

C++11 code 3.3.11: Sub-problem (3-3.g): Implementation of `testGlobalInverseQuad()`

→ [GitLab](#)

```
2 void testGlobalInverseQuad(const lf::mesh::Entity &quad, Eigen::Vector2d xh) {
3   LF_ASSERT_MSG(quad.RefEl() == lf::base::RefEl::kQuad(),
4                 "Cell must be a quadrilateral");
5   // get the coordinates of the corners of this cell
6   const lf::geometry::Geometry *geo_ptr = quad.Geometry();
7   auto vertices = lf::geometry::Corners(*geo_ptr);
8   // Image of point in unit square under parametric mapping
9   Eigen::Vector2d x = geo_ptr->Global(xh);
10  Eigen::Vector2d xh_comp = PointEvaluationRhs::GlobalInverseQuad(vertices, x);
11  EXPECT_NEAR((xh - xh_comp).norm(), 0.0, 1.0E-8)
12    << "quadl " << quad << ": Mismatch xh = " << xh << ", x = " << x
13    << ", xh_comp = " << xh_comp << std::endl;
14 }
```