

C++ code 3.18.7: Implementation of `trfLinToQuad()` → [GitLab](#)

```

2  Eigen::VectorXd trfLinToQuad(
3      std::shared_ptr<const If::uscalfe::FeSpaceLagrangeO1<double>> fes_lin_p ,
4      std::shared_ptr<const If::uscalfe::FeSpaceLagrangeO2<double>> fes_quad_p ,
5      const Eigen::VectorXd &mu) {
6      using gdof_idx_t = If::assemble::gdof_idx_t;
7      // Obtain local-to-global index mappings
8      const If::assemble::DofHandler &dh_lin{fes_lin_p->LocGlobMap()};
9      const If::assemble::DofHandler &dh_quad{fes_quad_p->LocGlobMap()};
10     LF_ASSERT_MSG(dh_lin.Mesh() == dh_quad.Mesh(),
11         "DofHandlers must be based on the same mesh");
12     LF_ASSERT_MSG(dh_lin.NumDofs() == mu.size(), "Vector length mismatch");
13     // Underlying mesh
14     std::shared_ptr<const If::mesh::Mesh> mesh_p{dh_lin.Mesh()};
15     const If::mesh::Mesh &mesh{*mesh_p};
16     LF_ASSERT_MSG(
17         (dh_lin.NumDofs() == mesh.NumEntities(2)) &&
18         (dh_quad.NumDofs() == mesh.NumEntities(2) + mesh.NumEntities(1)),
19         "#DOFs do not match #entities");
20     // The coefficients of a FE function in the quadratic Lagrangian FE
21     // space with
22     // respect to the nodal basis are simply its values at the nodes and
23     // the
24     // midpoints of the edges.
25     Eigen::VectorXd nu(dh_quad.NumDofs());
26     // Visit nodes (codimension =2) and copy values
27     for (const If::mesh::Entity *node : mesh.Entities(2)) {
28         std::span<const gdof_idx_t> lf_dof_idx{dh_lin.GlobalDofIndices(*node)};
29         LF_ASSERT_MSG(lf_dof_idx.size() == 1,
30             "Exactly one LFE basis functiona at a node");
31         LF_ASSERT_MSG(lf_dof_idx[0] < dh_lin.NumDofs(), "LFE dof number too large");
32         std::span<const gdof_idx_t> qf_dof_idx{dh_quad.GlobalDofIndices(*node)};
33         LF_ASSERT_MSG(qf_dof_idx.size() == 1,
34             "Exactly one QFE basis functiona at a node");
35         LF_ASSERT_MSG(qf_dof_idx[0] < dh_quad.NumDofs(),
36             "QFE dof number too large");
37         nu[qf_dof_idx[0]] = mu[lf_dof_idx[0]];
38     }
39     // Run through all edges
40     for (const If::mesh::Entity *edge : mesh.Entities(1)) {
41         // Obtain global numbers of LFE DOFs associated with the endpoints
42         // of
43         // the edge, that is, of those DOFs covering the edge.
44         std::span<const gdof_idx_t> lf_dof_idx{dh_lin.GlobalDofIndices(*edge)};
45         LF_ASSERT_MSG(lf_dof_idx.size() == 2,
46             "Exactly two basis functions must cover an edge");
47         LF_ASSERT_MSG((lf_dof_idx[0] < dh_lin.NumDofs()) &&
48             (lf_dof_idx[1] < dh_lin.NumDofs()),
49             "LFE dof numbers too large");
50         // Obtain global number of QFE basis function associated with the
51         // edge
52         std::span<const gdof_idx_t> qf_dof_idx{
53             dh_quad.InteriorGlobalDofIndices(*edge)};
54         LF_ASSERT_MSG(qf_dof_idx.size() == 1,
55             "A single QFE basis function is associated to an edge!");
56         LF_ASSERT_MSG(qf_dof_idx[0] < dh_quad.NumDofs(),
57             "QFE dof number too large");
58     }
59     // Value of p.w. linear continuous FE function at the midpoint of

```