

C++ code 9.7.16: Implementation of the constructor of WaveABC2DTimestepper → GitLab

```

2  template <typename FUNC_RHO, typename FUNC_MU0, typename FUNC_NU0>
3  WaveABC2DTimestepper<FUNC_RHO, FUNC_MU0, FUNC_NU0>::WaveABC2DTimestepper(
4      const std::shared_ptr<If::uscalfe::FeSpaceLagrangeO1<double>> &fe_space_p,
5      FUNC_RHO rho, unsigned int M, double T)
6
7      : fe_space_p_(fe_space_p),
8        M_(M),
9        T_(T),
10       step_size_(T / M),
11       timestepping_performed_(false) {
12     /* Creating coefficient-functions as Lehrfem++ mesh functions */
13     // Coefficient-functions used in the class template
14     // ReactionDiffusionElementMatrixProvider and MassEdgeMatrixProvider
15     auto rho_mf = If::mesh::utils::MeshFunctionGlobal(rho);
16     auto zero_mf = If::mesh::utils::MeshFunctionGlobal(
17         [](Eigen::Vector2d) -> double { return 0.0; });
18     auto one_mf = If::mesh::utils::MeshFunctionGlobal(
19         [](Eigen::Vector2d) -> double { return 1.0; });
20
21     // On construction no timestepping was yet performed
22
23     std::cout << "Assembling Galerkin matrices..." << std::endl;
24     If::assemble::COOMatrix<double> A_COO = computeGalerkinMat(
25         fe_space_p, one_mf, zero_mf, zero_mf); // stiffness matrix
26     If::assemble::COOMatrix<double> M_COO =
27         computeGalerkinMat(fe_space_p, zero_mf, rho_mf, zero_mf); // Mass matrix
28     If::assemble::COOMatrix<double> B_COO =
29         computeGalerkinMat(fe_space_p, zero_mf, zero_mf,
30             one_mf); // Boundary mass matrix
31
32     const If::assemble::DofHandler &dofh{fe_space_p->LocGlobMap()};
33     N_dofs_ = dofh.NumDofs();
34     std::cout << "Number of degrees of freedom : " << N_dofs_ << std::endl;
35
36     std::cout << "Assembling the evolution matrix..." << std::endl;
37     /* Assemble the full linear system matrix of the stepping method */
38     /*
39     //      L = [ M + (1/2)*tau*B      (1/2)*tau*A ]
40     //           [ - (1/2)*tau*I      I ]
41     // */
42     If::assemble::COOMatrix<double> L_COO(2 * N_dofs_, 2 * N_dofs_);
43     A_triplets_vec_ = A_COO.triplets();
44     M_triplets_vec_ = M_COO.triplets();
45     const std::vector<Eigen::Triplet<double>> B_triplets_vec = B_COO.triplets();
46     // Inserting M in L
47     for (auto &triplet : M_triplets_vec_) {
48         L_COO.AddToEntry(triplet.row(), triplet.col(), triplet.value());
49     }
50     // Inserting B in L
51     for (auto &triplet : B_triplets_vec) {
52         L_COO.AddToEntry(triplet.row(), triplet.col(),
53             0.5 * step_size_ * triplet.value());
54     }
55     // Inserting A in L
56     for (auto &triplet : A_triplets_vec_) {
57         L_COO.AddToEntry(triplet.row(), triplet.col() + N_dofs_,

```