

C++ code 3.16.4: Class definition of CompactStorageQR

```
2 class CompactStorageQR {
3 public:
4     /**
5      * \brief Construct a new CompactStorageQR object
6      *
7      * \param data raw array which has to be persistent
8      * \param n number of rows/columns of encoded matrix
9      */
10    CompactStorageQR(const double *data, unsigned int n) : n_(n), M_(data, n, n) {
11        for (unsigned int k = 0; k < n_ - 1; ++k) {
12            double sn{0};
13            for (unsigned int j = k + 1; j < n_; ++j) {
14                const double t{M_(j, k)};
15                sn += t * t;
16            }
17            if (sn > 1.0) {
18                throw std::runtime_error(
19                    "CompactStorageQR: Illegal subdiagonal column norm!");
20            }
21        }
22    }
23
24    /**
25     * \brief Determinant of the matrix
26     *
27     * \return double the determinant
28     */
29    double det() const;
30
31    /**
32     * \brief Right multiplication with another matrix
33     *
34     * \param X matrix to multiply
35     * \return Eigen::MatrixXd product
36     */
37    Eigen::MatrixXd matmult(const Eigen::MatrixXd &X) const;
38
39    /**
40     * \brief Solution of linear systems of equations
41     *
42     * \param B right hand side
43     * \return Eigen::MatrixXd solution matrix
44     */
45    Eigen::MatrixXd solve(const Eigen::MatrixXd &B) const;
46
47 private:
48     unsigned int n_; // Matrix dimensions
49     Eigen::Map<const Eigen::MatrixXd> M_; // Raw data wrapped into a matrix
50 };
```

Get it on  GitLab (compactstorageqr.hpp).