Scientific Programming
(Wissenschaftliches Programmieren)

Exercise 8

Preparation

- Download the pylintrc config file from the course web site and store in your HOME directory as `~/.pylintrc`

1. Testing with Pytest

- Rewrite the tests in `test_solvers.py` so that they can be used by PyTest.
- Run the PyTest tester from the shell command line.
- Run the PyTest tester within your IDE.
- Commit your changes.

2. Code analysis by Pylint

- Check the quality of your Python source files (`solvers.py`, `test_solvers.py`) with Pylint within Spyder.
- Change the source files to obtain a Pylint score of 10.0.
- Apply the Black formatter to obtain further stylistical changes.
- Commit your changes.

3. Parametrized tests

- Rewrite the tests in `test_solvers.py`, so that they read the data for A, b and the expected solution x from files as follows:
- Create for each test an input file (containing A and b) and whenever it makes sense also an output file (containing the expected x) in a `testdata/` subfolder within your project. (e.g. `testdata/simple.in`, `testdata/simple.out`, `tesdata/needs_pivot.in`, `testdata/needs_pivot.out`, `testdata/linearly_dependent.in`).
- Replace the tests in `test_solvers.py` with parameterized tests, which read their necessary data from those files. Create one parameterized test routine which checks for successful eliminations (calls the Gaussian elimination with A and b read from `testname.in` and compares the obtained result with the reference solution read from `testname.out`) and one parameterized test which checks for linearly dependant systems (calls the Gaussian elimination with A and b read from `testname.in` and checks whether the elimination indicated linear dependency by returning None) .The parametrized test functions should take the name of the actual test as parameter.
- Commit your changes (including the test data in the `testdata/` folder).

4. *Gaussian elimination with partial pivoting

- Extend the Gaussian elimination of the last exercise with partial pivoting: Inspect the absolute values of the current column in all rows below the current one. Exchange the current row with the one containing the highest absolute value before doing the elimination.
- Make sure that the first two tests in `test_solvers.py` return the right values.
- Commit your changes.

5. *Gaussian elimination with dependency detection*

- Extend the Gaussian elimination with detection for linear dependency.
- Make sure that all three tests in `test_solvers.py` return the right values.
- Commit your changes.