C++ code 11.5.11: Sub-problem (11-5.h): function G() ⇝ GitLab

```
template <typename FUNCTOR, typename NUMFLUX>
Eigen::VectorXd G(const Eigen::VectorXd &mu, FUNCTOR &&f, NUMFLUX &&F, int Ml,
                  int Mr, double h) {
  const int N_half = (Ml + Mr + 1);
  const int N = 2 * N_half;
  Eigen::VectorXd Gvec(N);
  double uN_xminus = 0.0;  // since we extend mu to the left by zero
  double uN_xplus = mu(0) - 0.5 * h * mu(1);
  double F_old;
  double F_new = F(uN_xminus, uN_xplus);
  const double w = h / (2.0 * std::sqrt(3.0));

  for (int i = 0; i < N_half - 1; ++i) {
    uN_xminus = mu(2 * i) + 0.5 * h * mu(2 * i + 1);
    uN_xplus = mu(2 * (i + 1)) - 0.5 * h * mu(2 * (i + 1) + 1);
    F_old = F_new;
    F_new = F(uN_xminus, uN_xplus);
    Gvec(2 * i) = F_new - F_old;

    double x_minus = mu(2 * i) - w * mu(2 * i + 1);
    double x_plus = mu(2 * i) + w * mu(2 * i + 1);
    double I = 0.5 * h * (f(x_minus) + f(x_plus));
    Gvec(2 * i + 1) = 0.5 * h * (F_new + F_old) - I;
  }

  uN_xminus = mu(2 * (N_half - 1)) + 0.5 * h * mu(2 * (N_half - 1) + 1);
  uN_xplus = 0.0;  // since we extend mu to the right by zero
  F_old = F_new;
  F_new = F(uN_xminus, uN_xplus);
  Gvec(2 * (N_half - 1)) = F_new - F_old;

  double x_minus = mu(2 * (N_half - 1)) - w * mu(2 * (N_half - 1) + 1);
  double x_plus = mu(2 * (N_half - 1)) + w * mu(2 * (N_half - 1) + 1);
  double I = 0.5 * h * (f(x_minus) + f(x_plus));
  Gvec(2 * (N_half - 1) + 1) = 0.5 * h * (F_new + F_old) - I;
  return Gvec;
}
```