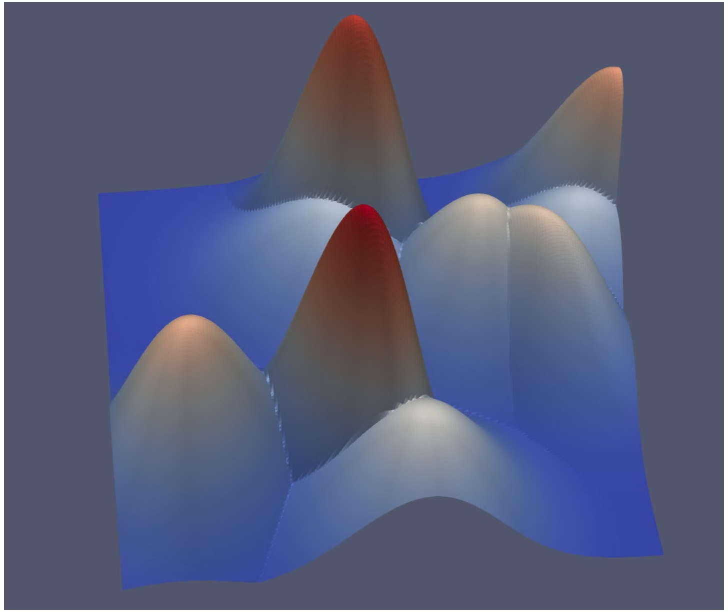
Reverse Engineer a Merge Tree from Topology Information

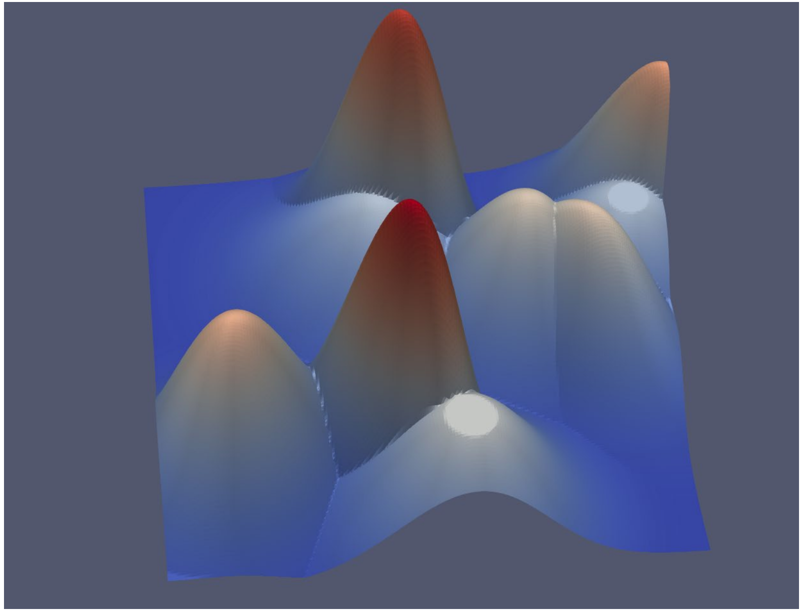
He Chen, Tart Patel

Merge Tree From Scalar Field

Topological Simplification

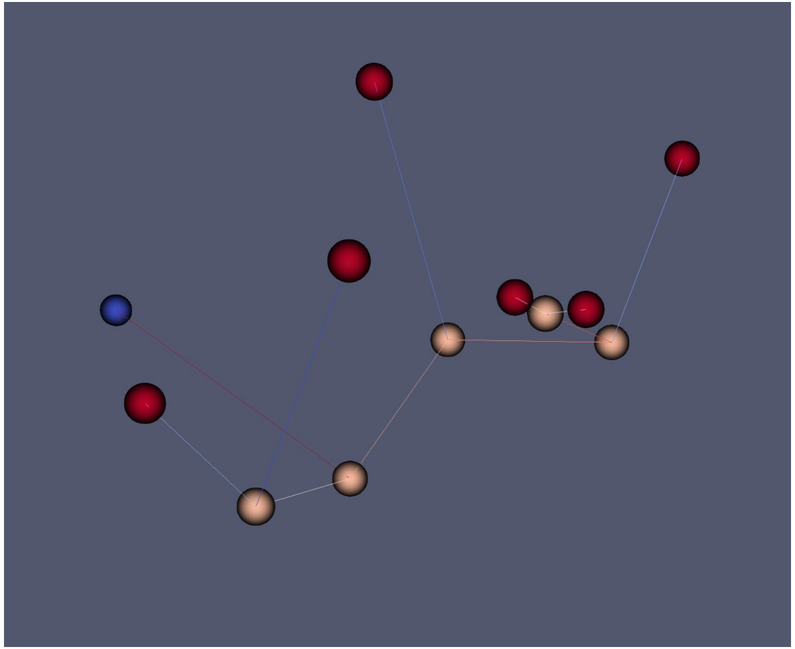


Input

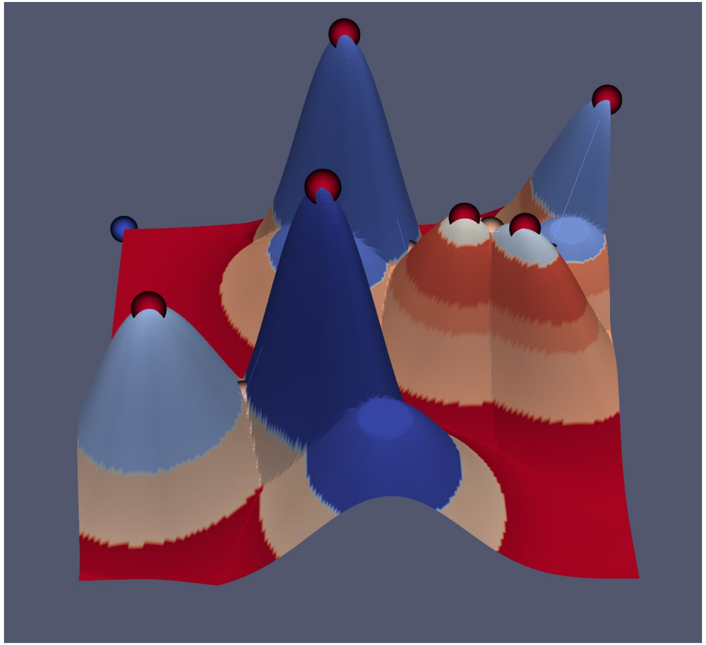


Simplified

Merge Tree

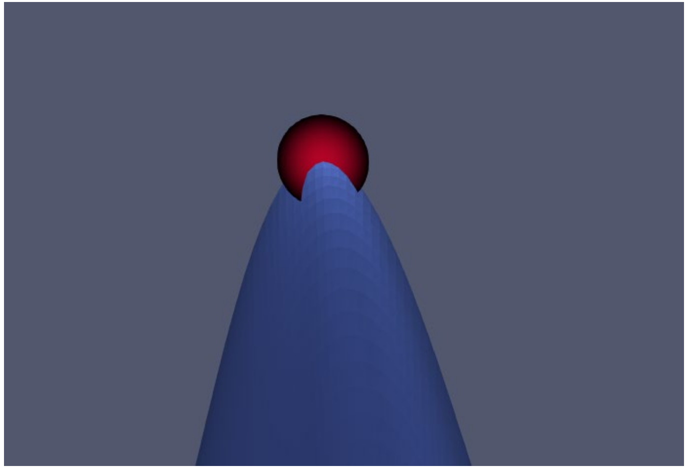


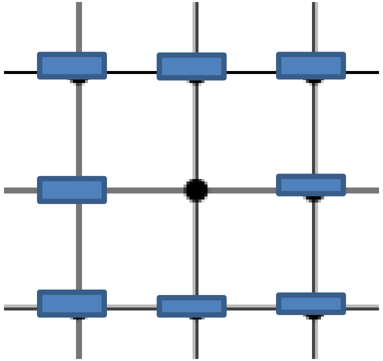
Merge tree



Segmentation

Critical Points: Local Maximum

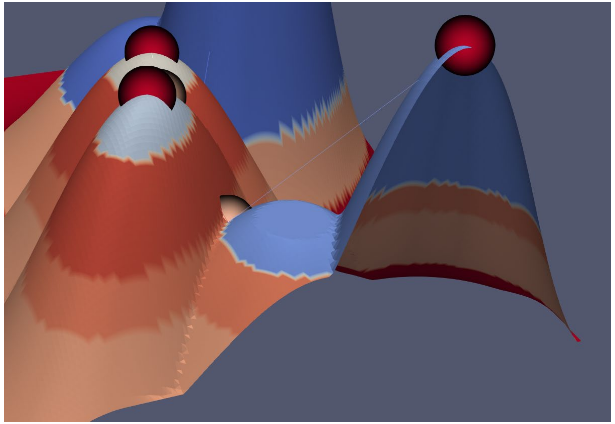


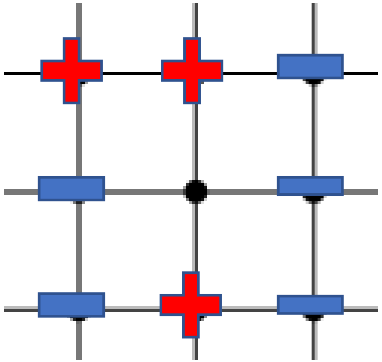


Local Maximum

Corresponds to Leaf Node in the Merge Tree

Critical Points: Saddle point



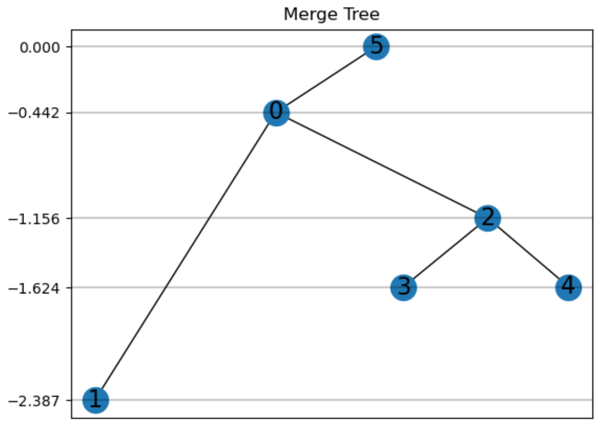


Saddle point

Where two or more connected component join

Merge Tree to Induced Matrix

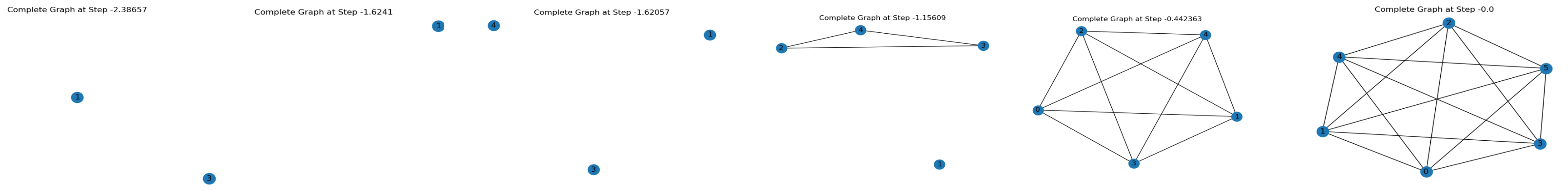
- Induced matrices are an intemediate form for interpolation
- Data Layout
  - Symmetric
  - The diagonal values M_{i,i} are the function value of node i
  - The value M_{i,j} is the function value where nodes i and j merge



0	0	0	0	0	5
	1	0	0	0	5
		2	2	2	5
			3	2	5
				4	5
					5

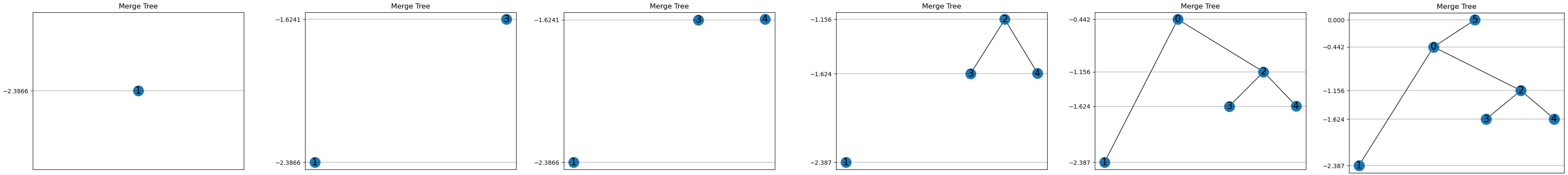
Induced Matrix to Merge Tree

- To construct a merge tree from an induced matrix, the nodes and edge of the tree must be sorted by the function value
- Then all of the function values are traversed in order and any nodes or edges are added to the graph
- The steps are represented by complete graphs that are fully connected by the last step



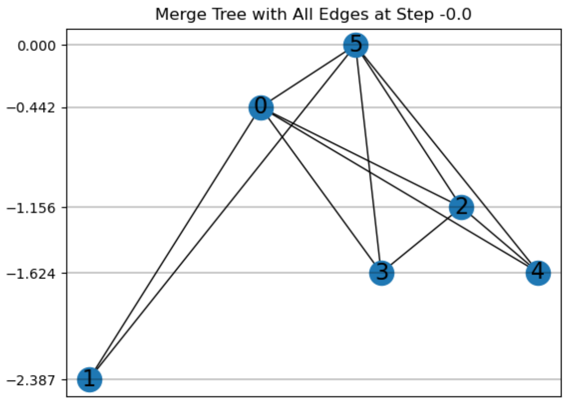
Induced Matrix to Merge Tree

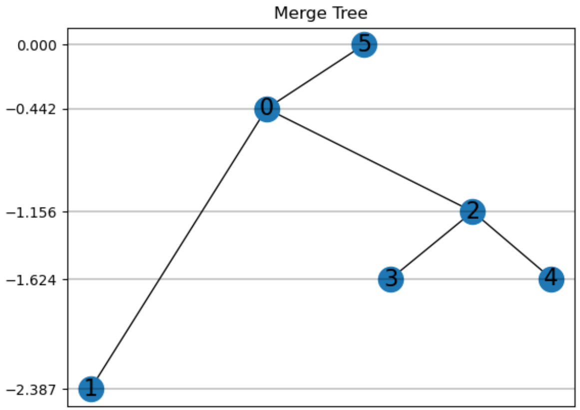
- To construct a merge tree from an induced matrix, the nodes and edge of the tree must be sorted by the function value
- Then all of the function values are traversed in order and any nodes or edges are added to the graph
- The steps are represented by complete graphs that are fully connected by the last step



Induced Matrix to Merge Tree

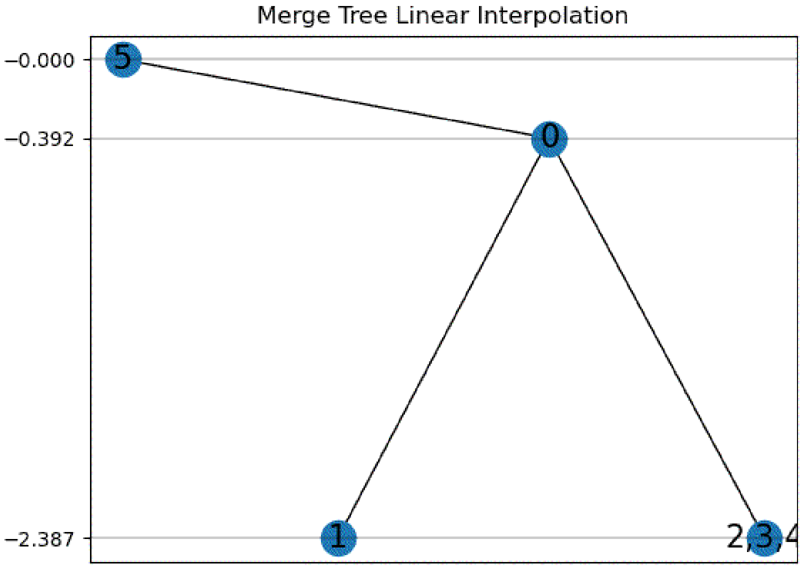
- Adding edges to the graph in the order of steps left redundant connections
  - With entirely labeled nodes, the only case is when the connecting node appears in the middle of the steps
  - With unlabeled nodes, the edge that needs to be removed depends on its function value





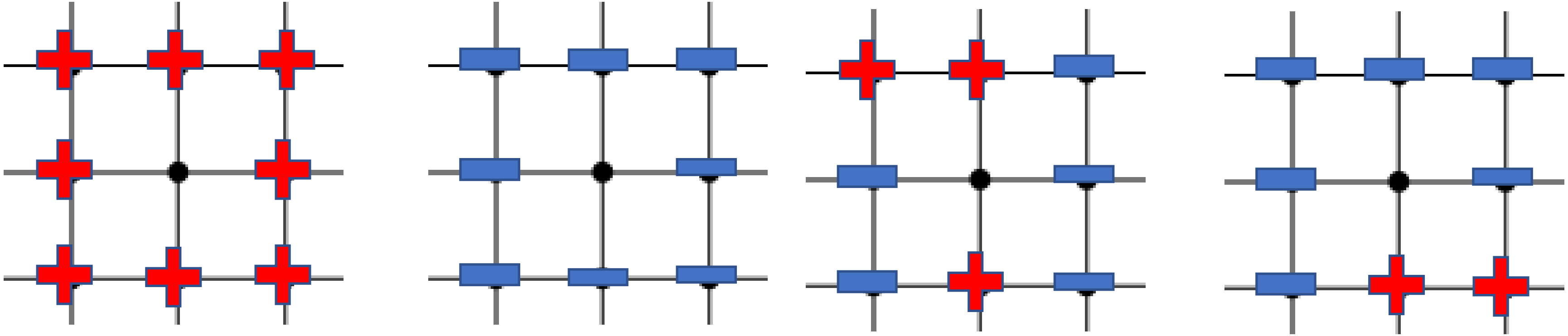
Merge Tree Interpolation

- Interpolation can be done in the induced matrix form
- Merge tree interpolation methods
  - Linear interpolation
    - Simple and fast
  - Geodesic interpolation
    - More computationally expensive,
    - but produces better results



Critical Points: Discrete Definition

[Charles et al. 2017] [T. F. Banchoff. et al. 1970]



Local maximum

Local minimum

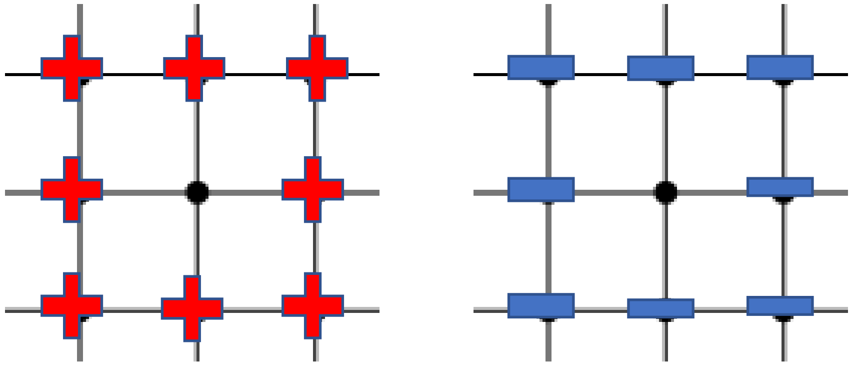
Saddle point:
Inclining directions are separatedby declining directions

Regular point:
Inclining directions and declining directions are simply connected

Points with higher value

Points with lower value

Critical Points: Discrete Definition



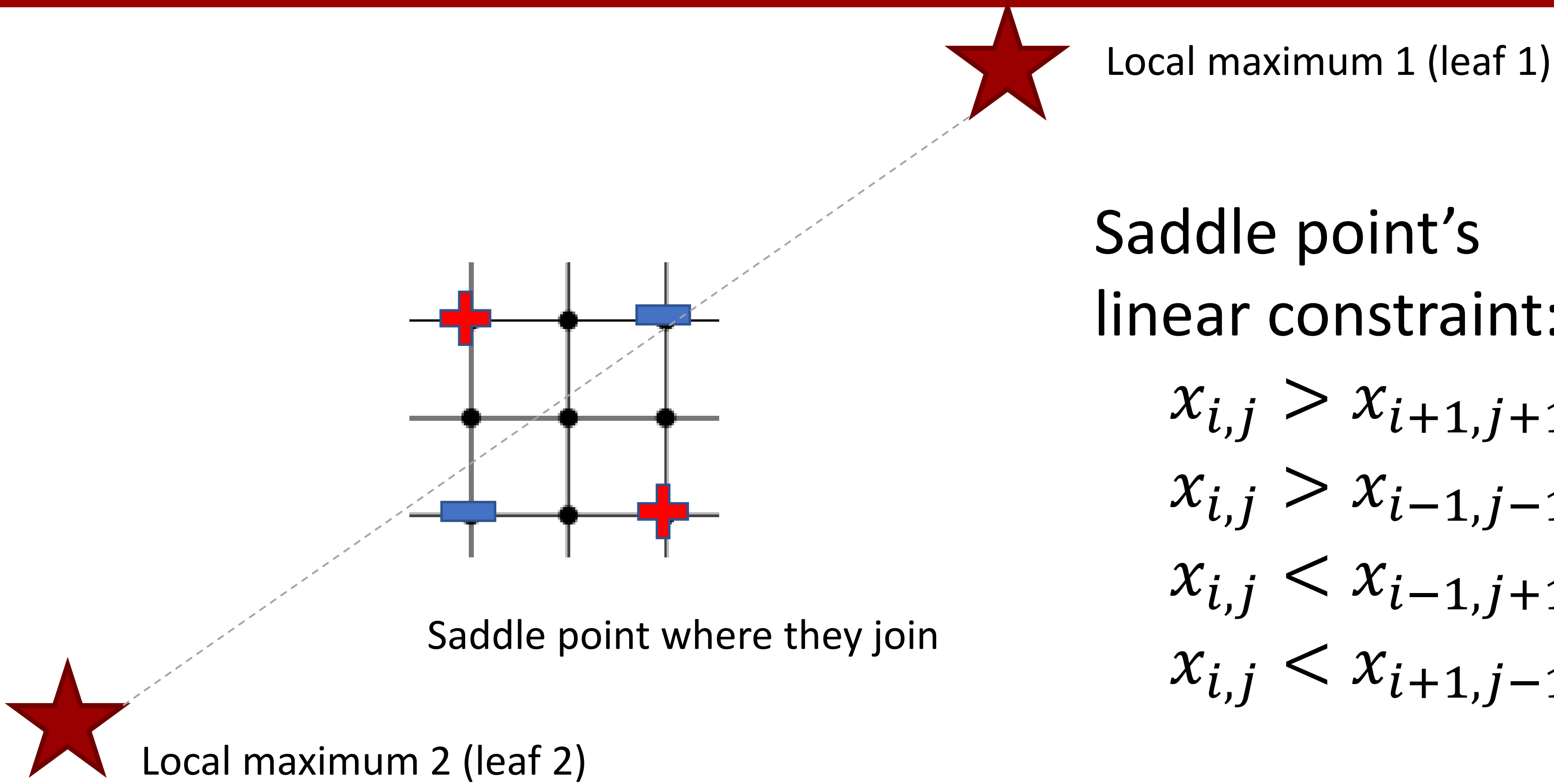
Local maximum   Local minimum

For local maximum/minimum:

$$x_{i,j} > x_{i+1,j}$$
$$x_{i,j} > x_{i,j+1}$$
$$x_{i,j} > x_{i+1,j+1}$$
$$x_{i,j} > x_{i,j-1}$$

.....

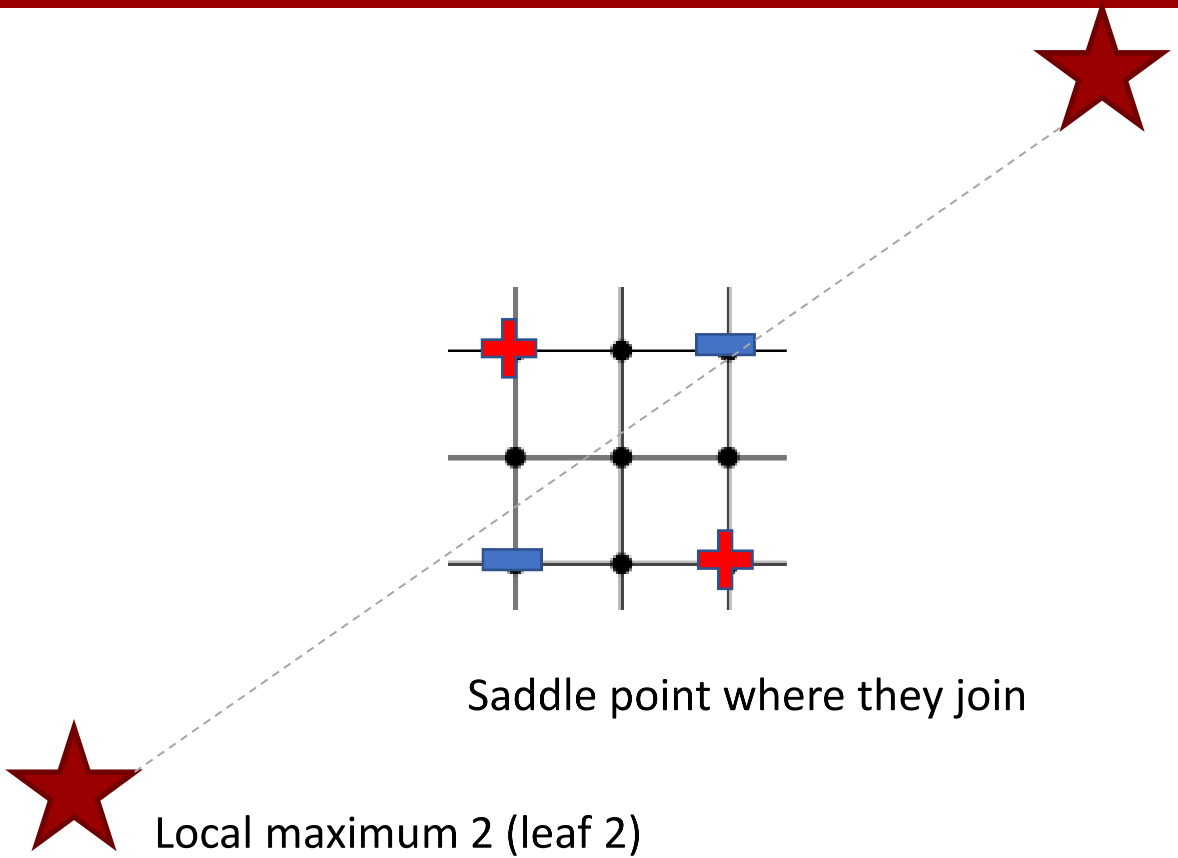
Critical Points: Discrete Definition



Saddle point's linear constraint:

$$x_{i,j} > x_{i+1,j+1}$$
$$x_{i,j} > x_{i-1,j-1}$$
$$x_{i,j} < x_{i-1,j+1}$$
$$x_{i,j} < x_{i+1,j-1}$$

Critical Points: Discrete Definition

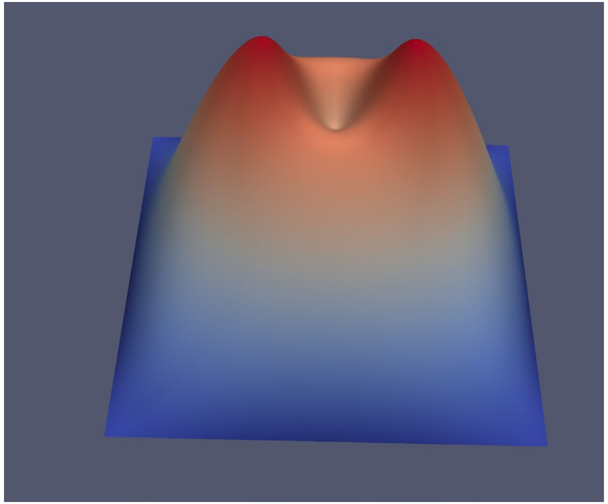


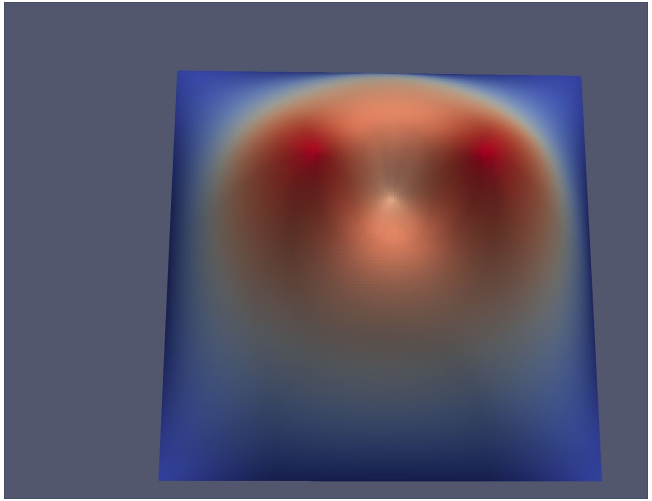
Saddle point's linear constraint:

$$x_{i,j} > x_{i+1,j+1}$$
$$x_{i,j} > x_{i-1,j-1}$$
$$x_{i,j} < x_{i-1,j+1}$$
$$x_{i,j} < x_{i+1,j-1}$$

Critical Points: Discrete Definition

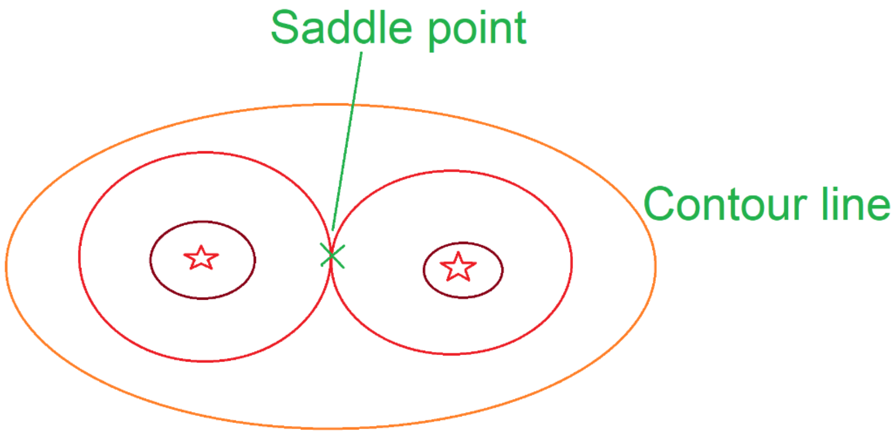
Local constraint is not sufficient for creating saddle points.





Contour Line Constraint

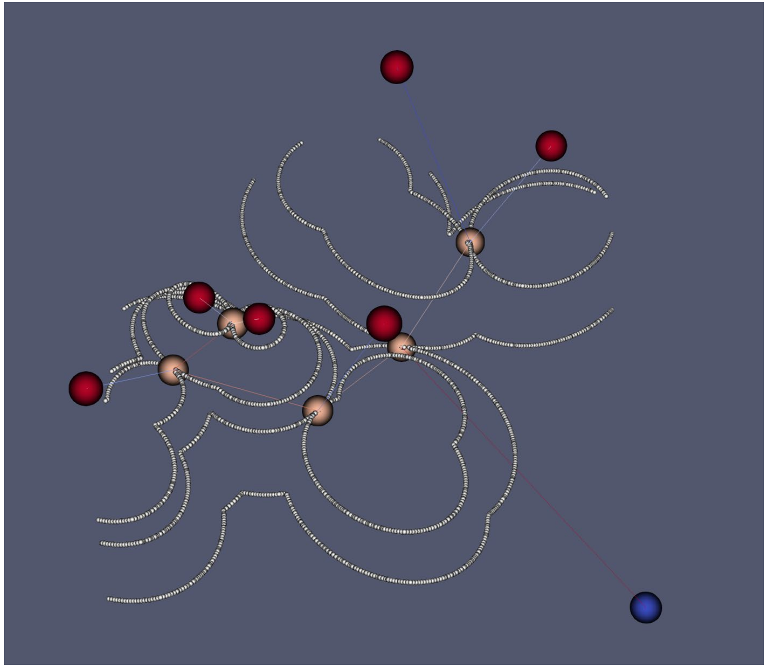
We have to use global constraint to make sure components join at a certain saddle point.

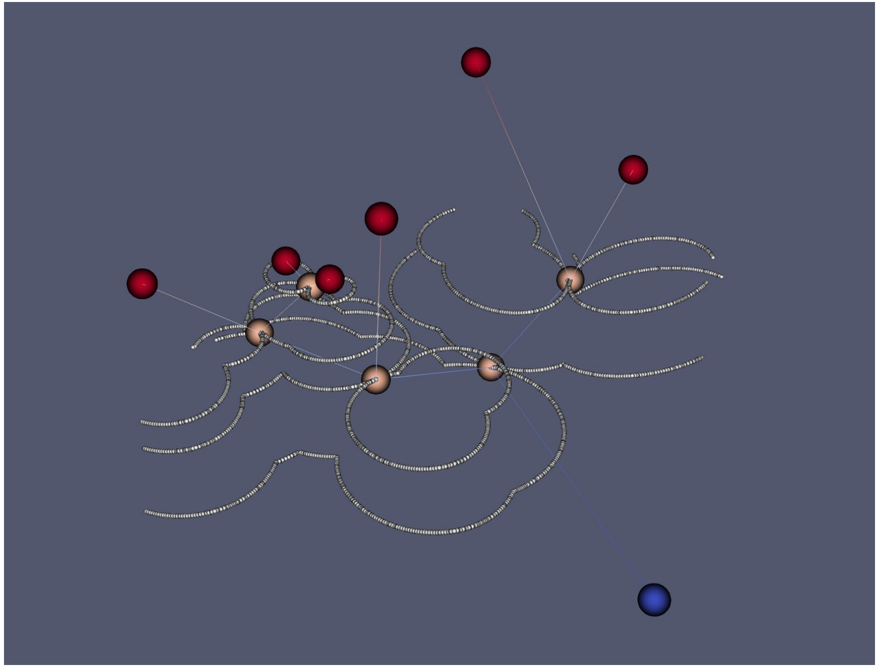


The solution is to add contour line to make sure the two component contacts exactly at the saddle point

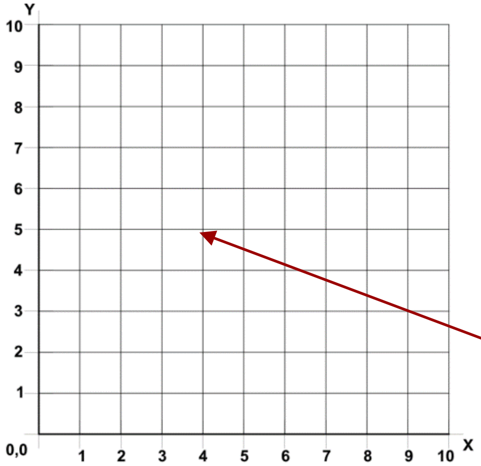
Constraints:

Critical points constraints (local, inequality) + contour line constraints (global, equility)





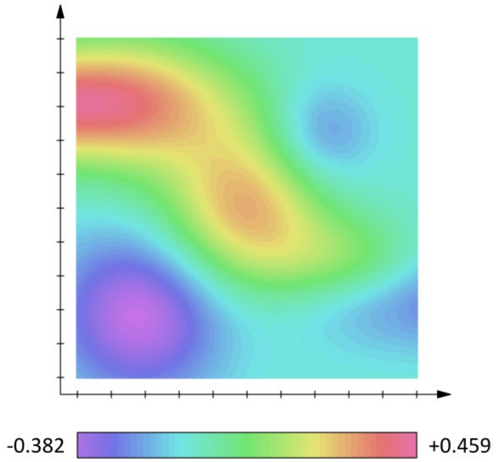
Fill the Rest of the Points



$$X = \begin{bmatrix} x_{1,1} & \cdots & x_{1,N} \\ \vdots & \ddots & \vdots \\ x_{N,1} & \cdots & x_{N,N} \end{bmatrix} \in R^{N \times N}$$

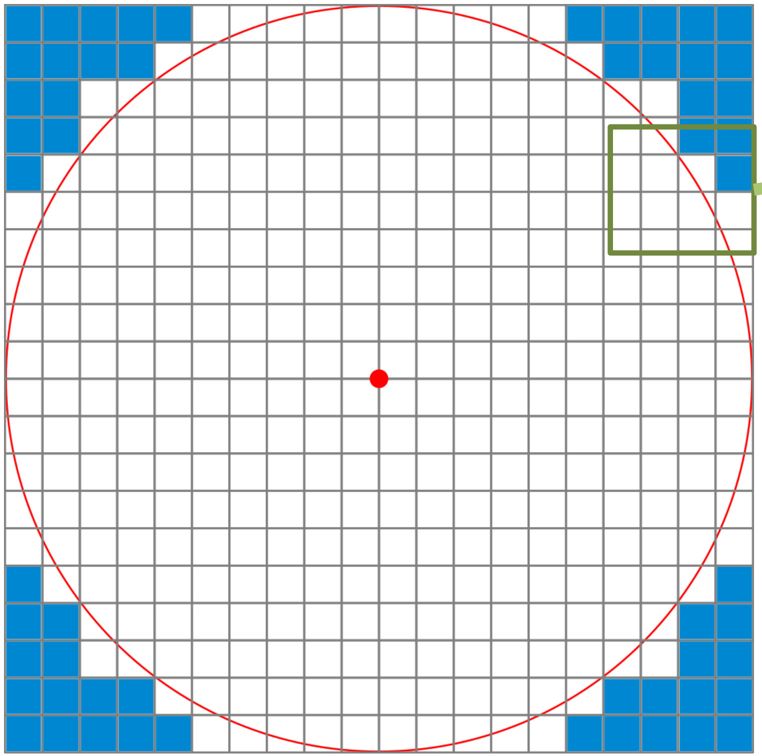
$$x_{i,j} \in R$$

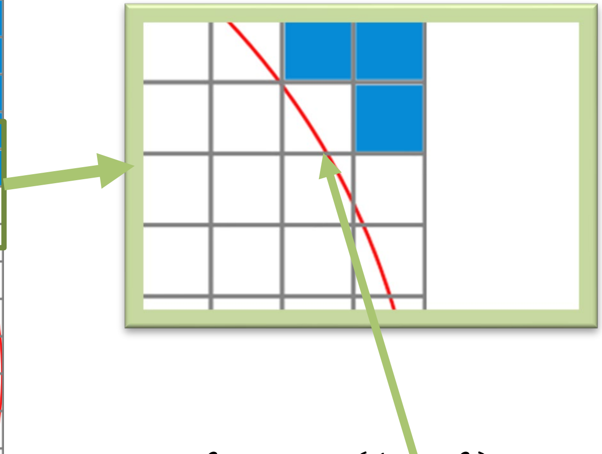
$$x_{5,4}$$



Define scalar field on a 2D regular grid (image)

Contour Line Constraint





$$\lambda x_{i,j} + (1 - \lambda) x_{i,j+1} = h$$

Fill the Rest of the Points

Sum of per-pixel Laplacian:

$$X = argmin_X \sum_{\substack{1 \le i \le N, \\ 1 \le j \le N}} (2x_{i,j} - x_{i-1,j} - x_{i+1,j})^2 + (2x_{i,j} - x_{i,j-1} - x_{i,j+1})^2$$

s.t.:

For local maximum:

$$x_{i,j} > x_{i+1,j}$$
$$x_{i,j} > x_{i,j+1}$$
$$x_{i,j} > x_{i+1,j+1}$$
$$x_{i,j} > x_{i,j-1}$$

.....

Saddle constraint:

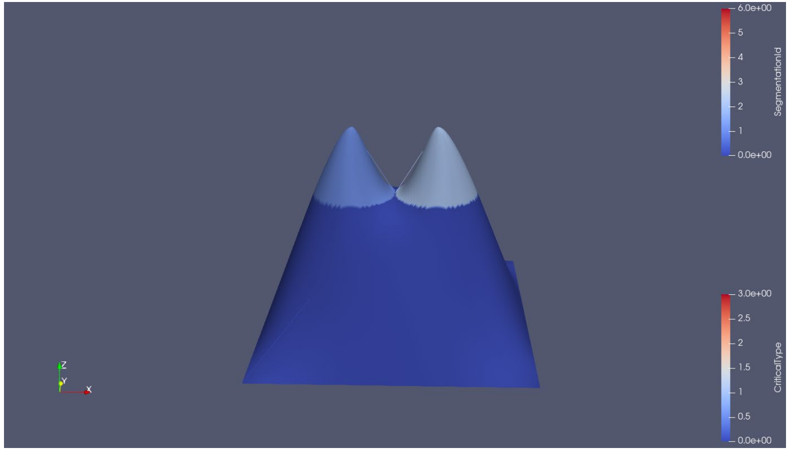
$$x_{i,j} > x_{i+1,j+1}$$
$$x_{i,j} > x_{i-1,j-1}$$
$$x_{i,j} < x_{i-1,j+1}$$
$$x_{i,j} < x_{i+1,j-1}$$

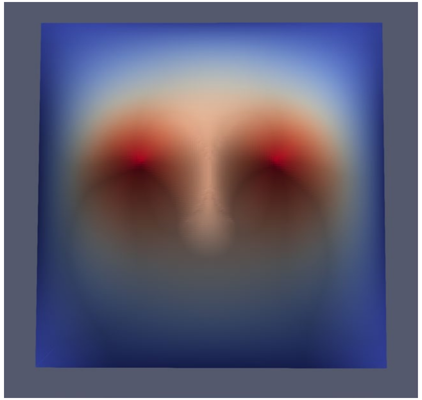
Contour line constraint:

$$\lambda x_{i,j} + (1 - \lambda) x_{i,j+1} = h$$

All linear equality / inequality constraint

Example


6.0e+00
5
4
3
2
1
0.0e+00
SegmentationId
3.0e+00
2.5
2
1.5
1
0.5
0.0e+00
CriticalType



Example