

C++ code 3.9.14: Implementation of `Eval()` method of `GradientProjectionVectorProvider`

→ [GITHUB](#)

```
2 Eigen::VectorXd GradientProjectionVectorProvider::Eval(  
3     const If::mesh::Entity &entity) {  
4     Eigen::VectorXd elVec(6); // for returning the element vector  
5     // Obtain local->global index mapping for current finite element space  
6     const If::assemble::DofHandler &dofh{_fe_space_p->LocGlobMap()};  
7     // Obtain global indices of the vertices of the triangle entity  
8     auto dof_idx_vec = dofh.GlobalDofIndices(entity);  
9     LF_ASSERT_MSG(dofh.NumLocalDofs(entity) == 3,  
10        "Too many global indices were returned for a triangle entity!");  
11  
12     // Obtain the gradients of the barycentric coordinate functions  
13     Eigen::Matrix<double, 2, 3> elgrad_Mat = gradbarycoordinates(entity);  
14     // Compute the local constant gradient of the finite element solution  
15     Eigen::Vector2d grad_vec(0.0, 0.0);  
16     for (int i = 0; i < 3; i++) {  
17         grad_vec = grad_vec + elgrad_Mat.col(i) * _mu(dof_idx_vec[i]);  
18     }  
19     // Assemble local element vector  
20     // Compute the area of the triangle cell  
21     const double area = If::geometry::Volume(*(entity.Geometry()));  
22     // clang-format off  
23     elVec << grad_vec(0),  
24             grad_vec(1),  
25             grad_vec(0),  
26             grad_vec(1),  
27             grad_vec(0),  
28             grad_vec(1);  
29     // clang-format on  
30     elVec *= area / 3.0;  
31     return elVec;  
32 } // GradientProjectionVectorProvider::Eval
```