C++ code 9.7.15: Class definition of WaveABC2DTimestepper ↬ GitLab

```cpp
template <typename FUNC_RHO, typename FUNC_MU0, typename FUNC_NU0>
class WaveABC2DTimestepper {
 public:
  // Main constructor; precomputations are done here
  WaveABC2DTimestepper(
      const std::shared_ptr<lf::uscalfe::FeSpaceLagrangeO1<double>> &fe_space_p,
      FUNC_RHO rho, unsigned int M, double T);

  // Public member functions
  Eigen::VectorXd solveWaveABC2D(FUNC_MU0 mu0, FUNC_NU0 nu0);
  double energies();

 private:
  double T_;          // final time
  unsigned int M_;    // nb of steps
  double step_size_;  // time inverval
  std::shared_ptr<lf::uscalfe::FeSpaceLagrangeO1<double>> fe_space_p_;
  bool timestepping_performed_;  // bool to assert that energies are computed
  lf::uscalfe::size_type N_dofs_;  // nb of degrees of freedom
                                   // only after timestepping
  // Precomputed objects
  std::vector<Eigen::Triplet<double>> A_triplets_vec_;  // stiffness matrix
  std::vector<Eigen::Triplet<double>> M_triplets_vec_;  // mass matrix
  Eigen::SparseMatrix<double> R_;                       // rhs evaluation matrix
  Eigen::SparseLU<Eigen::SparseMatrix<double>> solver_;  // linear solver
  Eigen::VectorXd
      full_sol_;  // Vector for discrete solution and discrete velocity
};  // class WaveABC2DTimestepper
```