2D V-cell Algorithm

- Let \( P \) be the set of input points in 2D space
- \( \text{DELAUNAY2D}(P) \) returns the triangulation for a given set of points \( P \), which represent the V-cell centers
- \( \text{CIRCUMCENTER}(t) \) returns the center of circle which passes through all vertices of the given triangle \( t \)
- \( \text{GETNEXTVERTEX}(v_1, v_2, t.id) \) takes two vertices, \( v_1 \) and \( v_2 \), and the id of the current triangle, \( t.id \), and returns the next triangle with forming points \( v_1 \) and \( v_2 \) and which also shares an edge with \( t.id \), without backtracking
- \( \text{COUNTERCLOCKWISE}(\text{pointsList}) \) takes a list of points and returns the points in sorted counterclockwise order
- \( \text{CREATEPOLYGON}(\text{pointsList}) \) takes a list of counterclockwise points and creates a polygon

procedure \( \text{CALCULATEVCELLS2D}(P) \)

\[
T \leftarrow \text{DELAUNAY2D}(P)
\]

for all \( p_i \in P \) do
  for all \( t_j \in T \) do
    if \( t_j \).contains\( (p_i) \) then
      centerFormingPoint \( \leftarrow t_j\).pointOne
      secondFormingPoint \( \leftarrow t_j\).pointTwo
      currentTriangle \( \leftarrow t_j\)
      startTriangle \( \leftarrow t_j\)
      triangleList.add\( (currentTriangle)\)
      circumcenterPointsList.add\( (\text{CIRCUMCENTER}(currentTriangle))\)
      repeat
        \( t_k \leftarrow \text{GETNEXTVERTEX}(\text{centerFormingPoint}, \text{secondFormingPoint}, \text{currentVertex})\)
        currentVertex \( \leftarrow t_k\)
        secondFormingPoint \( \leftarrow t_k\).pointTwo
        circumcenterPointsList.add\( (\text{CIRCUMCENTER}(currentTriangle))\)
      until currentTriangle.equals\( (startTriangle)\)
      \( v\text{Cells}.add(p_i\).id\)
      \( v\text{Centers}.add(circumcenterPointsList)\)
    endif
  end for
end for

for all circumcenterPointsList \( \in v\text{Centers} \) do
  counterClockwisePointsList \( \leftarrow \text{COUNTERCLOCKWISE}(\text{circumcenterPointsList}_i)\)
  \( v\text{CellPolygonList}.add(\text{CREATEPOLYGON}(\text{counterClockwisePointsList}))\)
  \( v\text{CellCentersList}.add(v\text{Cells}.get(i))\)
end for

return \( v\text{CellPolygonList}, v\text{CellCentersList} \)