

Scientific Programming (Wissenschaftliches Programmieren)

Exercise 7

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 Spyder.
- 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.
- 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.
- Create for each test separate input files (containing A and b) and whenever it makes sense also output files (containing the expected x) in a `testdata/` subfolder within your project. (e.g. `testdata/simple.in`, `testdata/simple.out`, `testdata/needs_pivot.in`, `testdata/needs_pivot.out`, `testdata/linearly_dependent.in`).
- Replace the tests in `test_solvers.py` with parameterized tests, reading their necessary data from those files. Create one parameterized test checking for successful eliminations (reading `testname.in` and `testname.out`) and one parameterized test checking for linearly dependant systems. The test functions should take the name of the test as argument.
- Set up your test file, so that the right testnames are passed to the right test functions using Pytests parameterized test framework.

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.