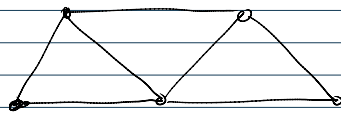


# Piece-wise linear function

attach a number to every vertex and interpolate linearly inside the triangle and on the edge



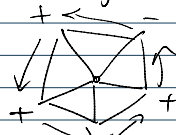
this is not exact, but can be made  
arbitrarily close

critical points must be the vertices

max :  $>$  all neighbors (if = happens, we break tie based on the order of the vertices)

min :  $<$  all neighbors

saddle :



walk around

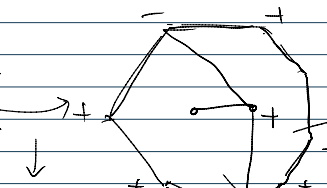
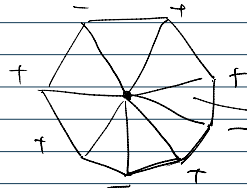
if there is interleaving of + and - then it is the regular point

else it is critical point

Count the number of connected components

Monkey saddle

make the going up and down to 2



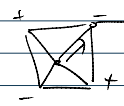
the other side has complexity -1

so we can do this recursively

this is the requirement (a monkey saddle would not have 2 gradient lines going through it)

fix the mesh by splitting the center point and reduce the number of going up and down

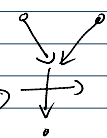
Start from saddle points, go up/down, and keep going in the same direction



2 problems

end up in another saddle

2 lines can collide



we can treat this as a narrow tube instead

use the same trick

to avoid going through another saddle

(sort the saddle points and go from the lowest)

and keep going down the line using a consistent rule (like the lowest vertex)



simulation differentiability