

C++ code 13.1.5: Computation of cell-edge incidence matrix D_1 → [GitLab](#)

```
2 Eigen::SparseMatrix<int> computeCellEdgeIncidenceMatrix(  
3     const If::mesh::Mesh &mesh) {  
4     // Store cell-edge incidence matrix here  
5     Eigen::SparseMatrix<int, Eigen::RowMajor> D;  
6  
7     // Mesh::NumEntities(unsigned codim) returns the number of elements  
8     // with given codimension. Codim(Edge) = 0, Codim(Node) = 1.  
9     const If::mesh::Mesh::size_type numCells = mesh.NumEntities(0),  
10         numEdges = mesh.NumEntities(1);  
11     // Following the demo for the reserve()-initialising the sparse matrix  
12     // given  
13     // in the exercise sheet. From (2.1a) we know that D has at most 4  
14     // entries  
15     // per row.  
16     D = Eigen::SparseMatrix<int, Eigen::RowMajor>(numCells, numEdges);  
17     D.reserve(Eigen::VectorXi::Constant(numCells, 4));  
18  
19     // To compute D efficiently we iterate over all cells and check the  
20     // orientations (+1 or -1, same as in the definition of the matrix D)  
21     // of its edges. For this we may use RelativeOrientations().  
22     for (const If::mesh::Entity *cell : mesh.Entities(0)) {  
23         // Get cell index  
24         If::mesh::Mesh::size_type cellIdx = mesh.Index(*cell);  
25         // Get edges and their orientations (these already the entries for  
26         // D!)  
27         auto edges = cell->SubEntities(1);  
28         auto edgeOrientations = cell->RelativeOrientations();  
29  
30         // Iterate over both and add to D  
31         auto edgelt = edges.begin();  
32         auto orntlt = edgeOrientations.begin();  
33         for (; edgelt != edges.end() && orntlt != edgeOrientations.end();  
34             ++edgelt, ++orntlt) {  
35             // Get the edge index and add its orientation to D  
36             If::mesh::Mesh::size_type edgIdx = mesh.Index(*edgelt);  
37             D.coeffRef(cellIdx, edgIdx) += If::mesh::to_sign(*orntlt);  
38         }  
39     }  
40  
41     return D;  
42 }
```