

C++ code 2.4.14: Implementation of `computeRHS()` → [GitLab](#)

```
2 template <typename FUNCTOR1>
3 Eigen::VectorXd computeRHS(const Eigen::VectorXd &mesh, FUNCTOR1 &&f) {
4     // Nodes are indexed as 0=x_0 < x_1 < ... < x_N = 1
5     unsigned N = mesh.size() - 1;
6     // Initializing right hand side vector
7     Eigen::VectorXd rhs_vec = Eigen::VectorXd::Zero(N + 1);
8
9     // Some tool variables
10    double dx;
11
12    /* Calculate the entries */
13    // First entry (left boundary node)
14    rhs_vec(0) = f(mesh(0)) * (mesh(1) - mesh(0)) * 0.5;
15    // Last entry (right boundary node)
16    rhs_vec(N) = f(mesh(N)) * (mesh(N) - mesh(N - 1)) * 0.5;
17    // All other entries associated to interior nodes
18    for (unsigned i = 1; i < N; ++i) {
19        dx = mesh(i + 1) - mesh(i - 1);
20        rhs_vec(i) = f(mesh(i)) * 0.5 * dx;
21    }
22
23    return rhs_vec;
24 } // computeRHS
```