

Machine learning in OPM Flow: hybrid Newton developments and practical applications

Antoine Lechevallier

Computational Geosciences and Modelling

anle@norceresearch.no

Outline

I. General advances of ML in OPM

II. HyOPML

III. Application example: History Matching

IV. Conclusion and future challenges

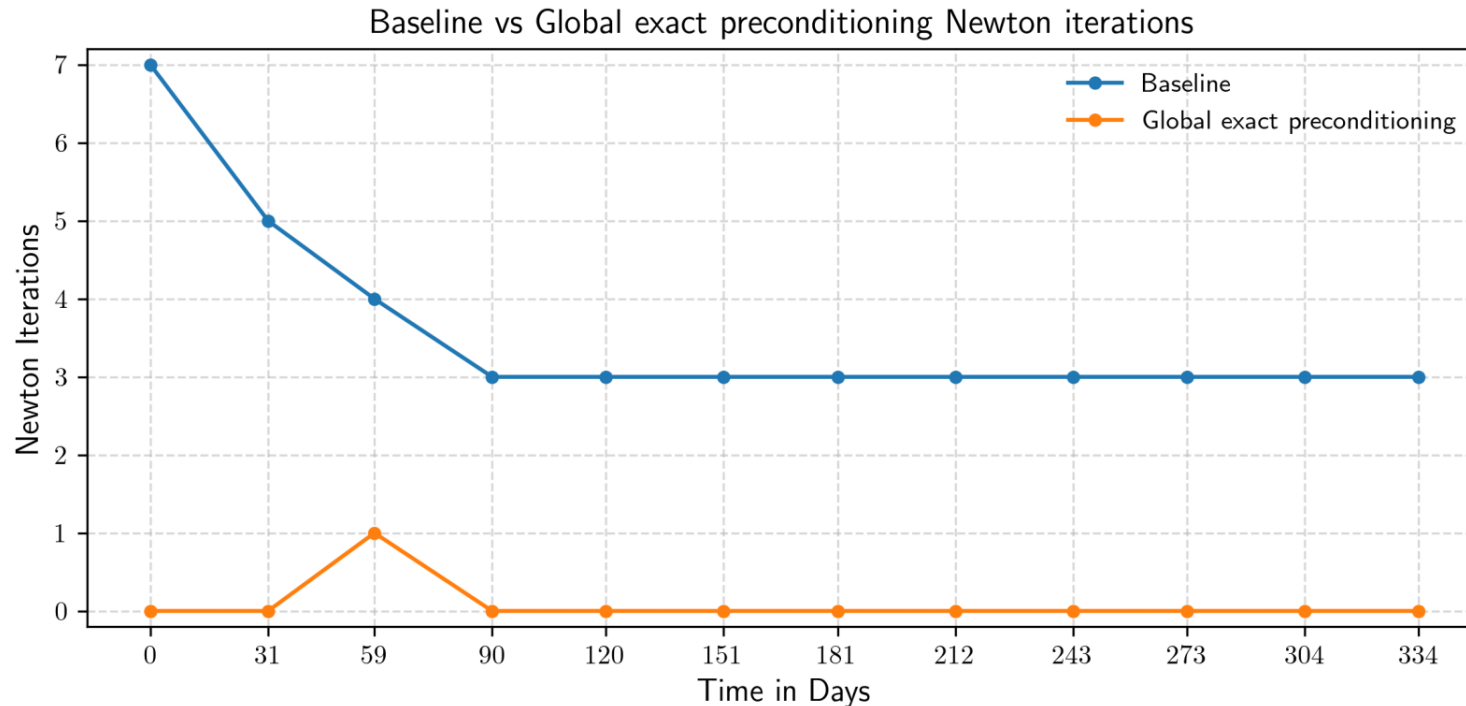
General advances of ML in OPM

Refactoring ML dependencies #4857

- Remove **tensorflow/keras** dependency
- **Numpy** wrapper for neural networks
- User workflow:
 - Train neural network in your favorite framework outside of OPM
 - Convert the neural network to Numpy
 - Export it to OPM compatible using **export_model**
 - Want to train it more? Load it back to Python using **load_model**

Hybrid Newton: description

- Right nonlinear preconditioning strategy
- A neural network updates Newton's method initial guess
- Select cells and variables to update



Hybrid Newton: implementation #6424

- New flags `--use-hybrid-newton=True` & `--hybrid-newton-config-file=file.json`

```
{
  "0": {
    "model_path": "En_ml_models/489_cluster_0.model",
    "cell_indices_file": "En_ml_models/489_cluster_0_active_cells.txt",
    "apply_times": {
      "0": 42249600.0
    },
    "features": {
      "inputs": {
        "RS": {
          "feature_engineering": "log1p",
          "scaling_params": {
            "min": 4.83956423828954,
            "max": 5.327647038489927
          }
        }
      },
      "SOIL": {},
      "SWAT": {},

```

```
    },
    "outputs": {
      "PRESSURE": {
        "feature_engineering": "log10",
        "scaling_params": {
          "min": 2.351185291213962,
          "max": 2.503219904096242
        }
      }
    },

```

```
  },
  "1": {
    "model_path": "En_ml_models/549_cluster_0.model",
    "cell_indices_file": "En_ml_models/549_cluster_0_active_cells.txt",
    "apply_times": {
      "0": 47433600.0
    },
    "features": {
      "inputs": {
        "RS": {

```

Hybrid Newton: implementation

- New files in **opm-simulators/opm/simulators/flow**:
 - ***HybridNewtonConfig.hpp/cpp*** : Parses the input json config file
 - ***HybridNewton.hpp***: Updates initial guess
- Only BlackOil is supported:

```
class FlowProblemBlackoil : public FlowProblem<TypeTag>

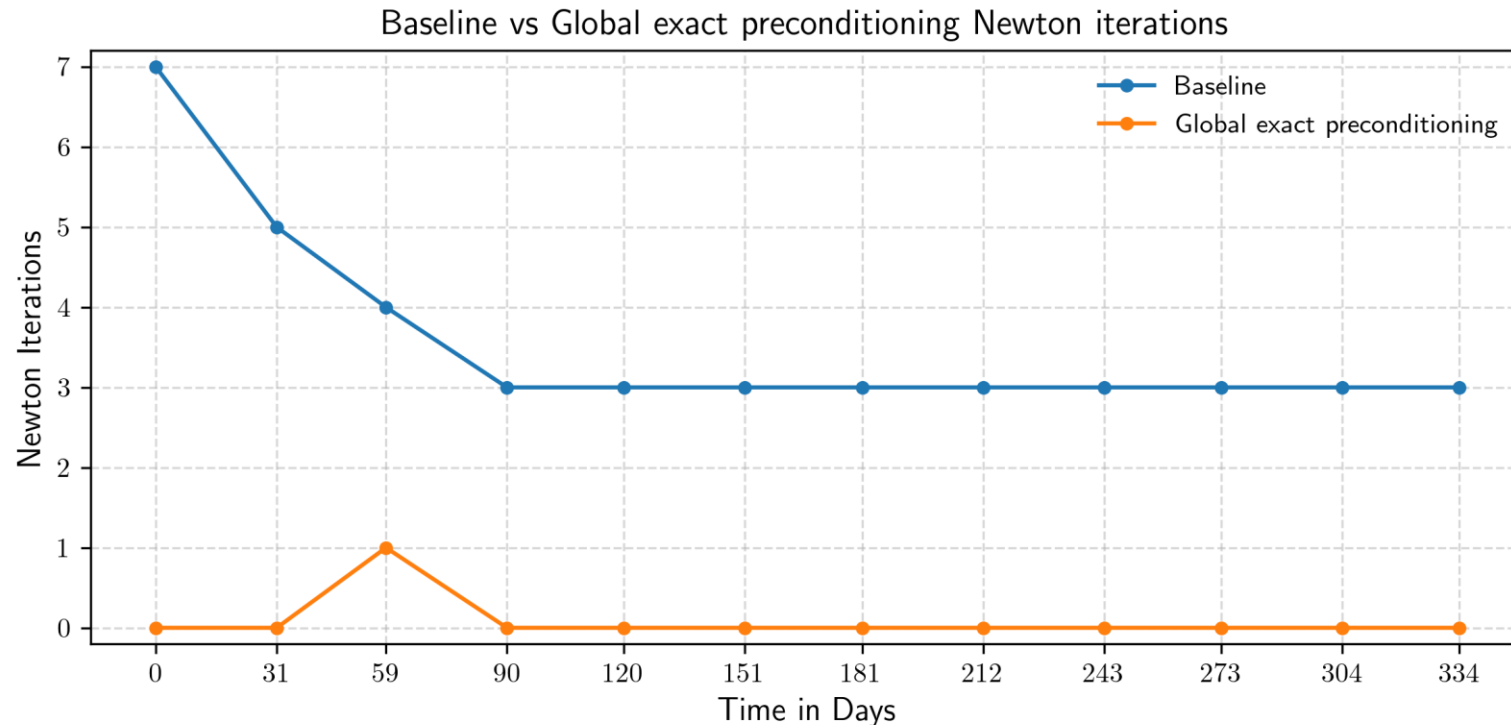
    /*!
     * \brief Called by the simulator before each time integration.
     */
    void beginTimeStep() override
    {
        FlowProblemType::beginTimeStep();
        hybridNewton_.tryApplyHybridNewton();
    }
```

HyOPML (cssr-tools/hyopml)

- Python package enabling integration of neural network-based methods into OPM
- Overall objective is to develop tools, and when they are mature enough, move them to OPM (e.g.: Numpy neural network wrapper)
- Convert model from **Jax/Pytorch/Keras/Tensorflow** to **Numpy**
- **Hybrid Newton** interface for creating **config files**, **extracting data**
- Update it following OPM releases

Global exact hybrid Newton for SPE1

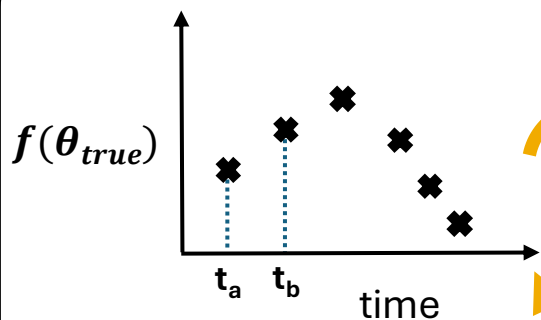
- Setup:
 1. Run SPE1: No relaxation, save all solutions
 2. Create a neural network that predicts exactly the solution for all cells and all steps
 3. Run SPE1 again with `--use-hybrid-newton=true` for all steps



Application example: History matching

Standard History Matching

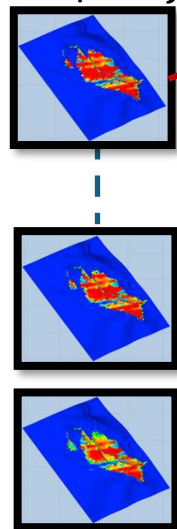
Historical data



Objective:
Find θ_{true} by iterating over a prior $\hat{\theta}$

Ensemble

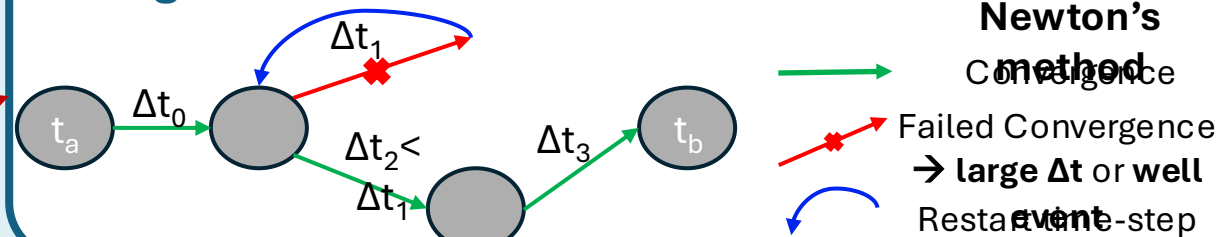
OPM forward simulations
→ Compute $f(\hat{\theta})$



**Predict
Newton initial
guess**

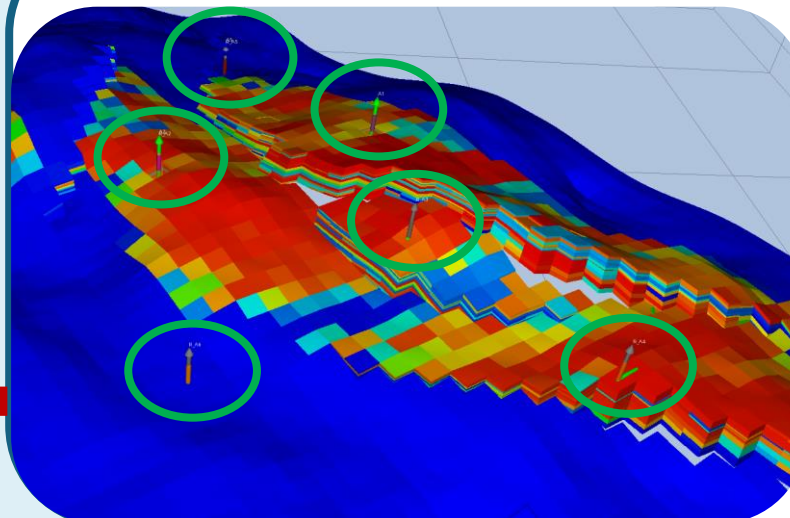
Data-driven Non-linear Preconditioning

Single Simulation



Lots of unused data

Train Neural Networks in Near-well Regions



Benefits:

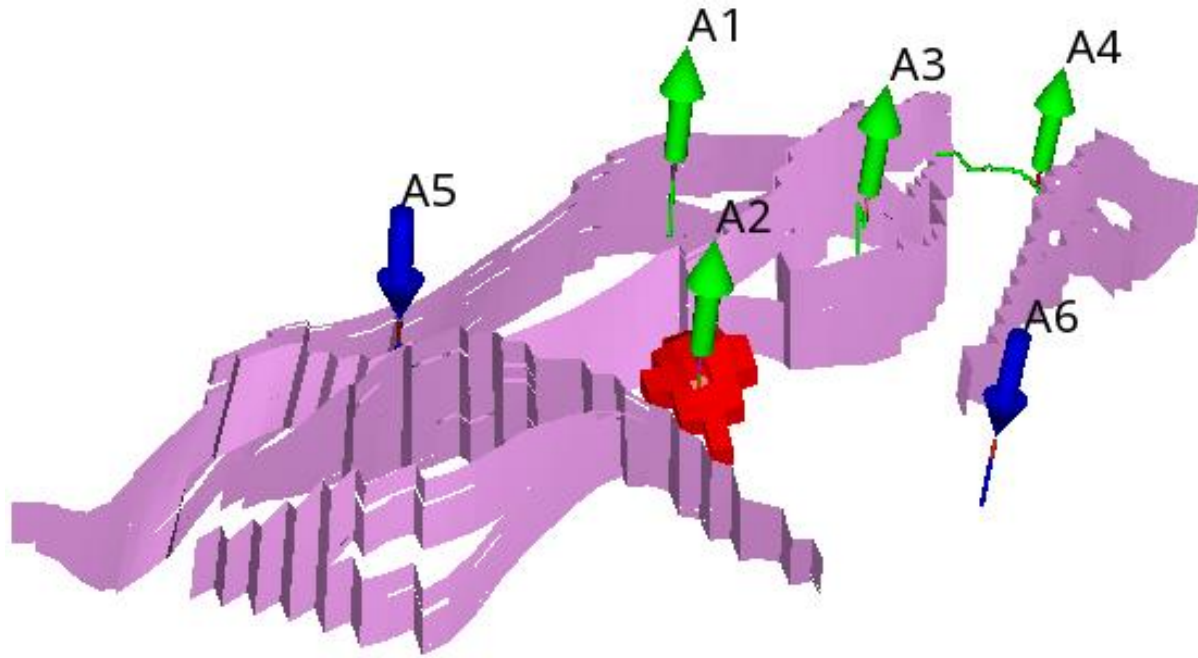
Accelerate →

Avoid ↩

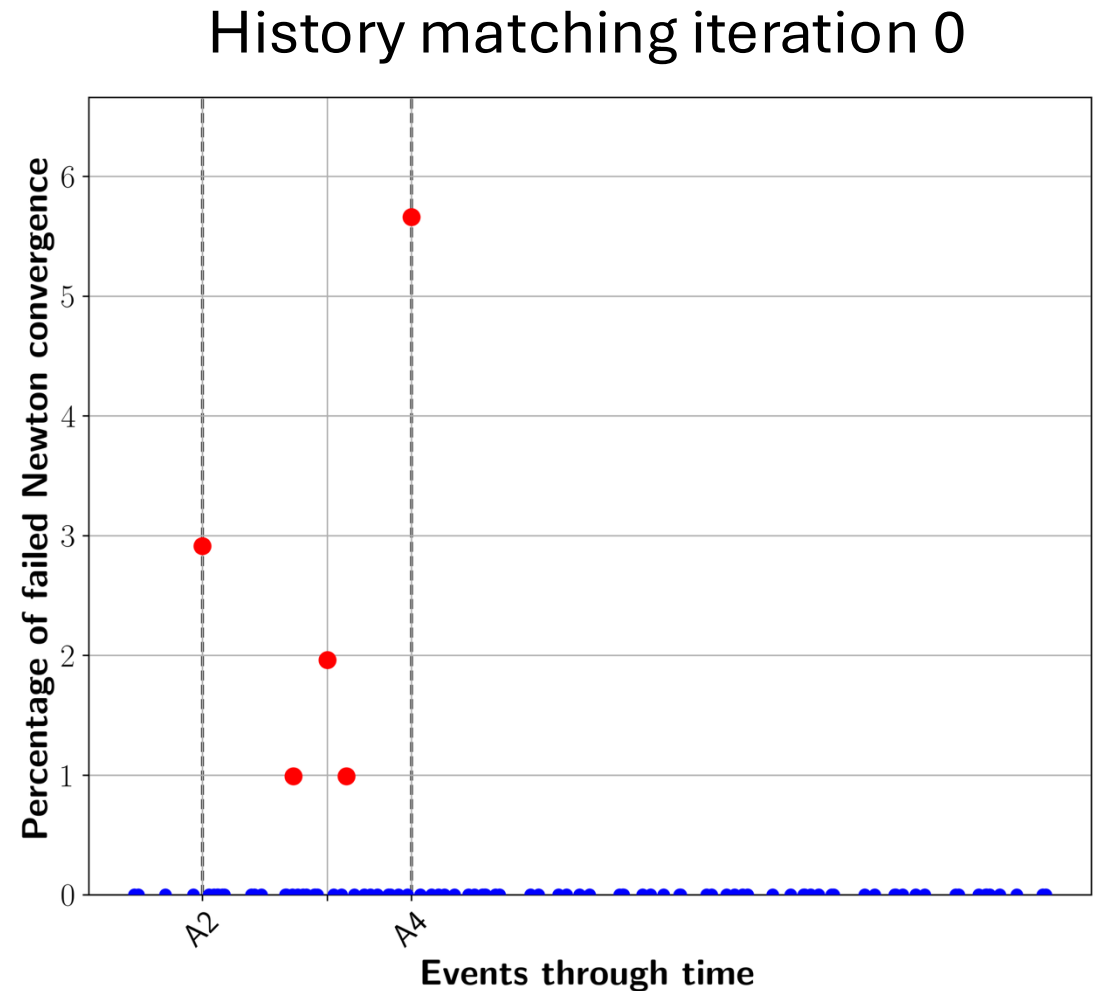
Implementation:

- Learn near-well behavior
- One model per well
- Memory efficient
- Fast parallel training

Application example: some results Drogon

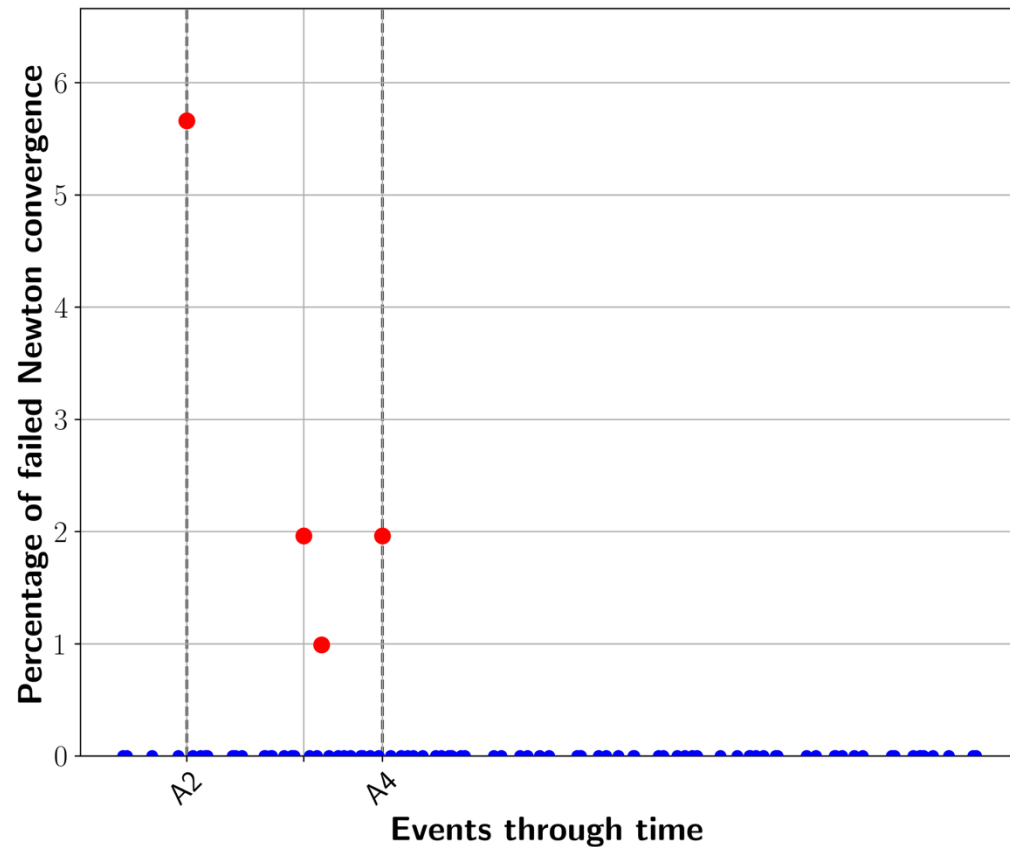


Local domain around A2 well:
BFS algorithm with distance 2

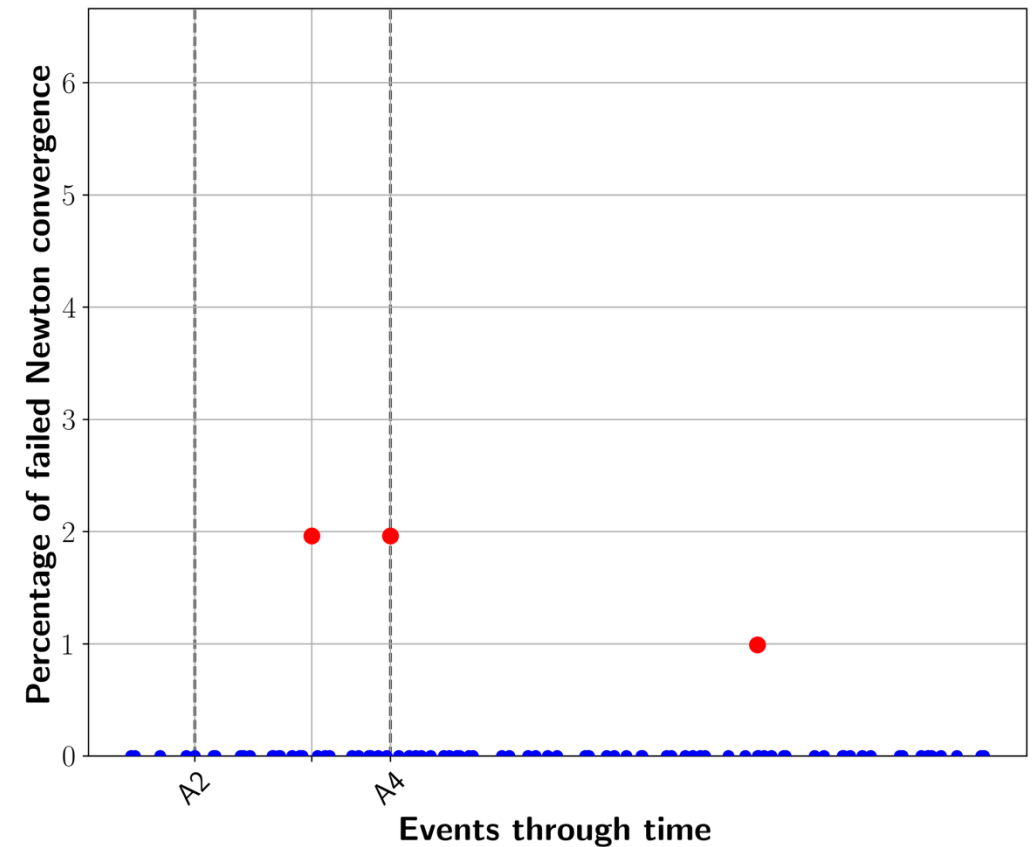


Application example: some results Drogon

History matching iteration 4

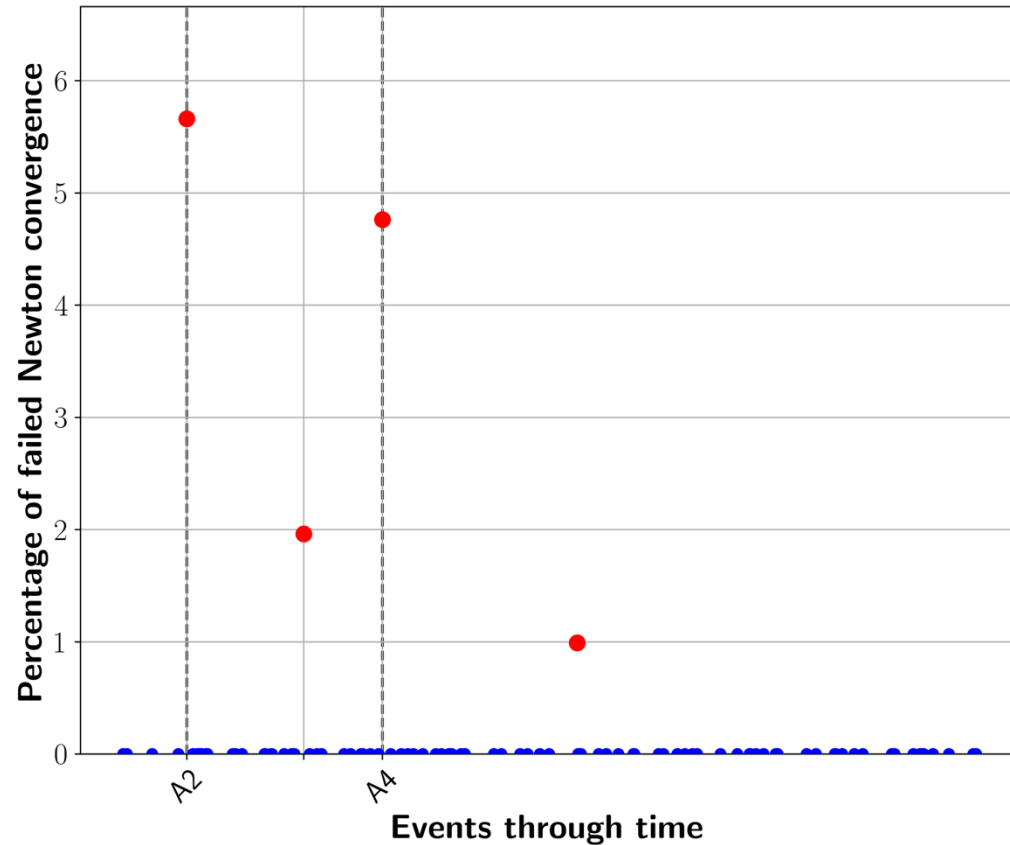


Standard



Hybrid

History matching iteration 3



Conclusion

OPM

- Method Integrated into OPM simulators: ***Hybrid newton flag #6424***
- Accessible through **hyopml**: high-level Python interface

Development of a Data-Driven Nonlinear Solver Preconditioner:

- Trained on data that are already produced by data-intensive workflows
- Automatically detects challenging events in time
- Training completes in just **0.1 second on CPU**

Results

- Tested on **Drogon** and **Norne** cases:
 - Drogon (1 event): 0% speed-up
 - Norne:
 - 1 event: **2.5% speed up**
 - 2 events: **3% speed up**

Futures challenges

OPM:

- **The potential future success of ML module in OPM will require some more mature methods**

Hybrid Newton:

- Automatic detection of nonlinear issues in space and time
- Link neural network performances to time stepping
- Implement for CO₂ storage

Neural networks

- Use better optimizer
- Add double precision