

C++ code 13.2.63: Implementation of `Eval()` member function computing M_K for a mesh cell entity. → [GitLab](#)

```
2 HodgeLaplacian2DElementMatrixProvider::ElemMat
3 HodgeLaplacian2DElementMatrixProvider::Eval(const If::mesh::Entity& cell) {
4     LF_VERIFY_MSG(cell.RefEl() == If::base::RefEl::kTria(),
5         "Unsupported cell type " << cell.RefEl());
6     LF_VERIFY_MSG(cell.Geometry()->isAffine(),
7         "Triangle must have straight edges");
8     // Area of the triangle
9     double area = If::geometry::Volume(*cell.Geometry());
10    // Compute gradients of barycentric coordinate functions and store
11    // them in the
12    // columns of the 2x3-matrix G
13    // clang-format off
14    const Eigen::MatrixXd dpt = (Eigen::MatrixXd(2, 1) << 0.0, 0.0).finished();
15    const Eigen::Matrix<double, 2, 3> G =
16        cell.Geometry()->JacobianInverseGramian(dpt).block(0, 0, 2, 2) *
17        (Eigen::Matrix<double, 2, 3>(2,3) << -1, 1, 0,
18            -1, 0, 1).finished();
19    // clang-format on
20    // Compute the element matrix for  $-\Delta$  and  $S_1^0$ .
21    // See also [Lecture → Code 2.4.5.13].
22    Eigen::Matrix<double, 3, 3> L = area * G.transpose() * G;
23    // Fill left upper block  $C_K$ , (13.2.50)
24    const Eigen::Matrix3d C =
25        (Eigen::Matrix3d() << 2, 1, 1, 1, 2, 1, 1, 1, 2).finished();
26    MK_.block(0, 0, 3, 3) = -(area / 12.0) * C;
27    // Initialize right upper block  $B_K$  and left lower block  $B^T$ ,
28    // see (13.2.47).
29    Eigen::Matrix3d B;
30    B.col(0) = (L.col(1) - L.col(0)) / 3.0;
31    B.col(1) = (L.col(2) - L.col(1)) / 3.0;
32    B.col(2) = (L.col(0) - L.col(2)) / 3.0;
33    MK_.block(0, 3, 3, 3) = B;
34    MK_.block(3, 0, 3, 3) = B.transpose();
35    // Set lower right block  $A_K$ , see (13.2.56)
36    MK_.block(3, 3, 3, 3) = Eigen::Matrix3d::Constant(1.0 / area);
37    // Correct for orientation mismatch, cf. Sub-problem (13-2.z).
38    auto relor = cell.RelativeOrientations();
39    LF_ASSERT_MSG(relor.size() == 3, "Triangle should have 3 edges!?");
40    for (int k = 0; k < 3; ++k) {
41        if (relor[k] == If::mesh::Orientation::negative) {
42            // Flip sign of 3+k-th row and column
43            MK_.col(3 + k) *= -1.0;
44            MK_.row(3 + k) *= -1.0;
45        }
46    }
47    return MK_;
48 }
```