

C++ code 9.6.10: Implementation of assembly for matrix $M \rightarrow$ [GitLab](#)

```
2  If :: assemble :: COOMatrix<double> buildM(  
3      const std :: shared_ptr<If :: uscalfe :: FeSpaceLagrangeO1<double>> &fe_space_p) {  
4      // I. TOOLS AND DATA  
5      // Pointer to current fe_space and mesh  
6      std :: shared_ptr<const If :: mesh :: Mesh> mesh_p(fe_space_p->Mesh());  
7      // Obtain local->global index mapping for current finite element space  
8      const If :: assemble :: DofHandler &dofh{fe_space_p->LocGlobMap()};  
9      // Dimension of finite element space  
10     const If :: uscalfe :: size_type N_dofs(dofh.NumDofs());  
11  
12     // II : ASSEMBLY  
13     // Matrix in triplet format holding Galerkin matrix, zero initially.  
14     If :: assemble :: COOMatrix<double> M(N_dofs, N_dofs);  
15     // Obtain an array of boolean flags for the edges of the mesh, 'true'  
16     // indicates that the edge lies on the boundary  
17     auto bd_flags{If :: mesh :: utils :: flagEntitiesOnBoundary(mesh_p, 1)};  
18     // Creating a predicate that will guarantee that the computations are carried  
19     // only on the edges of the mesh using the boundary flags  
20     // Actually a redundant step, because 'bdflags' is a predicate already.  
21     auto edges_predicate = [&bd_flags](const If :: mesh :: Entity &edge) -> bool {  
22         return bd_flags(edge);  
23     };  
24     // Coefficient function used in the class template MassEdgeMatrixProvider  
25     auto eta = If :: mesh :: utils :: MeshFunctionGlobal(  
26         [] (Eigen :: Vector2d x) -> double { return 1.0; });  
27     If :: uscalfe :: MassEdgeMatrixProvider<double, decltype(eta),  
28         decltype(edges_predicate)>  
29         edgemat_builder(fe_space_p, eta, edges_predicate);  
30     // Invoke assembly on edges by specifying co-dimension = 1  
31     If :: assemble :: AssembleMatrixLocally(1, dofh, dofh, edgemat_builder, M);  
32     return M;  
33 };
```