

C++ code 9.2.8: Sub-problem (9-2.f): function `assembleGalerkinMatrices()` → [GitLab](#)

```
2  std::pair<Eigen::SparseMatrix<double>, Eigen::SparseMatrix<double>>
3  assembleGalerkinMatrices(const If::assemble::DofHandler &dofh,
4                          double cool_coeff) {
5      std::pair<Eigen::SparseMatrix<double>, Eigen::SparseMatrix<double>>
6      sparse_pair;
7      std::cout << ">> Constructing SDIRK2Timestepper " << std::endl;
8      auto mesh_p = dofh.Mesh(); // pointer to current mesh
9      // Instantiating Galerkin matrices to be pre-computed
10     // Dimension of finite element space
11     const If::uscalfe::size_type N_dofs(dofh.NumDofs());
12
13     // Obtain an array of boolean flags for the edges of the mesh: 'true'
14     // indicates that the edge lies on the boundary. This predicate will
15     // guarantee that the computations are carried only on the boundary edges
16     auto bd_flags{If::mesh::utils::flagEntitiesOnBoundary(mesh_p, 1)};
17     // Creating predicate that will guarantee that the computations are carried
18     // only on the edges of the mesh using the boundary flags
19     auto edges_predicate = [&bd_flags](const If::mesh::Entity &edge) -> bool {
20         return bd_flags(edge);
21     };
22     // Matrices in triplet format holding Galerkin matrices, zero initially.
23     If::assemble::COOMatrix<double> A_COO(N_dofs, N_dofs);
24     If::assemble::COOMatrix<double> M_COO(N_dofs, N_dofs);
25
26     std::cout << "> Initializing the Galerkin local matrices builders"
27             << std::endl;
28     // Initialize classes containing the information required for the local
29     // computations of the Galerkin matrices. Simple implementations of
30     // LinFEMassMatrixProvider adapted to this particular problem was written to
31     // spare some of the overhead calculations involved in the use of the more
32     // general LehrFEM++ matrices providers.
33     If::uscalfe::LinearFELaplaceElementMatrix elLapMat_builder;
34     LinearMassEdgeMatrixProvider<decltype(edges_predicate)> el_MassEdge_builder(
35         edges_predicate, cool_coeff);
36     LinFEMassMatrixProvider elMassMat_builder;
37
38     std::cout << "> Assembling Galerkin matrices in COO format" << std::endl;
39     // Compute the Galerkin matrices
40     // Invoke assembly on cells and edges (co-dimension = 0 and 1 respectively as
41     // first argument). Information about the mesh and the local-to-global map is
42     // passed through a Dofhandler object, argument 'dofh'. This function call
43     // adds triplets to the internal COO-format representation of sparse matrices
44     If::assemble::AssembleMatrixLocally(0, dofh, dofh, elLapMat_builder, A_COO);
45     If::assemble::AssembleMatrixLocally(1, dofh, dofh, el_MassEdge_builder,
46                                         A_COO);
47     If::assemble::AssembleMatrixLocally(0, dofh, dofh, elMassMat_builder, M_COO);
48
49     std::cout << "> Converting to triplets to sparse matrices" << std::endl;
50     // Assembling the private Galerkin matrices
51     Eigen::SparseMatrix<double> A = A_COO.makeSparse();
52     Eigen::SparseMatrix<double> M = M_COO.makeSparse();
53     sparse_pair = std::make_pair(A, M);
54     return sparse_pair;
55 }
```