

C++ code 0.2.2: Swapping left and right parts of a matrix in EIGEN.

```
2 Eigen::MatrixXd swap_left_right_blocks(const Eigen::MatrixXd& A,
3                                         unsigned int p) {
4     assert(A.cols() >= p && "Can only swap at most A.cols() columns.");
5     Eigen::MatrixXd B, C;
6
7     // We can use .rows() and .cols() to get the number of rows and
8     // columns in A.
9     unsigned int q = A.cols() - p;
10
11    // Make a copy of A.
12    Eigen::MatrixXd Anew(A);
13
14    // TODO: (0-2.b) Swap the first p columns of A with the rightmost q
15    // columns
16    // and write them in Anew. The methods block(),
17    // leftCols()/rightCols() may
18    // come in handy.
19    // Hint: A.block( i, j, m, n ) returns the m by n block that has its
20    // top-left
21    // corner at the index (i,j) in A.
22    // START
23    Anew.leftCols(q) = A.block(0, p, A.rows(), q);
24    Anew.rightCols(p) = A.block(0, 0, A.rows(), p);
25    // END
26
27    // Tip: Many more methods exist that are special cases of block(),
28    // e.g. topRows(), bottomRows(), topLeftCorner(), bottomLeftCorner(),
29    // For vectors we have head(), tail(), and segment().
30
31    return Anew;
32 }
```

Get it on  GitLab (MatrixBlocks.hpp).