

C++11 code 3.1.3: Sub-problem (3-1.a): Implementation of `compGalerkinMatrix()`

→ [GitLab](#)

```
2 template <typename FUNC_ALPHA, typename FUNC_GAMMA, typename FUNC_BETA>
3 Eigen::SparseMatrix<double> compGalerkinMatrix(
4     const If::assemble::DofHandler &dofh, FUNC_ALPHA &&alpha,
5     FUNC_GAMMA &&gamma, FUNC_BETA &&beta) {
6     // obtain mesh and set up fe_space (p.w. linear Lagrangian FEM)
7     auto mesh = dofh.Mesh();
8     auto fe_space =
9         std::make_shared<If::uscalfe::FeSpaceLagrangeO1<double>>(mesh);
10
11     // get the number of degrees of freedom = dimension of FE space
12     const If::base::size_type N_dofs(dofh.NumDofs());
13     // Set up an empty sparse matrix to hold the Galerkin matrix
14     If::assemble::COOMatrix<double> A(N_dofs, N_dofs);
15     // Initialize ELEMENT_MATRIX_PROVIDER object
16     If::mesh::utils::MeshFunctionGlobal mf_alpha{alpha};
17     If::mesh::utils::MeshFunctionGlobal mf_gamma{gamma};
18     If::uscalfe::ReactionDiffusionElementMatrixProvider elmat_builder(
19         fe_space, std::move(mf_alpha), std::move(mf_gamma));
20     // Cell-oriented assembly over the whole computational domain
21     If::assemble::AssembleMatrixLocally(0, dofh, dofh, elmat_builder, A);
22
23     // Add contributions of boundary term in the bilinear form using
24     // a LehrFEM++ built-in high-level ENTITY_MATRIX_PROVIDER class
25     auto bd_flags{If::mesh::utils::flagEntitiesOnBoundary(mesh, 1)};
26     If::mesh::utils::MeshFunctionGlobal mf_beta{beta};
27     If::uscalfe::MassEdgeMatrixProvider edgemat_builder(
28         fe_space, std::move(mf_beta), bd_flags);
29     If::assemble::AssembleMatrixLocally(1, dofh, dofh, edgemat_builder, A);
30     Eigen::SparseMatrix<double> A_crs = A.makeSparse();
31     return A_crs;
32 }
```