C++ code 9.18.11: Implementation of `computeGalerkinMatrices()`

```
std::tuple<lf::assemble::COOMatrix<double>, lf::assemble::COOMatrix<double>,
           lf::assemble::COOMatrix<double>>
computeGalerkinMatrices(
    std::shared_ptr<const lf::uscalfe::FeSpaceLagrangeO2<double>> fe_space_p) {
  // Pointer to current mesh
  std::shared_ptr<const lf::mesh::Mesh> mesh_p = fe_space_p->Mesh();
  // Obtain local->global index mapping for current finite element space
  const lf::assemble::DofHandler &dofh{fe_space_p->LocGlobMap()};
  // Dimension of finite element space
  const lf::uscalfe::size_type N_dofs(dofh.NumDofs());
  // Instantiating N_dofs x N_dofs zero matrices in triplet format
  lf::assemble::COOMatrix<double> M_COO(N_dofs, N_dofs);
  lf::assemble::COOMatrix<double> B_COO(N_dofs, N_dofs);
  lf::assemble::COOMatrix<double> A_COO(N_dofs, N_dofs);
  /* Creating dummy coefficient-functions as Lehrfem++ mesh functions */
  // These coefficient-functions are needed in the class templates
  // ReactionDiffusionElementMatrixProvider and MassEdgeMatrixProvider
  auto zero_mf = lf::mesh::utils::MeshFunctionConstant(0.0);
  auto one_mf = lf::mesh::utils::MeshFunctionConstant(1.0);
  /* Initialization of local matrices builders */
  // Initialize objects taking care of local computations for volume integrals
  lf::uscalfe::ReactionDiffusionElementMatrixProvider M_locmat_builder(
      fe_space_p, zero_mf, one_mf);
  lf::uscalfe::ReactionDiffusionElementMatrixProvider A_locmat_builder(
      fe_space_p, one_mf, zero_mf);
  // Initialize objects taking care of local computations for boundary integrals
  // Creating a predicate that will guarantee that the computations for the
  // boundary mass matrix are carried only on the edges of the mesh
  lf::mesh::utils::CodimMeshDataSet<bool> bd_flags{
      lf::mesh::utils::flagEntitiesOnBoundary(mesh_p, 1)};
  lf::uscalfe::MassEdgeMatrixProvider B_locmat_builder(fe_space_p, one_mf,
                                                       bd_flags);
  /* Assembling the Galerkin matrices */
  // Information about the mesh and the local-to-global map is passed through
  // a Dofhandler object, argument 'dofh'. This function call adds triplets to
  // the internal COO-format representation of the sparse matrix.
  // Invoke assembly on cells (co-dimension = 0 as first argument)
  lf::assemble::AssembleMatrixLocally(0, dofh, dofh, M_locmat_builder, M_COO);
  lf::assemble::AssembleMatrixLocally(0, dofh, dofh, A_locmat_builder, A_COO);
  // Invoke assembly on edges (co-dimension = 1 as first argument)
  lf::assemble::AssembleMatrixLocally(1, dofh, dofh, B_locmat_builder, B_COO);
  return {M_COO, B_COO, A_COO};
}
```