

C++ code 6.7.10: Sub-problem (6-7.f): Convergence of Hermite interpolation with reconstructed slopes

```
2 std::vector<double> rspchipConvergence() {
3     auto f = [](double x) { return 1. / (1. + x * x); };
4
5     constexpr double a = 5; // Interval bounds will be (-a,a)
6     constexpr unsigned int M =
7         1000; // Number of points in which to evaluate the interpolant
8
9     // Precompute values at which evaluate f
10    Eigen::VectorXd x = Eigen::VectorXd::LinSpaced(M, -a, a);
11    Eigen::VectorXd fx = x.unaryExpr(f);
12
13    // Store error and number of nodes
14    std::vector<double> N_nodes, err_reconstr;
15
16    // TODO: (6-7.f) Compute err_reconstr.
17    // START
18    for (unsigned int i = 4; i <= 512; i = i << 1) {
19        // Define subintervals and evaluate f there (find pairs (t,y))
20        Eigen::VectorXd t = Eigen::VectorXd::LinSpaced(i, -a, a);
21
22        // Construct PCHI with zero and reconstructed slopes
23        CubicHermiteInterpolant s_reconstr(f, t);
24
25        // Evaluate interpolant
26        Eigen::VectorXd s_reconstr_x = s_reconstr.eval(x);
27
28        // Compute infinity norm of error
29        err_reconstr.push_back((s_reconstr_x - fx).lpNorm<Eigen::Infinity>());
30        N_nodes.push_back(i);
31
32        if (i == 16) {
33            plt::figure();
34            plt::title("Interpolant with reconstructed slope");
35            plt::xlabel("t");
36            plt::ylabel("y");
37            plt::plot(x, s_reconstr_x, "r", {{ "label", "s_reconstr" }});
38            plt::plot(x, fx, "b--", {{ "label", "f" }});
39            plt::legend();
40            plt::savefig("./cx_out/p_reconstr.png");
41        }
42    }
43    // END
44
45    return err_reconstr;
46 }
```

Get it on 🐙 [GitLab](#) (pchi.hpp).