SINTEF

Flexible routines for computing sensitivities by adjoints

MRST Symposium, September 14-15, 2021

S. Krogstad and M.A. Borregales Reverón

SINTEF

Motivation

- Parameter sensitivities are useful for model calibration and establishing influence of parameter uncertainties (amongst other things).
- Sensitivities for any number of parameters can be efficiently computed in a single adjoint simulation.
- Adjoint-based computation of sensitivities has been part of MRST for some years, but recent work on calibration of network-type/coarsened models revealed the need for a much more flexible implementation.
- Here we present the new **`ModelParameter`**–class (and accompanying functions), and illustrate its use by two simple examples.

The `ModelParameter`-class

- Main purpose is to keep track of and update parameters for adjoint simulations and optimization
- Any parameter belonging model, schedule or initial state can be set up
- Default setup available for many common parameters
- Options for setting up multipliers, subsets, groupings and parameter scaling for use in optimization.

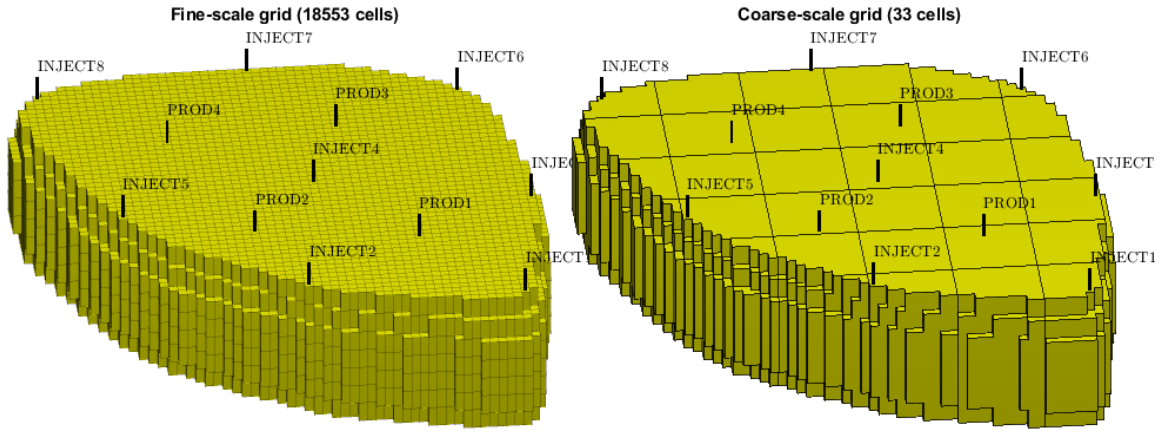
```
>> % struct of simulation setup
>> setup

setup =

  struct with fields:

      model: [1×1 GenericBlackOilModel]
   schedule: [1×1 struct]
     state0: [1×1 struct]

>> % default setup of pore volume parameter
>> p = ModelParameter(setup, 'name', 'porevolume')

p =

  ModelParameter with properties:

            name: 'porevolume'
            type: 'value'
          boxLims: [22.9880 5.109le+03]
          subset: ':'
         scaling: 'linear'
  referenceValue: []
       belongsTo: 'model'
        location: {'operators'  'pv'}
          nParam: 200
         lumping: []
          setfun: @(obj,loc,v)setfield(obj,loc{:},v)
     scalingBase: NaN
    controlSteps: []
```

Computing sensitivities

- For computing parameter sensitivities, some objective needs to be defined. For instance, `matchObservedOW` computes the mismatch between the output of a simulation (rates, BHP) to a supplied reference output.
- For a list of parameter objects, the corresponding sensitivities are computed by the function `computeSensitivitiesAdjointAD`
- For an optimization setting, everything is wrapped together in the function `evaluateMatch` that evaluates the objective and optionally its gradient with respect to the set of *scaled* parameters.

Example 1: tuning a coarse Egg-model

- We consider a very coarse upscaling of the first realization of the Egg ensemble*: 18533 cells -> 33 cells
- Initial coarse model created by `upscaleModelTPFA` (simple harmonic averaging of transmissibilities)
- For illustration, we introduce a large number (297) of coarse model parameters



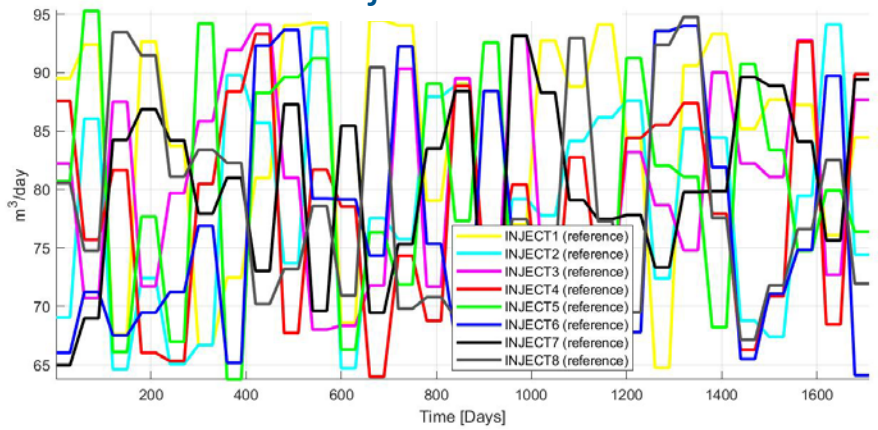

```
config = {...
     %name            include    scaling              boxlims    relativeLimits
    'porevolume',        1,      'linear',    [.01*pv, 1.5*pv],              []
    'conntrans',         1,        'log',                  [],     [1e-2, 1e2]
    'transmissibility', 1,         'log',                  [],     [1e-2  1e2]
    'swl',               1,      'linear',            [0, .3],              []
    'swcr',              1,      'linear',            [0, .4],              []
    'swu',               1,      'linear',            [.7, 1],              []
    'sowcr',             1,      'linear',            [0, .4],              []
    'krw',               1,      'linear',          [.5, 1.5],              []
    'kro',               1,      'linear',          [.5, 1.5],              []};
```

*Jansen, J.D., Fonseca, R.M., Kahrobaei, S., Siraj, M.M.,Van Essen, G.M., Van den Hof, P.M.J.: The egg model–a geological ensemble for reservoir simulation. Geo-science Data Journal1(2), 192–195 (2014).

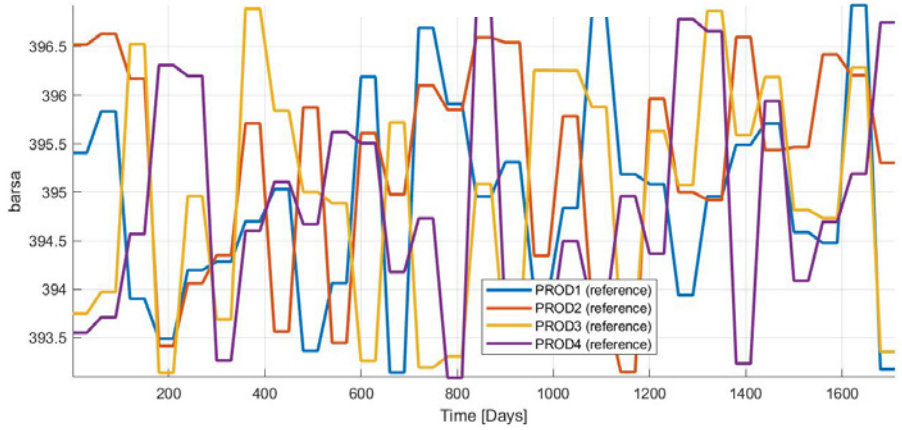
Example 1: tuning a coarse Egg-model

- We train the coarse model output to the fine model output using a randomly perturbed schedule
- We run the optimization for 30 iterations (34 coarse model simulations)
- The tuned coarse model is then validated for different schedules

Injector rates



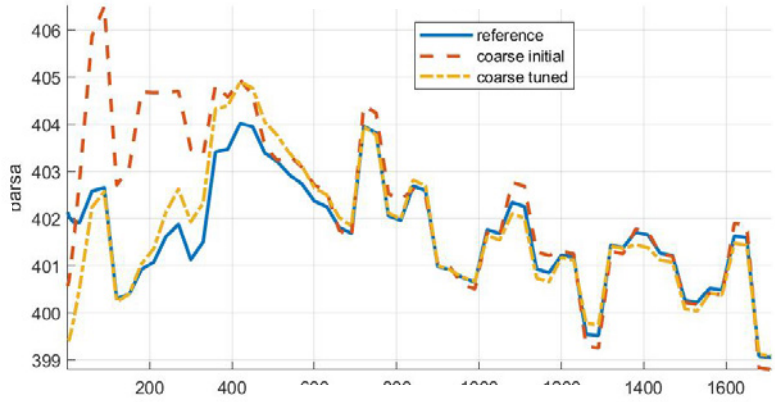
Producer BHPs



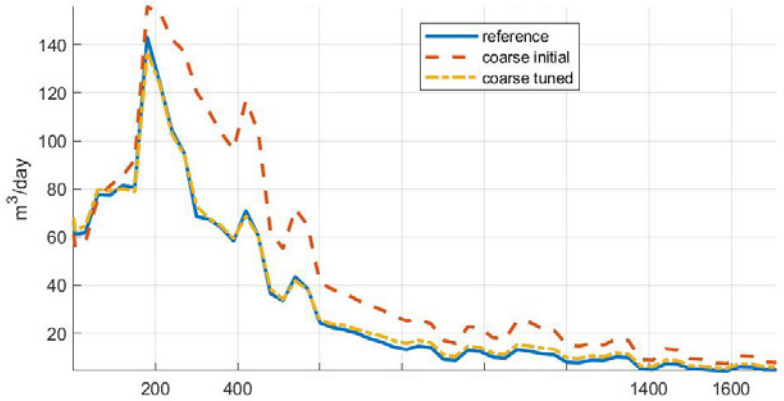
Example 1: tuning a coarse Egg-model

Training schedule

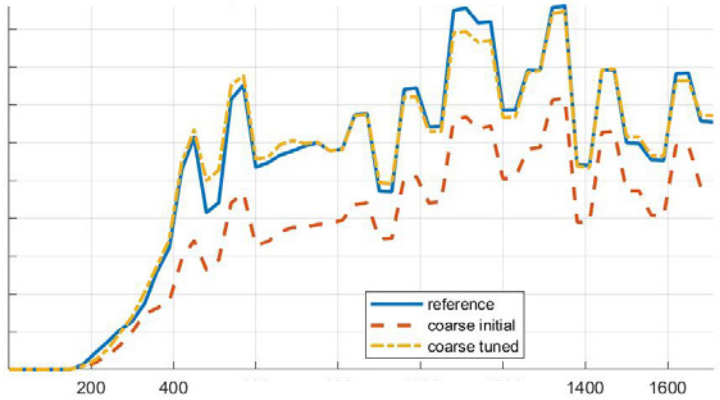
INJECT1: BHP



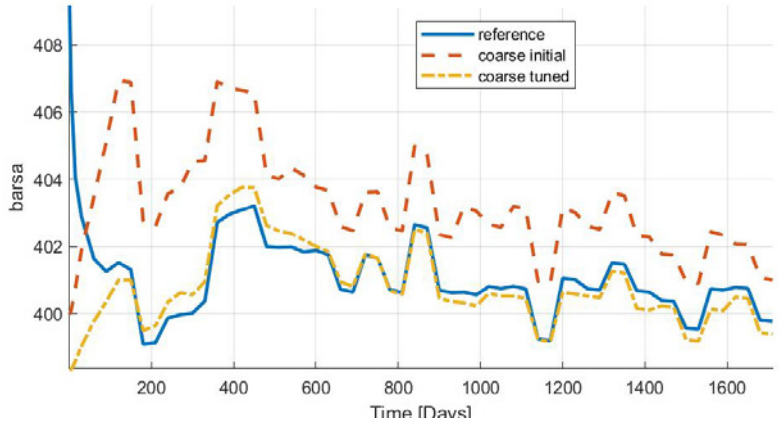
PROD2: OIL



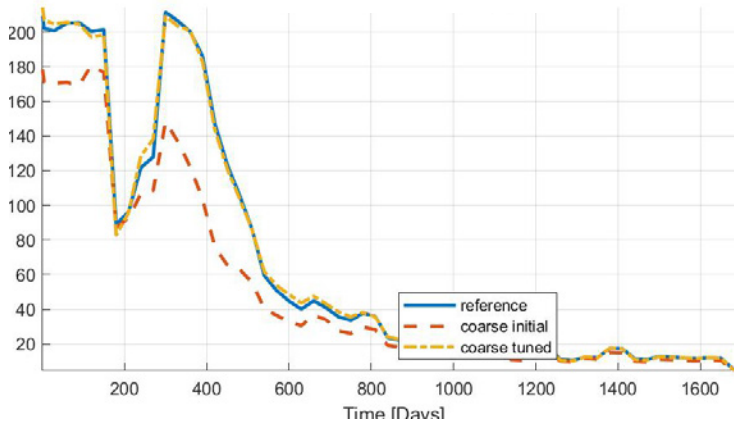
PROD2: WATER



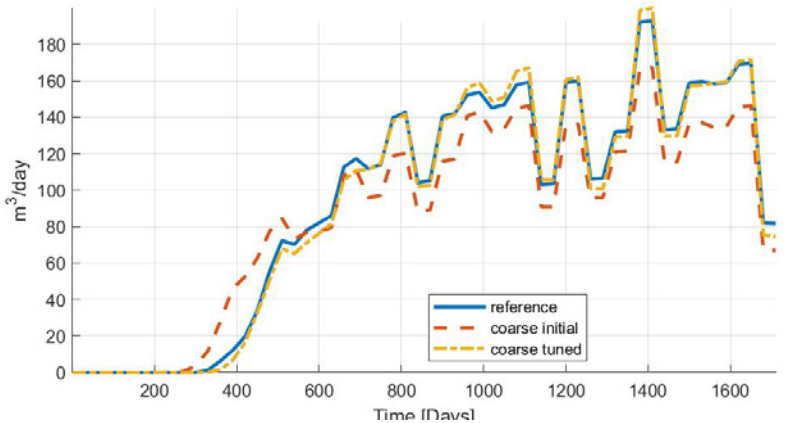
INJECT4:BHP



PROD4:OIL

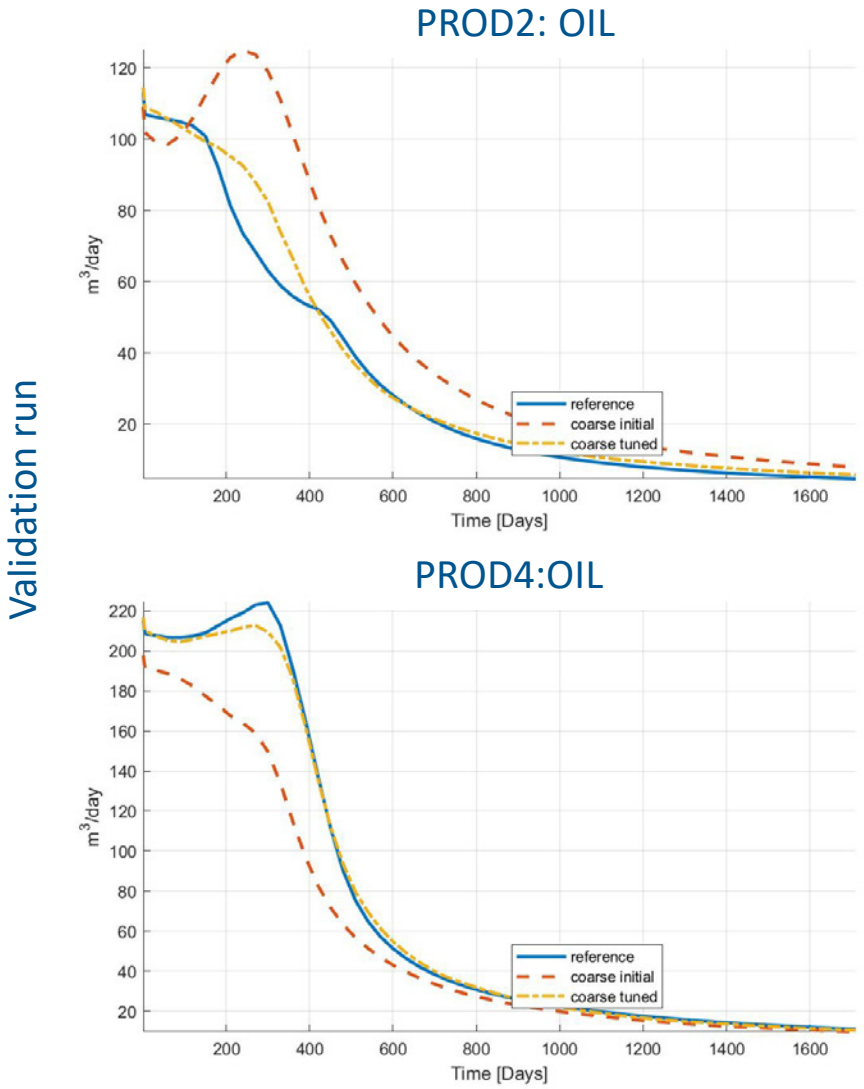


PROD4:WATER



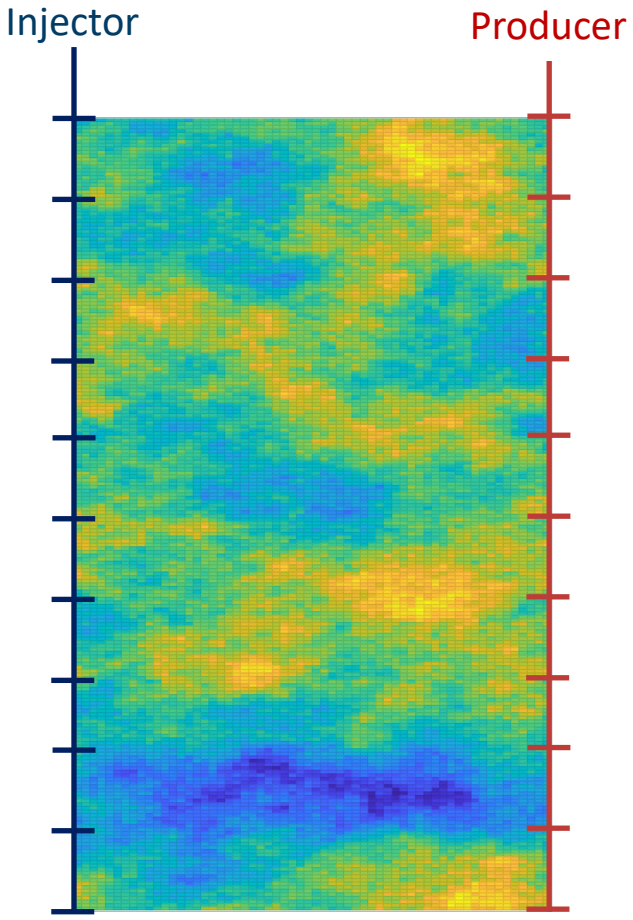
Example 1: tuning a coarse Egg-model

- Tuned coarse model able to capture dynamics of fine model well except for early transients.
- Tuned coarse model performs well also for validation runs, but clear (and obvious) tendency that match is best for simulations relatively *close* to the training.



Example 2: optimizing valve openings

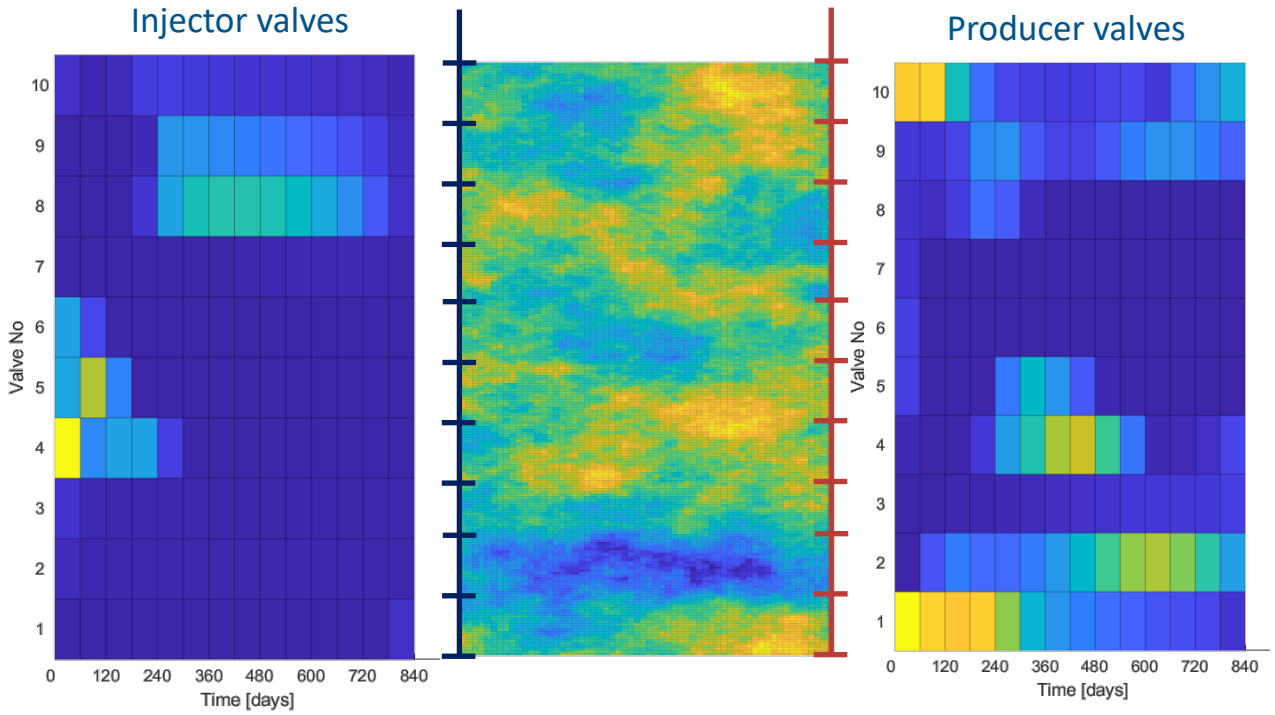
- To illustrate alternative use of sensitivity code, we consider a simplified example with valves.
- Rather than using a mismatch function as objective, we use net-present-value (NPV)
- Valve openings are approximated by well-index multipliers along ten sections of each well
- Multipliers are allowed to change every 60 days (2x10x14 = 280 parameters)



Example 2: optimizing valve openings

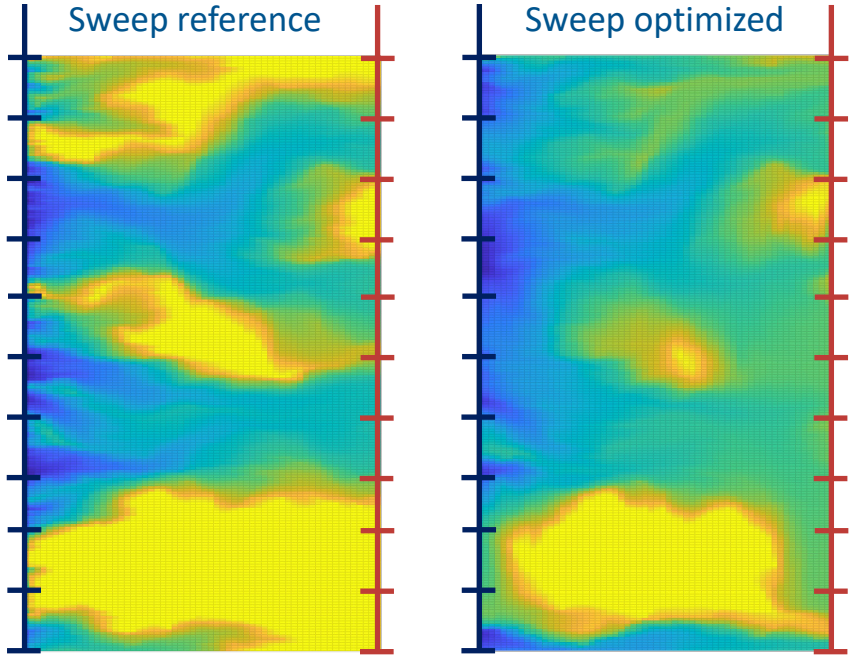
Optimized valve settings over time

yellow fully open, dark blue closed



Sweep at end of simulation

NPV increase of 33%



Final remarks

- New features for computation and use of parameter sensitivities in MRST is now available (Release 2021a). Further examples available in optimization-module and network-model-module. Examples presented here, and a few updates will be pushed to master-branch on bitbucket after release.
- Model tuning based on sensitivities obtained from adjoints is efficient, but one should be aware:
  - Accuracy of adjoint-based sensitivities/gradients is linked to accuracy of simulator. Consider using tighter tolerances than default in forward runs.
  - Some of the computed sensitivities may be extremely small, e.g. for very high transmissibilities/well-indexes relative to neighbouring parameters (may also be subject to round-off errors). This can to some extend be improved by using logarithmic scaling, but for optimization one should take this into account when selecting an initial guess.

SINTEF

Technology for a better society