

C++ code 0.2.1: Some function to be analyzed with GDB → [GITHUB](#)

```
2 void ReadAndOutputMesh(const char *filename) {
3     if (filename != nullptr) {
4         // Build full path to the mesh file
5         auto gmsfile_path = std::string(CURRENT_SOURCE_DIR) + "/" + filename;
6         // Load the mesh from a file produced by Gmsh
7         auto mesh_factory = std::make_unique<If::mesh::hybrid2d::MeshFactory>(2);
8         If::io::GmshReader reader(std::move(mesh_factory), gmsfile_path);
9         // Obtain pointer to read mesh
10        std::shared_ptr<const If::mesh::Mesh> mesh_p = reader.mesh();
11        const If::mesh::Mesh &mesh{*mesh_p};
12        // Run through all entities of the mesh and print entity information
13        for (int codim = 0; codim <= mesh.DimMesh(); ++codim) {
14            int cnt = 0;
15            Eigen::VectorXd c{Eigen::VectorXd::Zero(mesh.DimWorld())};
16            for (const If::mesh::Entity *entity : mesh.Entities(codim)) {
17                // Number of vertices
18                const int no_vertices = (entity->RefEl()).NumNodes();
19                // Obtain "convex hull" of an entity
20                const Eigen::MatrixXd vertices{
21                    If::geometry::Corners(*entity->Geometry())};
22                // Compute center of gravity
23                c += vertices.rowwise().sum() / no_vertices;
24                cnt++;
25            }
26            if (cnt != mesh.NumEntities(codim)) {
27                std::cerr << "Count mismatch for entities of codim = " << codim << ": "
28                    << cnt << " <-> " << mesh.NumEntities(codim) << std::endl;
29            }
30            c /= cnt;
31            std::cout << "Center of codim-" << codim
32                << " entities = " << c.transpose() << std::endl;
33        }
34        // Write mesh data to file for visualization with Python script
35        If::io::writeMatplotlib(mesh, CURRENT_BINARY_DIR "/ljoint.csv");
36        std::cout << "Wrote " CURRENT_BINARY_DIR "/ljoint.csv" << std::endl;
37        std::system("python3 " CURRENT_SOURCE_DIR
38                    "/plot_mesh.py " CURRENT_BINARY_DIR
39                    "/ljoint.csv " CURRENT_BINARY_DIR "/mesh.eps");
40    }
41 }
```