

C++ code 8.4.4: Testing the order of Steffensen's method

```
2 void orderSteffensen() {
3     auto f = [](double x) { return x * std::exp(x) - 1; };
4
5     constexpr double x_star = 0.567143290409784; // use as exact value
6
7     // TODO: (8-4.c) Tabulate values from which you can read the
8     // order of Steffensen's method.
9     // Hint: To approximate the convergence rate, use the formula
10    //  $(\log(e_i) - \log(e_{i-1})) / (\log(e_{i-1}) - \log(e_{i-2}))$ 
11    // START
12
13    std::vector<double> store;
14
15    auto log = [&store](double x) -> void { store.push_back(x); };
16
17    steffensen_log(f, 1.0, log);
18
19    const unsigned n = store.size();
20
21    Eigen::VectorXd errs(n), log_errs(n);
22    for (unsigned i = 0; i < n; ++i) {
23        errs(i) = std::abs(store[i] - x_star);
24        log_errs(i) = std::log(errs(i));
25    }
26
27    Eigen::VectorXd ratios = Eigen::VectorXd::Zero(n);
28    for (unsigned i = 2; i < n; ++i) {
29        ratios(i) =
30            (log_errs(i) - log_errs(i - 1)) / (log_errs(i - 1) - log_errs(i - 2));
31    }
32
33    // Print output
34    std::cout.precision(10);
35    std::cout << std::setw(20) << "x" << std::setw(20) << "errors"
36              << std::setw(20) << "ratios" << std::endl;
37    std::cout << "
38              << std::endl;
39    for (unsigned i = 0; i < n; ++i) {
40        std::cout << std::setw(20) << store[i] << std::setw(20) << errs(i)
41              << std::setw(20) << ratios(i) << std::endl;
42    }
43
44    // END
45 }
```

Get it on 🐱 GitLab ([quadraticconvergence.hpp](#)).