

C++11 code 3.7.9: Sub-problem (3-7.g): Auxiliary function for specialized assembly on a tensor-product triangular mesh. → [GitLab](#)

```
2 Eigen::SparseMatrix<double> assemble(int M, const Eigen::Matrix3d &B_K) {
3     int M2 = M * M;
4     Eigen::SparseMatrix<double> A(M2, M2);
5     double near_neighbour_contribution = 2.0 * B_K(0, 1);
6     double far_neighbour_contribution = 2.0 * B_K(1, 2);
7     double self_contribution = 2.0 * (B_K(0, 0) + B_K(1, 1) + B_K(2, 2));
8
9     std::vector<double> contribution = {
10         far_neighbour_contribution, near_neighbour_contribution,
11         near_neighbour_contribution, self_contribution,
12         near_neighbour_contribution, near_neighbour_contribution,
13         far_neighbour_contribution};
14
15     std::vector<Eigen::Vector2i> shift = {{-1, -1}, {0, -1}, {-1, 0}, {0, 0},
16                                           {1, 0}, {0, 1}, {1, 1}};
17
18     std::vector<Eigen::Triplet<double>> tripletList;
19     for (int i = 0; i < M; ++i) {
20         for (int j = 0; j < M; ++j) {
21             Eigen::Vector2i self = Eigen::Vector2i(i, j);
22             for (int k = 0; k < 7; ++k) {
23                 Eigen::Vector2i other = self + shift[k];
24                 if (0 <= other(0) && other(0) < M && 0 <= other(1) && other(1) < M) {
25                     tripletList.push_back(Eigen::Triplet<double>(
26                         self(0) + M * self(1), other(0) + M * other(1), contribution[k]));
27                 }
28             }
29         }
30     }
31     A.setFromTriplets(tripletList.begin(), tripletList.end());
32     return A;
33 }
```