

C++11 code 11.2.9: Sub-problem (11-2.h): function `solveCP()` → [GitLab](#)

```
2 Eigen::VectorXd solveCP(double a, double b, Eigen::VectorXd u0, double T) {
3     // Find the maximal speed of propagation
4     double A = u0.minCoeff();
5     double B = u0.maxCoeff();
6     double K = std::max(std::abs(std::sinh(A)), std::abs(std::sinh(B)));
7     // Set uniform timestep according to CFL condition
8     int N = u0.size();
9     double h = (b - a) / N;
10    double tau_max = h / K;
11    double timesteps = std::ceil(T / tau_max);
12    double tau = T / timesteps;
13
14    // Main timestepping loop
15    Eigen::VectorXd mu(N);
16    for (int i = 0; i < timesteps; ++i) {
17        mu.swap(u0);
18        double F_minus;
19        double F_plus = EngquistOsherNumFlux(mu(0), mu(0));
20        for (int j = 0; j < N - 1; ++j) {
21            F_minus = F_plus;
22            F_plus = EngquistOsherNumFlux(mu(j), mu(j + 1));
23            u0(j) = mu(j) - tau / h * (F_plus - F_minus);
24        }
25        F_minus = F_plus;
26        F_plus = EngquistOsherNumFlux(mu(N - 1), mu(N - 1));
27        u0(N - 1) = mu(N - 1) - tau / h * (F_plus - F_minus);
28    }
29
30    return u0;
31 }
```