

C++11 code 3.1.15: Sub-problem (3-1.h): Extended `main()` function → [GitLab](#)

```
2 int main() {
3     // read in mesh and set up finite element space
4     auto mesh_factory = std::make_unique<If::mesh::hybrid2d::MeshFactory>(2);
5     If::io::GmshReader reader(std::move(mesh_factory), "meshes/square.msh");
6     auto mesh = reader.mesh();
7     // obtain dofh for lagrangian finite element space
8     auto fe_space =
9         std::make_shared<If::uscalfe::FeSpaceLagrangeO1<double>>(mesh);
10    const If::assemble::DofHandler &dofh{fe_space->LocGlobMap()};
11    // Solve test problem
12    Eigen::VectorXd mu = AvgValBoundary::solveTestProblem(dofh);
13    // compute H1 seminorm of the solution
14    double h1s_norm = AvgValBoundary::compH1seminorm(dofh, mu);
15    // compute boundary functional
16    auto w = [](Eigen::Vector2d x) -> double { return 1.0; };
17    double boundary_functional =
18        AvgValBoundary::compBoundaryFunctional(dofh, mu, w);
19
20    std::cout << "H1s-norm: " << h1s_norm << "\n";
21    std::cout << "F: " << boundary_functional << "\n";
22
23    auto results = AvgValBoundary::approxBoundaryFunctionalValues(7);
24    double ground_truth = results[6].second;
25    std::cout << "N_Dofs    Error\n";
26    for (int i = 0; i < 6; i++) {
27        std::cout << results[i].first << "    "
28            << std::abs(results[i].second - ground_truth) << "\n";
29    }
30    return 0;
31 }
```