C++ code 11.9.14: Sub-problem (11-9.g): Implementation of `traceMass()` ⇝ GitLab

```cpp
template <typename U0Functor>
Eigen::VectorXd traceMass(U0Functor &&u0, unsigned int N) {
  // Spacial boundaries
  double a = -5.0;
  double b = 10.0;

  // Compute μ(t = 0)
  Eigen::VectorXd x = Eigen::VectorXd::LinSpaced(N, a, b);
  Eigen::VectorXd mu = x.unaryExpr(std::forward<U0Functor>(u0));

  // Get timestep size and number by CLF condition
  double h = (b - a) / N;
  double s_max = std::exp(1.0) - 1.0;  // since 0 ≤ u_0(x) ≤ 1
  double tau = h / s_max;              // stencil m = 1
  double T = 3.0;
  // Total number of timesteps
  int M = (int)std::ceil(T / tau);
  tau = 3.0 / M;  // Timestep size

  // Compute solution and total masses at different times
  auto totalMass = [h](const Eigen::VectorXd &mu) { return (mu * h).sum(); };
  Eigen::VectorXd m(M + 1);
  for (int i = 0; i < M; ++i) {
    m(i) = totalMass(mu);
    auto s = [](double u) { return -u; };
    mu = mu + tau * (-1.0 / h) * fluxdiffsource(mu, &godnfn, s, h);
  }
  m(M) = totalMass(mu);

  return m;
}
```