C++ code 5.5.2: Function realizing fixed-point update

```cpp
void fixedPointNextIt(
    std::shared_ptr<const lf::uscalfe::FeSpaceLagrangeO1<double>> fes_p,
    Eigen::VectorXd &mu_vec, Eigen::VectorXd &rhs_vec) {
  // Set up mesh functions for diffusion coefficient and reaction
     coefficient
  lf::mesh::utils::MeshFunctionGlobal mf_one(
      [](Eigen::Vector2d /*x*/) -> double { return 1.0; });
  const lf::fe::MeshFunctionFE mf_uh_prev(fes_p, mu_vec);
  const FunctionMFWrapper mf_coeff(mf_uh_prev, [](double xi) -> double {
    return (std::abs(xi) < 1.0E-16) ? 1.0 : std::sinh(xi) / xi;
  });

  const lf::mesh::Mesh &mesh{*(fes_p->Mesh())};
  const lf::assemble::DofHandler &dofh{fes_p->LocGlobMap()};
  const std::size_t N_dofs(dofh.NumDofs());
  // Assemble Galerkin matrix
  lf::assemble::COOMatrix<double> A(N_dofs, N_dofs);
  lf::uscalfe::ReactionDiffusionElementMatrixProvider<double, decltype(mf_one),
                                                      decltype(mf_coeff)>
      elmat_provider(fes_p, mf_one, mf_coeff);
  lf::assemble::AssembleMatrixLocally(0, dofh, dofh, elmat_provider, A);

  // Enforce homogeneous Dirichlet boundary conditions
  auto bd_flags{lf::mesh::utils::flagEntitiesOnBoundary(fes_p->Mesh(), 2)};
  // Elimination of degrees of freedom on the boundary.  Also sets the
  // corresponding entries of rhs_vec to zero.
  lf::assemble::FixFlaggedSolutionComponents<double>(
      [&bd_flags,
       &dofh](lf::assemble::glb_idx_t gdof_idx) -> std::pair<bool, double> {
        const lf::mesh::Entity &node{dofh.Entity(gdof_idx)};
        return (bd_flags(node) ? std::make_pair(true, 0.0)
                               : std::make_pair(false, 0.0));
      },
      A, rhs_vec);
  // Solve linear system
  Eigen::SparseMatrix<double> A_crs = A.makeSparse();
  Eigen::SparseLU<Eigen::SparseMatrix<double>> solver;
  solver.compute(A_crs);
  LF_VERIFY_MSG(solver.info() == Eigen::Success, "LU decomposition failed");
  mu_vec = solver.solve(rhs_vec);
  LF_VERIFY_MSG(solver.info() == Eigen::Success, "Solving LSE failed");
}
```