3D V-cell Algorithm

-Let P be the set of input points in 3D space

-DELAUNAY3D(P) returns the 3D triangulation (as a set of tetrahedra) for a given set of points P, which represent the V-cell centers

-CIRCUMCENTER(t) returns the center of sphere which passes through all vertices of the given tetrahedron t

-GETFACES(p,t) returns the 3 triangles from tetrahedron t which share a point with p

-GETADJACENTTETRA(p,tri) returns the tetrahedra which shares a point with p and a face with triangle tri

-ADDTRIANGLE(p_1,p_2,p_3) returns a triangle containing given points p_1, p_2, and p_3

```
procedure CALCULATEVCELLS3D(P)
T ← DELAUNAY3D(P)

for all $p_i \in P$ do
  for all $t_j \in T$ do
    if $t_j$.contains(p) then                     //if $p_i$ exists as a point in $t_j$
      $p_i$.tetras.add($t_i$)
    end if
  end for
end for

for all $p_i \in P$ do
  for all $t_j \in p_i$ do
    $t_j$.center ← CIRCUMCENTER($t_j$)
    $t_j$.faces ← GETFACES($p_i$,$t_j$)
    for all triangle$_k \in t_j$.faces do          //for the 3 triangle faces
      triangle$_k$.tetra ← GETADJACENTTETRA($p_i$,triangle$_k$)
      triangle$_k$.tetraCenter ← CIRCUMCENTER(triangle$_k$.tetra)
    end for
    vCellList($p_i$).ADDTRIANGLE($t_j$.center,triangle$_0$.tetraCenter,triangle$_1$.tetraCenter)
    vCellList($p_i$).ADDTRIANGLE($t_j$.center,triangle$_0$.tetraCenter,triangle$_2$.tetraCenter)
    vCellList($p_i$).ADDTRIANGLE($t_j$.center,triangle$_1$.tetraCenter,triangle$_2$.tetraCenter)
  end for
end for

return vCellList
```