

SOLUTION of (6-4.d):

As a consequence of [Lecture $\rightarrow$ Thm. 6.2.3.4], we already know that $\{T_0, \dots, T_n\}$ is a basis for $\mathcal{P}_n$. We check the orthogonality with the C++ code given in ??.

C++ code 6.4.6: Code [Lecture $\rightarrow$ Code 6.2.3.6]

```
2 // Computes the values of the Chebychev polynomials  $T_0, \dots, T_d$ 
3 // at points passed in x using the 3-term recursion
4 // (??). The values  $T_k(x_j)$ , are returned in
5 // row  $k+1$  of V.
6 void chebpolmult(const unsigned int d, const RowVectorXd& x, MatrixXd& V) {
7     const unsigned int n = x.size();
8     V = MatrixXd::Ones(d + 1, n); //  $T_0 \equiv 1$ 
9     if (d == 0) return;
10    V.block(1, 0, 1, n) = x; //  $T_1(x) = x$ 
11    if (d == 1) return;
12    for (unsigned int k = 1; k < d; ++k) {
13        const RowVectorXd p = V.block(k, 0, 1, n); //  $p = T_k$ 
14        const RowVectorXd q = V.block(k - 1, 0, 1, n); //  $q = T_{k-1}$ 
15        V.block(k + 1, 0, 1, n) =
16            2 * x.cwiseProduct(p) - q; // 3-term recursion
17    }
18 }
```

C++ code 6.4.7: Solution of (6-4.d)

```
2 bool checkDiscreteOrthogonality(unsigned int n) {
3     bool return_val = false;
4     constexpr double tol =
5         std::numeric_limits<double>::epsilon(); // machine epsilon
6
7     // TODO: (6-4.d) Check the orthogonality property of the Chebychev
8     // polynomials
9     // up to order n.
10    // START
11    Eigen::MatrixXd V;
12    // generate Chebychev nodes
13    Eigen::RowVectorXd x = Eigen::RowVectorXd::LinSpaced(n + 1, 0, n);
14    x = x.unaryExpr(
15        [n](double x) { return std::cos(M_PI * (2. * x + 1.) / 2. / (n + 1.)); });
16    chebpolmult(n, x, V); // compute values of polynomials
17
18    // check for orthogonality
19    // note that the second loop starts at k + 1 because this inner
20    // product is
21    // (anti-)symmetric
22    return_val = true;
23    for (unsigned int k = 0; k < n + 1; ++k) {
24        for (unsigned int l = k + 1; l < n + 1; ++l) {
25            return_val &= (std::abs(V.row(k).dot(V.row(l))) < tol * 100.);
26        }
27    }
28    // END
29 }
```

```
28     return return_val;  
29 }
```

Get it on  GitLab (chebpolyproperties.hpp).
