

C++11 code 2.8.13: Sub-problem (2-8.i): Implementation of `Eval()` for `MyLinearFEElementMatrix` → [GitLab](#)

```
2 Eigen::Matrix<double, 4, 4> MyLinearFEElementMatrix::Eval(  
3     const If::mesh::Entity &cell) {  
4     // Topological type of the cell  
5     const If::base::RefEl ref_el{cell.RefEl()};  
6  
7     // Obtain the vertex coordinates of the cell, which completely  
8     // describe its shape.  
9     const If::geometry::Geometry *geo_ptr = cell.Geometry();  
10    // Matrix storing corner coordinates in its columns  
11    auto vertices = geo_ptr->Global(ref_el.NodeCoords());  
12    // Matrix for returning element matrix  
13    Eigen::Matrix<double, 4, 4> elem_mat;  
14    // Initialize matrix containing the mass part of the element matrix  
15    Eigen::Matrix<double, 4, 4> mass_elem_mat;  
16    // Retrieve laplace part of element matrix from LehrFEM's built-in  
17    // laplace  
18    // element matrix builder  
19    If::uscalfe::LinearFELaplaceElementMatrix laplace_elmat_builder;  
20    auto laplace_elem_mat = laplace_elmat_builder.Eval(cell);  
21    // Computations differ depending on the type of the cell  
22    switch (ref_el) {  
23        case If::base::RefEl::kTria(): {  
24            double area = 0.5 * ((vertices(0, 1) - vertices(0, 0)) *  
25                                (vertices(1, 2) - vertices(1, 0)) -  
26                                (vertices(1, 1) - vertices(1, 0)) *  
27                                (vertices(0, 2) - vertices(0, 0)));  
28            // clang-format off  
29            mass_elem_mat << 2.0, 1.0, 1.0, 0.0,  
30            1.0, 2.0, 1.0, 0.0,  
31            1.0, 1.0, 2.0, 0.0,  
32            0.0, 0.0, 0.0, 0.0;  
33            // clang-format on  
34            mass_elem_mat *= area / 12.0;  
35            break;  
36        }  
37        case If::base::RefEl::kQuad(): {  
38            double area =  
39                (vertices(0, 1) - vertices(0, 0)) * (vertices(1, 3) - vertices(1, 0));  
40            // clang-format off  
41            mass_elem_mat << 4.0, 2.0, 1.0, 2.0,  
42            2.0, 4.0, 2.0, 1.0,  
43            1.0, 2.0, 4.0, 2.0,  
44            2.0, 1.0, 2.0, 4.0;  
45            // clang-format on  
46            mass_elem_mat *= area / 36.0;  
47            break;  
48        }  
49        default: {  
50            LF_ASSERT_MSG(false, "Illegal cell type");  
51        }  
52    } // end switch  
53    elem_mat = laplace_elem_mat + mass_elem_mat;  
54    return elem_mat;  
55 }
```