

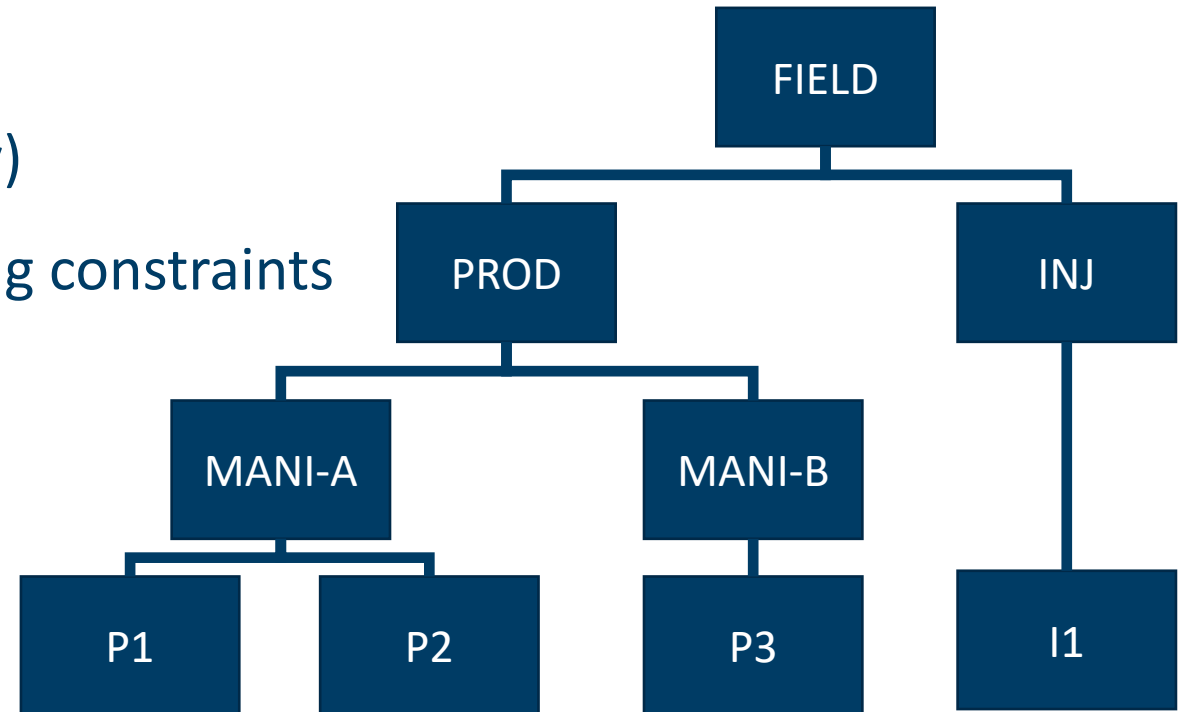


IMPROVEMENTS TO OPM FLOW (SINCE THE LAST MEETING)

Presenter: Atgeirr Flø Rasmussen

Improved well-group controls

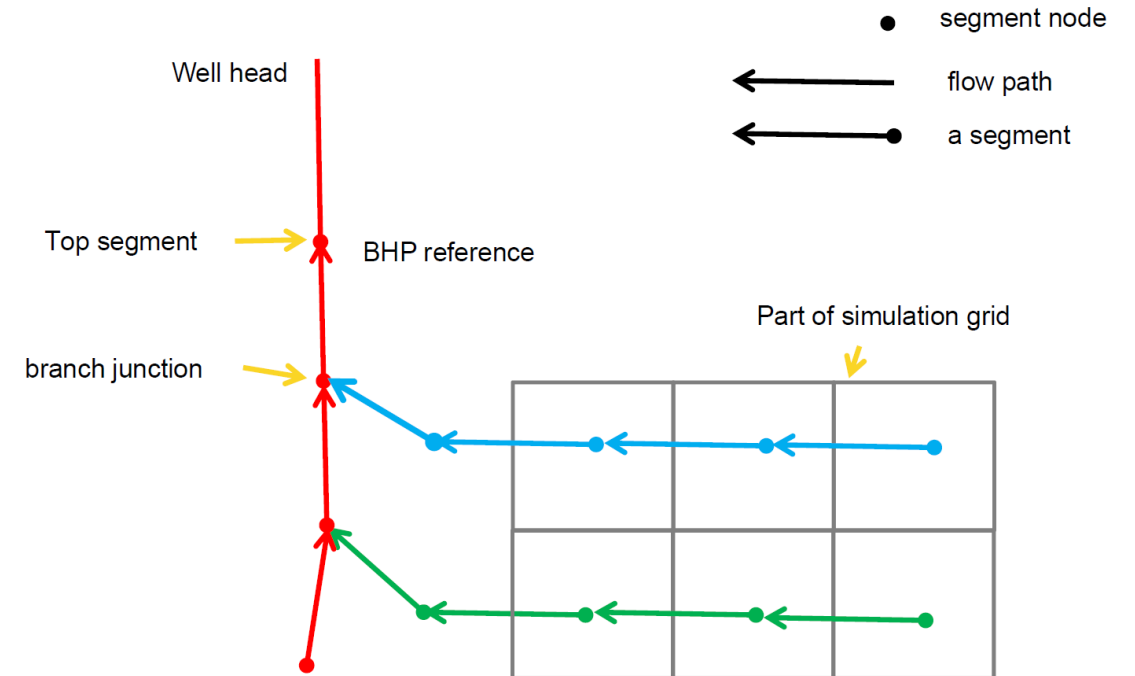
- Works in parallel
- Multi-level controls (in a group hierarchy)
- Improved logic for switching and checking constraints
- Much closer to Eclipse behaviour
 - Many, many corner cases...



Ongoing active development

Multi-segment wells

- Improvements in features, robustness
- WSEGSICD
 - Spiral Inflow Control Device (SICD)
 - Reduces inflow through connections with high rates
- WSEGVALV
 - Valve device in the well bore
 - Induces extra pressure drop in segment



Prediction-related features (I)

- Better THP (tubing head pressure) support
 - Connected to bottom-hole pressure (BHP) and well rates through vertical flow performance (VFP) relation: $bhp = f(thp, rates)$
 - Adds significant nonlinearity to the well model.
 - Supported also for multi-segment wells
- Potentials more robust
 - «How much could this well produce given only pressure constraints»
 - Used for distributing rate targets under group control.
 - Standard wells: really careful approach to ensure we do not miss the solution (still quick because no heavy numerical work)
 - Multisegment wells: a little less careful approach (since it requires re-solving the MSW equations which are much heavier)

Prediction-related features (II)

- RPTRST
 - Can deal with «output restart every 3 months» and similar
- WTEST
 - Re-opening shut wells if they become operable again
 - Now interacts more correctly with other keywords
- WECON
 - Economic limits: close a well if some constraint is broken
 - More comprehensive, covering a wider range of constraints

Better aquifer support

- Big general improvements (correctness, robustness)
- Fetkovitch aquifers
 - Improved sufficiently to run well on industrial models
 - Restart file support
- Carter-Tracy aquifers
 - Lacking «hard» testcase to verify on industrial case

Restart files and Eclipse compatibility

- Eclipse is able to restart from Flow restart files in most situations, and vice versa.

Separate talk by Jostein Alvestad and Torbjørn Skille

Linear solver

- GPU support
 - Using CUDA, for NVIDIA hardware only
- Flexible linear solver
 - Can be configured at runtime (using preconditioner factory), including complex multi-level preconditioners such as CPR with AMG.
 - Using `boost::property_tree` for options, read from JSON
 - Newest Dune version (2.7) also have a preconditioner factory.

```
{
  "tol" : "1e-2",
  "maxiter" : "20",
  "verbosity" : "0",
  "solver" : "bicgstab",
  "preconditioner" : {
    "type" : "cpr",
    "pressure_var_index" : "1",
    "finesmoothing" : {
      "type" : "ParOverILU0",
      "relaxation" : "1.0"
    },
  },
  "coarsesolver" : {
    "tol" : "1e-1",
    "maxiter" : "1",
    "verbosity" : "0",
    "solver" : "loopsolver",
    "preconditioner" : {
      "type" : "amg",
      "smoother" : "ILU0",
      "maxlevel" : "5",
      "coarsenTarget" : "1000",
      "alpha" : "0.2",
      "beta" : "0.0001",
      "verbosity" : "0",
      "relaxation" : "1.0",
      "iterations" : "1"
    },
  },
  "verbosity" : "0"
}
```

Also separate talk by Halvor Møll Nilsen

Added models

- Brine
 - Modeling the presence and transport of salt in the water phase
 - Salt concentration modifies fluid properties
- Foam
 - Simple model assuming surfactant/foam flows with the gas phase
 - Modifies the mobility of the gas phase
- Single-phase
 - With thermal, useful for geothermal properties

The brine model is still under development. The others are not extensively tested

Improved memory usage (esp. in parallel)

- Parsing the whole deck only on one process
 - Serialization of EclipseState
- Reduced memory consumption by storing only active cell properties
 - Example case: 6.1M total cells, 132k active

Effort still ongoing. Parallel I/O still in the future.

Support for ACTIONX and UDQ

- User Defined Quantities (UDQ) can be defined in the deck, and used in subsequent keywords.
- Conditional actions can be inserted into the Schedule using the ACTIONX keyword.

UDQ

```
-- WUPR3 sorts production wells from  
-- poorest (highest wct) to best.  
-- ACTIONX will shut #1 in this list  
DEFINE WUPR1 1/(WWCT 'OP*') /  
DEFINE WUPR3 SORTA(WUPR1) /  
/
```

ACTIONX

```
ACT01 10 /  
FMWPR >= 4 AND /  
WUPR3 'OP*' = 1 /  
/
```

WELOPEN

```
'?' SHUT 0 0 0 2* /  
/
```

ENDACTIO

Python bindings and related development

Separate talks by Joakim Hove and Robert Klöfkorn



Miscellaneous

- Improved logging and error handling in parallel
- More extensive warnings for unsupported keywords

- Two releases
 - 2019.04 (managed by Arne Morten Kvarving)
 - 2019.10 (managed by Markus Blatt)
- Improved manual (1751 pages, one thousand seven hundred and fifty one is a very large number!)
- 1225 PRs merged since Jan 26 2019 (by Jan 24 2020)



Technology for a better society