

pyLife@PragTic

Johannes Mueller, Daniel Kreuter

14-15th of December 2023



pyLife@PragTic Agenda

Day 1

1

About us (09-09:45)

About us, BOSCH, BOSCH Research, open source

2

About pyLife (10-11)

history, idea, content, usage, contribution

3

Deep dives (11-4:30)

damage calculation, Woehler evaluation, broadcaster, data model, FEM

5:00-5:45

Bas Aberkrom: Battelle structural stress method for fatigue analysis of weldments

Day 2

4

Hands on (9-12:15)

pyLife in your framework, write your own application, develop your own module

5

What's next (1:15-2:30)

further developments, our pipeline, issues, requests, collaboration

6

Wrap up (3-4)



01

About us

Bosch Research

About us



Daniel Kreuter

AI and Data based Engineering Methods

Senior Research Expert

Background:

M.E. studies and PhD @ University of Technology Dresden (Institute for Solid Mechanics)

R&D @ Tenneco Clean Air, Edenkoben

Research engineer for reliability and durability @ Corporate Research, Bosch, Renningen

danielchristopher.kreuter@de.bosch.com

Material modeling and AI in manufacturing

Johannes Mueller

Research Software Engineer



Background:

PhD in Material Sciences (Saarbrücken, Nancy, Luleå, Erlangen)

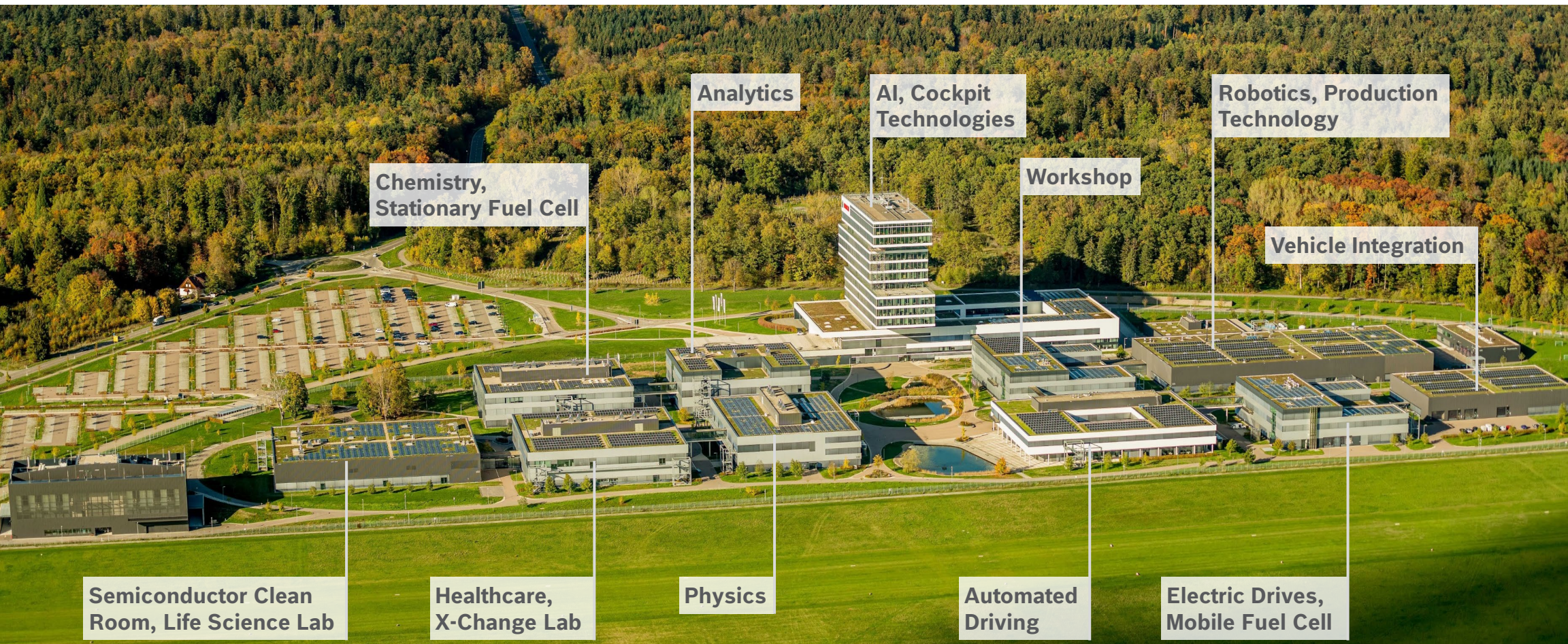
Research Software Engineer @ Corporate Research, Bosch, Renningen

johannes.mueller4@de.bosch.com

Bosch Research



Renningen Research Campus Facilities



Chemistry,
Stationary Fuel Cell

Analytics

AI, Cockpit
Technologies

Workshop

Robotics, Production
Technology

Vehicle Integration

Semiconductor Clean
Room, Life Science Lab

Healthcare,
X-Change Lab

Physics

Automated
Driving

Electric Drives,
Mobile Fuel Cell

Where we want to go

Our research and development

In 2022



7.2

billion euros
R&D expenditure



8.2%

R&D intensity



85,500

associates work
in R&D, including

44,000
software developers




136

R&D locations
worldwide









Bosch Research: Scientific Environment (selection)

R&D collaborations – connected to the best in the world



-  Carnegie Mellon University
-  University of California
-  IBM
-  Georgia Tech



-  University of Freiburg
-  IoC Lab University of Tübingen
-  University of Nottingham
-  University of Stuttgart
-  Delta Lab University (CWI) Amsterdam
-  Technical University of Wien
-  University of Cambridge
-  RWTH Aachen



-  Tsinghua University
-  Fudan University
-  University of Technology Sydney

Bosch Research

Thanks for your attention. Still curious?
Check us out online and visit our website
and LinkedIn account.



Website



LinkedIn

Who we are

Our guiding principle: Invented for life



“Improvements in the world of technology and business should always also be beneficial for mankind.”

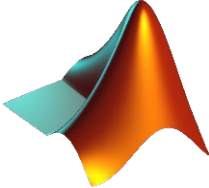
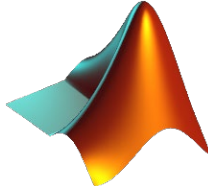
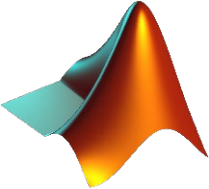
ROBERT BOSCH

02

About pyLife

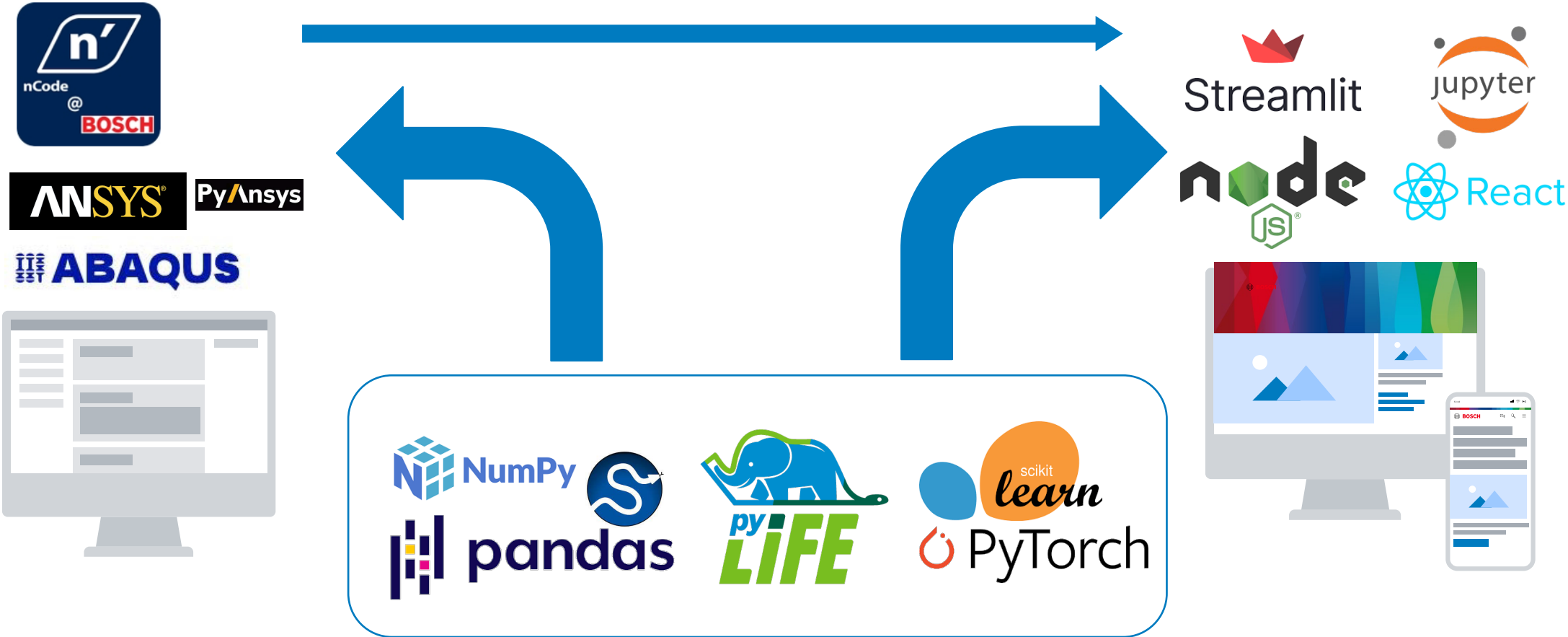
pyLife – a general library for fatigue and reliability

The world before pyLife (at least @Bosch)



pyLife – a general library for fatigue and reliability

The world with pyLife



pyLife – a general library for fatigue and reliability

Side note



<https://www.welt.de/kultur/literarischewelt/plus/74679846/Verschwundene-Alltagsdinge-Was-hat-sich-denn-da-in-Luft-aufgeloest.html>

pyLife – a general library for fatigue and reliability

Why OSS



https://github.com/pytorch/pytorch/blob/master/docs/source/_static/img/pytorch-logo-dark.png
<https://github.com/valohai/ml-logos/blob/master/tensorflow-text.svg>
<https://github.com/pandas-dev/pandas/blob/master/web/pandas/static/img/pandas.svg>
https://en.wikipedia.org/wiki/MATLAB#/media/File:Matlab_Logo.png
https://de.wikipedia.org/wiki/NumPy#/media/Datei:NumPy_logo_2020.svg

pyLife – a general library for fatigue and reliability

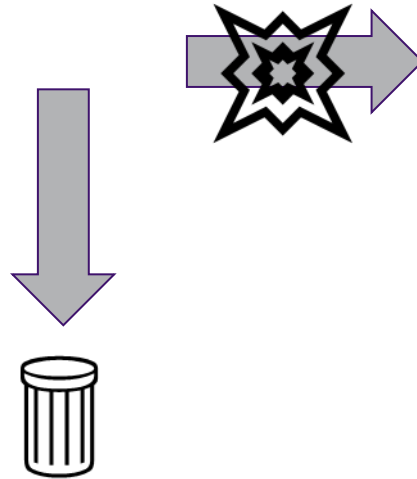
Two ways of programming

Explorative programming

- Learn something
- Try something out
- See if your idea is feasible

- No requirements analysis
- No unit tests
- No documentation

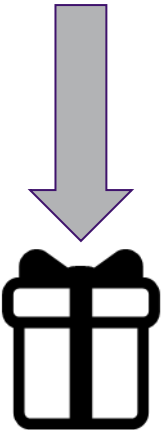
Must not accumulate for more than one week!



Software craftsmanship

- Deliver something to users
- User oriented
- Reader oriented

- Clear picture of the requirements
- Comprehensive unit tests
- Documentation
- CI/CD



pyLife – a general library for fatigue and reliability

pyLife

boschresearch/pylife: a general ... testsuite · Workflow runs · bosch ...

https://github.com/boschresearch/pylife

Fork 22 Starred 100

pyLife – a general library for fatigue and reliability

launch binder docs passing pypi v2.0.4 python 3 testsuite

pyLife is an Open Source Python library for state of the art algorithms used in lifetime assessment of mechanical components subject to fatigue load.

Purpose of the project

This library was originally compiled at [Bosch Research](#) to collect algorithms needed by different in house software projects, that deal with lifetime prediction and material fatigue on a component level. In order to further extent and scrutinize it we decided to release it as Open Source. Read [this article](#) about pyLife's origin.

So we are welcoming collaboration not only from science and education but also from other commercial companies dealing with the topic. We commend this library to university teachers to use it for education purposes.

The company [Viktor](#) has set up a [web application for Wöhler test analysis](#) based on pyLife code.

Status

pyLife-2.0.3 has been released. That means that for the time being we hope that we will not introduce *breaking* changes. That does not mean that the release is stable finished and perfect. We will do small improvements, especially with respect to documentation in the upcoming months and release them as 2.0.x releases. Once we have noticeable feature additions we will come up with a 2.x.0 release. No ETA about that.

Contents

About

a general library for fatigue and reliability

[pylife.readthedocs.io](#)

education engineering material reliability mechanical-engineering lifetime material-science fatigue

Readme

Apache-2.0 license

Activity

100 stars

14 watching

22 forks

Report repository

Releases 36

2.0.4 Latest 3 weeks ago

+ 35 releases



pyLife – a general library for fatigue and reliability

Content



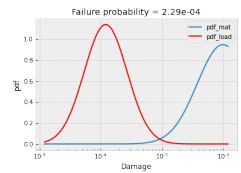
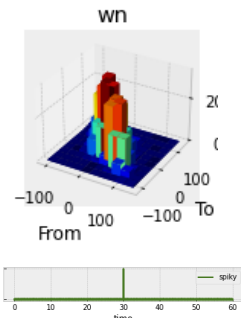
General

- pyLife core



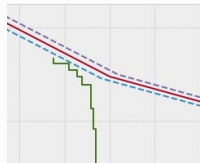
Stress

- The pyLife stress subpackage
- The `equistress` module
 - Equivalent Stresses
- The `rainflow` module
 - Overview over pyLife's rainflow counting module
 - API Documentation
- The `LoadCollective` class
- The `LoadHistogram` class
- The `stressignal` module
- The `timesignal` module
- The `frequencysignal` module



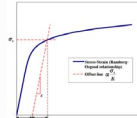
Strength

- The `Fatigue` class
- The `meanstress` module
 - Meanstress routines
- The `FailureProbability` class
- The `miner` module
 - Implementation of the miner rule for fatigue analysis



Materiallaws

- The `hookeslaw` module
- The `RambergOsgood` class
- The `woehlerCurve` class
- The `true_stress_strain` module

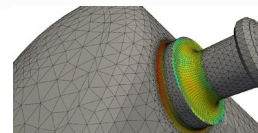


Materialdata

- The `woehler` module
 - Module description
 - Fatigue data handling
 - Analyzers
 - Helpers

Mesh utilities

- The `mesh` module
 - Overview
 - The signal classes
- The `HotSpot` class
- The `Gradient` class
- The `Meshmapper` class



Interfaces:



New:

FKM
linear/nonlinear

pyLife – a general library for fatigue and reliability Development



The pyLife deal

- What we want from you
 - Your ideas, your feedback
 - Code contributions
- What we will do with it
 - Integrate it into pyLife to make it better for everyone
- What we give you in return
 - Help in writing good code
 - Code review
 - Maintenance

✓ Test driven development

✓ Continuous integration

The screenshot displays a GitHub repository interface. At the top, there are two code snippets for pytest parametrized test functions. Below the code, a green checkmark indicates a successful test run for 'Fkm nonlinear #337'. The main part of the screenshot shows a CI/CD pipeline summary for a build on 'ubuntu-20.04, 3.9', which succeeded 3 weeks ago. The pipeline steps include 'Set up job', 'Run actions/checkout@v3', 'Set up Python 3.9', and 'Install dependencies'. The 'Unit tests' step is expanded, showing a terminal output for 'Run pytest' with a progress bar indicating the test session progress: 'collected 1435 items / 9 deselected / 1426 selected'. The progress bar shows 1% for 'tests/core/test_broadcaster.py', 3% for 'tests/core/test_signal.py', and 5% for 'tests/materialdata/woehler/test_analyzer.py'.

pyLife – a general library for fatigue and reliability

Examples



jupyter
nbviewer

pyLife / demos

pyLife – a general library for fatigue and reliability

pyLife is an Open Source Python library for state-of-the-art algorithms used in lifetime assessment of mechanical components subject to fatigue load. For further information on pyLife, check out the [documentation](#) and the [GitHub repository](#).

Disclaimer: The demos here are – well – demos. I.e. they are not meant to be used for real calculation. No guarantees whatsoever.

Tutorial notebooks

These notebooks describe some concepts of pyLife in a practical way.

- Wöhler curve**
The data structure to model Wöhler curves for e.g. damage calculations.
[Wöhler curve](#)
- Stress strength**
The concept of stress and strength that allows us to calculate damage sums and failure probabilities.
[Stress strength](#)
- Load Collective**
How to model load collectives in pyLife.
[Load Collective](#)

Demos

This page features a couple of demos on pyLife.

- Wöhler analysis**
Usual SN-curve data can be analyzed to get the SN-curve parameters.
[Wöhler analysis](#)
- Hotspot detection**
In a connected FEM mesh, hotspots of a scalar value like damage sum, or von Mises stress can be detected and classified.
[Hotspot detection](#)
- Gradients in a mesh**
In a connected mesh, gradients of scalar values can be calculated.
[Stress gradient](#)
- Lifetime calculation**
The damage sum and failure probability can be calculated using a nominal and local stress reliability approach. The two notebooks depend on each other. The *Time Series Handling* notebook needs to be executed first.
 - [Time Series Handling](#)
 - [Lifetime calculation](#)
- Local stress approach**
Damage calculation by local stress approach.

Maximum Likelihood Full

The Maximum Likelihood Full method just takes the elementary result as starting values but fits all the parameters.

```
[14]: maxlike_full_result = woehler.MaxLikeFull(fatigue_data).analyze()
maxlike_full_result
```

```
[14]: k_1      8.626164e+00
      ND     1.326971e+06
      SD     2.952540e+02
      TN     9.862007e+00
      TS     1.106611e+00
      failure_probability  5.000000e-01
      dtype: float64
```

```
[15]: wc = maxlike_full_result.woehler
      cycles = np.logspace(np.log10(df.cycles.min()), np.log10(df.cycles.max()), 100)
      maxlike_full_fig = copy.deepcopy(fig)
      maxlike_full_fig.add_scatter(x=cycles, y=wc.basquin_load(cycles), mode='lines', name='MaxLike')
      maxlike_full_fig.add_scatter(x=cycles, y=wc.basquin_load(cycles, failure_probability=0.1), mode='lines', name='MaxLike 10%')
      maxlike_full_fig.add_scatter(x=cycles, y=wc.basquin_load(cycles, failure_probability=0.9), mode='lines', name='MaxLike 90%')
      maxlike_full_fig
```

pyLife – a general library for fatigue and reliability Development



<https://github.com/boschresearch/pylife>

boschresearch / pylife Public

Notifications Fork 22 Star 100

<> Code Issues 5 Pull requests 1 Discussions Actions Projects Security Insights

develop 3 branches 39 tags Go to file Code

About
a general library for fatigue and reliability
pylife.readthedocs.io
education engineering material reliability mechanical-engineering lifetime material-science fatigue

Readme Apache-2.0 license Activity 100 stars 14 watching 22 forks Report repository

	johannes-mueller Switch from panel to trame (#30) ...	✓ a9fe148 3 weeks ago	🕒 1,111 commits
📁 .githubhooks	Use core.hooksPath to deploy prepare-commit-msg		2 years ago
📁 .github	Update CI actions and don't act twice on pushes to PRs		last month
📁 batch_scripts	Make pymc based tests work in CIs		10 months ago
📁 binder	Switch from panel to trame (#30)		3 weeks ago
📁 demos	Switch from panel to trame (#30)		3 weeks ago
📁 docs	Fix reference in cookbook		last month
📁 src/pylife	Fix indexing bugs in gradient.py and hotspot.py (#24)		last month
📁 tests	Introduce two notched beam for hotspot detection demo (#...		last month
📁 tools	Make odbclient pandas-2.x compliant		last month

pyLife – a general library for fatigue and reliability

Links



Code Repository



Documentation



Blog article

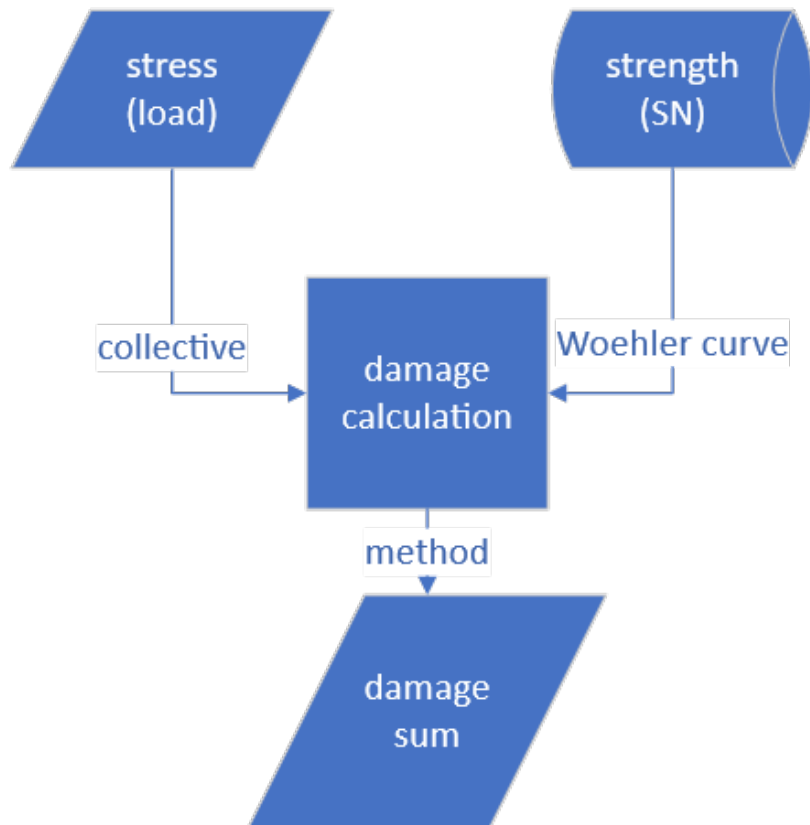


03

Deep dives

pyLife – Workshop

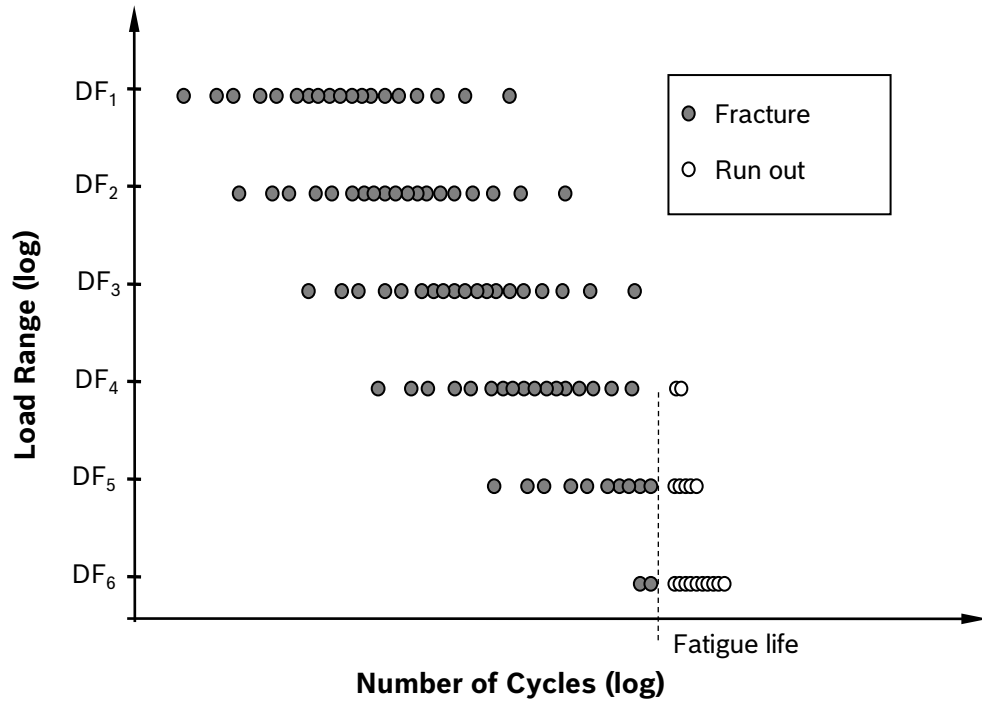
Example 1: damage calculation



- Jupyter notebook vs streamlit app
- Methods:
 - Miner original
 - Miner elementary
 - Miner Haibach

pyLife – Workshop

Example 2: SN data derivation



- Jupyter notebook vs streamlit app
- Methods:
 - Probit
 - Maximum Likelihood infinite
 - Maximum Likelihood full
 - Maximum Likelihood full with fixed parameters

04

Hands on

pyLife – Workshop

Inquiry of participants experience



Who of you ...

- ... can program Python
 - knows about numpy, pandas, scipy, ...
 - knows what a Jupyter notebook is
 - knows about anaconda/miniconda venvs, ...
 - knows about python packaging (setuptools, poetry, pdm, pip, pypi, ...)
 - knows about OOP
 - knows what unit tests are (is doing TDD)
- ... has general experience in Software Craftsmanship
 - knows git and has an installation of it available
 - knows what commits, branches, pull-requests, merges, rebases are
 - uses these as daily routine
 - is familiar with GitHub
 - knows what CI/CD means

05

Next steps

pyLife – Workshop

Outlook – next steps

Short term

- Finalize FKM non-linear
 - Stabilize API
 - Optimizations

Mid term

- Further FKM additions
- Cythonizing rainflow counters

Long term

- Integrate dask for parallelization
- Introduce type hints at strategic places

Your ideas?