

C++ code 9.7.13: Implementation of `scalarImplicitTimestepping()` → [GitLab](#)

```
2 Eigen::VectorXd scalarImplicitTimestepping(double epsilon, unsigned int M) {
3     /* PROBLEM SETUP */
4     double step_size = 1.0 / M; // time step "tau"
5     // Rows contain sequence of solutions states
6     Eigen::MatrixXd x(M + 1, 2);
7     // INITIAL DATA
8     x(0, 0) = 1.0;
9     x(0, 1) = 1.0;
10
11     Eigen::MatrixXd A(2, 2);
12     Eigen::VectorXd b(2);
13     // clang-format off
14     A << epsilon + 0.5*step_size*(1-epsilon), 0.5*step_size,
15         -0.5*step_size, 1.0;
16     // clang-format on
17     // TIMESTEPPING
18     for (int k = 1; k < M + 1; k++) {
19         // Update right hand side
20         b(0) = (epsilon - 0.5 * step_size * (1 - epsilon)) * x(k - 1, 0) -
21             0.5 * step_size * x(k - 1, 1);
22         b(1) = x(k - 1, 1) + 0.5 * step_size * x(k - 1, 0);
23         // Solve the linear system
24         x.row(k) = A.fullPivLu().solve(b);
25     }
26     return x.col(1);
27 } // scalarImplicitTimestepping
```