

C++ code 9.10.17: Implementation of `computeMQ()` → [GitLab](#)

```
2  Eigen::SparseMatrix<double> computeMQ(const If::assemble::DofHandler &dofh_Q) {
3      // TOOLS AND DATA
4      // Dimension of finite element space
5      const If::uscalfe::size_type N_dofs_Q(dofh_Q.NumDofs());
6      // Pointer to current mesh
7      std::shared_ptr<const If::mesh::Mesh> mesh_p = dofh_Q.Mesh();
8      // Sparse matrix to be returned
9      Eigen::SparseMatrix<double> M_Q(N_dofs_Q, N_dofs_Q);
10     // ASSEMBLY
11     // First option: assembly via triplet format
12     If::assemble::COOMatrix<double> M_Q_COO(N_dofs_Q, N_dofs_Q);
13     // Loop over all cells
14     for (const If::mesh::Entity *entity : mesh_p->Entities(0)) {
15         const double area = If::geometry::Volume(*(entity->Geometry()));
16         auto global_idx = dofh_Q.GlobalDofIndices(*entity);
17         M_Q_COO.AddToEntry(global_idx[0], global_idx[0], area);
18         M_Q_COO.AddToEntry(global_idx[1], global_idx[1], area);
19     }
20     M_Q = M_Q_COO.makeSparse();
21     std::cout << "Assembly: M_Q finished" << std::endl;
22     // Alternative option: reserve() method plus direct initialization
23     // of
24     // diagonal entries.
25     return M_Q;
26 }
```