

C++11 code 2.6.4: Computation of edge-node incidence matrix [G](#) → [GitLab](#)

```
2 Eigen::SparseMatrix<int> computeEdgeVertexIncidenceMatrix(  
3     const If::mesh::Mesh &mesh) {  
4     // Store edge-vertex incidence matrix here  
5     Eigen::SparseMatrix<int, Eigen::RowMajor> G;  
6  
7     // Mesh::NumEntities(unsigned codim) returns the number of elements  
8     // with given codimension. Codim(Edge) = 1, Codim(Node) = 2.  
9     const If::mesh::Mesh::size_type numEdges = mesh.NumEntities(1),  
10         numNodes = mesh.NumEntities(2);  
11     // Following the demo for the reserve()-initialising the sparse matrix  
12     // given  
13     // in the exercise sheet. From (2.1a) we know that G has exactly 2  
14     // entries  
15     // per row.  
16     G = Eigen::SparseMatrix<int, Eigen::RowMajor>(numEdges, numNodes);  
17     G.reserve(Eigen::VectorXi::Constant(numEdges, 2));  
18  
19     // To compute G efficiently we iterate over all edges and check the  
20     // index  
21     // of the nodes at its end. This is the efficient way to do the  
22     // assembly,  
23     // introduced as "distribute scheme" in class. We cannot iterate  
24     // over  
25     // vertices, because LehrFEM++ does not allow to visit the edges  
26     // adjacent to a vertex  
27     for (const If::mesh::Entity *edge : mesh.Entities(1)) {  
28         // Get index of this edge  
29         If::mesh::Mesh::size_type edgeIdx = mesh.Index(*edge);  
30         // Get the nodes and their indices.  
31         // Note, that seen from the edges the nodes have codim 1, not 2,  
32         // hence we call SubEntities(1). This is a relative codimension!  
33         auto nodes = edge->SubEntities(1);  
34         If::mesh::Mesh::size_type firstNodeIdx = mesh.Index(*nodes[0]);  
35         If::mesh::Mesh::size_type lastNodeIdx = mesh.Index(*nodes[1]);  
36         // Add the matrix entries according to the definition  
37         G.coeffRef(edgeIdx, firstNodeIdx) += 1;  
38         G.coeffRef(edgeIdx, lastNodeIdx) -= 1;  
39     }  
40  
41     return G;  
42 }
```