

C++11 code 2.10.11: Solution code for Sub-problem (2-10.g) → [GitLab](#)

```
2  template <typename FUNCTOR>
3  Eigen::Vector3d GradProjRhsProvider<FUNCTOR>::Eval(
4      const If::mesh::Entity &entity) {
5      LF_ASSERT_MSG(If::base::RefEl::kTria() == entity.RefEl(),
6                  "Function only defined for triangular cells");
7
8      const If::geometry::Geometry *geo_ptr = entity.Geometry();
9      Eigen::Vector3d loc_vec;
10
11     // get area of the entity
12     const double area = If::geometry::Volume(*geo_ptr);
13
14     // calculate center of mass
15     const Eigen::MatrixXd corners = If::geometry::Corners(*geo_ptr);
16     const Eigen::Vector2d c = corners.rowwise().sum() / 3.;
17
18     // value of the functor f at center of mass of the entity
19     const Eigen::Vector2d func_value = f_(c);
20
21     // calculate the gradients of the basis functions
22     Eigen::Matrix3d grad_helper;
23     grad_helper.col(0) = Eigen::Vector3d::Ones();
24     grad_helper.rightCols(2) = corners.transpose();
25     // vector of the gradients of the basis function evaluated at c
26     const Eigen::MatrixXd grad_basis = grad_helper.inverse().bottomRows(2);
27
28     loc_vec = area * (grad_basis.transpose() * func_value);
29     return loc_vec;
30 }
```