

C++ code 9.6.11: Implementation of assembly for matrix [M](#) → [GitLab](#)

```
2  If :: assemble :: COOMatrix<double> buildA(  
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> A(N_dofs, N_dofs);  
15     // Coefficient functions used in class ReactionDiffusionElementMatrixProvider  
16     auto alpha = If :: mesh :: utils :: MeshFunctionGlobal(  
17         [](Eigen :: Vector2d x) -> double { return 1.0 + x.dot(x); });  
18     auto gamma = If :: mesh :: utils :: MeshFunctionGlobal(  
19         [](Eigen :: Vector2d x) -> double { return 0.0; });  
20     // Initialize element matrix builder  
21     If :: uscalfe :: ReactionDiffusionElementMatrixProvider<double, decltype(alpha),  
22         decltype(gamma)>  
23         elmat_builder(fe_space_p, alpha, gamma);  
24     // Invoke assembly on cells (co-dimension = 0 as first argument)  
25     // Information about the mesh and the local-to-global map is passed through  
26     // a Dofhandler object, argument 'dofh'. This function call adds triplets to  
27     // the internal COO-format representation of the sparse matrix A.  
28     If :: assemble :: AssembleMatrixLocally(0, dofh, dofh, elmat_builder, A);  
29     return A;  
30 };
```