

C++ code 3.8.18: Computation of boundary version $F_{\partial}(d;v)$ of the output functional
→ [GITHUB](#)

```
2 double computeBoundaryOutputFunctional(  
3     const Eigen::VectorXd eta,  
4     const std::shared_ptr<If::uscalfe::FeSpaceLagrangeO1<double>> &fe_space_p,  
5     Eigen::Vector2d d) {  
6     double func_val = 0.0;  
7     // Pointer to current mesh  
8     std::shared_ptr<const If::mesh::Mesh> mesh_p = fe_space_p->Mesh();  
9     // Obtain local->global index mapping for current finite element space  
10    const If::assemble::DofHandler &dofh{fe_space_p->LocGlobMap()};  
11  
12    // Obtain an array of boolean flags for the edges of the mesh, 'true'  
13    // indicates that the edge lies on the boundary  
14    auto bd_flags{If::mesh::utils::flagEntitiesOnBoundary(mesh_p, 1)};  
15  
16    // Creating a predicate that will guarantee that the computations are carried  
17    // only on the interior boundary edges of the mesh using the boundary flags  
18    auto edges_predicate_RobinBC =  
19        [&bd_flags](const If::mesh::Entity &edge) -> bool {  
20        if (bd_flags(edge)) {  
21            auto endpoints = If::geometry::Corners(*(edge.Geometry()));  
22            return endpoints(0, 0) > 0.05 && 0.95 > endpoints(0, 0) &&  
23                endpoints(1, 0) > 0.05 && 0.95 > endpoints(1, 0);  
24        }  
25        return false;  
26    };  
27  
28    // Computing value of the functional  
29    for (const If::mesh::Entity *edge : mesh_p->Entities(1)) {  
30        if (edges_predicate_RobinBC(*edge)) {  
31            // Find the endpoints global indices  
32            auto dof_idx = dofh.GlobalDofIndices(*edge);  
33            assert(dofh.NumLocalDofs(*edge) == 2);  
34            // Value of linear function in the endpoints  
35            const double nu0 = eta(dof_idx[0]);  
36            const double nu1 = eta(dof_idx[1]);  
37            // Find coordinates of endpoints  
38            auto endpoints = If::geometry::Corners(*(edge->Geometry()));  
39            // Length of edge  
40            const double elen = (endpoints.col(1) - endpoints.col(0)).norm();  
41            LF_ASSERT_MSG(endpoints.cols() == 2, "Wrong no endpoints in " << *edge);  
42            func_val += elen / 6.0 *  
43                d.dot(2 * endpoints.col(0) * nu0 + endpoints.col(0) * nu1 +  
44                    endpoints.col(1) * nu0 + 2 * endpoints.col(1) * nu1);  
45        }  
46    }  
47    return func_val;  
48 };
```