

C++ code 11.5.17: Sub-problem (11-5.k): function `solveTrafficFlow()` → [GitLab](#)

```
2 Solution solveTrafficFlow() {
3     int Ml = 40;
4     int Mr = 40;
5     int N_half = Mr + Ml + 1;
6     int N = 2 * N_half;
7
8     double h = 0.05;
9     double tau = h / 3;
10    double T = 1.0;
11    unsigned int m = (unsigned int)(T / tau);
12
13    // Spatial mesh
14    Eigen::VectorXd x = Eigen::VectorXd::LinSpaced(N_half, -2.0, 2.0);
15    // Initial condition according to (11.5.16). Note that we just
16    // initialize the cell averages; a more refined implementation would
17    // also sample the local first moments of  $u_0$ .
18    Eigen::VectorXd mu0 = Eigen::VectorXd::Zero(N);
19    auto u0 = [](double x) { return 0.0 <= x && x <= 1.0 ? 1.0 : 0.0; };
20    for (int i = 0; i < N_half; ++i) {
21        mu0(2 * i) = u0(x(i));
22    }
23    // Flux function
24    auto f = [](double u) { return u * (1.0 - u); };
25    // Perform fully discrete evolution
26    Eigen::VectorXd mu = dgcl(mu0, f, Feo, T, Ml, Mr, h, m);
27    // Retrieve cell averages
28    Eigen::VectorXd u(N_half);
29    for (int i = 0; i < N_half; ++i) {
30        u(i) = mu(2 * i);
31    }
32
33    return Solution(std::move(x), std::move(u));
34 }
```