

C++ code 3.18.10: Implementation of `compHierSurplusSolution()` → [GitLab](#)

```
2 template <typename MESHFUNCTION_ALPHA, typename MESHFUNCTION_F>
3 Eigen::VectorXd compHierSurplusSolution(
4     const MESHFUNCTION_ALPHA &mf_alpha, const MESHFUNCTION_F &mf_f,
5     std::shared_ptr<const If::uscalfe::FeSpaceLagrangeO1<double>> fes_lin_p,
6     std::shared_ptr<const If::uscalfe::FeSpaceLagrangeO2<double>> fes_quad_p,
7     const Eigen::VectorXd &mu) {
8     // References to FE space
9     const If::uscalfe::FeSpaceLagrangeO2<double> &quad_space{*fes_quad_p};
10    const If::uscalfe::FeSpaceLagrangeO1<double> &lfe_space{*fes_lin_p};
11    // Get references to DofHandlers
12    const If::assemble::DofHandler &dh_quad{quad_space.LocGlobMap()};
13    const If::assemble::DofHandler &dh_lfe{lfe_space.LocGlobMap()};
14    LF_ASSERT_MSG(dh_lfe.Mesh() == dh_quad.Mesh(),
15        "DofHandlers must be based on the same mesh");
16    LF_ASSERT_MSG(dh_lfe.NumDofs() == mu.size(), "Vector length mismatch");
17    // Underlying mesh
18    std::shared_ptr<const If::mesh::Mesh> mesh_p{dh_lfe.Mesh()};
19    const If::mesh::Mesh &mesh{*mesh_p};
20    LF_ASSERT_MSG(
21        (dh_lfe.NumDofs() == mesh.NumEntities(2)) &&
22        (dh_quad.NumDofs() == mesh.NumEntities(2) + mesh.NumEntities(1)),
23        "#dof mismatch");
24    // Dimension of the quadratic finite element space
25    const If::base::size_type N_qdofs(dh_quad.NumDofs());
26    // Vector for returning the result
27    Eigen::VectorXd quad_surplus(N_qdofs);
28    // Step I: Assemble full Galerkin matrix for quadratic FE
29    // Object for sparse matrix to be filled by cell-oriented assembly
30    If::assemble::COOMatrix<double> A(N_qdofs, N_qdofs);
31    // Provider object for element matrices for scalar linear second-order
32    // diffusion operator with variable diffusion coefficient. Uses
33    // quadrature of order 3 and, thus, computes exact element matrices
34    // for locally constant diffusion coefficient.
35    If::fe::DiffusionElementMatrixProvider<double, MESHFUNCTION_ALPHA>
36        elmat_builder(fes_quad_p, mf_alpha);
37    // Invoke cell-oriented assembly of the finite-element Galerkin matrix
38    If::assemble::AssembleMatrixLocally(0, dh_quad, dh_quad, elmat_builder, A);
39
40    // II: Assembly of right-hand-side vector in quadratic FE space
41    Eigen::Matrix<double, Eigen::Dynamic, 1> phi(N_qdofs);
42    phi.setZero();
43    // Assemble right-hand side vector depending on the source function f.
44    // Initialize object taking care of local computations on all cells.
45    If::uscalfe::ScalarLoadElementVectorProvider<double, MESHFUNCTION_F>
46        elvec_builder(fes_quad_p, mf_f);
47    // Invoke assembly on cells (codim == 0)
48    AssembleVectorLocally(0, dh_quad, elvec_builder, phi);
49
50    // III.: Compute residual vector in quadratic FE space
51    // We take for granted that the components of mu corresponding to
52    // nodes on the
53    // boundary vanish
54    Eigen::VectorXd residual_quad =
55        phi + A.MatVecMult(-1.0, trfLinToQuad(fes_lin_p, fes_quad_p, mu));
```