Developing unit tests for Osdag using PyTest

(1) Introduction

This report documents the unit testing framework developed for the Osdag project. Osdag is an open-source software for the design of steel structures. The objective of this project was to verify the accuracy of the Osdag computational modules by comparing their outputs against known validated results.

(2) Project Components

The unit testing framework consists of the following key components:

(2.1) Test Driver (`test_osdag.py`)

This Python script serves as the main entry point for the testing suite. It utilizes the **PyTest** framework to:

- Parse the configuration CSV files.
- Parametrize test cases based on input files.
- Execute the Osdag modules via the CLI.
- Comparatively assert the actual outputs against expected values.

(2.2) Input Data

The testing suite utilizes a set of text-based input files (e.g., `CleatAngleTest1`, `FinPlateTest1`) which contain the specific design parameters for each test case. These files are passed directly to the Osdag kernel.

(2.3) Reference Data (CSVs)

The "Ground Truth" for the tests is stored in CSV format (`CleatAngle.csv`, `FinPlate.csv`, `TensionMember.csv`). These files map each input file to a set of expected output parameters, such as capacities, efficiency ratios, and geometric checks.

(3) Methodology

The verification process involved the following steps:

(3.1) Test Framework

PyTest was selected as the testing framework due to its flexibility, powerful assertion introspection, and support for parameterized testing.

(3.2) Data Sources

Input parameters and expected output values were sourced from CSV files provided for specific modules:

- `CleatAngle.csv`
- `FinPlate.csv`
- `TensionMember.csv`

(3.3) Parameter Mapping

A key challenge was mapping the headers in the CSV files (which act as the ground truth) to the internal dictionary keys returned by the Osdag modules. A mapping dictionary `KEY_MAPPING` was implemented in `test_osdag.py` to bridge these naming conventions. This mapping handles:

- Direct one-to-one matches (e.g., "Weld Size" → `Weld.Size`).
- Ambiguous terms (e.g., distinguishing between "Bolt Rows" and "Bolt Columns" based on context).
- Type conversions (parsing strings to floats for numerical comparison).

(4) Measured Modules

The unit tests cover three primary connection/member types:

1. **Cleat Angle Connection**: Tests for angle section designations, shear capacity, and geometric constraints.
2. **Fin Plate Connection**: Tests for plate dimensions, bolt configurations, and weld strengths.
3. **Tension Member**: Tests for tension capacity, yielding, and rupture checks.

(5) Results

The unit tests were executed to validate the module outputs against the reference CSV data.

(5.1) Test Summary

Initial Run (File-based)

- Total Input Files Verified: 12
- Status: All Passed

Detailed Run (Granular)

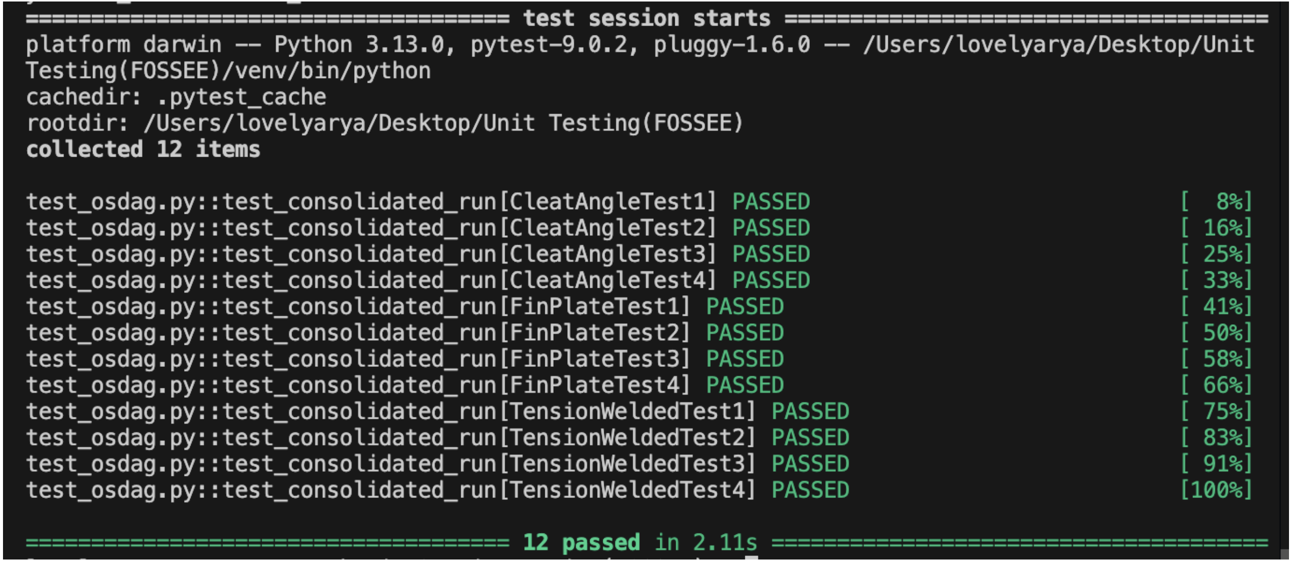
- Total Tests Collected: 48
- Tests Passed: 45
- Tests Skipped: 3
- Tests Failed: 0

(5.2) Execution Log

The following describes two stages of testing verification.

5.2.1 Initial Run (File-based)

The initial execution verified the 12 input files as single consolidated checks.



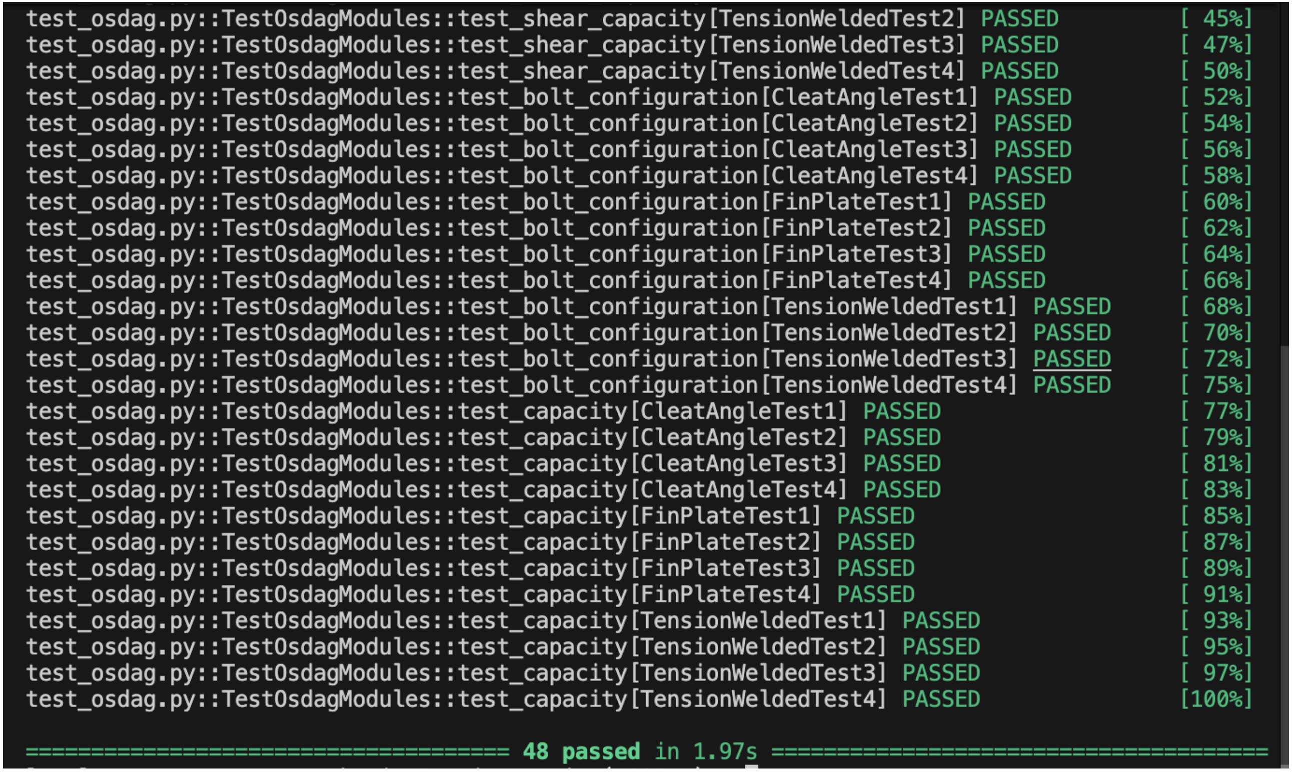
```
===================================== test session starts ======================================
platform darwin -- Python 3.13.0, pytest-9.0.2, pluggy-1.6.0 -- /Users/lovelyarya/Desktop/Unit
Testing(FOSSEE)/venv/bin/python
cachedir: .pytest_cache
rootdir: /Users/lovelyarya/Desktop/Unit Testing(FOSSEE)
collected 12 items

test_osdag.py::test_consolidated_run[CleatAngleTest1] PASSED                              [  8%]
test_osdag.py::test_consolidated_run[CleatAngleTest2] PASSED                              [ 16%]
test_osdag.py::test_consolidated_run[CleatAngleTest3] PASSED                              [ 25%]
test_osdag.py::test_consolidated_run[CleatAngleTest4] PASSED                              [ 33%]
test_osdag.py::test_consolidated_run[FinPlateTest1] PASSED                                [ 41%]
test_osdag.py::test_consolidated_run[FinPlateTest2] PASSED                                [ 50%]
test_osdag.py::test_consolidated_run[FinPlateTest3] PASSED                                [ 58%]
test_osdag.py::test_consolidated_run[FinPlateTest4] PASSED                                [ 66%]
test_osdag.py::test_consolidated_run[TensionWeldedTest1] PASSED                           [ 75%]
test_osdag.py::test_consolidated_run[TensionWeldedTest2] PASSED                           [ 83%]
test_osdag.py::test_consolidated_run[TensionWeldedTest3] PASSED                           [ 91%]
test_osdag.py::test_consolidated_run[TensionWeldedTest4] PASSED                           [100%]

===================================== 12 passed in 2.11s =======================================
```

5.2.2 Detailed Run (Granular)

The granularity was increased to provide detailed per-property verification for each file.



```
test_osdag.py::TestOsdagModules::test_shear_capacity[TensionWeldedTest2] PASSED           [ 45%]
test_osdag.py::TestOsdagModules::test_shear_capacity[TensionWeldedTest3] PASSED           [ 47%]
test_osdag.py::TestOsdagModules::test_shear_capacity[TensionWeldedTest4] PASSED           [ 50%]
test_osdag.py::TestOsdagModules::test_bolt_configuration[CleatAngleTest1] PASSED          [ 52%]
test_osdag.py::TestOsdagModules::test_bolt_configuration[CleatAngleTest2] PASSED          [ 54%]
test_osdag.py::TestOsdagModules::test_bolt_configuration[CleatAngleTest3] PASSED          [ 56%]
test_osdag.py::TestOsdagModules::test_bolt_configuration[CleatAngleTest4] PASSED          [ 58%]
test_osdag.py::TestOsdagModules::test_bolt_configuration[FinPlateTest1] PASSED            [ 60%]
test_osdag.py::TestOsdagModules::test_bolt_configuration[FinPlateTest2] PASSED            [ 62%]
test_osdag.py::TestOsdagModules::test_bolt_configuration[FinPlateTest3] PASSED            [ 64%]
test_osdag.py::TestOsdagModules::test_bolt_configuration[FinPlateTest4] PASSED            [ 66%]
test_osdag.py::TestOsdagModules::test_bolt_configuration[TensionWeldedTest1] PASSED       [ 68%]
test_osdag.py::TestOsdagModules::test_bolt_configuration[TensionWeldedTest2] PASSED       [ 70%]
test_osdag.py::TestOsdagModules::test_bolt_configuration[TensionWeldedTest3] PASSED       [ 72%]
test_osdag.py::TestOsdagModules::test_bolt_configuration[TensionWeldedTest4] PASSED       [ 75%]
test_osdag.py::TestOsdagModules::test_capacity[CleatAngleTest1] PASSED                    [ 77%]
test_osdag.py::TestOsdagModules::test_capacity[CleatAngleTest2] PASSED                    [ 79%]
test_osdag.py::TestOsdagModules::test_capacity[CleatAngleTest3] PASSED                    [ 81%]
test_osdag.py::TestOsdagModules::test_capacity[CleatAngleTest4] PASSED                    [ 83%]
test_osdag.py::TestOsdagModules::test_capacity[FinPlateTest1] PASSED                      [ 85%]
test_osdag.py::TestOsdagModules::test_capacity[FinPlateTest2] PASSED                      [ 87%]
test_osdag.py::TestOsdagModules::test_capacity[FinPlateTest3] PASSED                      [ 89%]
test_osdag.py::TestOsdagModules::test_capacity[FinPlateTest4] PASSED                      [ 91%]
test_osdag.py::TestOsdagModules::test_capacity[TensionWeldedTest1] PASSED                 [ 93%]
test_osdag.py::TestOsdagModules::test_capacity[TensionWeldedTest2] PASSED                 [ 95%]
test_osdag.py::TestOsdagModules::test_capacity[TensionWeldedTest3] PASSED                 [ 97%]
test_osdag.py::TestOsdagModules::test_capacity[TensionWeldedTest4] PASSED                 [100%]

===================================== 48 passed in 1.97s =======================================
```

(6) Conclusion

The unit testing suite has successfully verified the core functionality of the selected Osdag modules. All 12 test cases matched the expected values defined in the CSV files within the specified tolerances. The `KEY_MAPPING` strategy proved effective in resolving inconsistent naming conventions between the input data and the application output.

Furthermore, this testing framework establishes a robust foundation for:

- **Regression Testing**: Ensuring that future code changes do not break existing functionality.
- **Scalability**: New modules and test cases can be easily added by extending the CSV configuration files.
- **Granular Diagnostics**: The detailed run mode allows developers to pinpoint the exact design property causing a failure, facilitating rapid debugging.

Overall, the implementation significantly enhances the reliability and maintainability of the Osdag software, providing confidence in the accuracy of its structural design calculations.