# A CI/CB SETUP FOR A QUANTUM TRANSPORT CODE

June 13, 2025 | Christoph Conrads | Jülich Supercomputing Centre

# Context

libNEGF is the quantum transport code of the simulation software DFTB+. It is currently undergoing significant improvements to its physics models and its scalability to allow simulation of real-world solar cells on most available HPC systems.

# RSE Status

## Project start

- Usage of a revision control system
- End-to-end (whole molecule) tests present
- Continuous integration (CI) set up
- JUBE setup created
- A small number of integration tests
- Set of benchmark problems under construction

# RSE Status

## Current status

- Usage of a revision control system
- End-to-end (whole molecule) tests present
- Continuous integration (CI) set up
- JUBE setup created
- A small number of integration tests
- Set of benchmark problems under construction

# CI Setup

libNEGF is built and run

- on Debian 12, Rocky Linux 9
- on CPU, GPU
- with OpenMPI, MPICH

in Docker containers on a dedicated PC.

# JUBE Setup

The JUBE setup was hacked to enable greater flexibility:

```
env LIBNEGF_INPUT_SIZE=3x3 \
  LIBNEGF_NUM_NODES=1,2,4,8 \
  LIBNEGF_TIMELIMIT_MIN=10 \
  jube run \
    --outpath="$SCRATCH/$USER/jube-2025-05/" \
    --tag jedi gpu \
    -- JUBE/benchmark-libnegf.xml
```

# Context

## Performance Aspects

- libNEGF is comparatively small at  10k lines of code.
- Performance is well understood in the RAM model.
- libNEGF experienced unexplained performance fluctuations when deployed.

# Continuous Benchmarking

## Goals

- Diagnose unexpected HPC system behavior instead of exploring performance
- Rely on system-provided metrics
- How to sample at regular intervals?
- Can we compute robust statistics for pipeline pass/fail?

# The End

Thank you for your attention. Questions?

# References


[1] Christoph Alt et al. “A continuous benchmarking infrastructure for high-performance computing applications.” In: International Journal of Parallel, Emergent and Distributed Systems 39.4 (2024), pp. 501–523. DOI: 10.1080/17445760.2024.2360190.

[2] Hartwig Anzt et al. “Towards Continuous Benchmarking. An Automated Performance Evaluation Framework for High Performance Software.” In: Proceedings of the Platform for Advanced Scientific Computing Conference. PASC ’19. Zürich, Switzerland: Association for Computing Machinery, 2019. DOI: 10.1145/3324989.3325719.

[3] R. Speck D. Brömmel J. Fritz. Integrated Continuous Benchmarking. Connecting gitlab, Jacamar, and JUBE. 2024. DOI: 10.34734/FZJ-2024-01995.