

C++ code 11.8.16: Sub-problem (11-8.g): function `computeCellNormals()` → [GitLab](#)

```
2  std::shared_ptr<
3      If::mesh::utils::CodimMeshDataSet<Eigen::Matrix<double, 2, Eigen::Dynamic>>>
4  computeCellNormals(std::shared_ptr<const If::mesh::Mesh> mesh_p) {
5      // Appears only in mastersolution
6
7      // Initialize datastructure for the result
8      If::mesh::utils::CodimMeshDataSet<Eigen::Matrix<double, 2, Eigen::Dynamic>>
9          result(mesh_p, 0);
10
11     // Compute normal vectors
12     for (const If::mesh::Entity *cell : mesh_p->Entities(0)) {
13         const If::geometry::Geometry *geo_p = cell->Geometry();
14         const Eigen::MatrixXd corners = If::geometry::Corners(*geo_p);
15
16         if (corners.cols() == 3) {
17             Eigen::Matrix<double, 2, Eigen::Dynamic> normal_vectors(2, 3);
18
19             // Compute normal vectors
20             Eigen::Matrix<double, 2, 3> bary_coord = gradbarycoordinates(corners);
21
22             // Reorder computed normal vectors
23             // normal_vectors[0] -> normal_vector of edge[0]
24             normal_vectors.col(0) = -(bary_coord.col(2)).normalized();
25             normal_vectors.col(1) = -(bary_coord.col(0)).normalized();
26             normal_vectors.col(2) = -(bary_coord.col(1)).normalized();
27
28             // Save the matrix in our datastructure
29             result(*cell) = normal_vectors;
30         } else { // corners.cols() == 4
31             Eigen::Matrix<double, 2, Eigen::Dynamic> normal_vectors(2, 4);
32
33             // Split the quadrilateral into two triangles (1,2,3) and (1,3,4)
34             Eigen::Matrix<double, 2, 3> tria_1;
35             Eigen::Matrix<double, 2, 3> tria_2;
36             tria_1 << corners.col(0), corners.col(1), corners.col(2);
37             tria_2 << corners.col(0), corners.col(2), corners.col(3);
38
39             // Compute normal vectors
40             Eigen::Matrix<double, 2, 3> bary_coord_1 = gradbarycoordinates(tria_1);
41             Eigen::Matrix<double, 2, 3> bary_coord_2 = gradbarycoordinates(tria_2);
42
43             // Reorder computed normal vectors
44             // normal_vectors[0] -> normal_vector of edge[0]
45             normal_vectors.col(0) = -(bary_coord_1.col(2)).normalized();
46             normal_vectors.col(1) = -(bary_coord_1.col(0)).normalized();
47             normal_vectors.col(2) = -(bary_coord_2.col(0)).normalized();
48             normal_vectors.col(3) = -(bary_coord_2.col(1)).normalized();
49
50             // Save the matrix in our datastructure
51             result(*cell) = normal_vectors;
52         }
53     }
54     return std::make_shared<If::mesh::utils::CodimMeshDataSet<
55         Eigen::Matrix<double, 2, Eigen::Dynamic>>>(result);
56
57 }
```