

C++ code 11.8.21: Sub-problem (11-8.I): function `computeHmin()` → [GitLab](#)

```
2 double computeHmin(std::shared_ptr<const If::mesh::Mesh> mesh_p) {
3     // Vector to store the distances between cells
4     std::vector<double> min_h;
5
6     // Get Adjacent Cells
7     std::shared_ptr<If::mesh::utils::CodimMeshDataSet<
8         std::array<const If::mesh::Entity *, 4>>>
9         adjacentCells = AdvectionFV2D::getAdjacentCellPointers(mesh_p);
10
11     // Iterate over all cells
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         // Compute the barycenter of the cell
17         Eigen::Vector2d cur_midpoint = barycenter(corners);
18
19         // Iterate over all adjacent cells
20         for (const If::mesh::Entity *neighbour_cell : (*adjacentCells)(*cell)) {
21             // Check that the neighbor exists
22             if (neighbour_cell != nullptr) {
23                 const If::geometry::Geometry *geo_p_neighbour =
24                     neighbour_cell->Geometry();
25                 const Eigen::MatrixXd neighbour_corners =
26                     If::geometry::Corners(*geo_p_neighbour);
27
28                 // Compute barycenter of neighbour cell
29                 Eigen::Vector2d neighbour_midpoint = barycenter(neighbour_corners);
30
31                 // Compute distances between cells
32                 double distance = (cur_midpoint - neighbour_midpoint).norm();
33
34                 // Store value in vector
35                 min_h.push_back(distance);
36             }
37         }
38     }
39
40     // Find the minimum value in the vector and return it
41     return *std::min_element(std::begin(min_h), std::end(min_h));
42 }
```