

### C++11 code 3.2.3: Sub-problem (3-2.a): function `interpolateOntoQuadFE()` → [GitLab](#)

```
2 template <typename FUNCTOR>
3 Eigen::VectorXd interpolateOntoQuadFE(const If::assemble::DofHandler &dofh,
4                                     FUNCTOR &&f) {
5     // get mesh and set up finite element space
6     auto mesh = dofh.Mesh();
7
8     // initialize result vector
9     const size_type N_dofs(dofh.NumDofs());
10    Eigen::VectorXd result = Eigen::VectorXd::Zero(N_dofs);
11
12    for (const If::mesh::Entity *cell : mesh->Entities(0)) {
13        // get local to global map for the cell
14        auto glob_ind = dofh.GlobalDofIndices(*cell);
15        // This is a rather clumsy implementation. A much better way to accomplish
16        // this is to pass a full matrix of reference coordinates to the Global()
17        // method of the current cell entity. This will immediately give the global
18        // coordinates of all local interpolation nodes and we can dispense with the
19        // auxiliary function. Afterwards we can loop over the columns of the
20        // matrix.
21        for (int i = 0; i < 6; i++) {
22            // update the result vector
23            auto coords = globalCoordinate(i, *cell);
24            result(glob_ind[i]) = f(coords);
25        }
26    }
27    return result;
28 }
```