

C++11 code 2.13.11: Sub-problem (2-13.e): Implementation of `Eval()` member function of `ImpedanceBoundaryEdgeMatrixProvider` → [GitLab](#)

```
2 Eigen::MatrixXd ImpedanceBoundaryEdgeMatrixProvider::Eval(  
3     const If::mesh::Entity &edge) {  
4     Eigen::MatrixXd element_matrix(2, 2);  
5  
6     /* TOOLS AND DATA */  
7     // Obtain local->global index mapping for current finite element space  
8     const If::assemble::DofHandler &dofh{fe_space_->LocGlobMap()};  
9     // Obtain edge data  
10    auto edge_global_idx = dofh.GlobalDofIndices(edge);  
11  
12    // I. COMPUTE LOCAL INTEGRATION DATA  
13    Eigen::Vector2d w;  
14    // Evaluate w(x) at the degrees of freedom (endpoints of the edge)  
15    for (int i = 0; i < 2; i++) {  
16        w(i) = coeff_expansion_(edge_global_idx[i]);  
17    }  
18  
19    // II. COMPUTE LOCAL INTEGRATION DATA  
20    double edge_length = If::geometry::Volume(*edge.Geometry());  
21    // It is assumed here that the edge is straight!  
22    Eigen::MatrixXd m_1(2, 2), m_2(2, 2), m_3(2, 2);  
23    m_1 << 24, 6, 6, 4;  
24    m_2 << 6, 4, 4, 6;  
25    m_3 << 4, 6, 6, 24;  
26  
27    // III. PERFORM NUMERICAL QUADRATURE  
28    element_matrix =  
29        (w(0) * w(0)) * m_1 + (w(0) * w(1)) * m_2 + (w(1) * w(1)) * m_3;  
30    element_matrix *= edge_length / 120.;  
31  
32    return element_matrix;  
33 }
```