

C++ code 3.9.18: Implementation of `Eval()` method of `VectorProjectionMatrixProvider`

→ [GITHUB](#)

```
2 Eigen::VectorXd computeLumpedProjection(  
3     const If::assemble::DofHandler &scal_dofh, const Eigen::VectorXd &mu,  
4     const If::assemble::DofHandler &vec_dofh) {  
5     // Obtain shared_ptr to mesh  
6     std::shared_ptr<const If::mesh::Mesh> mesh_p = scal_dofh.Mesh();  
7     // Dimension of vector-valued finite element space  
8     const If::uscalfe::size_type N_vec_dofs(vec_dofh.NumDofs());  
9     // Initialize vector FE basis expansion coefficient vector with zeros  
10    Eigen::VectorXd proj_vec(N_vec_dofs);  
11    proj_vec.setZero();  
12    // Initialize temporary helper nodal DataSet (codim 2)  
13    auto nodal_sum_of_areas =  
14        If::mesh::utils::CodimMeshDataSet<double>(mesh_p, 2, 0.0);  
15  
16    // Loop over the triangular cells of the mesh in the spirit of  
17    // cell oriented assembly  
18    for (const If::mesh::Entity *cell : mesh_p->Entities(0)) {  
19        LF_VERIFY_MSG(cell->RefEl() == If::base::RefEl::kTria(),  
20            "Unsupported cell type " << cell->RefEl());  
21        // Obtain global scalar-FE indices of the vertices  
22        const auto scal_dof_idx_vec = scal_dofh.GlobalDofIndices(*cell);  
23        // Obtain the gradients of the barycentric coordinate functions  
24        const Eigen::Matrix<double, 2, 3> elgrad_Mat = gradbarycoordinates(*cell);  
25        // Obtain area of the triangular cell  
26        const double area = If::geometry::Volume(*(cell->Geometry()));  
27        // Compute the gradient of the passed coefficient vector  
28        const Eigen::Vector2d grad_mu =  
29            elgrad_Mat.col(0) * mu(scal_dof_idx_vec[0]) +  
30            elgrad_Mat.col(1) * mu(scal_dof_idx_vec[1]) +  
31            elgrad_Mat.col(2) * mu(scal_dof_idx_vec[2]);  
32        // Local contribution to the area of the cell patch surrounding a node  
33        for (const If::mesh::Entity *node : cell->SubEntities(2)) {  
34            LF_VERIFY_MSG(node->RefEl() == If::base::RefEl::kPoint(),  
35                "Expected kPoint type!" << node->RefEl());  
36            auto vec_dofh_idx = vec_dofh.GlobalDofIndices(*node);  
37            proj_vec[vec_dofh_idx[0]] += area * grad_mu[0];  
38            proj_vec[vec_dofh_idx[1]] += area * grad_mu[1];  
39            nodal_sum_of_areas(*node) = nodal_sum_of_areas(*node) + area;  
40        }  
41    }  
42  
43    // Scaling of components of vector of dofs  
44    for (const If::mesh::Entity *node : mesh_p->Entities(2)) {  
45        LF_VERIFY_MSG(node->RefEl() == If::base::RefEl::kPoint(),  
46            "Expected kPoint type!" << node->RefEl());  
47        const double area_scal_fac = 1.0 / nodal_sum_of_areas(*node);  
48        auto vec_dofh_idx = vec_dofh.GlobalDofIndices(*node);  
49        proj_vec[vec_dofh_idx[0]] *= area_scal_fac;  
50        proj_vec[vec_dofh_idx[1]] *= area_scal_fac;  
51    }  
52    return proj_vec;  
53 }; // computeLumpedProjection
```