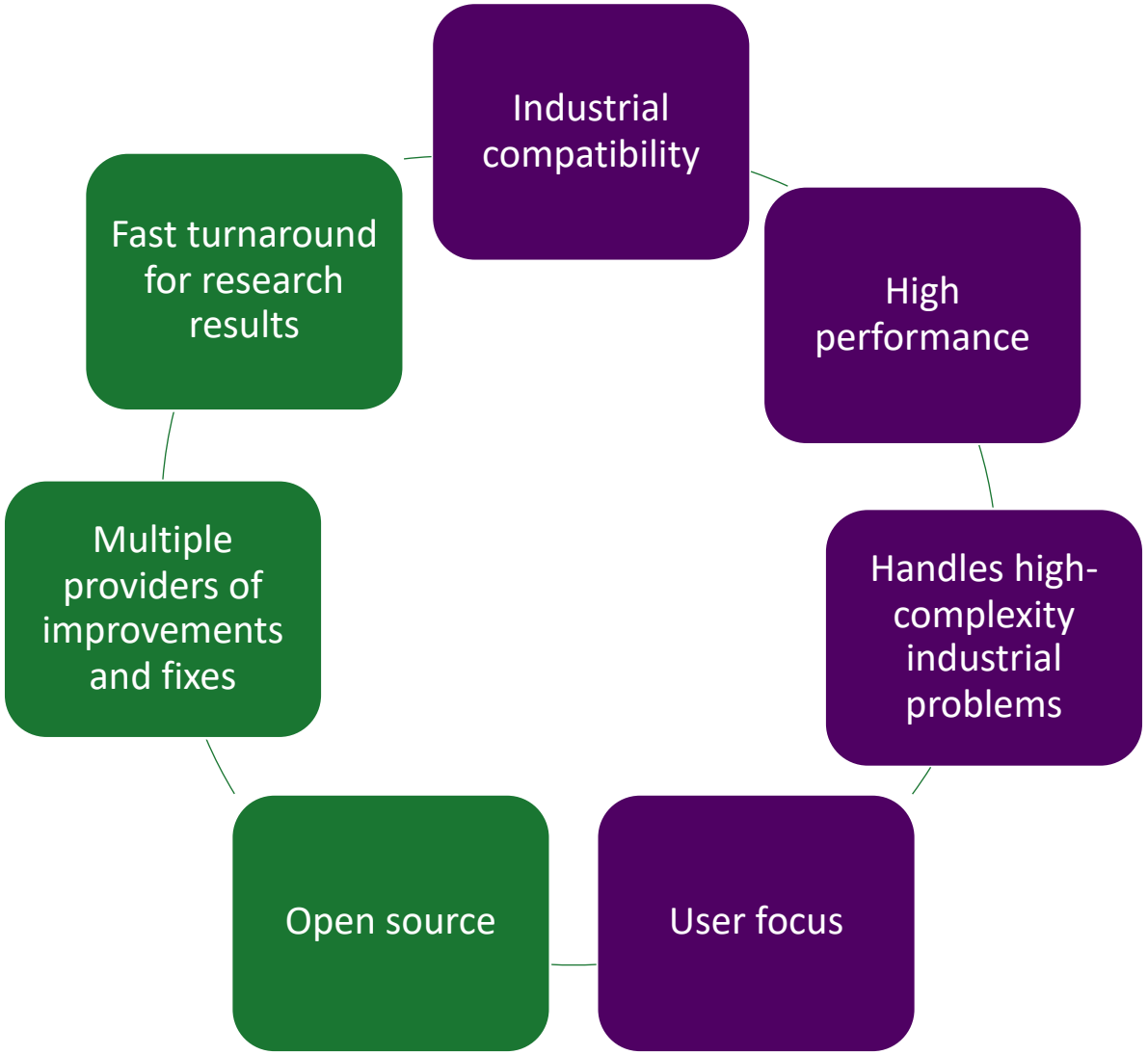
OPM Flow, status and ongoing work

OPM Summit, Bergen, May 26, 2025

Atgeirr Flø Rasmussen, SINTEF

Why is OPM Flow a success?



= Unusual for research software

= Unusual for commercial software

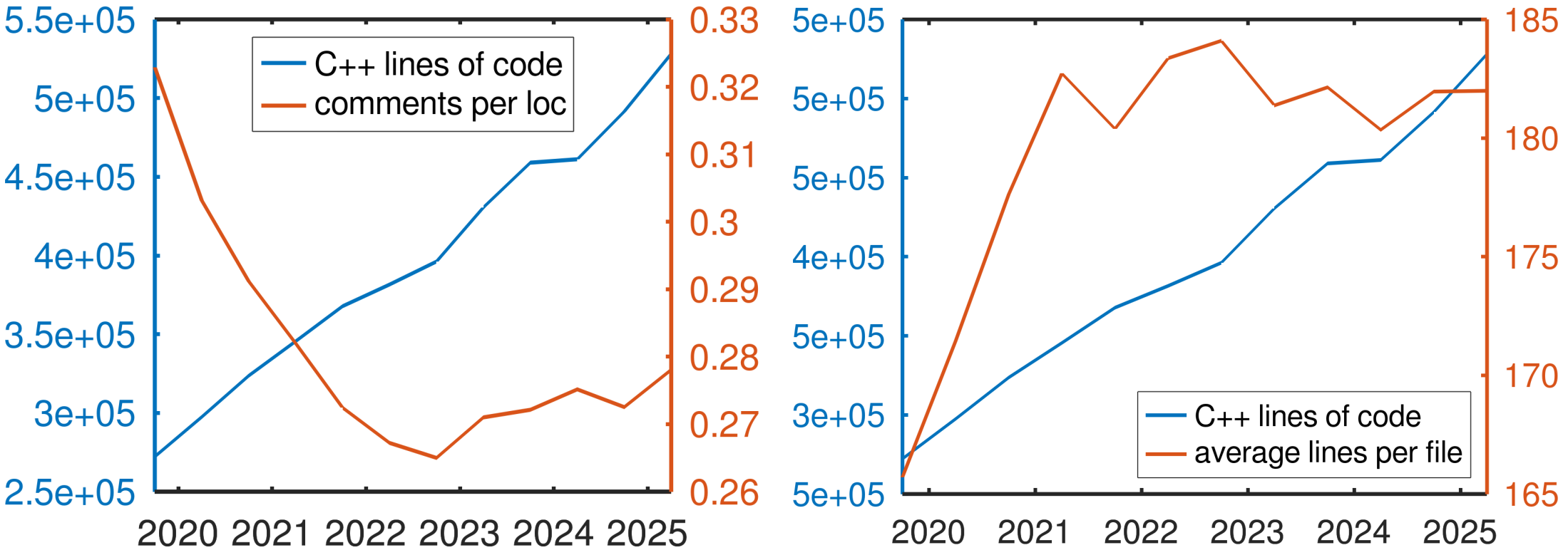
A few highlights and numbers

How large is OPM?

Language	Files	Blanks	Comments	Lines of code	Comment/LoC
C++	2902	135842	146860	**528142**	0.2781*
Python	72	1529	1519	**6566**	0.2313
CMake	117	1502	2613	**13233**	0.1975
JSON (keywords)	1192	15	0	**32973**	0

* Includes license block at top of files

C++ filesizes and comments over time



Probably too imprecise measures to make dramatic claims?

Commits and PRs

Repo	Commits	PRs	Janitorial PRs...
Opm-common	1661	506	39
Opm-grid	430	109	8
Opm-simulators	3469	794	61

Data from April 9, 2024 – May 22, 2025

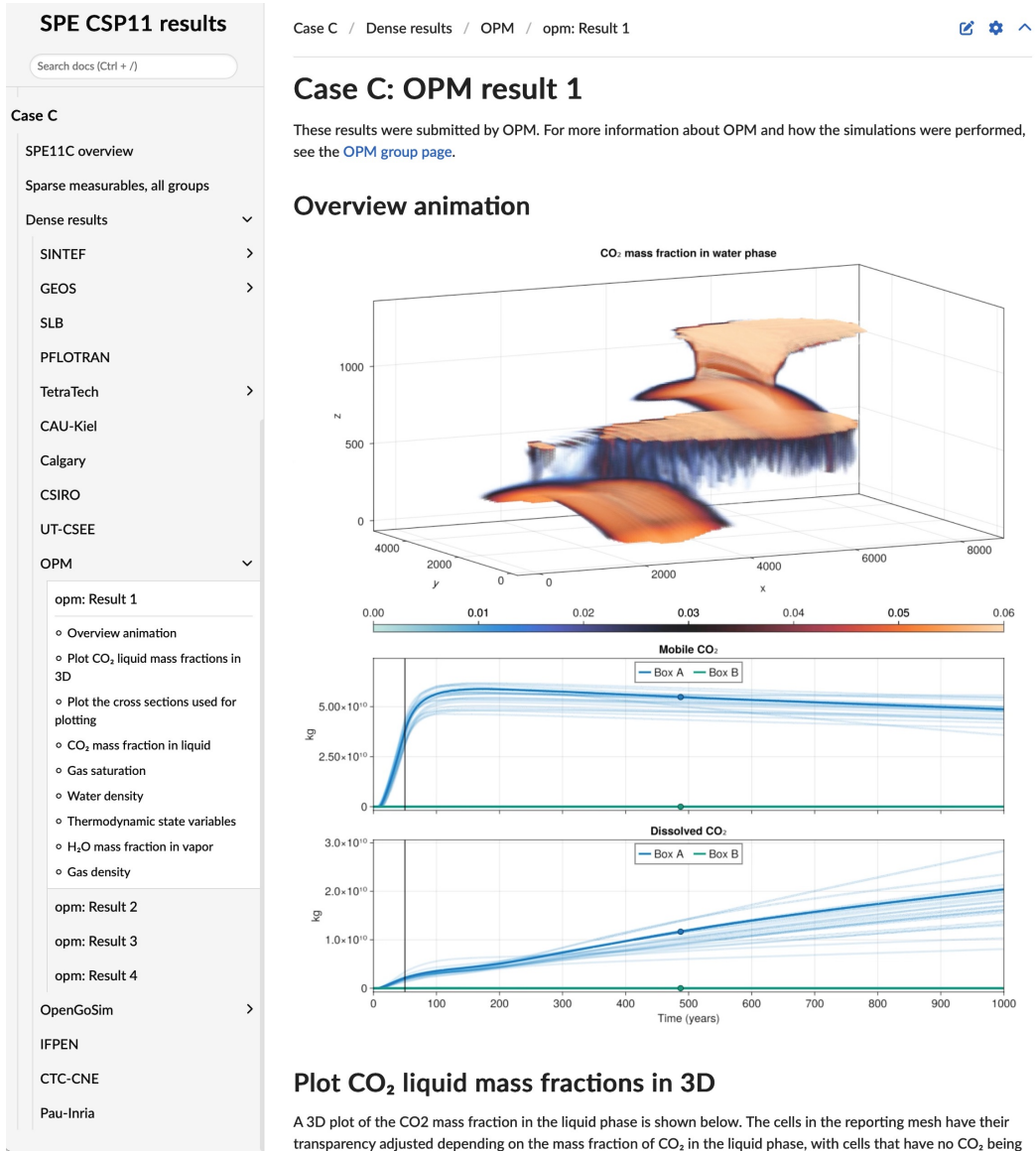
SPE11

The OPM Team:

- Only that submitted results for all cases
- Submitted the highest resolutions
  - Can now do ~170M cells (100M last year)
- Results similar to «middle of the pack»

Thanks for your effort!

Screenshot from official results page



Industrial usage at Equinor

”Equinor is using OPM Flow on one asset and is evaluating it for use on more.” (Equinor 2023)

”The black-oil simulation models are now run with OPM Flow in the assets to establish input to technical decisions.” (Reiso et.al 2025)

4 oil assets and 1 CO_2 storage asset discussed in paper

Paper: *Lessons Learned In Using Open-source Simulation Software On Real Asset Models*

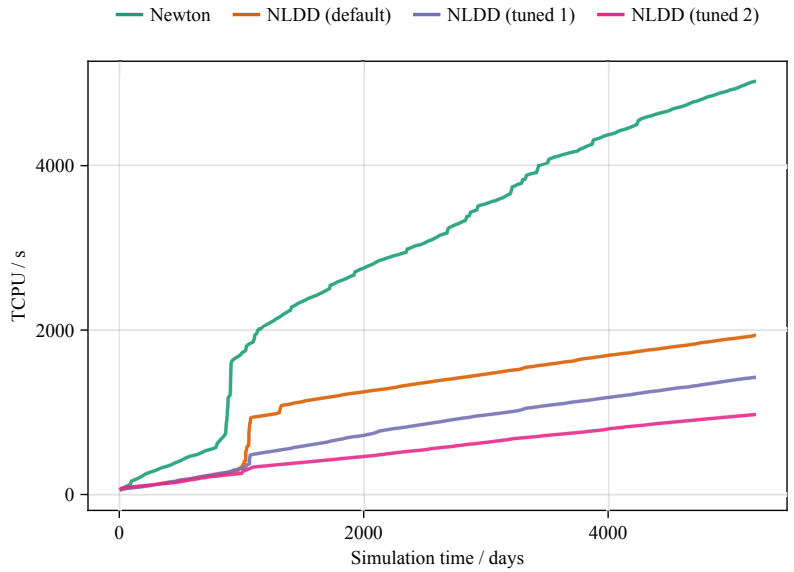
Talks by **TBA** and **Edel Reiso** today, **Alf Birger Rustad** tomorrow!

Recent and ongoing work

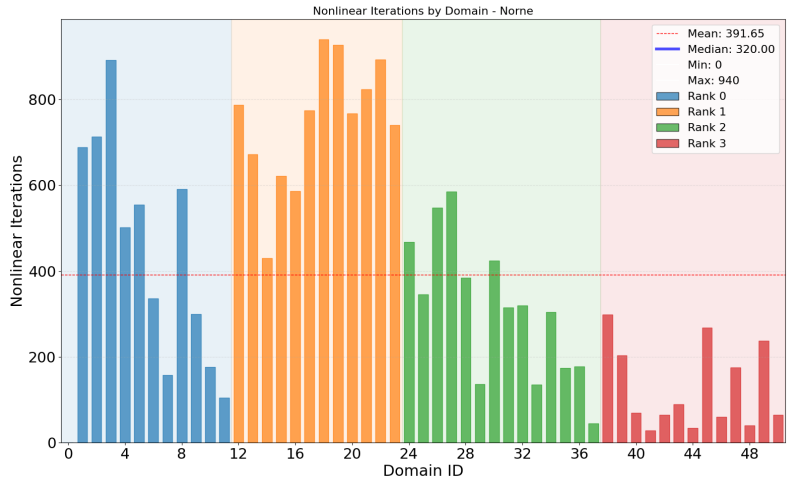
Improved nonlinear solvers

- Nonlinear domain decompositioning (NLDD) improvements:
  - Much better logging/reporting
  - More robust, works well for most cases out of the box
  - Ongoing: group controls in parallel

Runtimes for Sleipner with Newton and NLDD



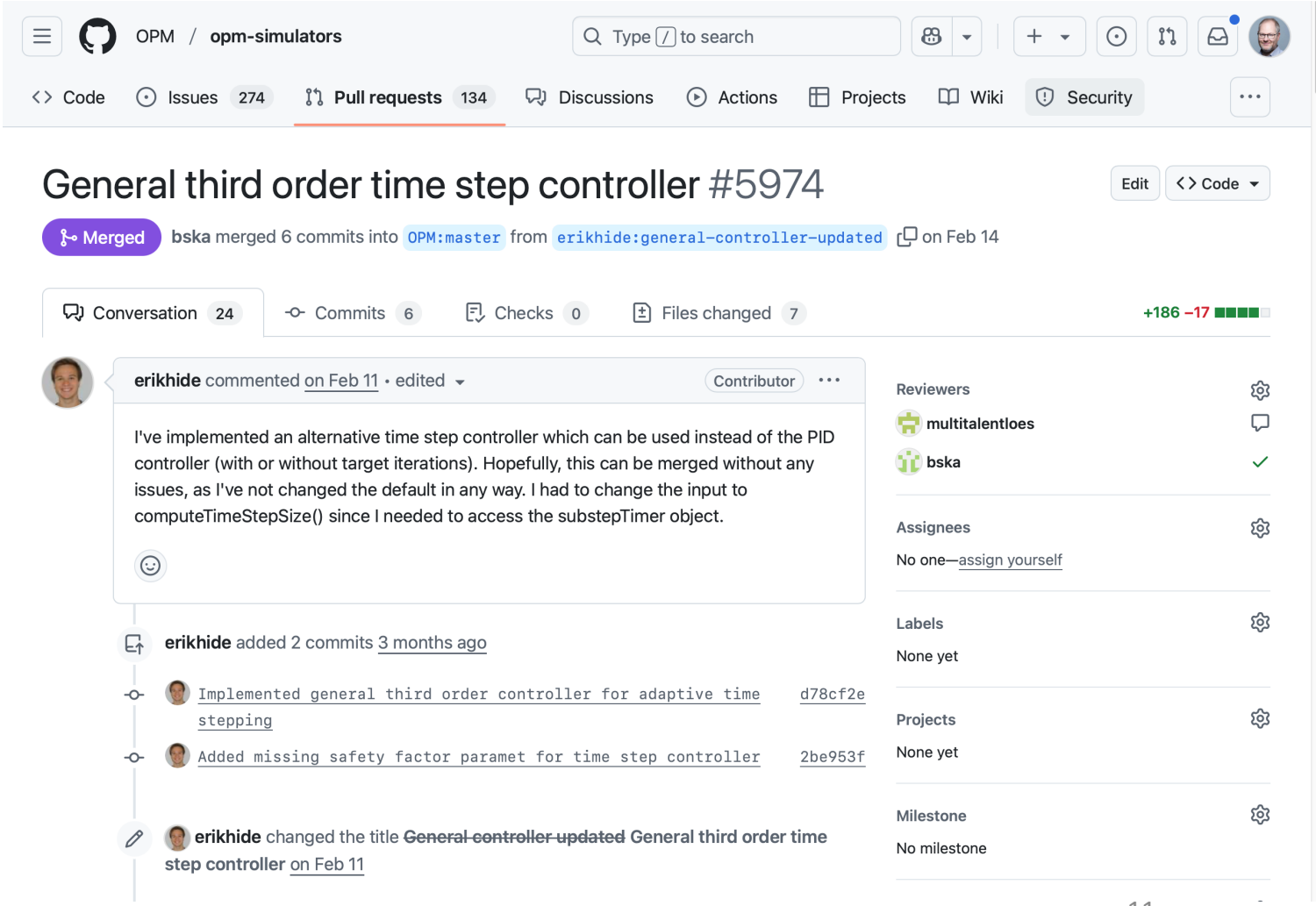
Nonlinear iterations for local solves per NLDD subdomain and MPI domain on Norne



Paper: *A distributed field-scale parallel nonlinear domain decomposition solver*

Timestepping: fixes and new algorithms

- New timestepping algorithm added
- Existing ones bugfixed
- Ongoing work: using convergence diagnostics



Talk by **Erik Hide Sæternes** later today!

GPU-accelerated linear solvers

- Performance improvements for ILU0 and DILU, autotuning by default!
- Testing facilitated by gpu-istl (easy to test new components)
  - Wrapped GPU preconditioners to be CPU preconditioners
  - Wrapped CPU preconditioners to be GPU preconditioners
  - ... But wrappers no longer *required* -> better performance!
- CPR on GPU (two-level preconditioner)
- AMG on GPU (from AMGX, Hypre, and BDA)
- Well-operators: MS wells in progress, standard wells in prototype

Papers:

- *gpu-ISTL - Extending OPM Flow with GPU Linear Solvers*
- *A Comparison of DILU And ILU(0) as GPU-Accelerated Preconditioners*

Talks by **Vinicius Martins** and **Jakob Torben** tomorrow!

GPU-accelerated properties

- Can now evaluate all properties of some models on GPU
  - CO2STORE, blackoil table input all supported
  - NOT supported: endpoint scaling, hysteresis, three-phase relperm, multiplexed pvt properties
  - Initial prototype target: SPE11-like cases
- Minimal changes to existing code
  - Typical change: templatize on storage (std::vector -> template argument)
- Almost identical code runs on CPU and GPU!

```
template<class TraitsT>
class PiecewiseLinearTwoPhaseMaterialParams
```

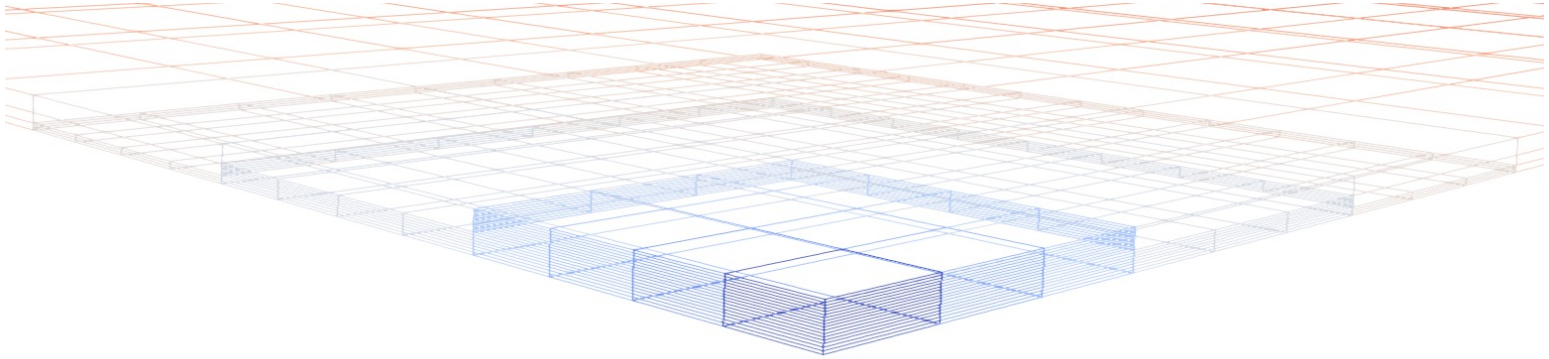
```
template <class TraitsT, class VectorT = std::vector<typename TraitsT::Scalar>>
class PiecewiseLinearTwoPhaseMaterialParams
```

```
template <class TraitsT, template <class> class Vector = std::vector>
class PiecewiseLinearTwoPhaseMaterialParams
```

Talk by **Tobias Meyer Andersen** tomorrow!

Local Grid Refinement (LGR)

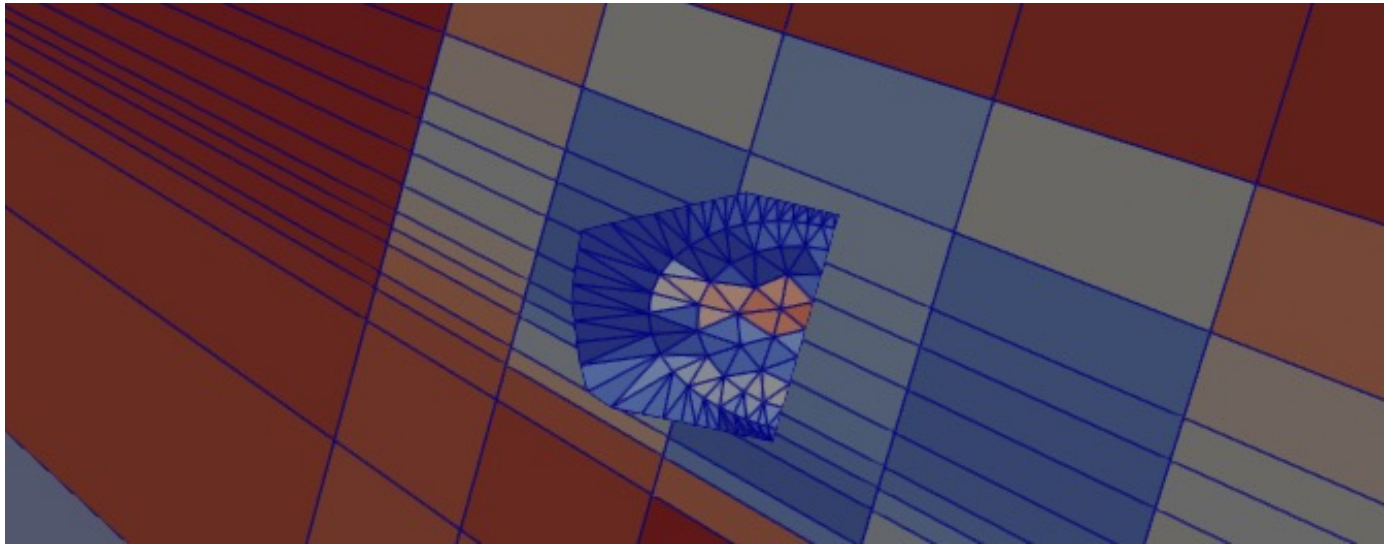
- More flexible *parent cell sets*
  - not necessary block-shaped
  - not necessary all-active cells
  - LGRs sharing faces/corners is supported now
- MPI parallel CpGrid with LGRs in progress
- Wells aware of LGRs
- Initial output file support (EGRID)



Talks by **Antonella Ritorto** and **Artur Castiel** tomorrow!

Geomechanics

- Coupled flow and geomechanics, fractures
- Improved grid processing
  - Geomech (and thermal) grids must include zero-porosity cells
  - NNCs for flow mostly meaningless for mechanics



Fracture surface embedded in corner point geometry

Talk by **Halvor Møll Nilsen** later today!

Parallel performance

- Distributed multisegment wells
- Improved MPI partitioning
  - Revised graph: each well's connected cells are one single node
- Lock-free OpenMP/thread parallelization
  - Used for general assembly, tracer, some output
  - Makes high --threads-per-process more feasible

```
Number of MPI processes:          1
Threads per MPI process:         10
Setup time:                       0.68 s
  Deck input:                     0.32 s
Number of timesteps:            331
Simulation time:                 95.31 s
  Assembly time:                  4.52 s (Wasted: 0.0 s; 0.0%)
    Well assembly:                0.70 s (Wasted: 0.0 s; 0.0%)
  Linear solve time:             42.57 s (Wasted: 0.0 s; 0.0%)
    Linear setup:                19.35 s (Wasted: 0.0 s; 0.0%)
  Props/update time:              4.88 s (Wasted: 0.0 s; 0.0%)
  Pre/post step:                 42.19 s (Wasted: 0.0 s; 0.0%)
  Output write time:              0.27 s
Overall Linearizations:         1374      (Wasted:      0; 0.0%)
Overall Newton Iterations:      1043      (Wasted:      0; 0.0%)
Overall Linear Iterations:      1919      (Wasted:      0; 0.0%)
```

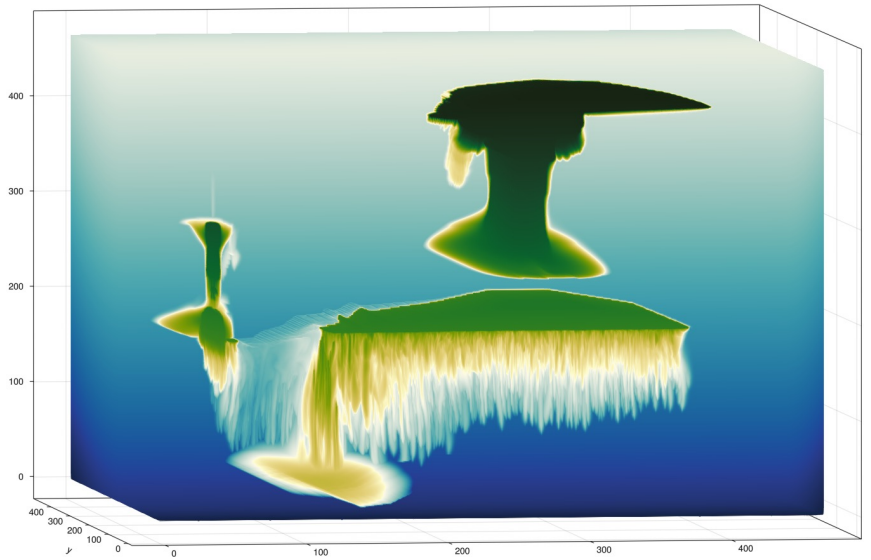
Assembly time same for 10 processes and 1 thread each and 1 process with 10 threads

```
Number of MPI processes:         10
Threads per MPI process:          1
Setup time:                       1.00 s
  Deck input:                     0.50 s
Number of timesteps:            333
Simulation time:                 28.62 s
  Assembly time:                  4.47 s (Wasted: 0.0 s; 0.2%)
    Well assembly:                0.40 s (Wasted: 0.0 s; 0.2%)
  Linear solve time:              9.79 s (Wasted: 0.1 s; 0.8%)
    Linear setup:                 4.22 s (Wasted: 0.0 s; 0.2%)
  Props/update time:              2.77 s (Wasted: 0.0 s; 0.1%)
  Pre/post step:                  8.40 s (Wasted: 0.0 s; 0.0%)
  Output write time:              1.78 s
Overall Linearizations:         1387      (Wasted:      2; 0.1%)
Overall Newton Iterations:      1055      (Wasted:      2; 0.2%)
Overall Linear Iterations:      2202      (Wasted:     34; 1.5%)
```

Talks by **Lisa Julia Nebel** and **Michal Tóth** tomorrow!

CO_2 simulation

- Advanced modeling capabilities, f.ex:
  - Salt precipitation
  - Upcaling of convecting mixing
- Good performance and scaling
- Successful SPE11 participation!



Brine density from SPE11C, 101-million cell simulation

Example papers:

New Sub-grid Model for Convective Mixing in Field-Scale Storage Simulation

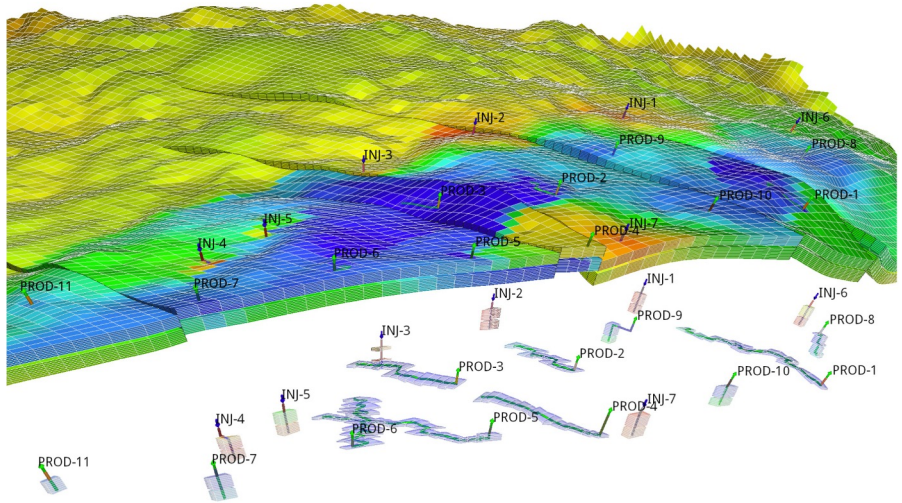
Impact of Intermittency on Salt Precipitation During CO2 Injection

Talks by **Tor Harald Sandve** today and **Trine Mykkeltvedt** tomorrow!

Robustness, wells, groups++

- Improved PVT extrapolation at low pressures
  - Avoid unphysical behaviour
- Improved initialization of wells and network
- Revised algorithms for finding solution before full Newton iteration
  - Dozens of small changes
  - A lot of ongoing work in this area
- Completing features (example: GSATPROD)

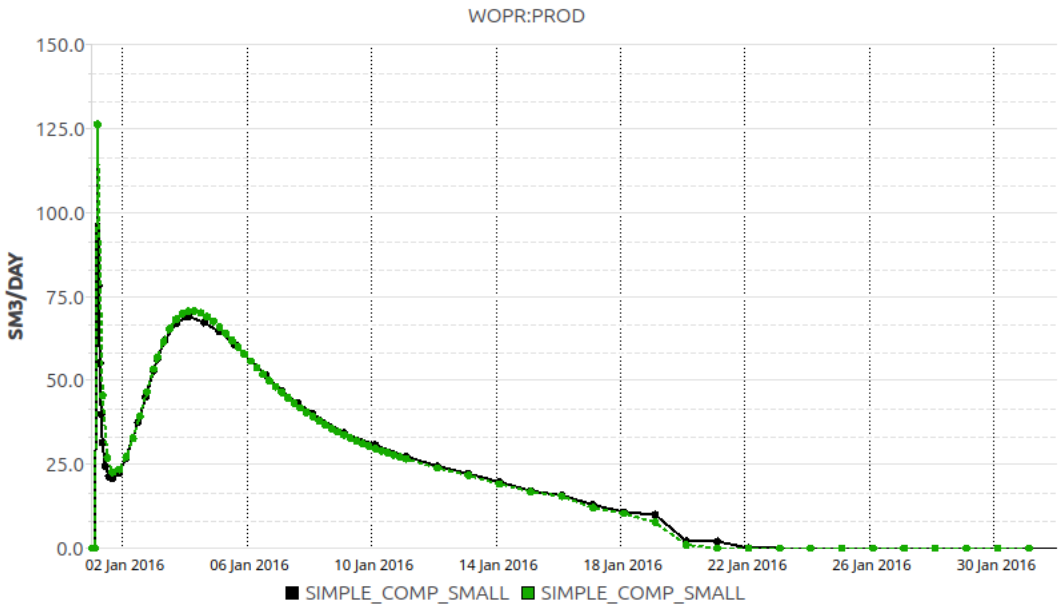
- Robustness and performance go together!



Well cells from an OLYMPUS case

Compositional simulation

- General compositional simulation!
  - Arbitrary number of components
  - Additional water phase
  - Multiple equations of state (EOS) available
  - Standard well model implemented (some limitations for now)
- Output to VTK and Eclipse formats



1D validation of compositional behaviour

Talk by **Kai Bao** later today!

… and much more!

- Input/Output improvements
  - Much improved consistency checks for saturation functions
  - Support updated restart format (E100 version 2022.4)
  - Some region support for UDQs
  - UDQ checks and validation
- Reservoir coupling
  - Prototype available
- Integrated machine learning
  - can replace arbitrary relation with ML model (some assembly required...)
- Python coupling, API and tools
  - PET, Pycopm, plopm, pyopmspe11++, python docs

Talks by **Rolf Johan Lorentzen** later today and **David Landa Marban** tomorrow!

SINTEF

Technology for a better society