C++ code 11.4.11: Sub-problem (11-4.g): function `numexpLaxWendroffRP()` ➙ GITHUB

```cpp
// Build spatial grid
Eigen::VectorXd getXValues(double T, unsigned int M) {
  double tau = T / M;
  double h = Constant::e * tau;
  int j_max = (int)(std::ceil((3.0 * T + 1.0) / h) + 0.5);
  int j_min = (int)(std::floor(-3.0 * T / h) - 0.5);
  unsigned int N = j_max - j_min + 1;
  return Eigen::VectorXd::LinSpaced(N, j_min * h, j_max * h);
}
Eigen::VectorXd numexpLaxWendroffRP(const Eigen::VectorXi &M) {
  const double T = 1.0;
  const int M_size = M.size();
  Eigen::VectorXd error(M_size);
  // Initial values for the Riemann problem
  auto u_initial = [](double x) { return 0.0 <= x ? 1.0 : 0.0; };
  // Exact solution (??) at time T = 1.0
  auto u_exact = [](double x) {
    return (x <= 1.0) ? 0.0 : ((Constant::e <= x) ? 1.0 : std::log(x));
  };
  for (int i = 0; i < M_size; ++i) {
    Eigen::VectorXd x = getXValues(T, M(i));
    Eigen::VectorXd u0 = x.unaryExpr(u_initial);
    Eigen::VectorXd uT = solveLaxWendroff(u0, T, M(i));

    double tau = T / M(i);
    double h = Constant::e * tau;
    error(i) = h * (x.unaryExpr(u_exact) - uT).lpNorm<1>();
  }
  return error;
}
```