

C++ code 3.14.12: Implementation of `findKp()` → [GitLab](#)

```
2  If::mesh::utils::CodimMeshDataSet<
3      std::pair<const If::mesh::Entity *, unsigned int>>
4  findKp(std::shared_ptr<const If::mesh::Mesh> mesh_p) {
5      // Variable for returning result
6      If::mesh::utils::CodimMeshDataSet<
7          std::pair<const If::mesh::Entity *, unsigned int>>
8          KpMeshDataSet(mesh_p, 2);
9      // Auxiliary array storing size of largest triangle adjacent to a node
10     If::mesh::utils::CodimMeshDataSet<double> sizeMeshDataSet(mesh_p, 2);
11     // loop over all cells
12     for (const If::mesh::Entity *triangle : mesh_p->Entities(0)) {
13         LF_ASSERT_MSG(triangle->RefEl() == If::base::RefEl::kTria(),
14             "Only implemented for triangles");
15         // Fetch coordinates of vertices
16         const Eigen::MatrixXd corners{If::geometry::Corners(*triangle->Geometry())};
17         // Determine size of triangle
18         const double newSize = std::max({(corners.col(1) - corners.col(0)).norm(),
19             (corners.col(2) - corners.col(1)).norm(),
20             (corners.col(0) - corners.col(2)).norm()});
21         // Obtain array of pointers to vertex objects of current triangle
22         std::span<const If::mesh::Entity *const> vertices{triangle->SubEntities(2)};
23         // Loop over vertices and update size of largest adjacent triangle.
24         for (unsigned int i = 0; i < 3; ++i) {
25             // Note that 'size' is a reference!
26             double &size = sizeMeshDataSet(*vertices[i]);
27             // Current triangle is larger than those recorded earlier
28             if (newSize > size) {
29                 // Update entry of auxiliary array
30                 size = newSize;
31                 // Store pointer to current triangle and local vertex index
32                 KpMeshDataSet(*vertices[i]) = std::make_pair(triangle, i);
33             }
34         }
35     } // end of loop over triangles
36     return KpMeshDataSet;
37 }
```