

C++ code 9.2.33: Part of `main()`-function reading the mesh and running the numerical experiment. → [GitLab](#)

```
2 // Load mesh into a Lehfem++ object
3
4 auto mesh_factory = std::make_unique<lf::mesh::hybrid2d::MeshFactory>(2);
5 const lf::io::GmshReader reader(std::move(mesh_factory),
6                                 "meshes/square64_bnd.msh");
7 auto mesh_p = reader.mesh(); // type shared_ptr< const lf::mesh::Mesh>
8
9 // Finite element space
10 auto fe_space =
11     std::make_shared<lf::uscalfe::FeSpaceLagrangeO1<double>>(mesh_p);
12 // Obtain local->global index mapping for current finite element space
13 const lf::assemble::DofHandler &dofh{fe_space->LocGlobMap()};
14 // Dimension of finite element space
15 const lf::uscalfe::size_type N_dofs(dofh.NumDofs());
16
17 // Building initial condition vector
18 Eigen::VectorXd initial_temperature_vec(N_dofs);
19 for (int idx = 0; idx < N_dofs; idx++) {
20     // Obtain coordinates of vertex at global index idx
21     auto coords = lf::geometry::Corners(*(dofh.Entity(idx).Geometry()));
22     LF_ASSERT_MSG(coords.cols() == 1, "Wrong no of coords in vertex");
23     initial_temperature_vec(idx) = 5.0;
24 }
25
26 // SDIRK-2 evolution of parabolic problem
27 unsigned int m = 100;
28 std::pair<Eigen::VectorXd, Eigen::VectorXd> solution_pair =
29     solveTemperatureEvolution(dofh, m, 1.0, initial_temperature_vec);
30 Eigen::VectorXd discrete_temperature_sol = solution_pair.first;
31 LF_ASSERT_MSG(
32     discrete_temperature_sol.size() == N_dofs,
33     "Size of discrete solution and dimension of FE space mismatch.");
34 Eigen::VectorXd energies = solution_pair.second;
35 LF_ASSERT_MSG(energies.size() == m + 1, "Wrong number of energie values.");
36 // Define output file format for the energies
37 const static Eigen::IOFormat CSVFormat(Eigen::StreamPrecision,
38                                         Eigen::DontAlignCols, " ", "\n");
39
40 // Corresponding time grid for plotting
41 Eigen::VectorXd time = Eigen::VectorXd::LinSpaced(m + 1, 0.0, 1.0);
42
43 // Write .csv file of energy vs. time
44 std::ofstream file;
45 file.open("energies.csv");
46 file << time.transpose().format(CSVFormat) << std::endl;
47 file << energies.transpose().format(CSVFormat) << std::endl;
48 file.close();
49 std::cout << "Generated energies.csv" << std::endl;
50
51 // Plot from .csv file using python
52 systemcall::execute(
53     "python3 ms_scripts/plot_energies.py energies.csv "
54     "energies.eps");
```