



## C++11 code 9.5.22: Implementation of function `wavePropSimulation()` → [GitLab](#)

```

2 void wavePropSimulation(unsigned int m) {
3     double T = 10.0; // final time
4
5     // Load mesh into a Lehigh++ object
6     auto mesh_factory = std::make_unique<lf::mesh::hybrid2d::MeshFactory>(2);
7     const lf::io::GmshReader reader(std::move(mesh_factory), "meshes/hex4.msh");
8     auto mesh_p = reader.mesh();
9
10    // Finite element space
11    auto fe_space =
12        std::make_shared<lf::uscalfe::FeSpaceLagrangeO1<double>>(mesh_p);
13    // Obtain local->global index mapping for current finite element space
14    const lf::assemble::DofHandler &dofh{fe_space->LocGlobMap()};
15    // Dimension of finite element space
16    const lf::uscalfe::size_type N_dofs(dofh.NumDofs());
17
18    // Building initial condition vector
19    Eigen::VectorXd u0_vec(N_dofs);
20    Eigen::VectorXd v0_vec(N_dofs);
21    u0_vec.setOnes();
22    v0_vec.setZero();
23
24    // Creating coefficient function
25    auto c = [] (Eigen::Vector2d x) -> double { return 1.0 + x.dot(x); };
26
27    // Solve the initial value problem
28    std::pair<Eigen::VectorXd, Eigen::VectorXd> solution_pair =
29        solvewave(fe_space, c, u0_vec, v0_vec, T, m);
30    Eigen::VectorXd discrete_wave_sol = solution_pair.first;
31    LF_ASSERT_MSG(
32        discrete_wave_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
37    // Define output file format for the energies
38    const static Eigen::IOFormat CSVFormat(Eigen::FullPrecision,
39        Eigen::DontAlignCols, " ", " ", "\n");
40    std::string errors_file_name = "waves_energies.csv";
41    std::ofstream file(errors_file_name.c_str());
42    if (file.is_open()) {
43        file << energies.format(CSVFormat);
44    }
45
46    std::cout << ">>The energies were written to:" << std::endl;
47    std::cout << "\t waves_energies.csv" << std::endl;
48    /* SAM_LISTING_BEGIN_2 */
49    // Output results for the wave function to vtk file
50    lf::io::VtkWriter vtk_writer(mesh_p, "discrete_wave_sol.vtk");
51    // Write nodal data taking the values of the discrete solution at the
52    // vertices
53    auto nodal_data = lf::mesh::utils::make_CodimMeshDataSet<double>(mesh_p, 2);
54    for (int global_idx = 0; global_idx < N_dofs; global_idx++) {
55        nodal_data->operator()(dofh.Entity(global_idx)) =
56            discrete_wave_sol[global_idx];
57    };
58
59    vtk_writer.WritePointData("discrete_wave_sol" * nodal_data);

```