction

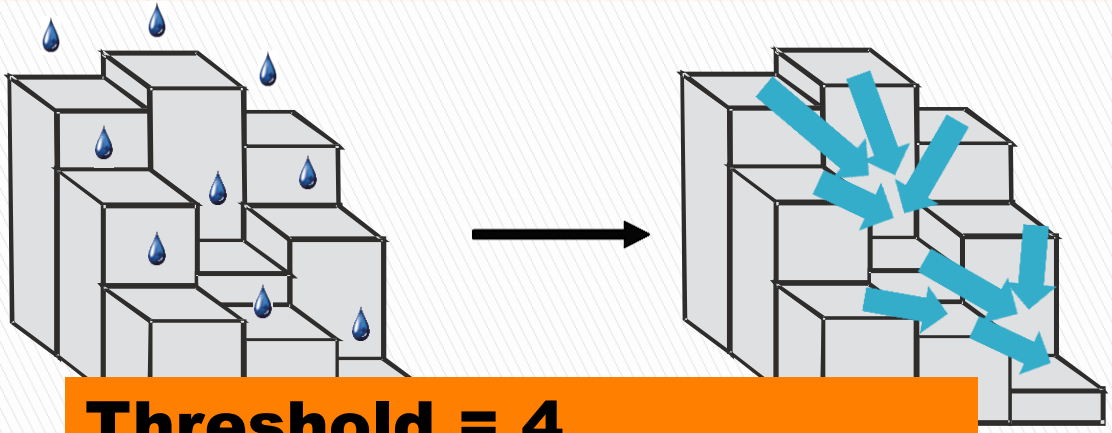
1	1	1
1	5	1
1	2	9

Flow accumulation

Drainage network computation

71	72	67
68	62	65
63	61	58

DEM



71	72	67
68	62	65
63	61	58

Flow direction

Threshold = 4
the drainage network is composed by all cells with "*flow accum ≥ 4*"

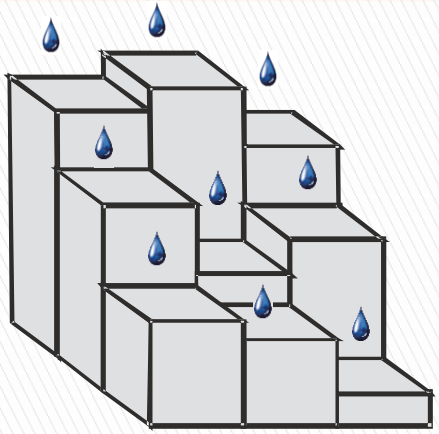
1	1	1
1	5	1
1	2	9

Flow accumulation

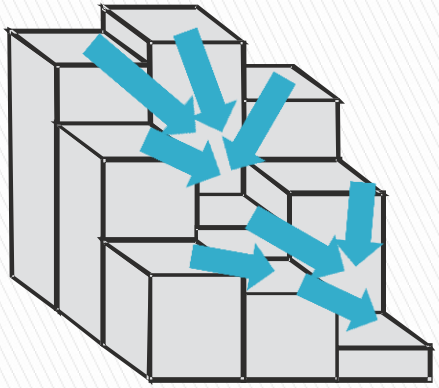
Drainage network computation

71	72	67
68	62	65
63	61	58

DEM



3D Viewing

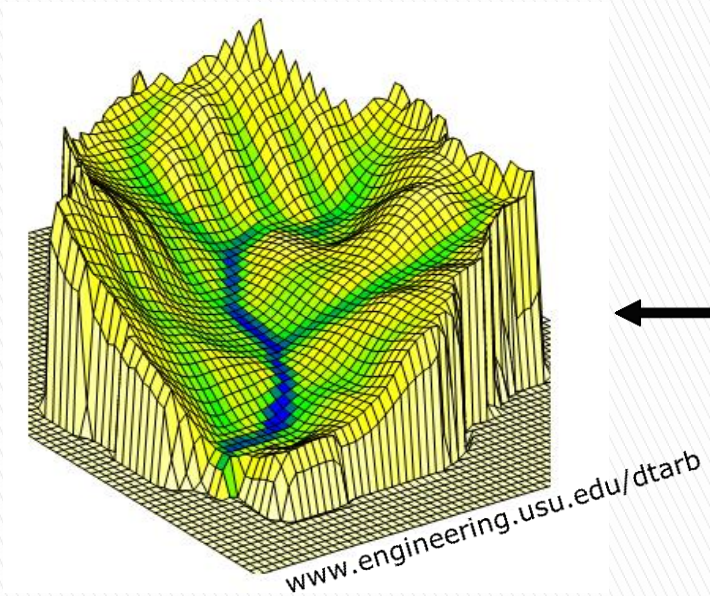


71	72	67
68	62	65
63	61	58

Flow direction

1	1	1
1	5	1
1	2	9

Flow accumulation



Drainage network

Challenges

- In some cases, it is not possible to determine the flow direction in a cell:

71	72	67	71	72
68	62	65	68	62
63	61	58	63	61
68	62	65	68	62
71	72	67	71	72

Depression

71	72	67	71	72
67	68	68	68	62
63	68	68	68	61
68	62	65	68	62
71	72	67	71	72

Flat area

- In general, these two cases are treated by a very time-consuming preprocessing step

Preprocessing step

- A depression is removed by filling it; that is, its elevation is raised to the elevation of its lowest neighbor.
- For example,

71	72	67	71	72
68	62	65	68	62
63	61	58	63	61
68	62	65	68	62
71	72	67	71	72

Depression

Preprocessing step

- A depression is removed by filling it; that is, its elevation is raised to the elevation of its lowest neighbor.
- For example,

Lowest neighbor

71	72	67	71	72
68	62	65	68	62
63	61	58	63	61
68	62	65	68	62
71	72	67	71	72

Preprocessing step

- A depression is removed by filling it; that is, its elevation is raised to the elevation of its lowest neighbor.
- For example,

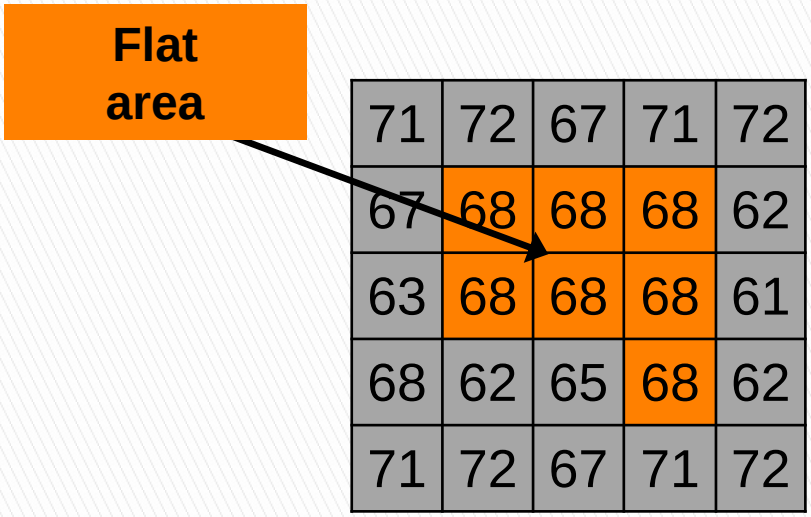
71	72	67	71	72
68	62	65	68	62
63	61	58	63	61
68	62	65	68	62
71	72	67	71	72

71	72	67	71	72
68	62	65	68	62
63	61	61	63	61
68	62	65	68	62
71	72	67	71	72

Raised cell

Preprocessing step

- And, the flow direction in flat areas is oriented to the lowest neighbor cell. For example,



Preprocessing step

- And, the flow direction in flat areas is oriented to the lowest neighbor cell. For example,

71	72	67	71	72
67	68	68	68	62
63	68	68	68	61
68	62	65	68	62
71	72	67	71	72

Lowest neighbor

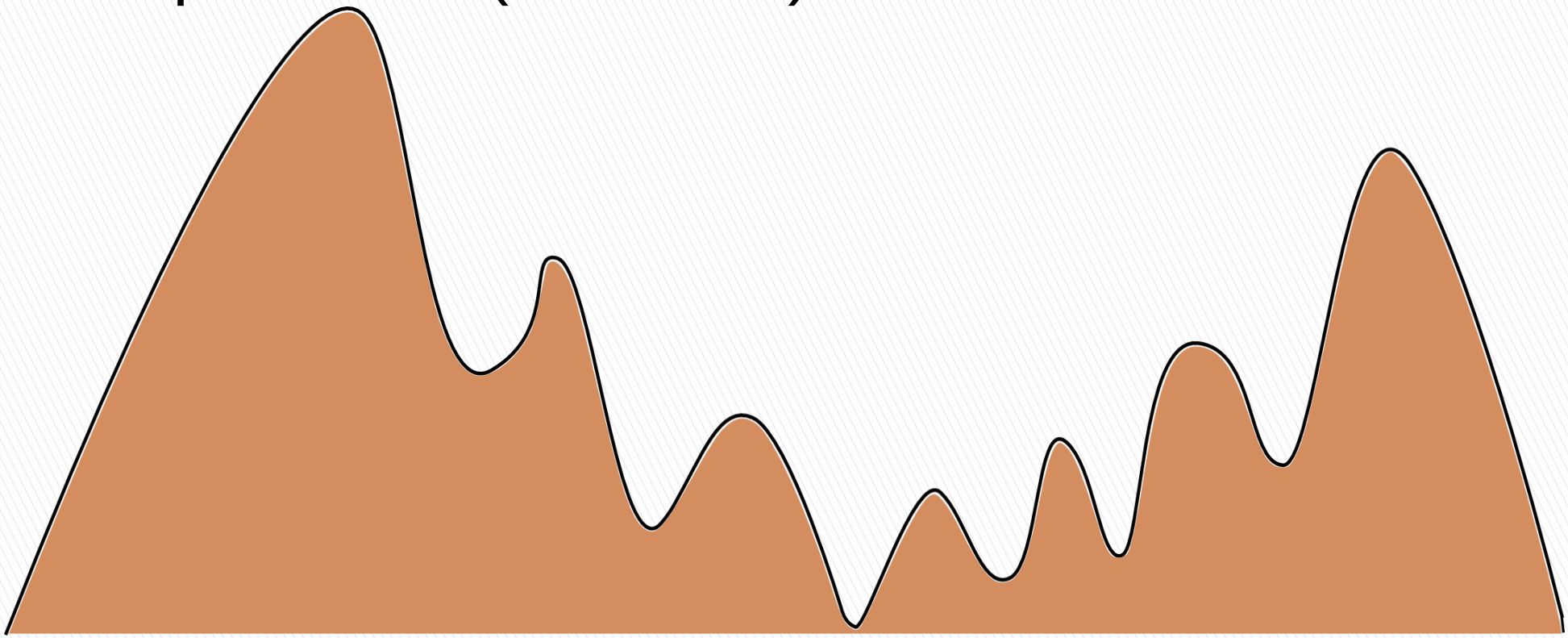
Preprocessing step

- And, the flow direction in flat areas is oriented to the lowest neighbor cell. For example,

71	72	67	71	72
67	68	68	68	62
63	68	68	68	61
68	62	65	68	62
71	72	67	71	72

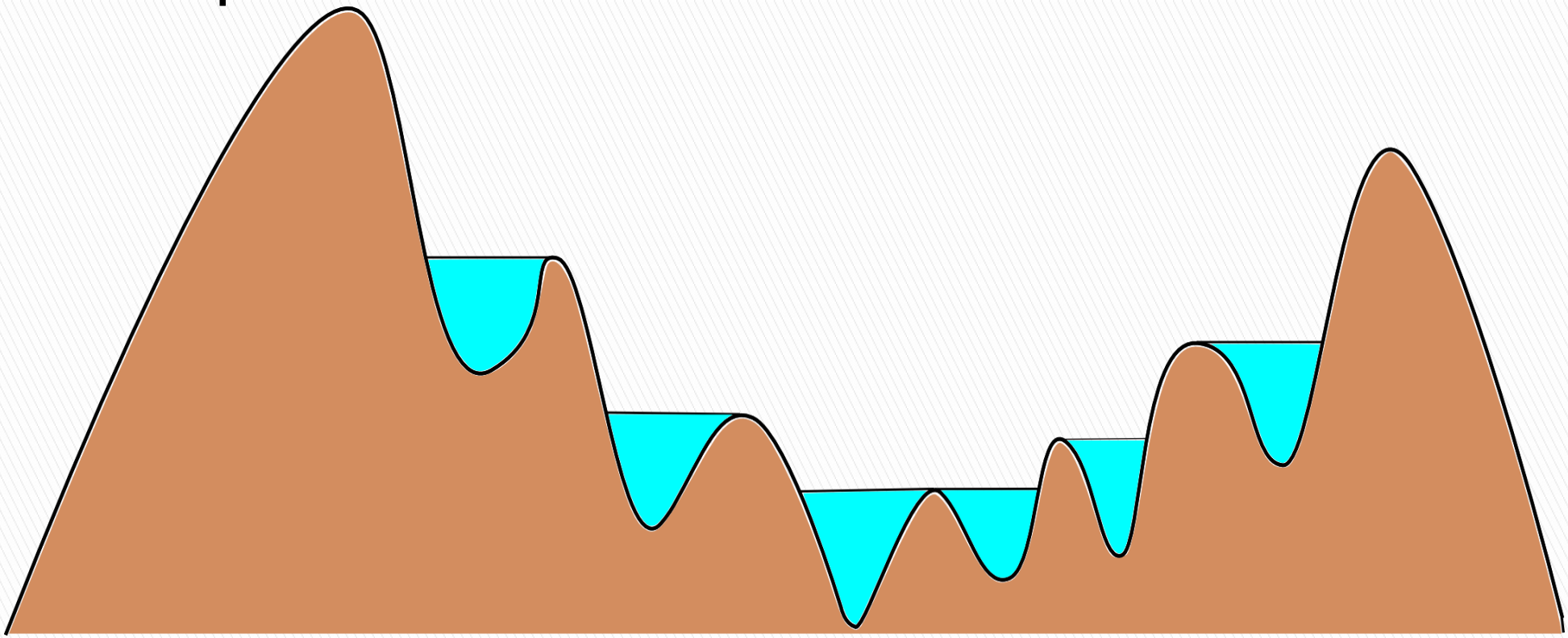
Preprocessing step

- It is not very rare to have some nested depressions (or basins)



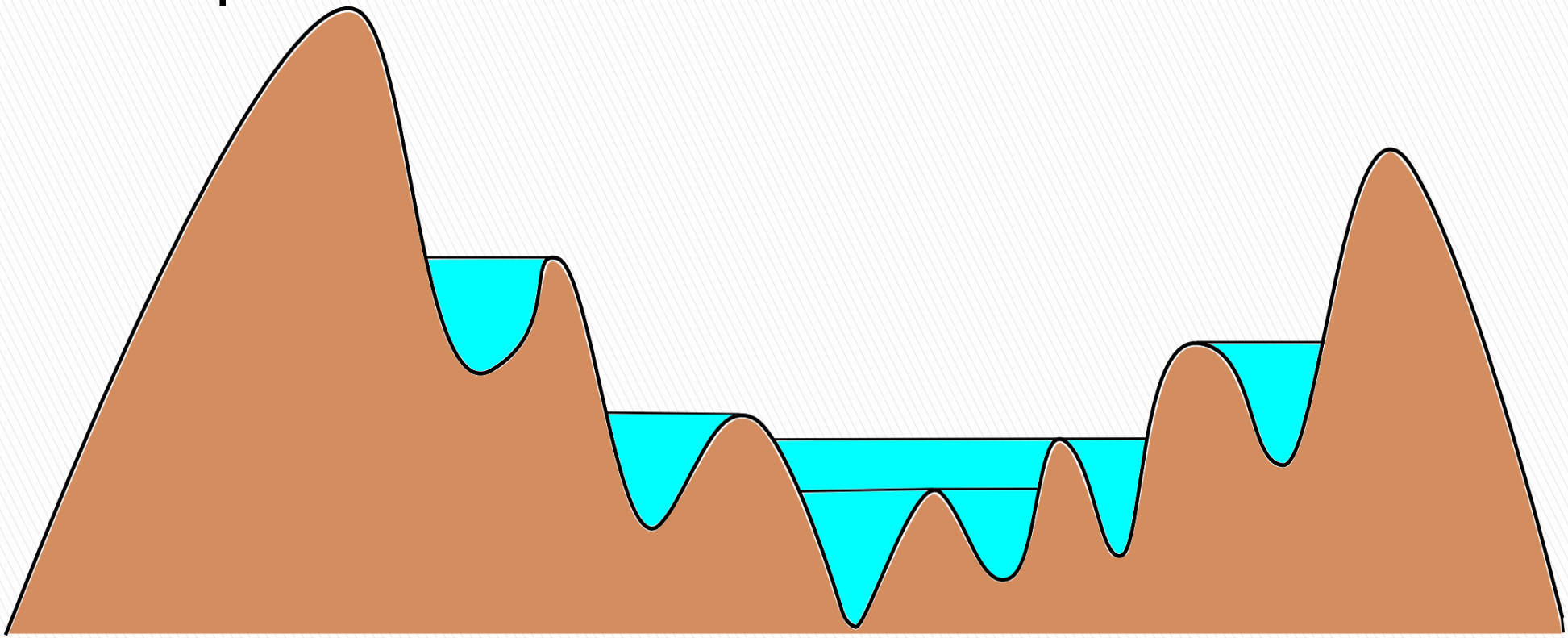
Preprocessing step

- If the basins are filled in sequence, the time can be quadratic in the number of cells



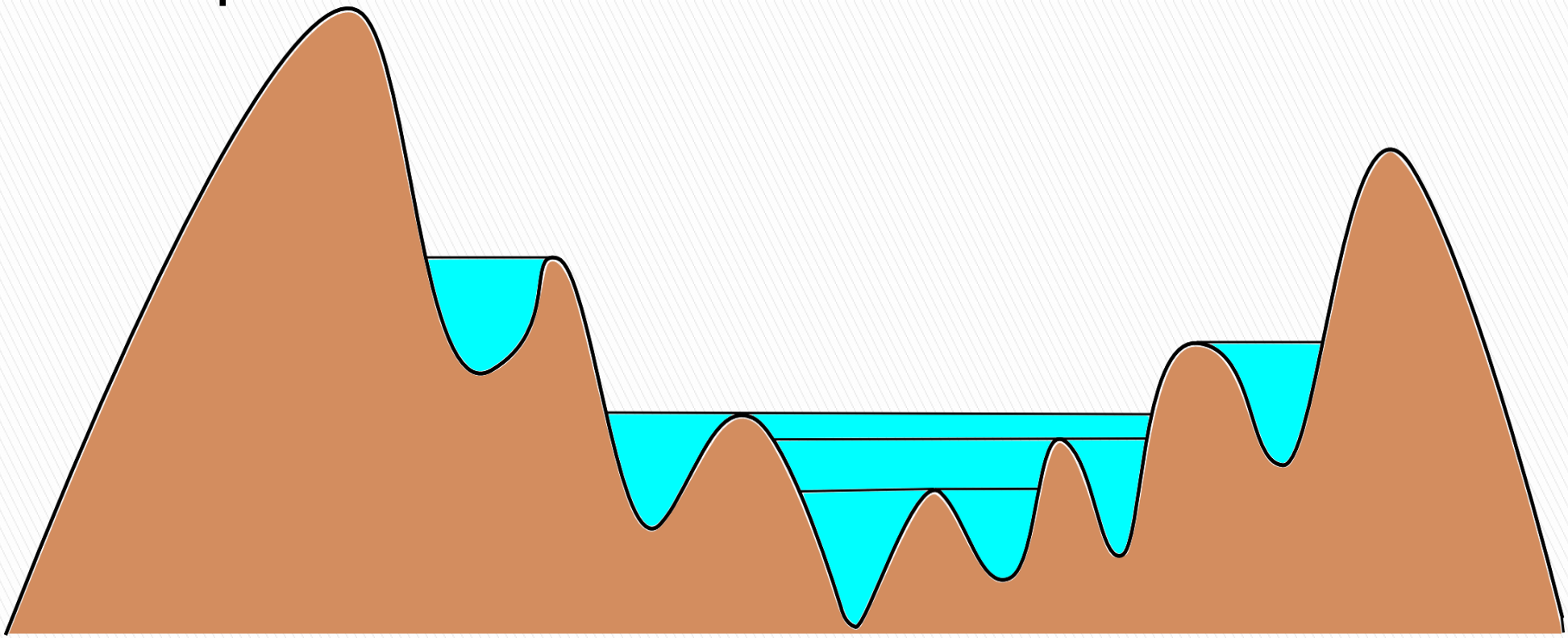
Preprocessing step

- If the basins are filled in sequence, the time can be quadratic in the number of cells



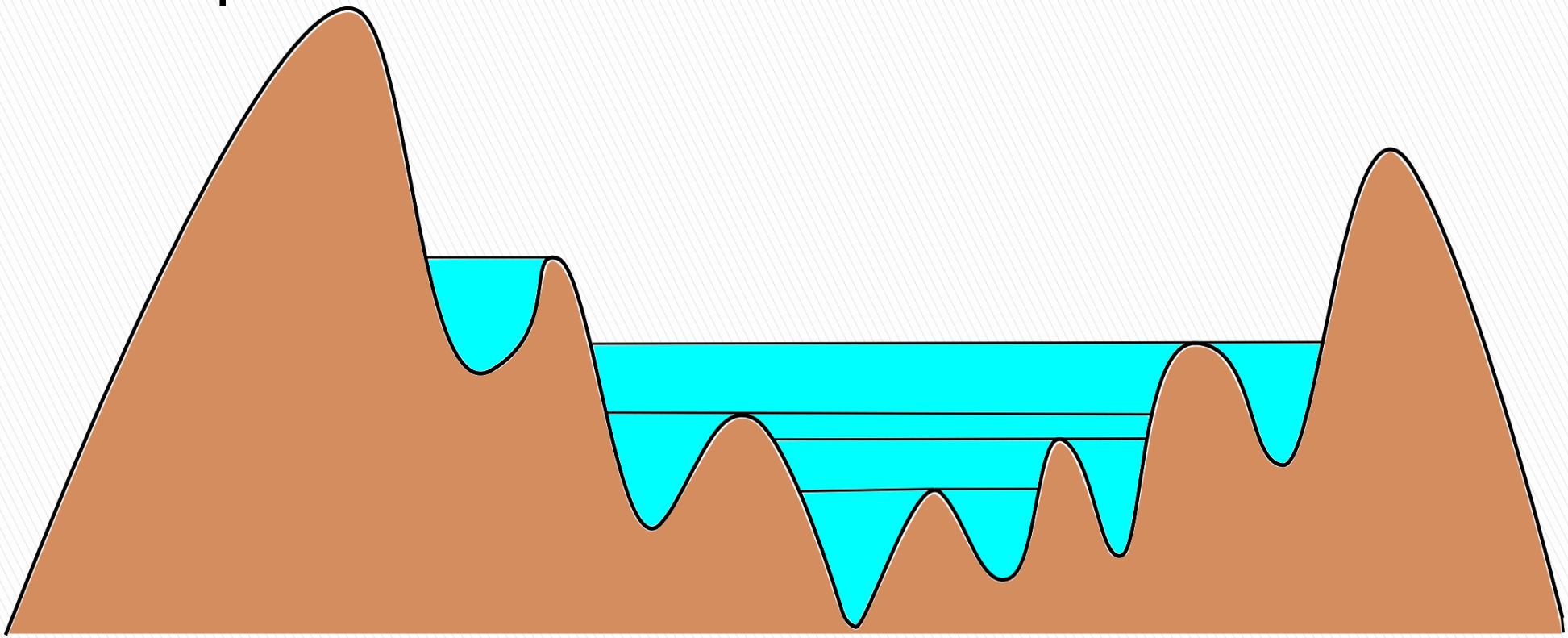
Preprocessing step

- If the basins are filled in sequence, the time can be quadratic in the number of cells



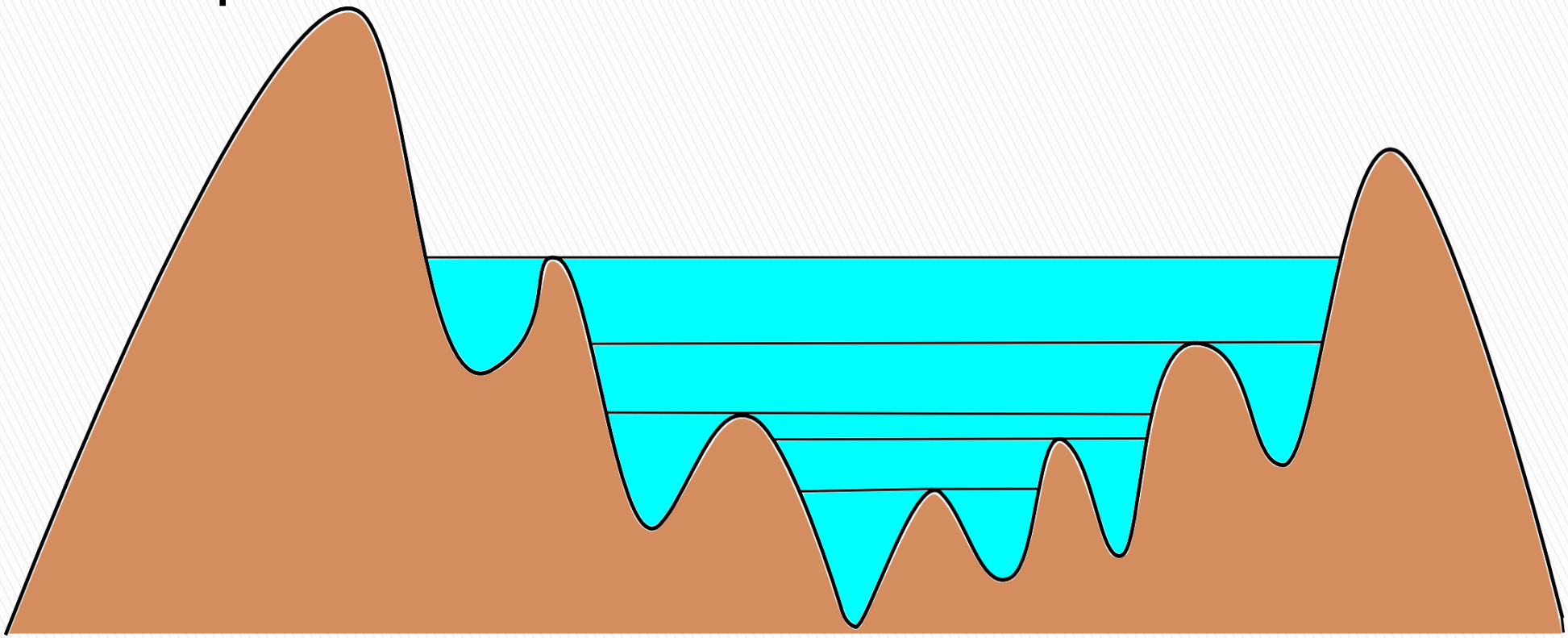
Preprocessing step

- If the basins are filled in sequence, the time can be quadratic in the number of cells



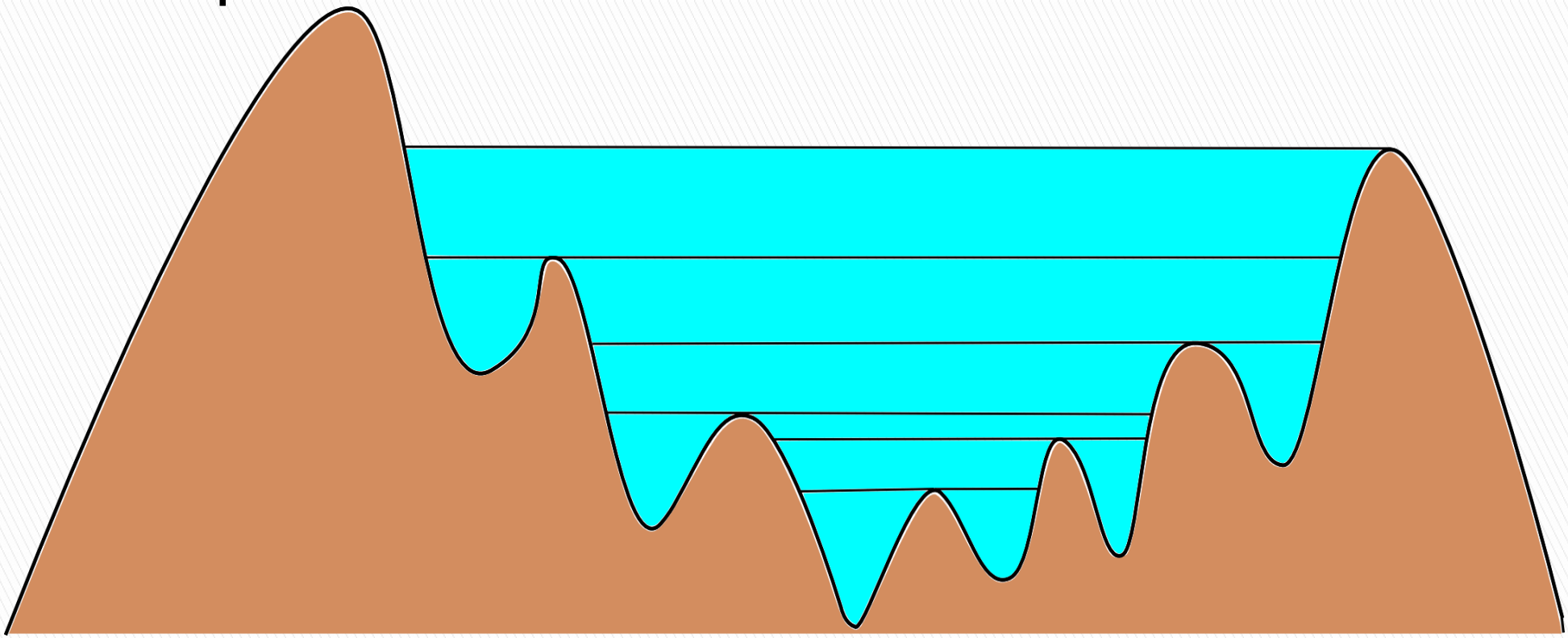
Preprocessing step

- If the basins are filled in sequence, the time can be quadratic in the number of cells



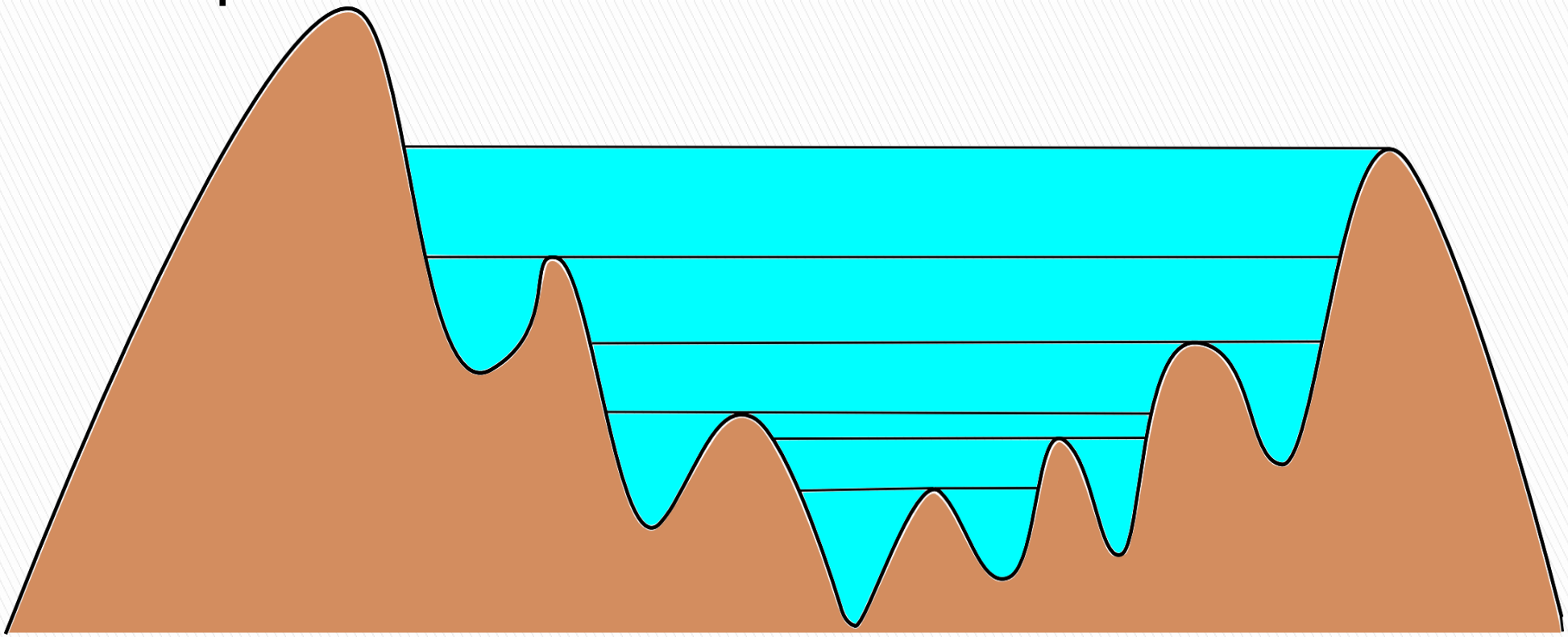
Preprocessing step

- If the basins are filled in sequence, the time can be quadratic in the number of cells



Preprocessing step

- If the basins are filled in sequence, the time can be quadratic in the number of cells



- The time could be sub-quadratic using, for example, a *Union-Find* data structure

Proposed method: *RWFlood*

- To avoid this time-consuming preprocessing step, we proposed a new method, named *RWFlood – Raising Water for Flooding* a terrain;

- The basic idea is:
  - supposing a terrain being flooded by water coming from outside and getting into the terrain through its boundary;
  - so, the course of the water getting into the terrain will be the same as the water coming from rain and flowing downhill (that is, the flow direction).

Proposed method: *RWFlood*

- Then, in other words, the idea is to suppose the terrain surrounded by water (as an island) and to simulate a flooding process raising the water level

Monu Island Fiji

RWFlood description

- Initially, the water level is set to the elevation of the lowest cell in the terrain boundary

- And then, two actions are executed iteratively:
  - flooding a cell
  - raising the water level

RWFlood description

- Flooding a cell *c* ➡ all cells neighbors of *c* are processed as following: given a neighbor cell *d*
  - if the elevation of *d* is smaller than the elevation of *c* then the elevation of *d* is raised to the elevation of *c.*
  - the flow direction of *d* is set to the cell *c*

RWFlood description

- After flooding all cells with the same elevation as *c*, the water level is raised to the elevation of the lowest cell higher than *c* and the process continues from this cell.

- To get this next cell quickly, the cells that need to be processed later are stored in an array of queues *F*.

- *F* has a queue for each elevation and a cell with elevation *m* is stored in the queue *F*[*m*].

RWFlood description

- After flooding all cells with the same elevation as *c*, the water level is raised to the elevation of the lowest cell higher than *c* and the process continues from this cell.

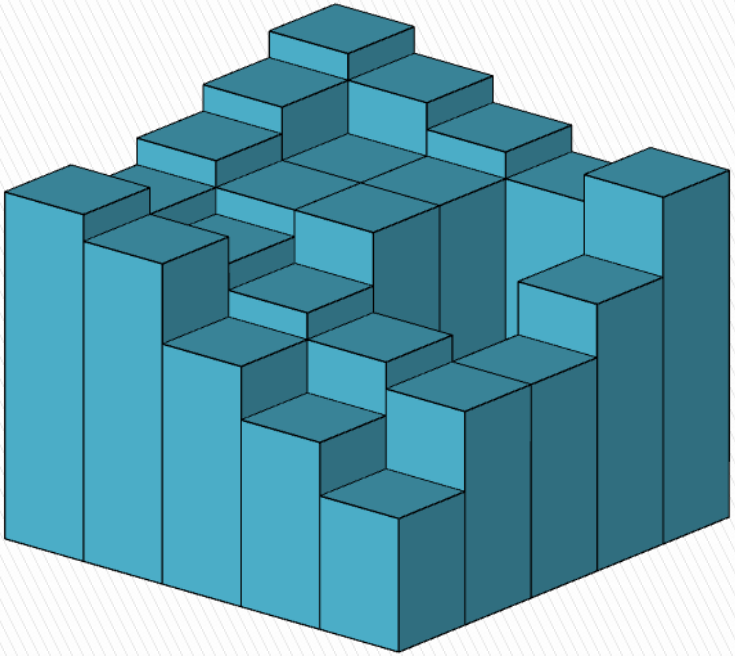
- To get this next [illegible] at need to be processed [illegible] array of queues *F*.

This is one of the most important contribution of our paper

- *F* has a queue [illegible] a cell with elevation *m* is stored in the queue *F*[*m*].

RWFlood description

	1	2	3	4	5
1	68	67	68	69	70
2	67	66	67	67	69
3	64	65	67	67	68
4	62	64	60	60	67
5	60	63	63	65	68

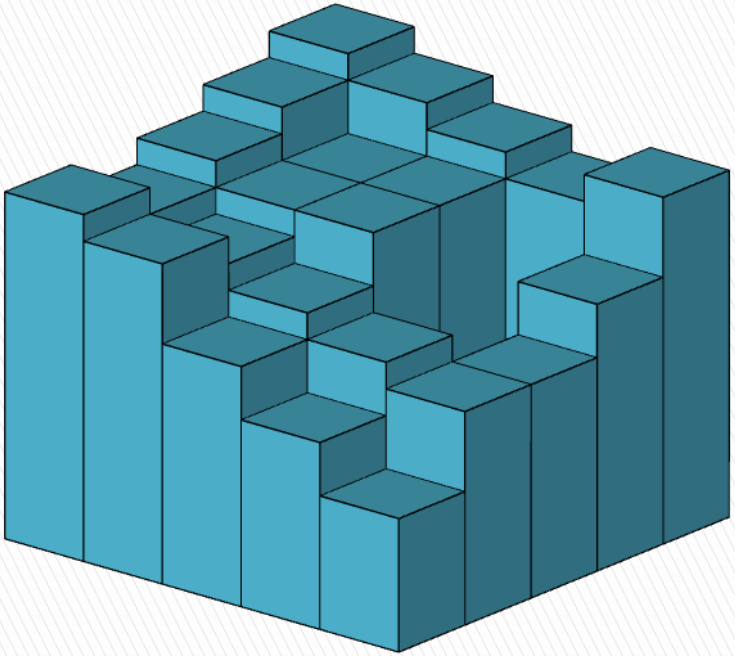


***F*=**

60	61	62	63	64	65	66	67	68	69	70

RWFlood description

	1	2	3	4	5
1	68	67	68	69	70
2	67	66	67	67	69
3	64	65	67	67	68
4	62	64	60	60	67
5	60	63	63	65	68

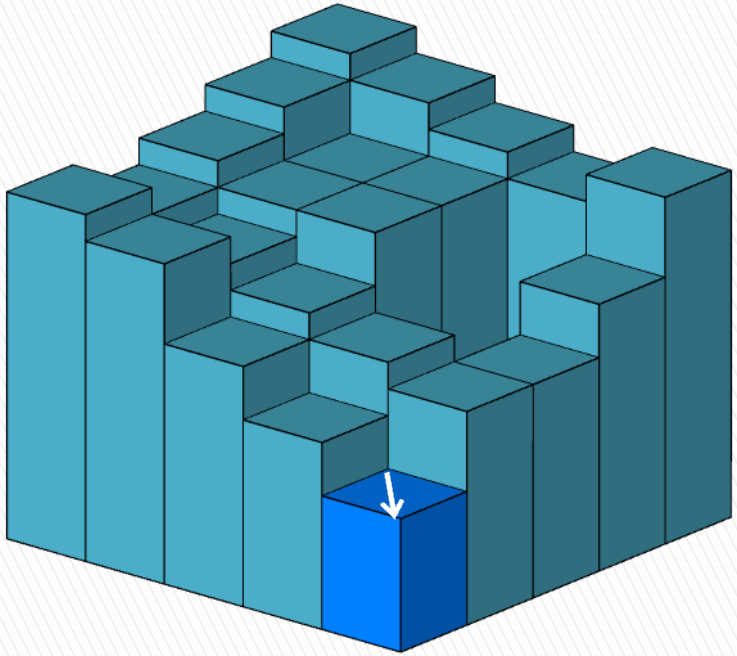


***F*=**

60	61	62	63	64	65	66	67	68	69	70
(5,1)		(4,1)	(5,2) (5,3)	(3,1)	(5,4)		(1,2) (2,1) (4,5)	(1,1) (1,3) (3,5) (5,5)	(1,4) (2,5)	(1,5)

RWFlood description

	1	2	3	4	5
1	68	67	68	69	70
2	67	66	67	67	69
3	64	65	67	67	68
4	62	64	60	60	67
5	60	63	63	65	68

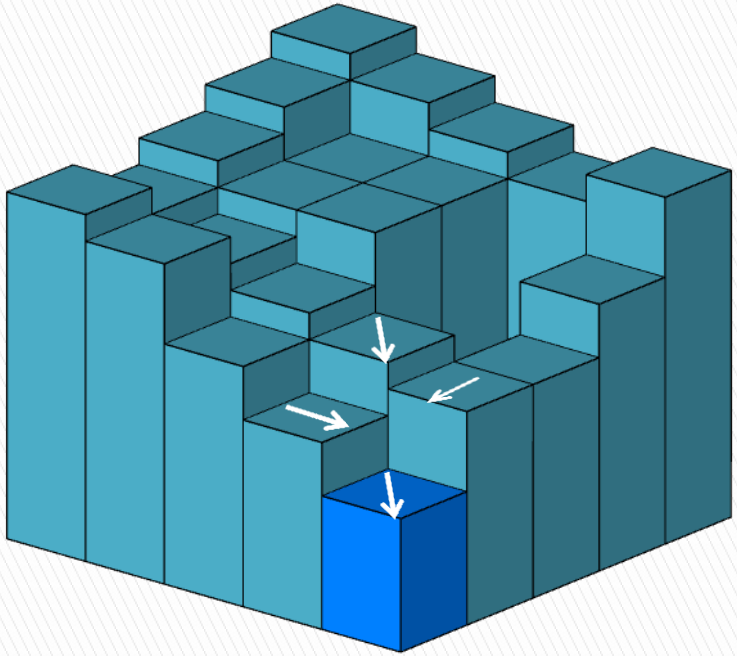


***F*=**

60	61	62	63	64	65	66	67	68	69	70
(5,1)		(4,1)	(5,2) (5,3)	(3,1)	(5,4)		(1,2) (2,1) (4,5)	(1,1) (1,3) (3,5) (5,5)	(1,4) (2,5)	(1,5)

RWFlood description

	1	2	3	4	5
1	68	67	68	69	70
2	67	66	67	67	69
3	64	65	67	67	68
4	62	64	60	60	67
5	60	63	63	65	68

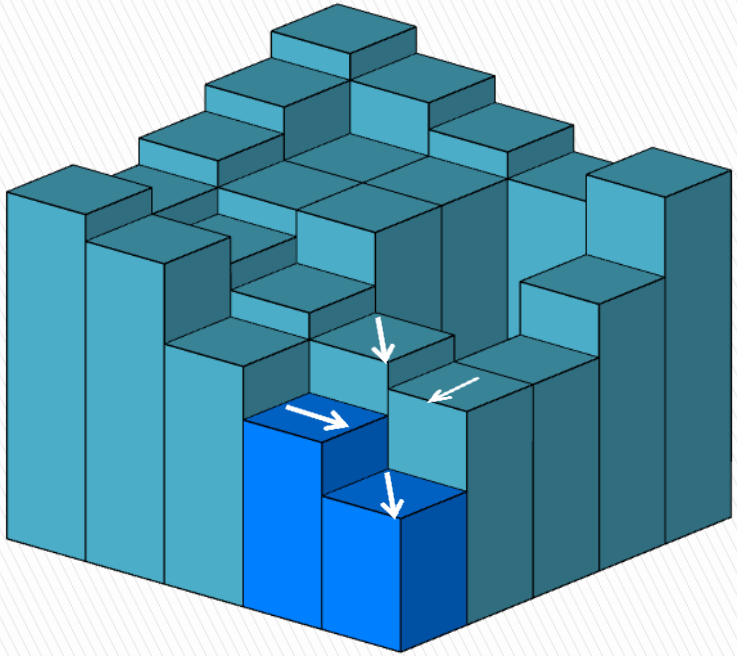


***F*=**

60	61	62	63	64	65	66	67	68	69	70
(5,1)		(4,1)	(5,2) (5,3)	(3,1) (4,2)	(5,4)		(1,2) (2,1) (4,5)	(1,1) (1,3) (3,5) (5,5)	(1,4) (2,5)	(1,5)

RWFlood description

	1	2	3	4	5
1	68	67	68	69	70
2	67	66	67	67	69
3	64	65	67	67	68
4	62	64	60	60	67
5	60	63	63	65	68

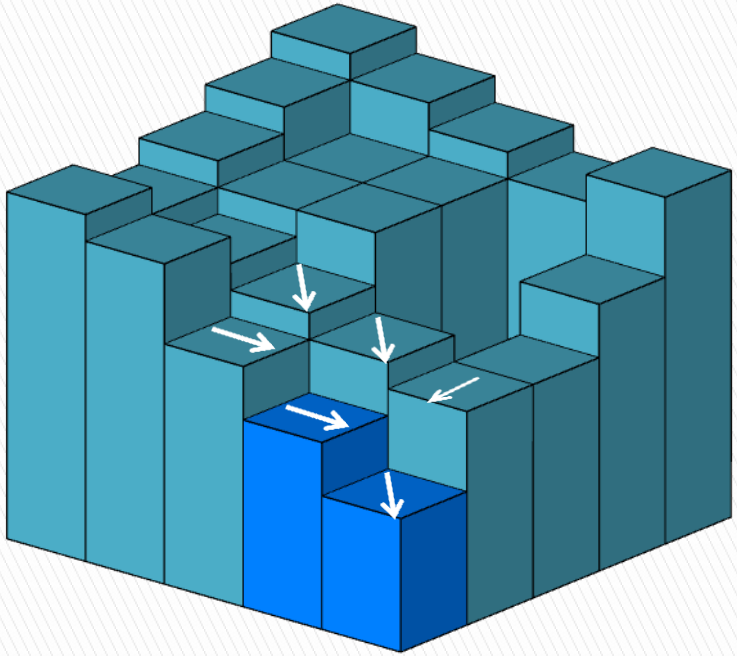


***F*=**

60	61	62	63	64	65	66	67	68	69	70
		(4,1)	(5,2) (5,3)	(3,1) (4,2)	(5,4)		(1,2) (2,1) (4,5)	(1,1) (1,3) (3,5) (5,5)	(1,4) (2,5)	(1,5)

RWFlood description

	1	2	3	4	5
1	68	67	68	69	70
2	67	66	67	67	69
3	64	65	67	67	68
4	62	64	60	60	67
5	60	63	63	65	68

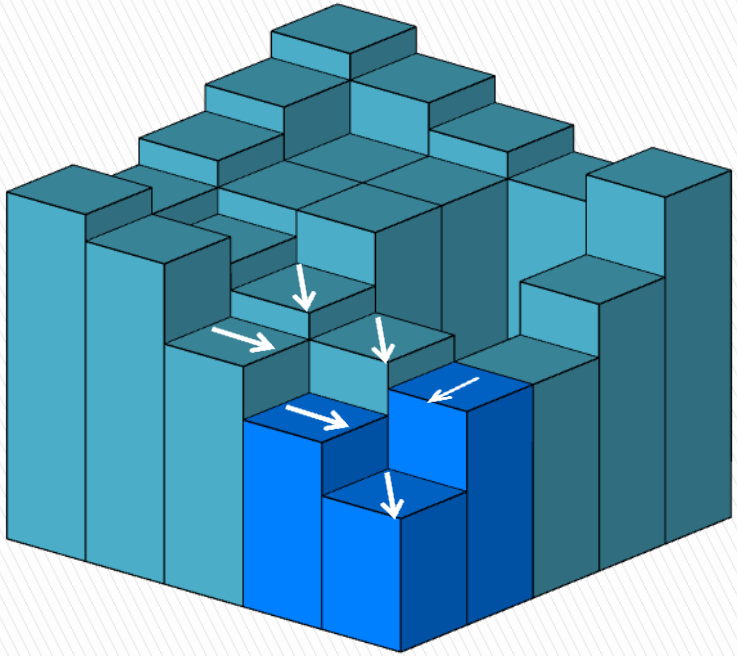


***F*=**

60	61	62	63	64	65	66	67	68	69	70
		(4,1)	(5,2) (5,3)	(3,1) (4,2)	(5,4) (3,2)		(1,2) (2,1) (4,5)	(1,1) (1,3) (3,5) (5,5)	(1,4) (2,5)	(1,5)

RWFlood description

	1	2	3	4	5
1	68	67	68	69	70
2	67	66	67	67	69
3	64	65	67	67	68
4	62	64	60	60	67
5	60	63	63	65	68

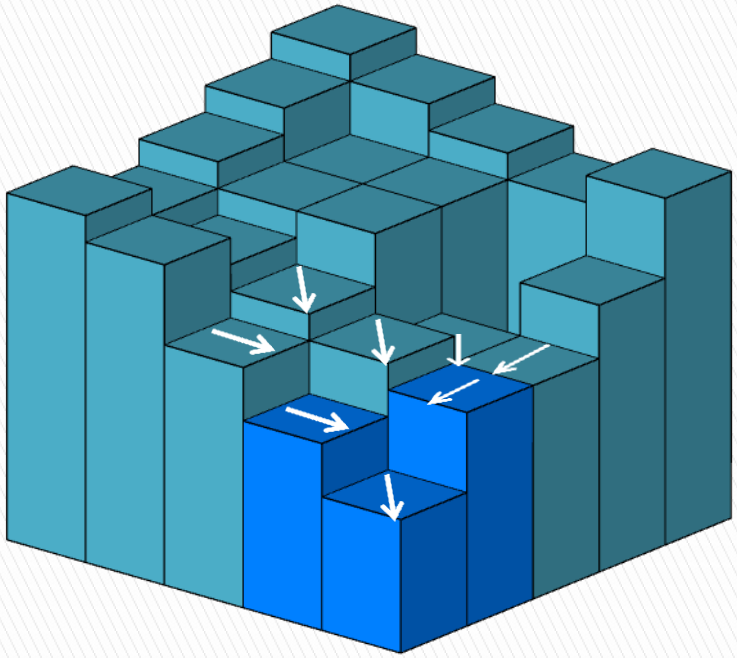


***F*=**

60	61	62	63	64	65	66	67	68	69	70
			(5,2)	(3,1)	(5,4)		(1,2)	(1,1)	(1,4)	(1,5)
			(5,3)	(4,2)	(3,2)		(2,1)	(1,3)	(2,5)	
							(4,5)	(3,5)		
								(5,5)		

RWFlood description

	1	2	3	4	5
1	68	67	68	69	70
2	67	66	67	67	69
3	64	65	67	67	68
4	62	64	63	60	67
5	60	63	63	65	68

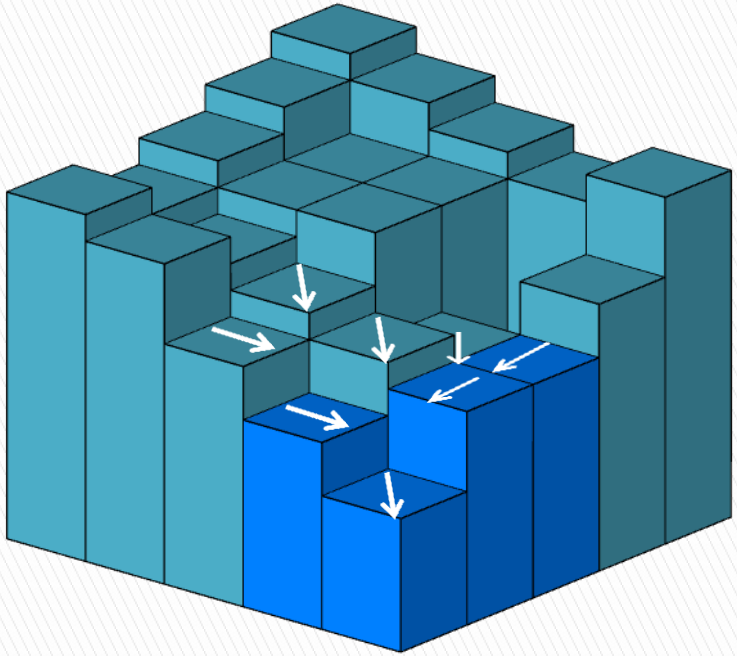


***F*=**

60	61	62	63	64	65	66	67	68	69	70
			(5,2)	(3,1)	(5,4)		(1,2)	(1,1)	(1,4)	(1,5)
			(5,3)	(4,2)	(3,2)		(2,1)	(1,3)	(2,5)	
			(4,3)				(4,5)	(3,5)		
								(5,5)		

RWFlood description

	1	2	3	4	5
1	68	67	68	69	70
2	67	66	67	67	69
3	64	65	67	67	68
4	62	64	63	60	67
5	60	63	63	65	68

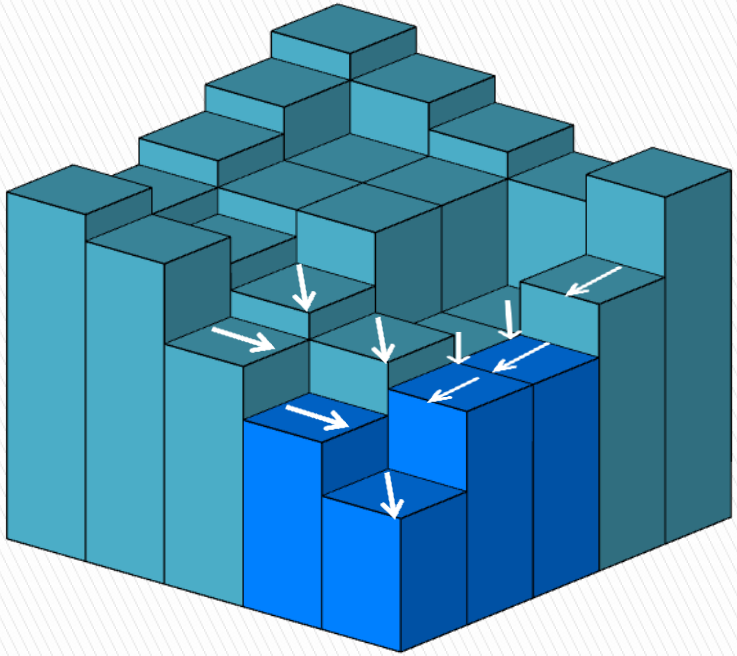


***F*=**

60	61	62	63	64	65	66	67	68	69	70
			(5,3)	(3,1)	(5,4)		(1,2)	(1,1)	(1,4)	(1,5)
			(4,3)	(4,2)	(3,2)		(2,1)	(1,3)	(2,5)	
							(4,5)	(3,5)		
								(5,5)		

RWFlood description

	1	2	3	4	5
1	68	67	68	69	70
2	67	66	67	67	69
3	64	65	67	67	68
4	62	64	63	63	67
5	60	63	63	65	68

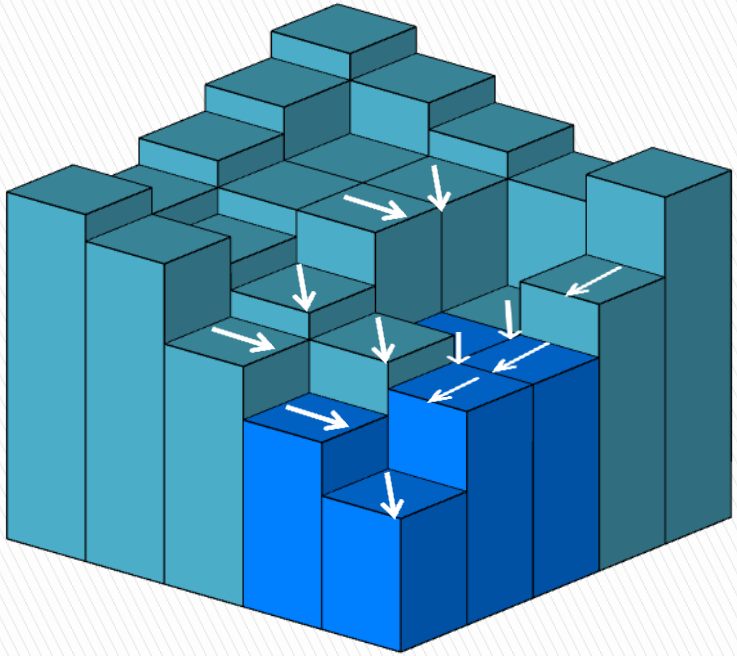


***F*=**

60	61	62	63	64	65	66	67	68	69	70
			(5,3)	(3,1)	(5,4)		(1,2)	(1,1)	(1,4)	(1,5)
			(4,3)	(4,2)	(3,2)		(2,1)	(1,3)	(2,5)	
			(4,4)				(4,5)	(3,5)		
								(5,5)		

RWFlood description

	1	2	3	4	5
1	68	67	68	69	70
2	67	66	67	67	69
3	64	65	67	67	68
4	62	64	63	63	67
5	60	63	63	65	68

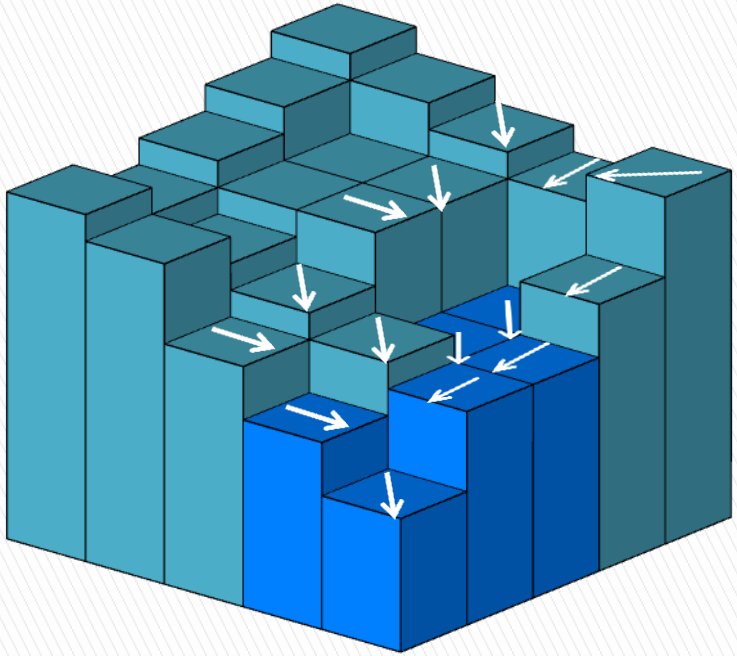


***F*=**

60	61	62	63	64	65	66	67	68	69	70
			(4,3)	(3,1)	(5,4)		(1,2)	(1,1)	(1,4)	(1,5)
			(4,4)	(4,2)	(3,2)		(2,1)	(1,3)	(2,5)	
							(4,5)	(3,5)		
							(3,3)	(5,5)		
							(3,4)			

RWFlood description

	1	2	3	4	5
1	68	67	68	69	70
2	67	66	67	67	69
3	64	65	67	67	68
4	62	64	63	63	67
5	60	63	63	65	68

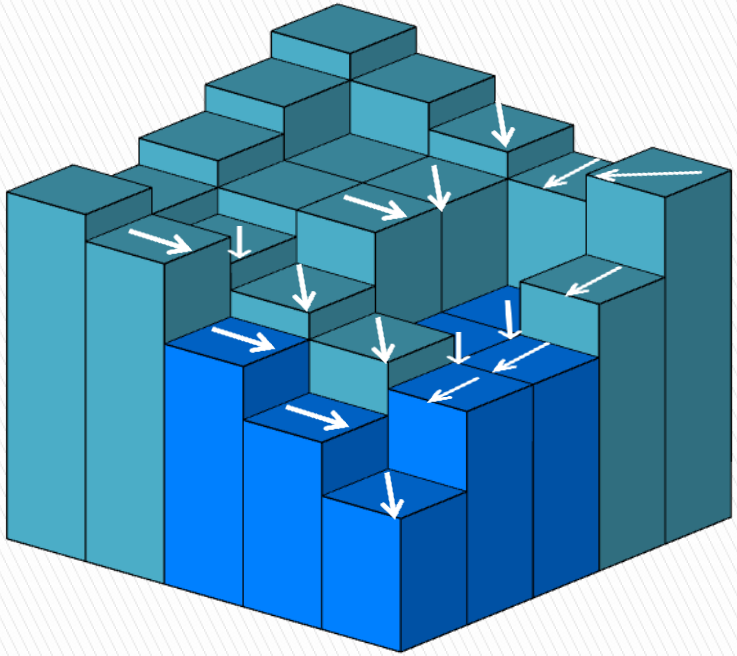


***F*=**

60	61	62	63	64	65	66	67	68	69	70
			(4,4)	(3,1)	(5,4)		(1,2)	(1,1)	(1,4)	(1,5)
				(4,2)	(3,2)		(2,1)	(1,3)	(2,5)	
							(4,5)	(3,5)		
							(3,3)	(5,5)		
							(3,4)			

RWFlood description

	1	2	3	4	5
1	68	67	68	69	70
2	67	66	67	67	69
3	64	65	67	67	68
4	62	64	63	63	67
5	60	63	63	65	68

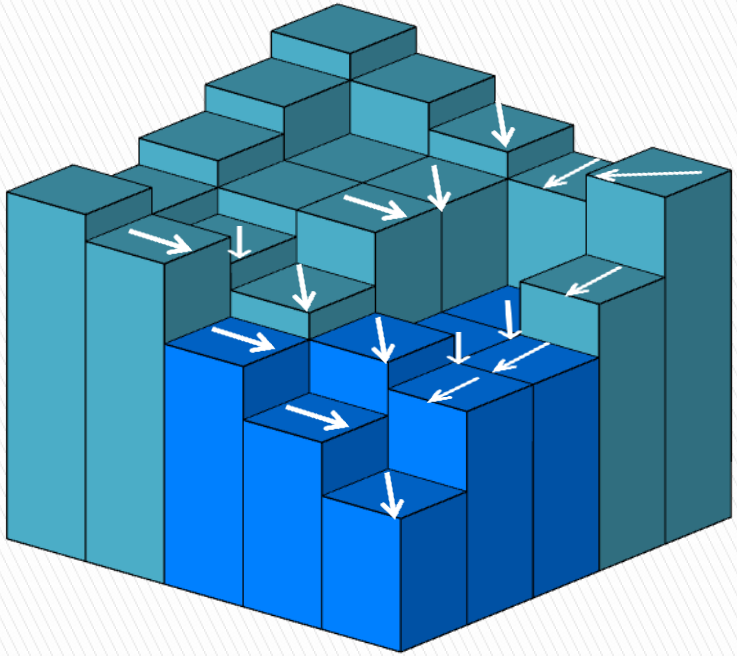


***F*=**

60	61	62	63	64	65	66	67	68	69	70
				(3,1)	(5,4)	(2,2)	(1,2)	(1,1)	(1,4)	(1,5)
				(4,2)	(3,2)		(2,1)	(1,3)	(2,5)	
							(4,5)	(3,5)		
							(3,3)	(5,5)		
							(3,4)			

RWFlood description

	1	2	3	4	5
1	68	67	68	69	70
2	67	66	67	67	69
3	64	65	67	67	68
4	62	64	63	63	67
5	60	63	63	65	68

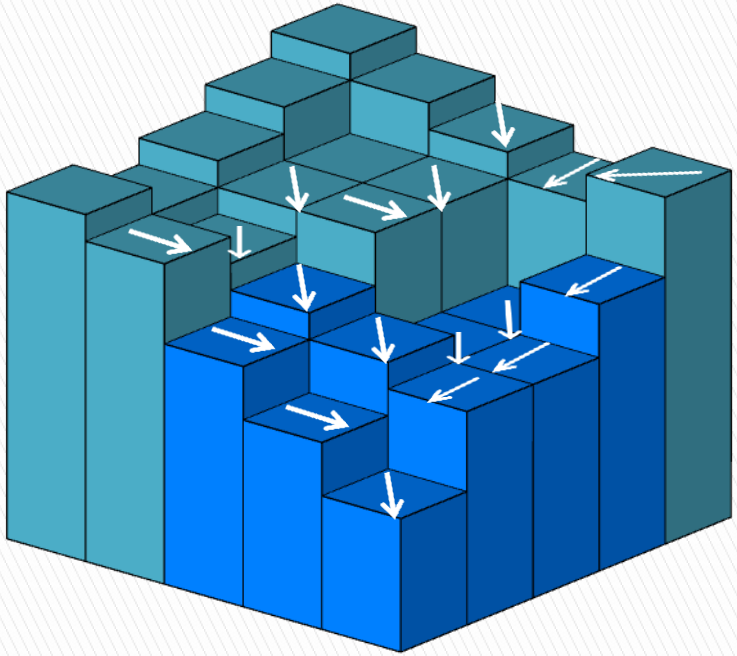


***F*=**

60	61	62	63	64	65	66	67	68	69	70
				(4,2)	(5,4)	(2,2)	(1,2)	(1,1)	(1,4)	(1,5)
					(3,2)		(2,1)	(1,3)	(2,5)	
							(4,5)	(3,5)		
							(3,3)	(5,5)		
							(3,4)			

RWFlood description

	1	2	3	4	5
1	68	67	68	69	70
2	67	66	67	67	69
3	64	65	67	67	68
4	62	64	63	63	67
5	60	63	63	65	68

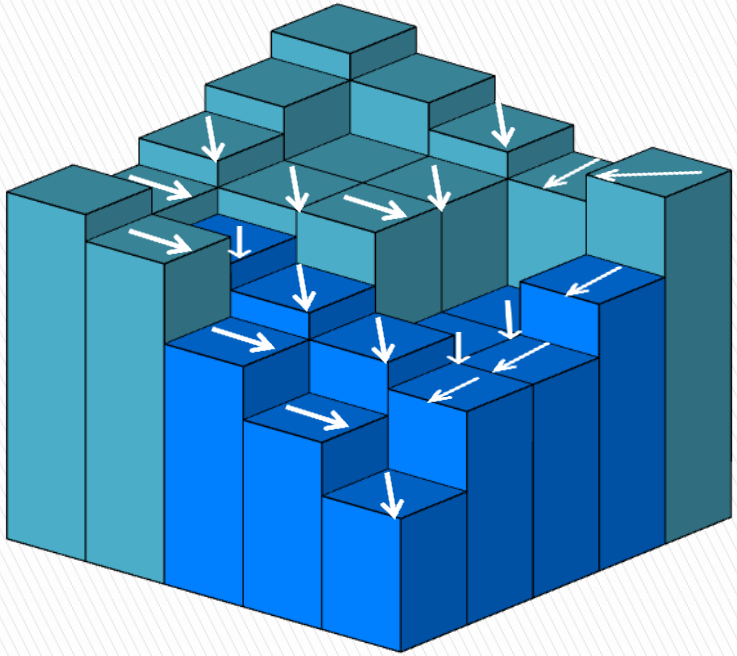


***F*=**

60	61	62	63	64	65	66	67	68	69	70
					(5,4) (3,2)	(2,2)	(1,2) (2,1) (4,5) (3,3) (3,4) (2,3)	(1,1) (1,3) (3,5) (5,5)	(1,4) (2,5)	(1,5)

RWFlood description

	1	2	3	4	5
1	68	67	68	69	70
2	67	66	67	67	69
3	64	65	67	67	68
4	62	64	63	63	67
5	60	63	63	65	68

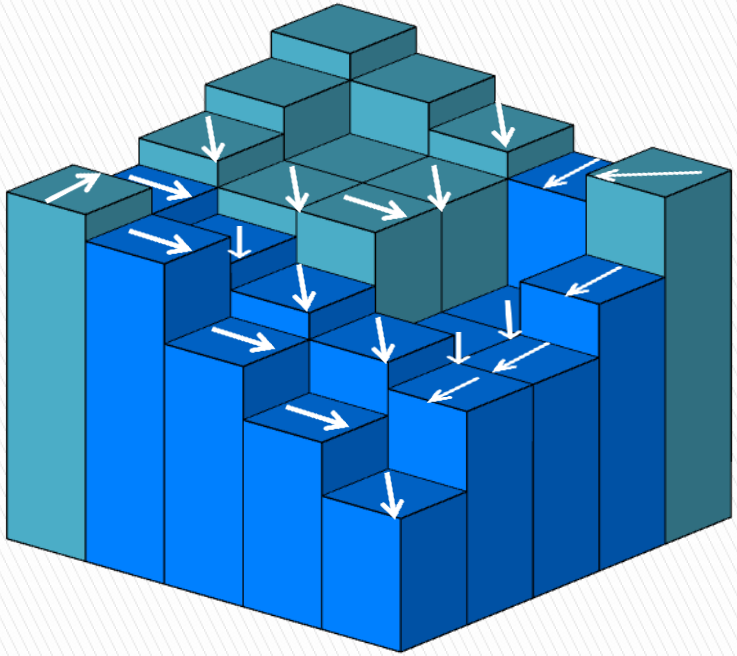


***F*=**

60	61	62	63	64	65	66	67	68	69	70
						(2,2)	(1,2)	(1,1)	(1,4)	(1,5)
							(2,1)	(1,3)	(2,5)	
							(4,5)	(3,5)		
							(3,3)	(5,5)		
							(3,4)			
							(2,3)			

RWFlood description

	1	2	3	4	5
1	68	67	68	69	70
2	67	66	67	67	69
3	64	65	67	67	68
4	62	64	63	63	67
5	60	63	63	65	68

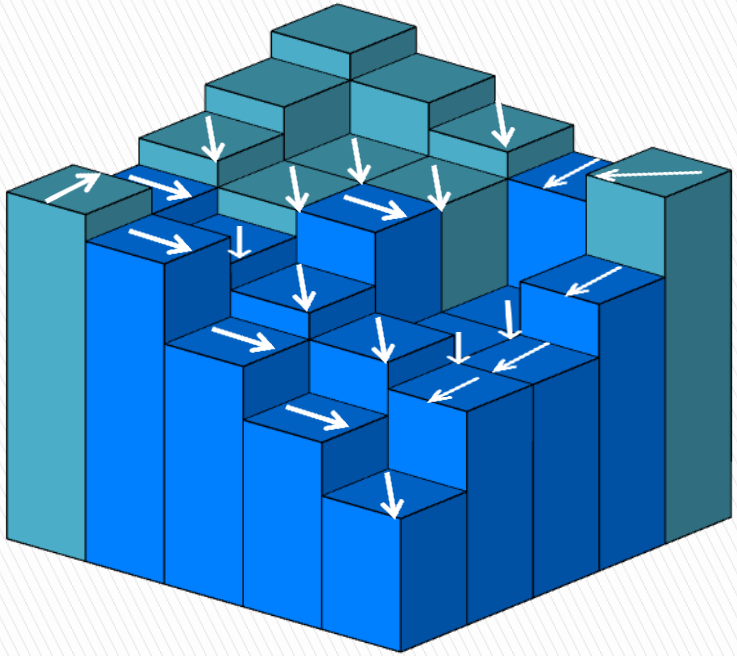


***F*=**

60	61	62	63	64	65	66	67	68	69	70
							(1,2)	(1,1)	(1,4)	(1,5)
							(2,1)	(1,3)	(2,5)	
							(4,5)	(3,5)		
							(3,3)	(5,5)		
							(3,4)			
							(2,3)			

RWFlood description

	1	2	3	4	5
1	68	67	68	69	70
2	67	66	67	67	69
3	64	65	67	67	68
4	62	64	63	63	67
5	60	63	63	65	68

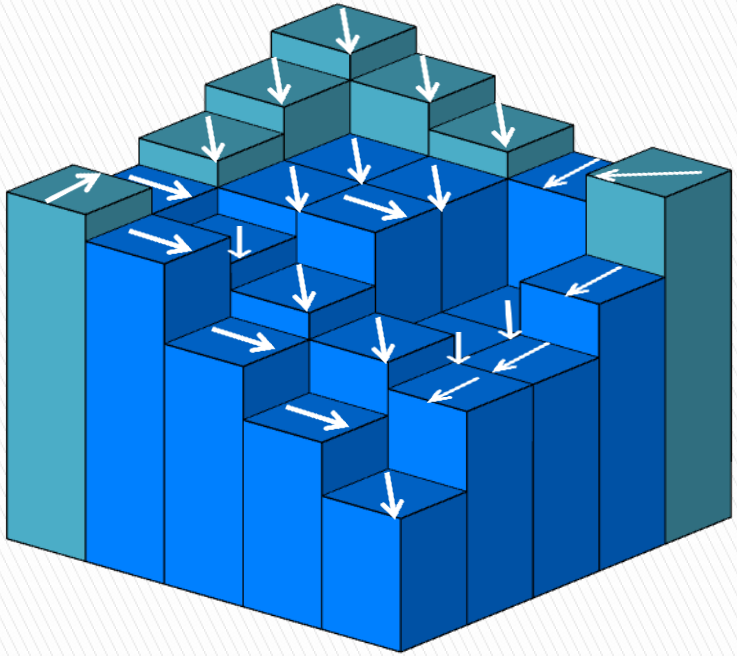


***F*=**

60	61	62	63	64	65	66	67	68	69	70
							(3,3)	(1,1)	(1,4)	(1,5)
							(3,4)	(1,3)	(2,5)	
							(2,3)	(3,5)		
							(2,4)	(5,5)		

RWFlood description

	1	2	3	4	5
1	68	67	68	69	70
2	67	66	67	67	69
3	64	65	67	67	68
4	62	64	63	63	67
5	60	63	63	65	68

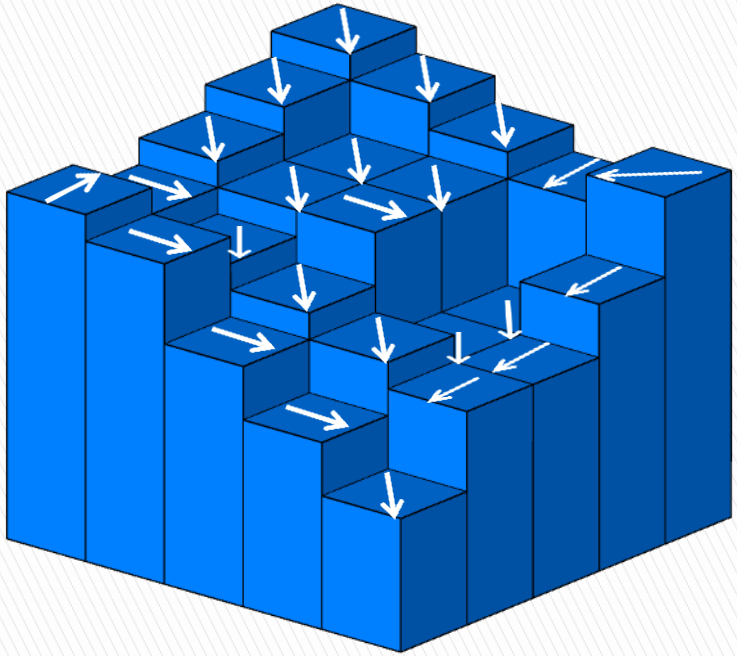


***F*=**

60	61	62	63	64	65	66	67	68	69	70
							(3,4)	(1,1)	(1,4)	(1,5)
							(2,3)	(1,3)	(2,5)	
							(2,4)	(3,5)		
								(5,5)		

RWFlood description

	1	2	3	4	5
1	68	67	68	69	70
2	67	66	67	67	69
3	64	65	67	67	68
4	62	64	63	63	67
5	60	63	63	65	68



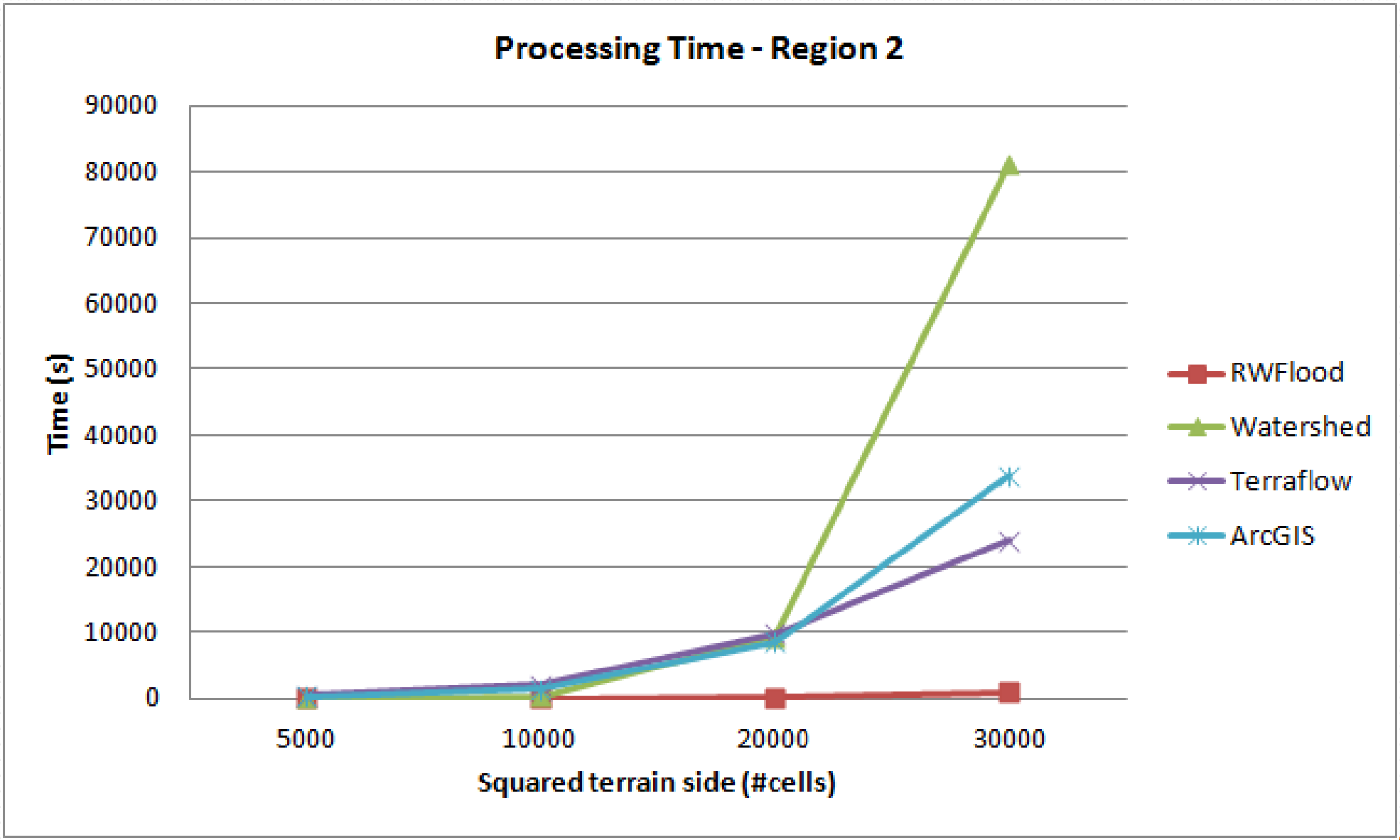
***F*=**

60	61	62	63	64	65	66	67	68	69	70
								(1,1)	(1,4)	(1,5)
								(1,3)	(2,5)	
								(3,5)		
								(5,5)		

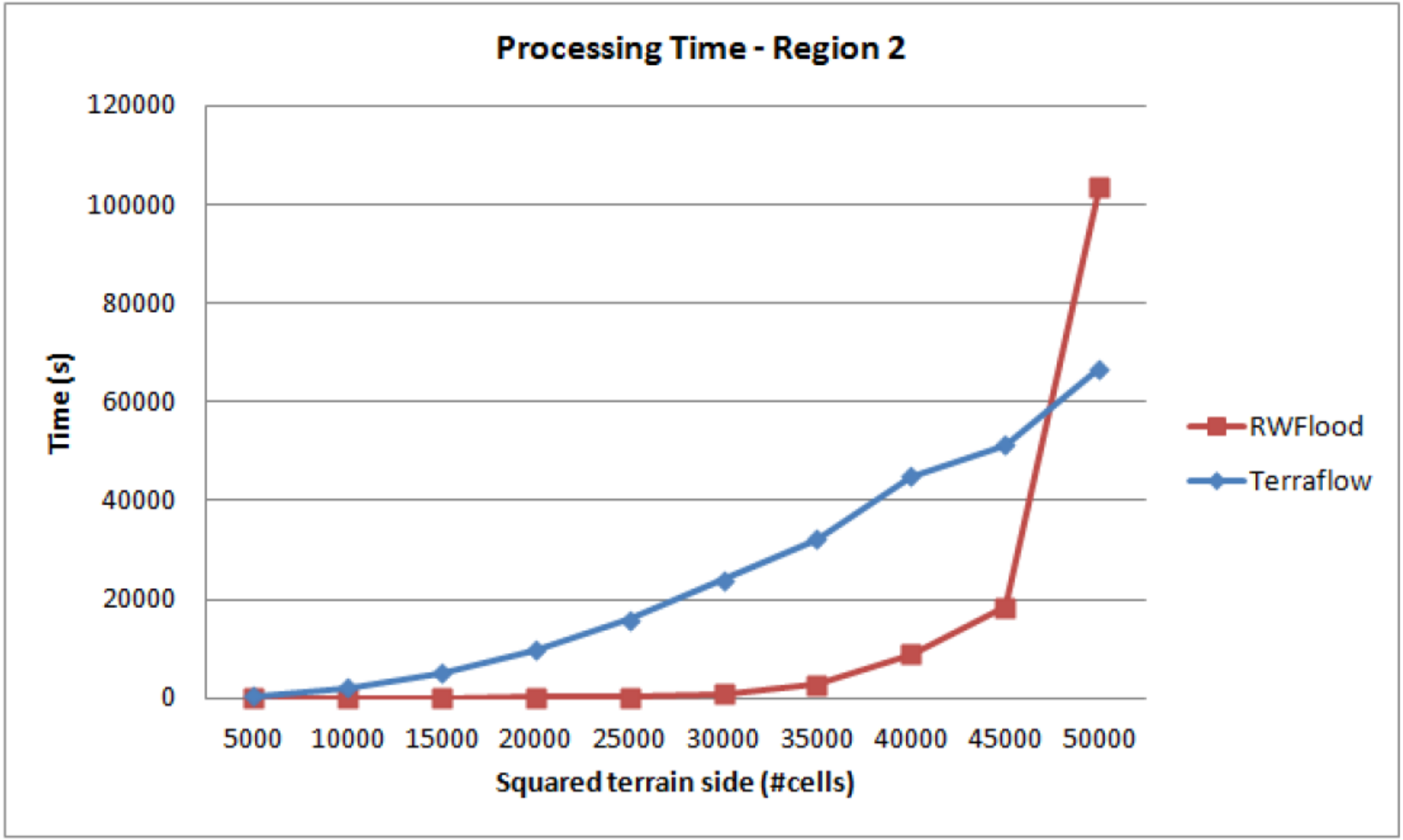
RWFlood : Experimental analysis

Terrain	Size [# cells]	Processing time (in seconds) RWFlood	r.watershed	r.terraflow	ArcGIS
Tres Marias	5000^2	5	47	405	293
	10000^2	14	233	2075	3860
	20000^2	68	8776	9924	17509
Tapajos	5000^2	3	48	401	376
	10000^2	16	242	2059	2869
	20000^2	73	9063	10015	13707
Region 3	5000^2	5	44	411	219
	10000^2	27	231	2106	1586
	20000^2	125	9185	10140	7693
	30000^2	1062	74135	24746	26338
Region 2	5000^2	5	46	389	264
	10000^2	27	246	2038	1449
	20000^2	145	9374	9804	8546
	30000^2	912	81195	24013	33829

RWFlood : Experimental analysis



RWFlood : Experimental analysis



RWFlood : Experimental analysis

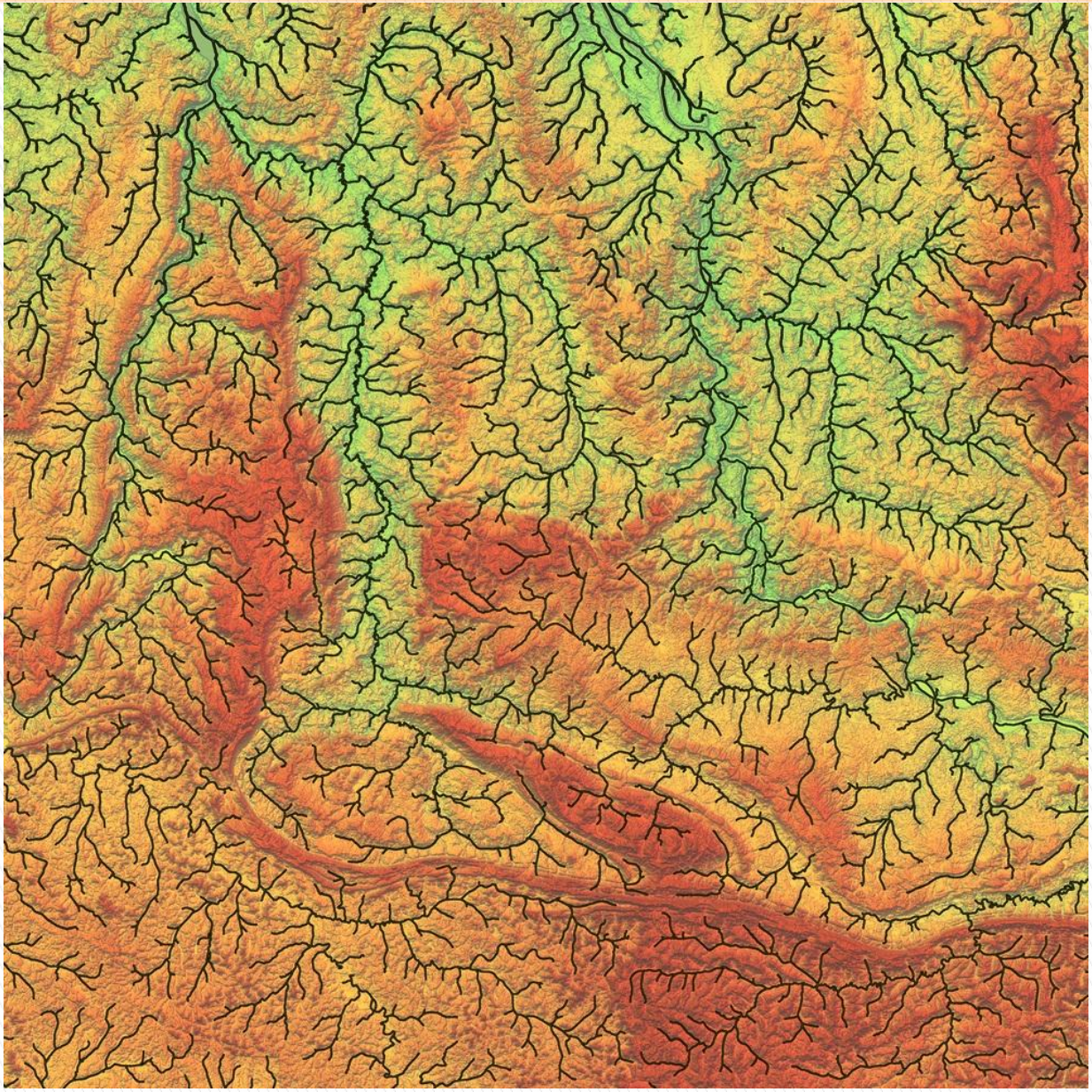
- Table comparing the memory used by the 3 methods: r.watershed, r.terraflow and RWFlood
- --- Tests are still running ----

RWFlood memory usage is given by:

$$56000 + 6N,$$

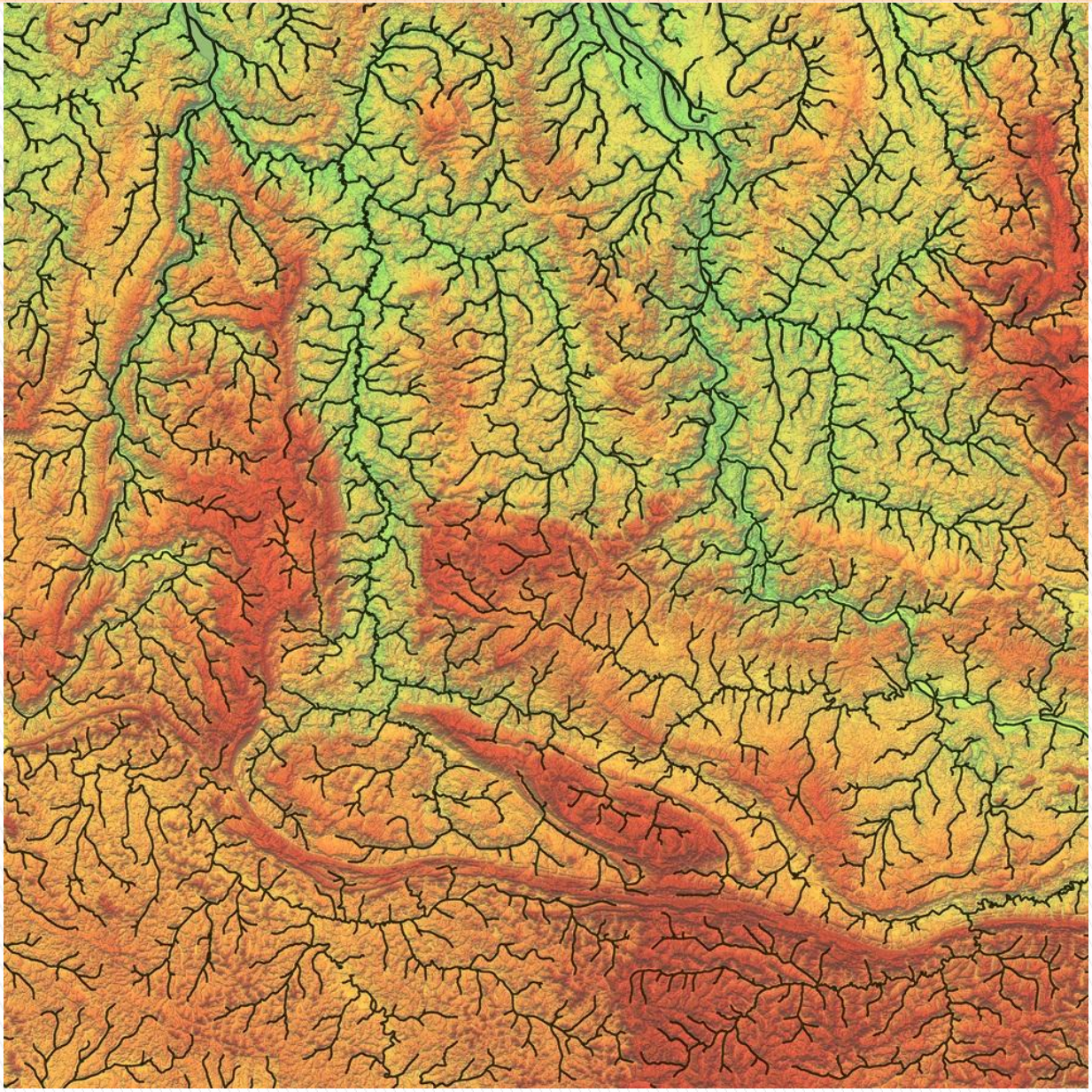
where N is the number of cells

RWFlood : Experimental analysis



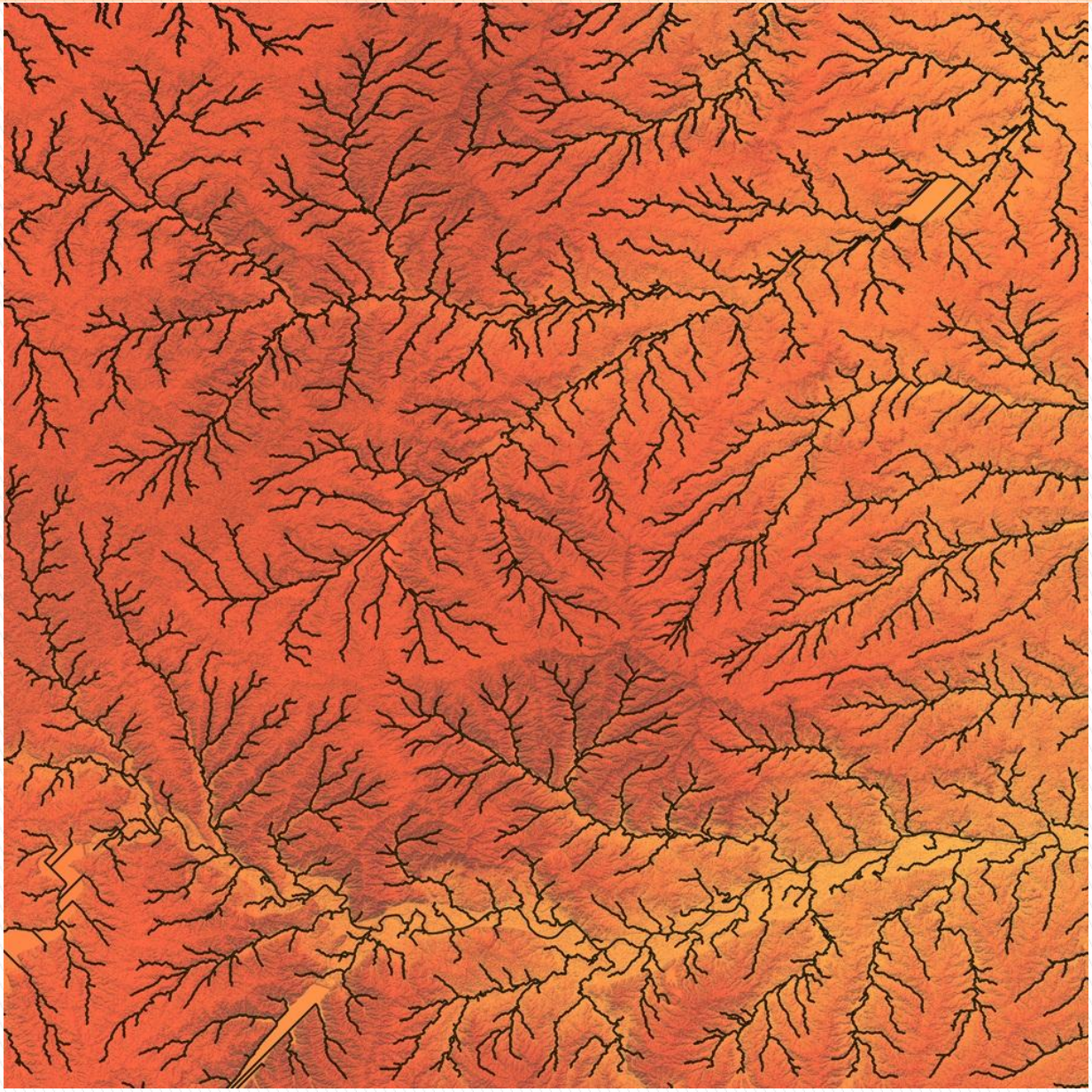
Drainage network computed by RWFlood

RWFlood : Experimental analysis



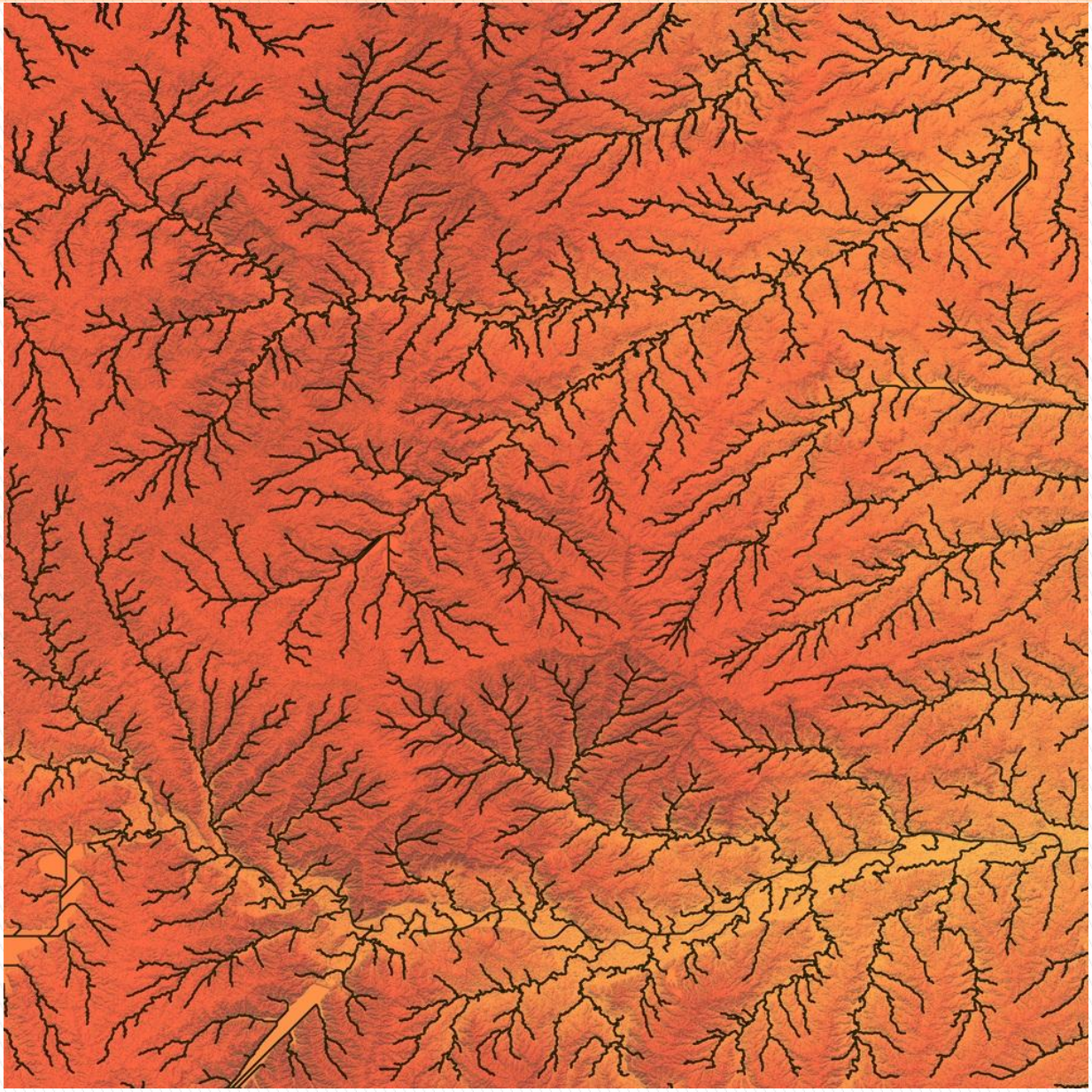
Drainage network computed by GRASS

RWFlood : Experimental analysis



Drainage network computed by RWFlood

RWFlood : Experimental analysis



Drainage network computed by GRASS

Conclusions and future work

- We developed a very fast and simple algorithm to compute the drainage network on grid terrains
- The algorithm doesn´t require a preprocessing step to remove depressions and flat areas
- It is linear in the number of cells in the terrain
- As a next step, the algorithm has been adapted for external memory processing

Conclusions and future work

- RWFlood code, in C++, is available in:

  www.dpi.ufv.br/~marcus/RWFlood

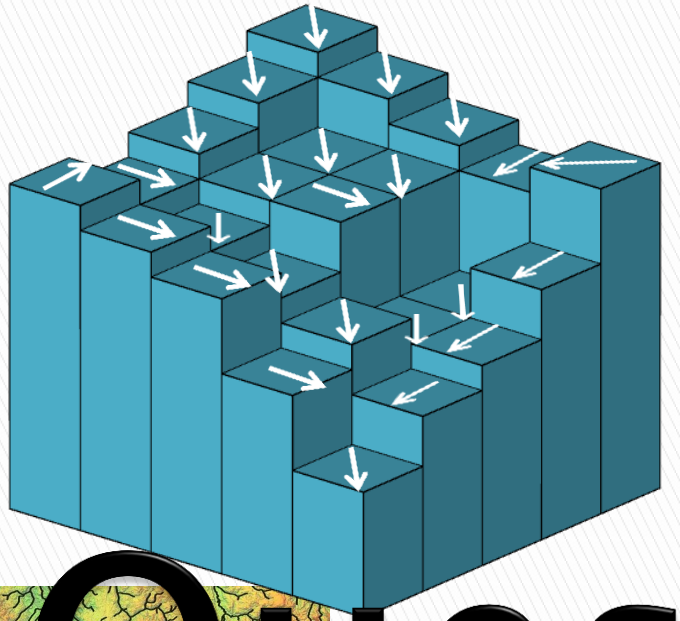
- Contact:

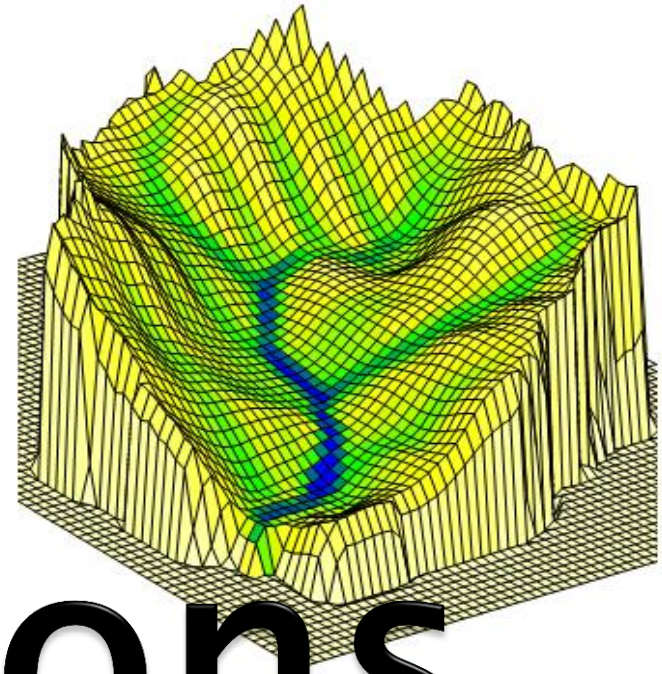
  marcus.ufv@gmail.com
  marcus@dpi.ufv.br

Acknowledgements

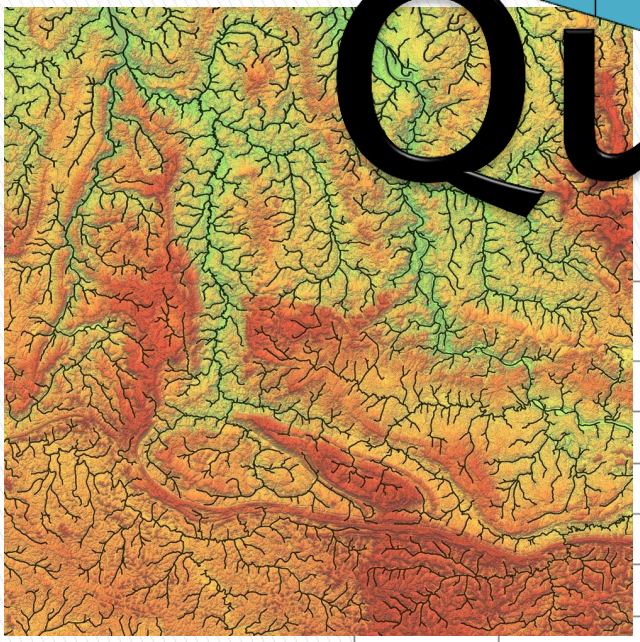
Universidade Federal de Viçosa

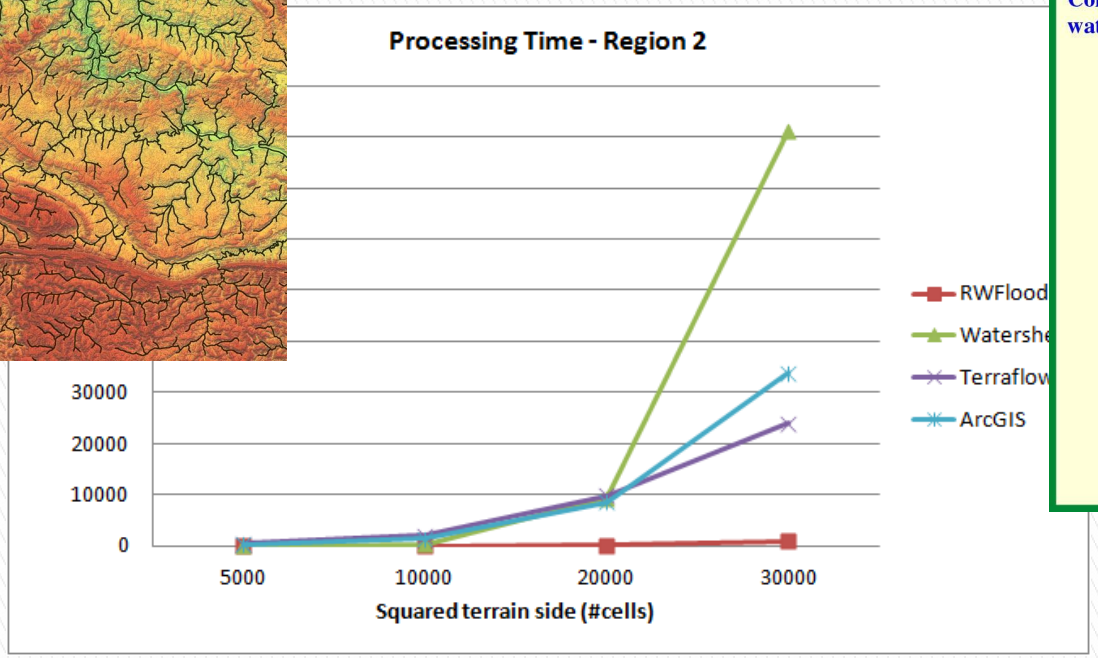
Grants CMMI-0835762 and IIS-1117277

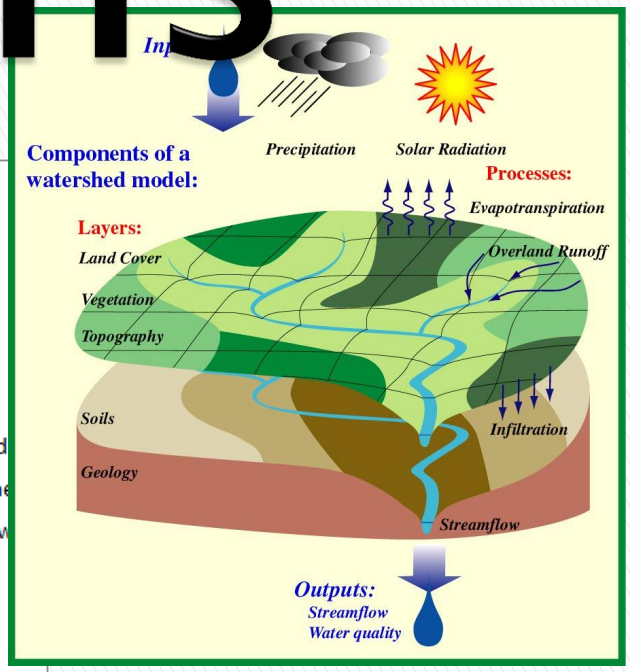




Questions




Processing Time - Region 2
30000
20000
10000
0
5000
10000
20000
30000
Squared terrain side (#cells)
RWFlood
Watershe
Terraflow
ArcGIS


Inp
Precipitation
Solar Radiation
Components of a watershed model:
Processes:
Evapotranspiration
Layers:
Land Cover
Overland Runoff
Vegetation
Topography
Soils
Infiltration
Geology
Streamflow
Outputs:
Streamflow
Water quality