

C++11 code 2.15.5: Implementation of getGalerkinLSE

```

2  template <typename FUNCT_F, typename FUNCT_H>
3  std::pair<Eigen::SparseMatrix<double>, Eigen::VectorXd> getGalerkinLSE(
4      const std::shared_ptr<If::uscalfe::UniformScalarFESpace<double>> fe_space,
5      const FUNCT_F &f, const FUNCT_H &h) {
6      const If::assemble::DofHandler &dofh{fe_space->LocGlobMap()};
7
8      const std::size_t N_dofs = dofh.NumDofs();
9      // Right-hand-side vector; don't forget to set to zero initially!
10     Eigen::VectorXd rhs_vec(N_dofs);
11     rhs_vec.setZero();
12
13     // ASSEMBLE MATRIX
14     If::assemble::COOMatrix<double> A_aux(N_dofs, N_dofs);
15     If::uscalfe::LinearFELaplaceElementMatrix my_mat_provider{};
16     // co-dimension 0 because we locally assemble on cells
17     If::assemble::AssembleMatrixLocally(0, dofh, dofh, my_mat_provider, A_aux);
18
19     // ASSEMBLE GLOBAL RHS VECTOR
20     // Volume part  $v \mapsto \int_{\Omega} f v \, dx$  of the rhs linear
21     // functional
22     If::uscalfe::ScalarLoadElementVectorProvider my_vec_provider(fe_space, f);
23     // co-dimension 0 because we locally assemble on cells
24     If::assemble::AssembleVectorLocally(0, dofh, my_vec_provider, rhs_vec);
25
26     // Boundary part  $v \mapsto \int_{\partial\Omega} h v \, dS$  of the rhs
27     // linear functional. Only edges on the boundary should be included
28     // in
29     // assembly, which is achieved by passing a suitable selector
30     // predicate to the
31     // builder object through a lambda function
32     auto bd_edges{If::mesh::utils::flagEntitiesOnBoundary(dofh.Mesh(), 1)};
33     If::uscalfe::ScalarLoadEdgeVectorProvider my_vec_provider_edge(
34         fe_space, h,
35         [&bd_edges](const If::mesh::Entity &edge) { return bd_edges(edge); });
36     // co-dimension 1 because we locally assemble on edges !
37     If::assemble::AssembleVectorLocally(1, dofh, my_vec_provider_edge, rhs_vec);
38
39     return std::make_pair(A_aux.makeSparse(), rhs_vec);
40 }

```