source/utils.hpp

```cpp
1   #pragma once
2
3   #include "firstparty/proto_utils.hpp"
4   #include "thirdparty/Eigen/Dense"
5   #include "thirdparty/Eigen/src/Core/Matrix.h"
6   #include "thirdparty/Eigen/src/Core/util/Meta.h"
7   #include <limits>
8
9
10
11  using Matrix    = Eigen::MatrixXd;
12  using Vector    = Eigen::VectorXd;
13  using RowVector = Eigen::RowVectorXd;
14  using Idx       = Eigen::Index; // Eigen uses signed (!) indeces
15
16  constexpr bool collapse_small_values = false;
17
18  // Eigen has formatting options built-in, but I prefer the style of my own
    package.
19  // Eigen stores matrices as col-major so we do a matrix view into the CR memory
    layout.
20  inline std::string stringify_matrix(const Matrix& eigen_matrix) {
21      using namespace utl;
22
23      if constexpr (collapse_small_values) {
24          utl::mvl::Matrix<double> mvl_matrix(
25              eigen_matrix.rows(), eigen_matrix.cols(),
26              [&](std::size_t i, std::size_t j) { return (std::abs(eigen_matrix(i,
    j)) < 1e-12) ? 0. : eigen_matrix(i, j); });
27          return utl::mvl::format::as_matrix(mvl_matrix);
28      } else {
29          mvl::ConstMatrixView<double, mvl::Checking::BOUNDS, mvl::Layout::CR> view(
30              eigen_matrix.rows(), eigen_matrix.cols(), eigen_matrix.data());
31          return mvl::format::as_matrix(view);
32      }
33  }
```