Power Grid Model
A High-Performance Distribution Grid Calculation Library

Nitish Bharambe
Scientific Software Engineer, Alliander N.V
Summary

- Power Grid Model: an open-source project for distribution power system calculation
  - https://github.com/PowerGridModel

- In this presentation
  - Why a new project?
  - What is Power Grid Model?
  - How does it perform?
  - Deployment inside Alliander
  - Road to open-source
Traditional workflow for power system analysis

Data files → Commercial software* → Built-in function* → Results

Modern workflow for power system analysis

Database → Programming/scripting → Cloud deployment

Publish results

input_data = import_data()
model = Model(input_data)
result = model.calculate()
Modern workflow for power system analysis

What makes a good power system calculation model/library?

input_data = import_data()
model = Model(input_data)
result = model.calculate()
Why Power Grid Model (PGM)?

- Unique propositions of Power Grid Model

- Well-defined Software API
- High performance
- Native parallelization
- Cross-platform
- Scalability
Power Grid Model

- Power System Calculation Functionalities
- Symmetric and asymmetric calculation
- Power flow
  - Newton-Raphson
  - Iterative current (equivalent to backwards/forwards for radial network)
  - Linear current (approximation)
  - Linear impedance (approximation)
- State estimation
  - Iterative linear method
- Short circuit calculation
Power Grid Model

• Power System Calculation Functionalities
  • Efficient implementation in C++
    • Native shared-memory multi-threading for parallelization in batch calculations
• API in Python
  - Stable and easy-to-use
  - Well-documented
• Cross-platform
  - Publish binary Python packages in PyPI and conda-forge
    - https://pypi.org/project/power-grid-model/
    - https://anaconda.org/conda-forge/power-grid-model
  - Built for Windows (x64), Linux (x64/arm64), macOS (x64/arm64)
Model Validation

• Validation of the library against reference models with 80+ test cases
  - Hand calculation
  - Vision
  - Gaia
  - PowerFactory
  - PandaPower
• Continuous validation as part of CI pipeline in GitHub Actions
Performance Benchmark

• Compare performance of Power Grid Model, PandaPower, and OpenDSS
  - [https://github.com/PowerGridModel/power-grid-model-benchmark](https://github.com/PowerGridModel/power-grid-model-benchmark)
  - 1000 nodes radial network
  - Time-series symmetric and asymmetric power flow calculation in 1000 steps
  - Testing environment: Intel i7-12700H, 64 GB RAM, single-thread in Linux (WSL)
  - Library version:
    - power-grid-model == 1.4.65
    - Pandapower == 2.12.1
    - dss-python == 0.14.1
Performance Benchmark

Relative performance for symmetric calculation

- PandaPower Newton-Raphson: 1
- PGM Newton-Raphson: 24
- PGM Iterative Current: 46
- PGM Linear Impedance: 65
- PGM Linear Current: 67
Performance Benchmark

OpenDSS Fix point algorithm is comparable to PGM iterative current
Current Deployment

- Data conversions
  - CIM
  - Vision
  - GridCal
  - PandaPower
  - Gaia (pending)
Current Deployment

A fundamental building block for Alliander

- Deployed in 10+ applications inside Alliander
Road to Open Source

- Ways of collaboration and contribution*

* https://github.com/PowerGridModel/.github/blob/main/CONTRIBUTING.md
Road to Open Source

- Current active partner