

C++ code 2.9.4: Sub-problem (2-9.f): Implementation of `IntegrateLinearFEFunction`

→ [GitLab](#)

```
2 double integrateLinearFEFunction(  
3     const If::assemble::DofHandler& dofhandler,  
4     const Eigen::VectorXd& mu) {  
5     double l = 0;  
6     std::shared_ptr<const If::mesh::Mesh> mesh = dofhandler.Mesh();  
7     for (const auto *cell : mesh->Entities(0)) {  
8         if (cell->RefEl() == If::base::RefEl::kQuad()) {  
9             throw "Only triangular meshes are allowed!";  
10        }  
11        // check if we the FE space is really  $S_1^0$   
12        if (dofhandler.NumLocalDofs(*cell) != 3) {  
13            throw "Not a  $S_1^0$  FE space on a triangular mesh!";  
14        }  
15        // Retrieve area of the triangle by calling  
16        // a LehrFEM++ utility function  
17        const double area = If::geometry::Volume(*(cell->Geometry()));  
18        // iterate over dofs  
19        auto int_dofs = dofhandler.GlobalDofIndices(*cell);  
20        for (auto dof_idx_p = int_dofs.begin(); dof_idx_p < int_dofs.end();  
21            ++dof_idx_p) {  
22            // local integral of the basis function associated with this dof:  
23            // in linear Lagrangian FE, the integral over the basis functions  
24            // over  
25            // a triangle K is:  $1/3 * vol(K)$   
26            const double l_bary = 1.0 / 3.0 * area;  
27            // multiply by the value at the dof to get local contribution  
28            l += l_bary * mu(*dof_idx_p);  
29        }  
30    }  
31    return l;  
}
```