source/utils.hpp

```cpp
#pragma once

#include "firstparty/proto_utils.hpp"
#include "thirdparty/Eigen/Dense"
#include "thirdparty/Eigen/src/Core/Matrix.h"
#include "thirdparty/Eigen/src/Core/util/Meta.h"
#include <limits>



using Matrix    = Eigen::MatrixXd;
using Vector    = Eigen::VectorXd;
using RowVector = Eigen::RowVectorXd;
using Idx       = Eigen::Index; // Eigen uses signed (!) indeces

constexpr bool collapse_small_values = false;

// Eigen has formatting options built-in, but I prefer the style of my own package.
// Eigen stores matrices as col-major so we do a matrix view into the CR memory layout.
inline std::string stringify_matrix(const Matrix& eigen_matrix) {
    using namespace utl;

    if constexpr (collapse_small_values) {
        utl::mvl::Matrix<double> mvl_matrix(
            eigen_matrix.rows(), eigen_matrix.cols(),
            [&](std::size_t i, std::size_t j) { return (std::abs(eigen_matrix(i, j)) < 1e-12) ? 0. : eigen_matrix(i, j); });
        return utl::mvl::format::as_matrix(mvl_matrix);
    } else {
        mvl::ConstMatrixView<double, mvl::Checking::BOUNDS, mvl::Layout::CR> view(
            eigen_matrix.rows(), eigen_matrix.cols(), eigen_matrix.data());
        return mvl::format::as_matrix(view);
    }
}
```