

C++ code 7.12.6: Implementation of `determineOrderCCF()`

```
2 unsigned int determineOrderCCF(unsigned int n) {
3     const CCFQuadRule ccqqr(n);
4     unsigned int d =
5         0;           // Degree of monomial, whose integral value is being
6         tested
7     double quaderr; // Quadrature error
8     double l_ex;    // Exact integral value
9     double l_qr;    // Value computed with CCF quadrature rule
10    // Predefined fixed tolerances
11    const double abstol = std::numeric_limits<double>::epsilon() * n * 2;
12    const double reltol = std::numeric_limits<double>::epsilon() * n * 20;
13    // Main quadrature loop
14    do {
15        // Compute powers of nodes:  $c_j^d$ 
16        const Eigen::ArrayXd pdvals = ccqqr.nodes().array().pow(d);
17        // Evaluate quadrature formula: Implicit summation via dot product
18        l_qr = ccqqr.weights().dot(pdvals.matrix());
19        // Exact integral value according to (7.12.4)
20        l_ex = (d % 2 == 0) ? (2.0 / (d + 1)) : 0.0;
21        quaderr = std::abs(l_qr - l_ex);
22        d = d + 1;
23    } while ((quaderr <= abstol) ||
24            (quaderr <= reltol * l_ex)); //
25    return d - 1;
26 }
```

Get it on  GitLab (clenshawcurtisfejer.hpp).