

# Local Updating of Fluvial Realisations

Arne Skorstad    Øivind Skare    Ragnar Hauge    Lars Holden

December 1, 1999

---

## Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Model</b>	<b>2</b>
<b>3</b>	<b>Objectives</b>	<b>3</b>
3.1	Restart . . . . .	3
3.2	Rate conditioning . . . . .	4
<b>4</b>	<b>The Barrier programme</b>	<b>4</b>
4.1	Drawing observed barriers . . . . .	4
4.2	Description of new model parameters . . . . .	6
<b>5</b>	<b>The gridder</b>	<b>6</b>
5.1	Description of model parameters . . . . .	6
<b>6</b>	<b>The Fluvial programme</b>	<b>7</b>
6.1	Description of model parameters . . . . .	8
6.2	Parameter discussion . . . . .	10
<b>7</b>	<b>Test Examples</b>	<b>15</b>
7.1	First test reservoir . . . . .	16
7.2	Second test reservoir . . . . .	19

---

<b>8 Conclusion</b>	<b>21</b>
<b>References</b>	<b>21</b>
<b>A Production data plots — First Reservoir</b>	<b>22</b>
<b>B Production data plots — Second Reservoir</b>	<b>27</b>
<b>C Model file extensions</b>	<b>32</b>
<b>D STORM REALISATION FORMAT</b>	<b>33</b>

## 1 Introduction

The project “Restarting and local updating of Fluvial realisations” has had two main, and separate, objectives, although the restart option was necessary for the local updating part.

Restarting a realisation can be useful in many ways. An immediate benefit is that it makes it possible to continue a previously generated realisation by adding more iterations. More important in a history matching perspective is that it enables a new simulation of the reservoir, that includes new data and parameters, and which starts with the previous “good” realisation as initial state. The simulations will therefore likely be close to the converged state at the beginning, which gives faster convergence.

A part of the project has been to design a new parametric realisation format, that provide the possibility of restarting simulations from the model.

Ideally, a reservoir should be conditioned on observed production history. In a Bayesian setting, this would lead to very slow MCMC algorithms, as the time spent for each iteration would be very high. However, it is computationally feasible to condition on one-phase steady-state well rates. Instead of a full reservoir simulation, a linear pressure equation is solved at each iteration of the MCMC algorithm. Because each iteration of the MCMC algorithm involves only small changes in the reservoir, the solution at the previous iteration will be a good initial value for the pressure equation solver. Well rates are directly related to the connectivity of the reservoir and, hence, also related to the two-phase flow properties of the reservoir. Well rates are not observed directly, but may to some extent be induced from observed production data.

When testing the implemented methodology, the particular problem of how to update an already history matched reservoir with new well observations was considered. We do not condition on observed well rates, but instead on preservation of the well rates of the history matched reservoir. The new updated reservoir will then satisfy all well observations and maintain the connectivity of the history matched reservoir. Since the initial state, the already history matched reservoir, is good; the number of iterations needed in the MCMC simulations will be limited. The method is demonstrated on two complex fluvial reservoirs, where a history matched reservoir is up-

dated by conditioning on well rates and facies observations from 8 producers and 5 injectors. This “local updating” aspect of this project has also been discussed in Skorstad, Hauge, Holden & Skare (2000).

## 2 Model

The model is an extension of the model discussed in detail in Holden, Hauge, Skare & Skorstad (1998). It is an object model that describes the distribution of facies, mainly channel facies (sand) and background facies (shale), in a fluvial reservoir. The posterior distribution for realisation  $r$  given well data  $w$ , sand/gross restrictions  $\gamma$ , seismic data  $s$ , and well rates  $\rho_0$  reads

$$\pi(r|s, w, \gamma, \rho_0) = cf(r|s, w, \gamma)f_P(\rho_0|r). \quad (1)$$

The posterior model  $f$  was described in Holden et al. (1998). The production likelihood,  $f_P$ , was chosen to be Gaussian:

$$f_P(\rho_0|r) \propto \exp\left(-0.5(\rho - \rho_0)^T \Sigma_\rho^{-1}(\rho - \rho_0)\right) \quad (2)$$

where  $\Sigma_\rho$  is a diagonal matrix and  $\rho$  is the  $p$ -dimensional vector of steady-state well rates from  $r$ . The steady state properties  $\rho_0$  may be found from a previously history matched realisation or specified manually based on observed production data. The matrix  $\Sigma_\rho$  is the variance matrix for the rates, indicating how much the final rates are allowed to deviate from the original rates. Since small rates should have smaller variance than larger rates, a default scaling factor has been implemented. This factor gives

$$\Sigma_{\rho ii} = \left(100 - 99 \exp\left(\frac{1 - (\max_j |\rho_{0,j}|)/|\rho_{0,j}|}{99}\right)\right)^{-1} \sigma^2, \quad (3)$$

where  $\sigma^2$  is the variance for the largest rate. To evaluate  $\rho$ , permeability values must be assigned to the facies. Constant permeability for the channel facies (sand), the crevasse facies and the background facies (shale) is used. This choice limits the computation time. Although permeability values in a real application do vary within each of these facies, the average permeabilities differ with several magnitudes and will therefore dominate the flow pattern. Moreover, this constant permeability assumption is also made for computing  $\rho_0$ . This means that no bias should be introduced by this choice.

The well rates  $\rho$  are obtained after specifying the pressure in the wells and computing the steady state pressure.

### 3 Objectives

In this section the description of the algorithm and its limitations are given for both parts of this project.

#### 3.1 Restart

A restart of a realisation means that the program must be able to read an old realisation file, and continue the simulation with the old realisation as the initial state. The simulation must be on an area that is not exceeding (but could be smaller than) the reservoir boundaries for the old realisation, and the parameters of the new model may be changed from the old. Possible new well data can also be used in the conditioning, in addition to the well data used for the old realisation. A well used in the old realisation can be excluded from the new simulation if it is explicitly stated. A new interpretation of a well used in the old realisation can be used, but only if the well file is given a new name.

When a smaller area is used in the restarted realisation, it is crucial that the wells used to condition the original realisation must be either totally within or outside the new area. Top and bottom zone border maps and erosion maps must be the same as those used in the original realisation, as this otherwise would affect the well conditioning in the old wells.

Before the restarted simulation begins, the programme removes all families where channels violate the facies pattern seen in new wells. This is necessary since the simulation algorithm do not accept illegal families and channels.

## 3.2 Rate conditioning

A Metropolis-Hastings algorithm with simulated annealing is used to simulate from the posterior (1). At each iteration, well rates for a new proposed realisation are found by solving the pressure equation. The computation involved in solving this equation will dominate the total cpu time used. It is thus important to have a fast equation solver. In our implementation, a preconditioned conjugate-gradient method is used. Since the equation system from the realisations from the model forms a sparse matrix system, the library SPARSELIB, specifically designed for such systems, was used in the equation solver.

Once the pressure have been found throughout the realisation, the well rate is found by

$$\rho = \int_S \nabla p \cdot k \, dS$$

where the area  $S$  is the border of the well blocks.

## 4 The Barrier programme

The Barrier programme now assumes that the input realization file is on the new format. Conditioning routines have been modified slightly to handle the new well format. As before, the conditioning is only approximate for non-vertical wells. There is also a minor change in the model file for the Barrier programme.

### 4.1 Drawing observed barriers

Each barrier observation is a part of the well path, and is, thus, a collection of piecewise linear parts. For each barrier observation the following is done:

1. The minimum and maximum  $z$  value ( $z_{min}$  and  $z_{max}$ ) of the observation is found.
2. The observation is split in its monotone (decreasing or increasing) segments.

3. For each segment a corresponding barrier is drawn as follows:
  - (a) Length and width of the barrier is drawn from prior.
  - (b) The x-y coordinates of the barrier centre point is set equal to the mean x-y coordinates of the segment.
  - (c) The vertical position and thickness of the barrier is drawn as follows
    - i. If the segment attains both  $z_{min}$  and  $z_{max}$ , then the thickness is set equal to  $z_{max} - z_{min}$ . Otherwise, the thickness is drawn from the prior constrained so that the thickness is at least the difference in  $z$  value for this segment.
    - ii. The vertical centre point of the barrier is drawn uniformly inside that interval that ensures that  $z_{min}$  and  $z_{max}$  are covered by the barrier. If the segment attains  $z_{min}$  ( $z_{max}$ ), then the top (bottom) position of the barrier equals  $z_{min}$  ( $z_{max}$ ).

Figure 1 show a barrier observation that consists of five segments. This implies that five barriers are generated. Here,  $z_{min} = z_2$  and  $z_{max} = z_4$ . For neither of these segments, both the minimum and maximum  $z$  value will be attained. Thus, the thickness will be drawn from the constrained prior for all five barriers. For example, the barrier corresponding to the first segment will be at least  $z_1 - z_0$  thick. For segment two and three,  $z_{min}$  is attained. Therefore, the top position of these barriers will equal  $z_{min}$ . The last segment attains  $z_{max}$ ; thus, the bottom position of this barrier will equal  $z_{max}$ .

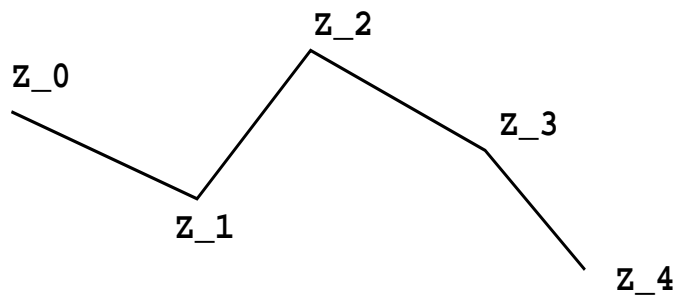


Figure 1: A barrier observation that consist of five linear segments.



This new conditioning scheme is only approximative. If the observation follows an almost horizontal well path; neither the width, length or thickness are drawn correctly. The barrier will always be at least as thick as the observed thickness, but may fail to cover the extent of the observation in the  $xy$ -plane. If the well path is vertical, the the width, length and thickness are drawn correctly.

## 4.2 Description of new model parameters

The modification of the Barrier programme results in three more parameters to the **SIMU** command. These specifies the dimension of the internal sampling grid, used when computing barrier proportion.

<**sampnx(int)**> Description: Number of grid nodes in the  $x$ -direction.  
Legal values: Integer > 0.

<**sampny(int)**> Description: Number of grid nodes in the  $y$ -direction.  
Legal values: Integer > 0.

<**sampnz(int)**> Description: Number of grid nodes in the  $z$ -direction.  
Legal values: Integer > 0.

## 5 The gridder

The gridder is modified so that it now assumes that Fluvial realisations are written on the new general parametric file format. The new grid programme is independent of the Fluvial programme code. It may be used to grid realisations from other programmes that use the same parametric file format.

### 5.1 Description of model parameters

**FILES** Has two (necessary) parameters:

**input-filename** The (Fluvial) parametric realisation file.

**output-filename** The grid realisation file containing the resulting grid from this programme.

**GRID** <nx> <ny> <nz>

The number of grid blocks in x-, y-, and z-direction respectively.

**ASCII** No parameters. If specified, the output grid file is on ascii format, otherwise binary.

**SUBVOLUME** Command specifying the simulation sub volume in world-coordinates to be gridded. The simulation sub volume must be contained in the simulation volume of the input parametric realisation. The command consists of the following parameters:

- *x0*
- dx*
- y0*
- dy*
- angle*
- top-surface-file* or a *constant*
- bottom-surface-file* or a *constant*
- top-erosion-file* or a *constant*
- bottom-erosion-file* or a *constant*
- dz*

## 6 The Fluvial programme

This extension of the Fluvial module has given two more options for the programme. Their names corresponds to the keyword necessary for the parameter file/model file:

- **RESTART** — Allows the program to start the simulation from any valid state for a parametric realization.
- **WELLRATE** — Allows the program to condition the simulations on one-phased steady-state well rates in well locations.

## 6.1 Description of model parameters

The necessary parameters for the commands are given in <angular-brackets>, while the optional parameters are given in [square-brackets]. If no brackets surround the parameter name, it means that parameter name should not be specified in the model file.

### RESTART

Status: Optional command.

Description: Has only one (necessary) parameter:

**filename** Any legal string without blanks or commas.

Note however that this parametric realization must be on the new format, designed in this project. Restarting from realizations on the former format will therefore not be possible. The new format will be easily distinguishable from the former by recognising that a new line has been inserted at the beginning of the file reading “parametric\_realization”.

### WELLRATE

Status: Optional command.

Description: Gives the necessary parameters for the rate conditioning. As this option has several records, of only the first is necessary, lead texts are necessary before the corresponding parameter(s):

1. <well-pressure-file(string)>

Description: The file has two or three columns: The first gives the well names as given in the well files, the second gives the steady-state pressures, and the third gives optional steady-state well rates in the well. The latter column is ignored if the RESTART option is used. In that case, the steady-state well rates are computed from the input realization/initial state.

Legal values: A legal text string.

2. [chn-perm(real)]

Description: Constant permeability of channel facies. Used for computation of steady-state well rates.

Legal values: Real number  $\geq 0.0$

Default values: In prioritised order; 100 times the background

permeability, 2 times the crevasse permeability, or 1 D if no permeability is set.

3. [crev-perm(real)]

Description: Constant permeability of crevasse facies. Used for computation of steady-state well rates.

Legal values: Real number  $\geq 0.0$

Default values: In prioritised order; half the channel permeability, 50 times the background permeability, or 0.5 D if no permeability is set.

4. [back-perm(real)]

Description: Constant permeability of background facies (shale) Used for computation of steady-state well rates.

Legal values: Real number  $\geq 0.0$

Default value: 1% of the channel permeability.

5. [ps-precision(real)]

Description: Convergence precision level for the iterative steady-state reservoir pressure equation solver that produces the well rates on which to condition. A too high precision will imply long cpu-consumption.

Legal values: Real number  $> 0.0$

Default value:  $10^{-6}$

6. [rate-variance([string/\*][real])]

Description: Parameter for controlling the variance in the rate conditioning. A string with a file name indicates a file including as many values as there are wells in the necessary well pressure file (see above). These values gives relative factors for the contribution from the wells to the total rate potential. An asterix, “\*”, indicates that an automatic relative factor is computed. A constant value gives an overall variance being multiplied in the total rate potential. Five possible combinations are valid:

**c** A constant. Implies using only the overall variance. This variance is used for the rate in all wells, meaning that the relative variances is larger in wells with low rates.

**file** A file giving the rate variance in each well.

**\*** Automatic scaling of the variance between wells. The computed factors are depending on their initial well rates, with extremal values 1 and 100, making the relative variances more equal, without giving small rates too much weight.

**file c** Giving both internal weighting of the contributions from the wells and an overall strength factor. This is equal to multiplying each value in the file with the constant, and specifying only the file name.

**\* c** Recommended. The  $c$  here is the variance of the well with the largest rate; the variances in other wells are scaled automatically.

Default value:  $c = 1$ .

7. [burn-in-time(int)]

Description: The iteration the program should start with the well rate conditioning. Default value is 1, which means that rate conditioning is made in the entire simulation. If the simulation is without the restart option, the default value most likely implies an undesirably long cpu time.

Legal values: Integer  $> 0$

Default value: 1

## 6.2 Parameter discussion

The different parameters should be chosen with care in order to limit the cpu consumption, and to get convergence in the simulations.

By including the rate conditioning, the cpu time will increase dramatically for each iteration. The rate conditioning is therefore generally not included before a good state is achieved. This will normally mean that a restart is made on a previously generated realisation known to be good.

Of all the model parameter set, we will concentrate the discussion on the most important ones for the “local updating” setting.

---

sampnx(int)	}	The internal sampling grid resolution. For previous versions of the Fluvial module, the internal sampling grid was used for sampling the sand/gross ratio and for holding the dimension of the seismic attribute grid. Now also the pressure equations from steady-state reservoir needed to compute the well rates are found on a grid of this size. The cpu consumption of the simulation will therefore perhaps be most dominated by the choice of grid resolution. It is important not to use a too coarse grid resolution, at least not coarser than the grid resolution of the reservoir simulator used at a later stage.
sampny(int)		
sampnz(int)		
ps-precision(real)		The error tolerance for the iterative pressure solver of the steady-state reservoir. By increasing the precision level from $10^{-6}$ (i.e. smaller error tolerance) the steady-state reservoir well rates are found with higher accuracy. This will however increase the cpu consumption. Lowering the precision could increase the simulation speed, but then again the results are more inaccurate.

min-temp(real) }  
start-temp(real) }

The annealing temperature. The temperature in iteration step  $k$  is  $T(k) = T_{start} \alpha^k$ , where the temperature rate  $\alpha = (T_{stop}/T_{start})^{-N}$  for the maximum number of iterations  $N$ . It is therefore a close relationship between setting the temperatures and the number of iterations.

If the simulation should do just a local updating, the starting temperature should be low, such that the simulation is kept near the initial state. If an uncertainty analysis should be made, the starting temperature should be high in order to allow for realisations further from the initial state.

A too fast annealing is not suggested. In the cases where few iterations are set, it is advised to let the start and stop temperatures be rather close. The default values are  $T_{stop} = 0.00001$  and  $T_{start} = 0.2$ . In the restart and local updating applications, it is most usual to lower the starting temperature considerably. Then the simulation will continue more similar to the behaviour before it terminated in the original simulation.

nIterations(int) }  
nMaxIterations(int) } The number of iterations. The choices here will depend on the initial state of the simulation. If a good starting reservoir is used through the RESTART option, very few iterations (say 100) could be sufficient. If rate conditioning is used, the number of iterations will depend on how far from the initial state the simulation should be allowed to go. For an investigation of the possible state space of the model, high numbers should be used, while low numbers are sufficient for just locally updating the reservoir due to new well data. Note that the choices will affect the annealing temperature, if these are not specified directly. Generally a high number of iterations could move the simulated realisation further from the initial state, but the possibility of reaching an even better state increases.

chn-perm(real) }  
crev-perm(real) }  
back-perm(real) } The constant permeabilities for the different facies in the steady-state reservoir. In order not to generate varying petrophysics, constant values are used in the pressure solving of the steady-state reservoir. Each facies should have a separate value. It is important to distinguish between the different facies such that the flow pattern will be dominated by the channels (and partly crevasses), but not setting values too far apart as this will introduce numerical instable pressure equations. At the extreme, the simulation could terminate with an error. Large differences will also normally increase the cpu consumption. It is advised not to separate channel and background with more than a factor 1000.



rate-variance(real)

The rate variance. Note that if this variance is too small, other components of the model may suffer. This can be investigated with the listing in the monitor-file, if this output is specified. A plot will reveal the rate potential importance in the total potential, and the rate variance can be adjusted accordingly.

The automatic computation of the distribution of variance between the wells can be included to increase the importance of wells where the steady-state rates are low. The weight for well  $i$  is found from the formula

$$w_i = 100 - 99 \exp \left( \frac{1 - (\max_i |\rho_{0,i}|)/|\rho_{0,i}|}{99} \right).$$

Note that  $w_i = 1/(2 * \Sigma_{\rho_{ii}})$  from equation (3). This formula is a trade-off between absolute and relative variances from the wells. The limitation is that the absolute variance should not vary more than with a factor 100 between wells. It is also possible to specify the weighting of the different wells in the rate potential. A file with rate variances for each well in the well rate conditioning must then be specified.

All the choices will be affected by the actual reservoir model. This is essential to be aware of especially when using the WELLRATE conditioning. Models with channels that are intersecting much and with high sand/gross will behave quite different than low sand/gross models with almost non-intersecting channels. Since the number of possibilities is so large, no general advice can be given, and most is left to trial and error. The user should play around with different parameter settings, to find something suitable for the reservoir and model in question.

## 7 Test Examples

A synthetic reservoir of area 3 by 11 km and thickness 50 meter was defined. It initially contained 13 vertical wells, of which 5 were injectors and the remaining 8 were producers. After a history match of this reservoir, a new well (X1) is included. The task is to update the reservoir description with the new data from well X1, which have conflicting channel observations with the reservoir description. The reservoir area is shown in Figure 2.

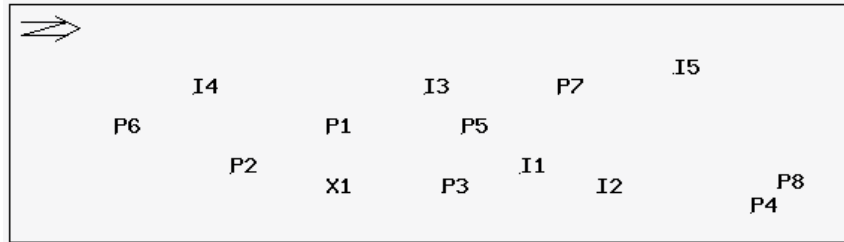


Figure 2: Well locations in the synthetic reservoir, which is rotated so that north is to the right. I1, I2, . . . , I5 are injectors; P1, P2, . . . , P8 are producers; while X1 is the new well.

Two different fluvial models were used while testing the new functionality of the Fluvial module of STORM. The same concept was used in both tests:

- Generate an unconditional realisation of the actual fluvial module.
- Extract facies data from a set of 14 vertical well locations (8 producers, named P1, . . . , P8, 5 injectors named I1, . . . , I5 and one new well X1).
- Reiterate further conditioned on 13 of the wells, to establish a realisation  $r_0$  that is used as a history matched realisation, and that has conflicting channel intersections with the well not used in the conditioning (X1).
- Repeat  $N$  times
  - A Generate a new simulation of the fluvial model conditioned only on the facies data from the 14 wells.

- B** Do a restart/local updating of the history matched realisation  $r_0$ , conditioned on the facies data from the 14 wells.
- C** Do a local updating of  $r_0$  conditioned on the 13 steady-state well rates of  $r_0$  and the facies data from the 14 wells.
- Assign permeability values to the two facies types, channel and background.
  - Run the reservoir simulator MORE on each realisation of each approach A, B and C.
  - Compute the water-cut and oil rate norms.
  - Plot the water-cut and oil rate to compare with the values of  $r_0$ .

## 7.1 First test reservoir

This reservoir was also studied in Skorstad et al. (2000). Table 1 summarises the most critical model parameters.

Sand/gross ratio	0.30
Average channel width (std.dev)	600 m (200 m)
Average channel thickness (std.dev)	6 m (2 m)
Std. dev. of horizontal amplitude	15 m
Average channel direction	75° from E-W parallel
Constant permeability in channels	100 mD
Constant permeability in shale	1 mD
Simulator grid size	60x50x20

Table 1: Some critical reservoir parameters for the synthetic reservoir.

For this reservoir, approach C was run twice, with a slight difference in the rate conditioning. It was run both with and without automatic rate factor computation, see Section 6.1. Both results are shown below. The number of independent simulations  $N$  was 20.

A monitoring of the potentials active in the posterior (1) gives information about the convergence of the simulation. Stability of the graphs will indicate

convergence of the potentials and hence convergence of the chain itself. In Figure 3, one simulation of approach C is shown.

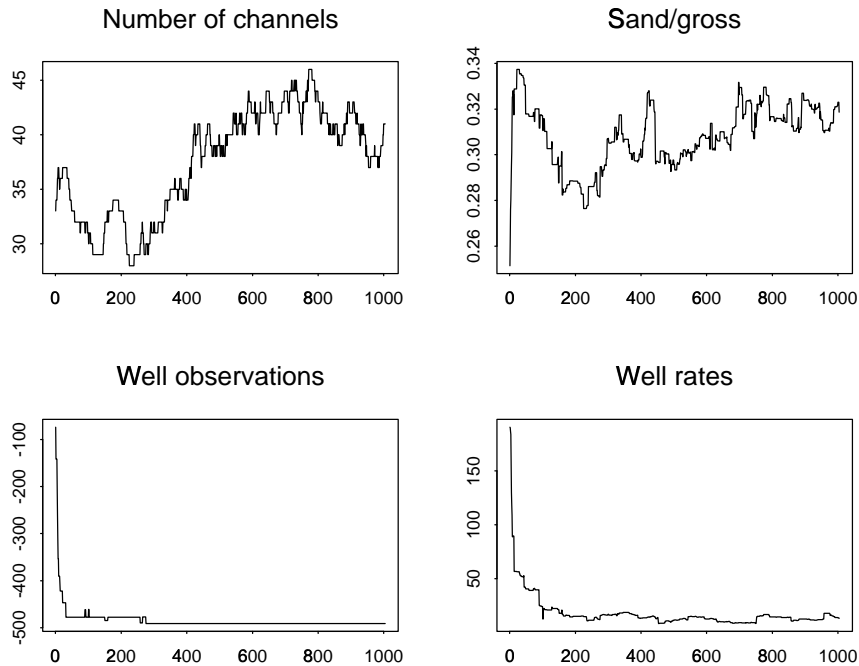


Figure 3: Factors in the model for approach C. For the number of channels and the sand/gross, the real values are given, while the potential is given for the well observation and well rate parts. The iteration number is along the abscissa, while the values/potentials are shown at the ordinate.

It is a typical behaviour seen in the figure. The original reservoir had 39 channels. At the beginning of the restarted simulation, some channels have been removed due to inconsistencies along the new well path. Since the well rate potential is high in the beginning, more channels are removed in order to meet the rate conditioning. As the observed channels are conditioned, new unobserved channels are included to meet the sand/gross requirement. These channels are however acceptable for the rate conditioning part. The simulation has been allowed to iterate a while at stable potentials, and can be viewed as converged to a state within the model.

The l2 norms for the different runs are found in Table 2.

Approach	Sum			Best		
	A	B	C	A	B	C
$\ P_{or}(r_0) - P_{or}(r)\  [10^6]$	153.9	119.8	56.2 (65.3)	3.82	2.25	1.01 (1.72)
$\ P_{wo}(r_0) - P_{wo}(r)\ $	1203	1164	941 (705.2)	10.50	4.71	7.67 (3.10)

Table 2: Simulation results for approaches A, B, and C comparing oil rate (*or*) and water cut (*wo*) at production wells. For approach C the values for running without automatic rate factor computation is given in parenthesis. The values in 'Sum' are summed l2 norms over all production wells and all 20 realisations, while the values in 'Best' are summed over all production wells for the realisation with minimum l2 norm.

Production data corresponding to the results in Table 2 are plotted in Figures 5 to 12 in Appendix A. The first columns correspond to approach A, the second to approach B, and the last two columns represent approach C.

The results show that C do give favourable results, both when comparing the overall l2 norm, and when comparing the best realisation of each approach. Both A and B seems more biased than C. Furthermore, A has more variation in the production than C. For B, the variations depend upon if the well is influenced or not by X1. Recall the main channel direction  $75^\circ$  and Figure 2. In wells influenced by X1 as well P2 in Figure 6, the variation is large, while in the opposite case the variation is low as in well P1 in Figure 5. It is also seen that a scaling of the rate contribution factors is favourable. Only the water cut norm for the best realisation shows a larger l2 error for the scaled C approach than for the non-scaled version. Note also that the values of the water cut norms are low compared to the oil rate norms.

When comparing the results from these three approaches, it should be taken into account that conditioning on well rates is time consuming. In our example, the facies simulations for approach C took about twice the time as one MORE simulation, while for approach A and B the computation time was negligible compared to the reservoir simulation. We used a simple bootstrapping algorithm for the comparison. A sample of given size was drawn from the population, and the l2 oil rate norm for the best realization in this sample was found. This was repeated, thus obtaining an average for the norm of the best realisation. This was done for different sample sizes.

For approach A and B, we used three times as large samples and populations as for approach C, thus compensating for the difference in CPU time. The results are shown in Figure 4. We see that approach C has the lowest l2 norm for all sample sizes. The total time for achieving a good new history

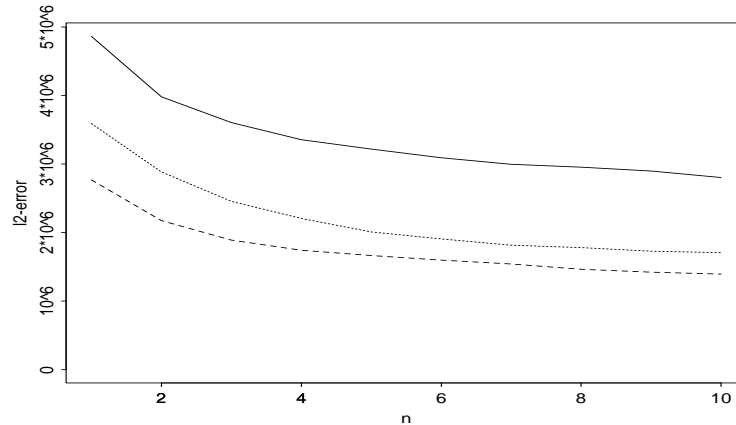


Figure 4: Comparable mean l2 oil rate norm for approach A, B and C for different sample sizes. Lines are solid (A), dotted (B) and dashed (C). Sample size is for approach C, the A and B samples are three times as large.

matched facies realisation is in this synthetic reservoir therefore more than accounted for.

## 7.2 Second test reservoir

Table 3 shows the model parameters that were changed from the first test reservoir. Here, the test was only run for  $N = 15$  simulations.

Average channel width (std.dev)	600 m (300 m)
Std. dev. of horisontal amplitude	20 m
Average channel direction	45° from E-W parallel

Table 3: Fluvial model parameters changed for the second test reservoir.

The l2 norms for the three approaches are found in Table 4.

Approach	Sum			Best		
	A	B	C	A	B	C
$\ P_{or}(r_0) - P_{or}(r)\  [10^6]$	737.94	737.79	717.13	31.96	38.42	28.27
$\ P_{wo}(r_0) - P_{wo}(r)\ $	1934	1571	1486	60.30	49.16	38.77

Table 4: Simulation results for approaches A, B, and C comparing oil rate (*or*) and water cut (*wo*) at production wells. The values in 'Sum' are summed l2 norms over all production wells and all 15 realisations, while the values in 'Best' are summed over all production wells for the realisation with minimum l2 norm.

Production data corresponding to the results in Table 2 are plotted from Figure 13 to Figure 20 in Appendix B. The first columns correspond to approach A, the second to approach B, and the last approach C.

In this second reservoir the rate conditioned local updating is only judged slightly better than the other approaches. Note however when comparing with the results from the first reservoir that the automatic rate potential scaling that showed to be the best approach for the first reservoir, was not used in this second reservoir. The water cut norms get better as the approach changes from the A to C, but this is not the case for the oil rate where A and B are equally good in the summed norm, but where the best realisation for A is better than B. It is also seen that the values of the norms are considerably larger than for the first reservoir. One major difference with the reservoirs is that this second has a main channel direction more skew to the elongation of the reservoir area. This means that the channels are seen shorter, and thereby an increase in the number of channels (from 39 to 54 in  $r_0$ ) and channel intersection complexity. This illustrates that the actual model of the reservoir is vital for the performance of the simulation. The model for the second reservoir was run only 30% as long as in the first reservoir, and the results implies that this was not long enough for reaching convergence within the desired level.

## 8 Conclusion

The results indicate that preservation of well rates is sufficiently correlated with preservation of production. Therefore better results than a traditional approach like A can be obtained by conditioning on the steady-state well rates. This is even true if we take into account that conditioning on well rates is time consuming. Updating a history matched reservoir by preserving well rates is, furthermore, a good alternative to condition directly on production data. In that case, a reservoir simulation had to be run for each Metropolis-Hastings iteration. For our example, assuming that we run the the same number of iterations as in approach C; this would take approximately 54 days instead of the three hours in approach C.

By implementing the possibility of restarting the simulation of a realisation, an important functionality has been added to the Fluvial module. The new realisation format has been designed to also meet the requirements from the GMPP module, such that they can have a common parametric file format.

## Acknowledgements

The authors wish to thank Norsk Hydro and Saga Petroleum for financing this work. Also, we thank Bjørn F. Nielsen for providing the pressure equation solver and Harald H. Soleng for assisting us with the flow simulations that were used to evaluate the presented method.

## References

- Holden, L., Hauge, R., Skare, Ø. & Skorstad, A. (1998), 'Modelling of fluvial reservoirs with object models', *Math. Geol.* **30**(5), 473–496.
- Skorstad, A., Hauge, R., Holden, L. & Skare, Ø. . (2000), 'Conditioning fluvial models on steady-state well rates'. 10 pages. To be presented at the Geostat2000 Congress Cape Town.



## A Production data plots — First Reservoir

Common for all plots are that the production data in the well are shown with the water cut (top) and oil rates (bottom) for approach A, B and C both without and with automatic rate factor computation (left to right). The bold lines indicates  $r_0$  while the dotted lines indicate the results of the 20 simulations of each approach.

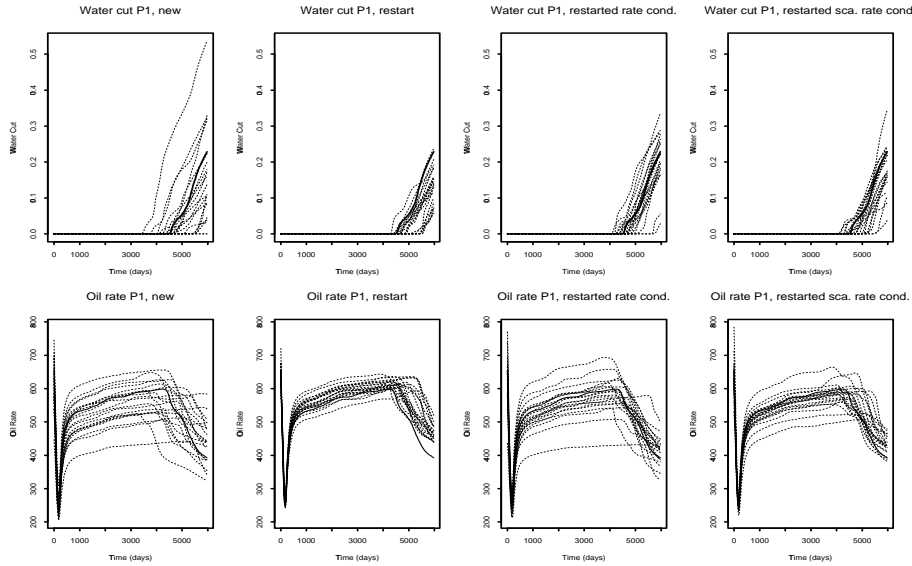


Figure 5: First reservoir: Production data for well P1.

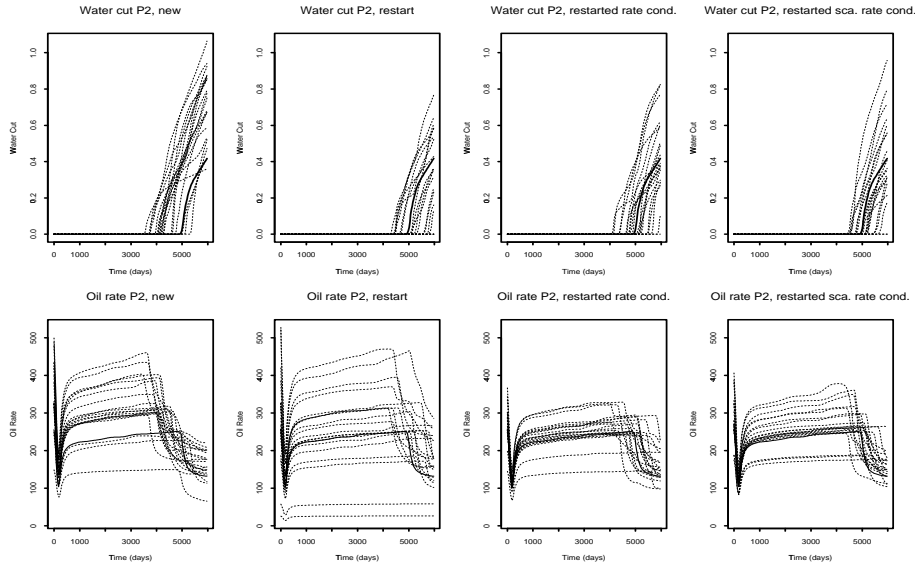


Figure 6: First reservoir: Production data for well P2.

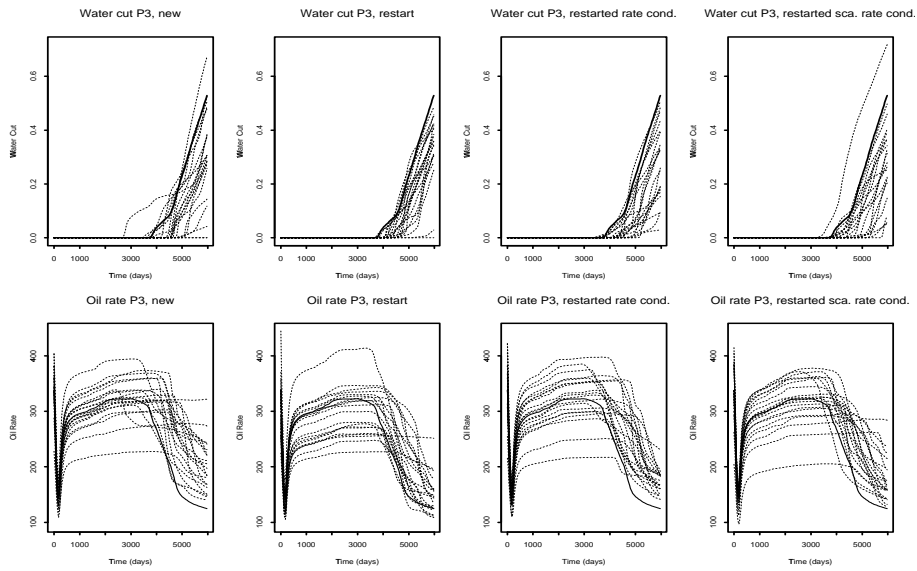


Figure 7: First reservoir: Production data for well P3.

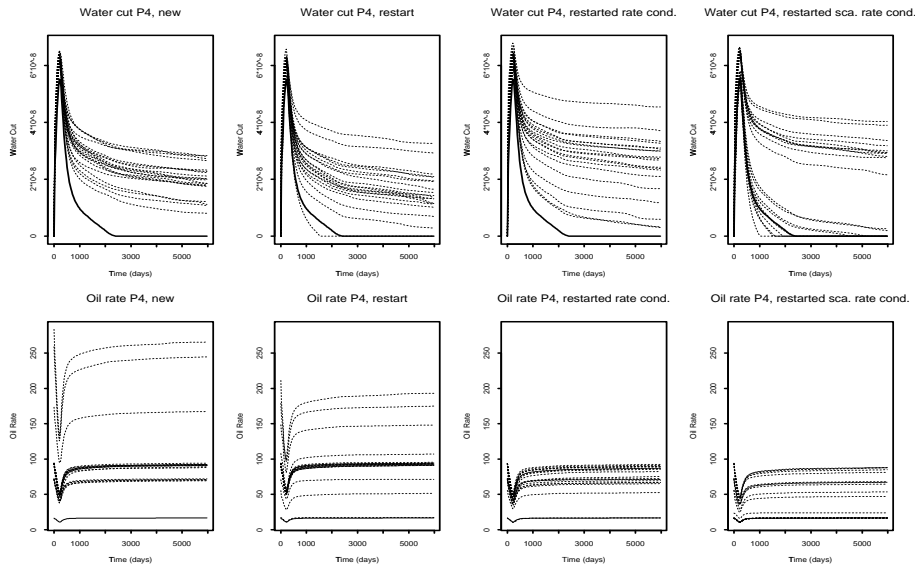


Figure 8: First reservoir: Production data for well P4.

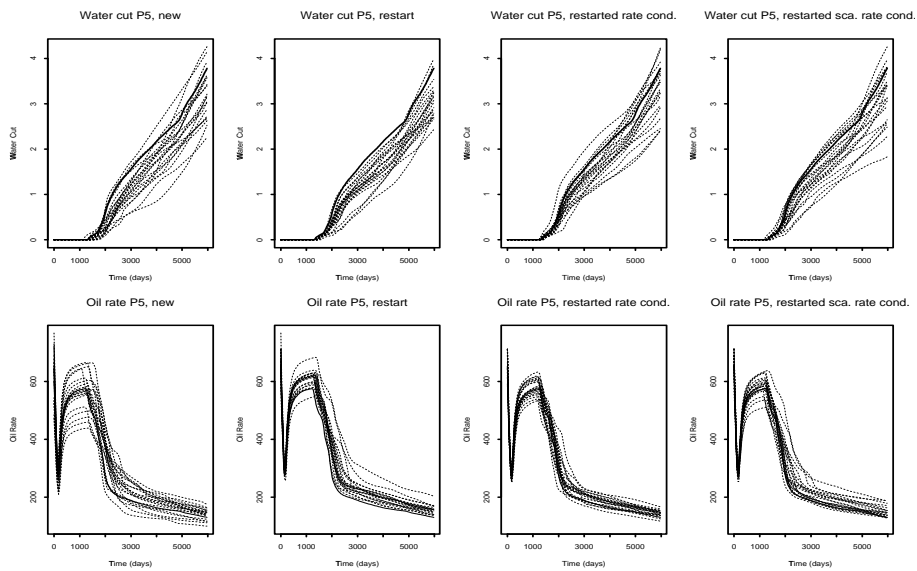


Figure 9: First reservoir: Production data for well P5.

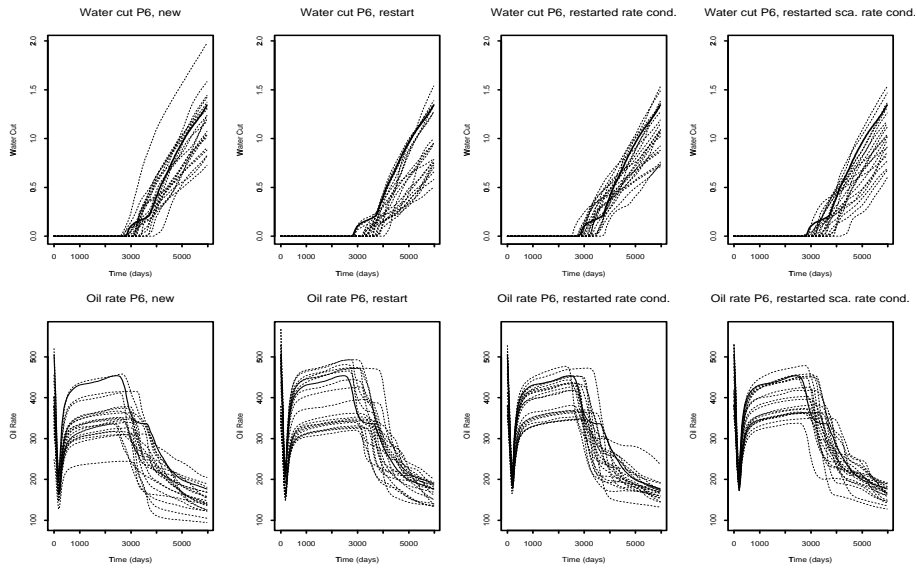


Figure 10: First reservoir: Production data for well P6.

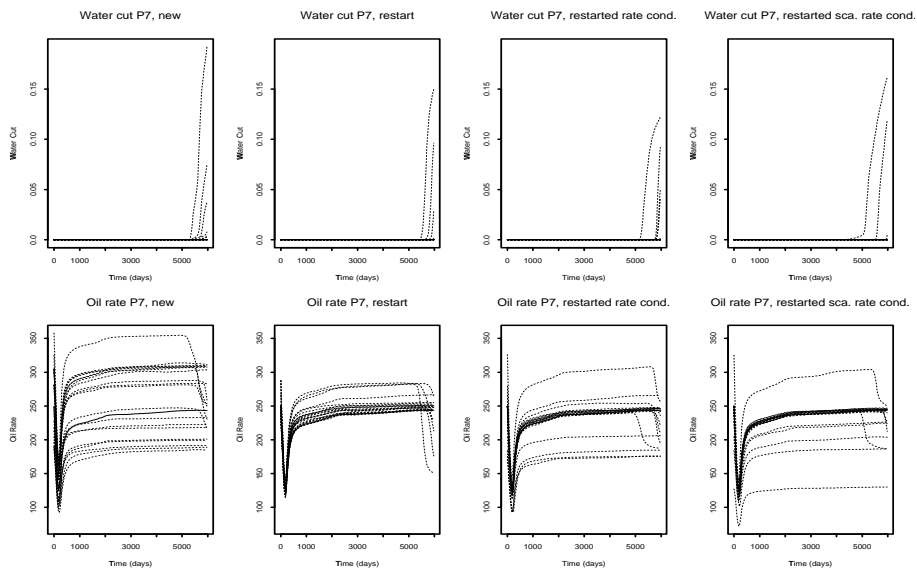


Figure 11: First reservoir: Production data for well P7.

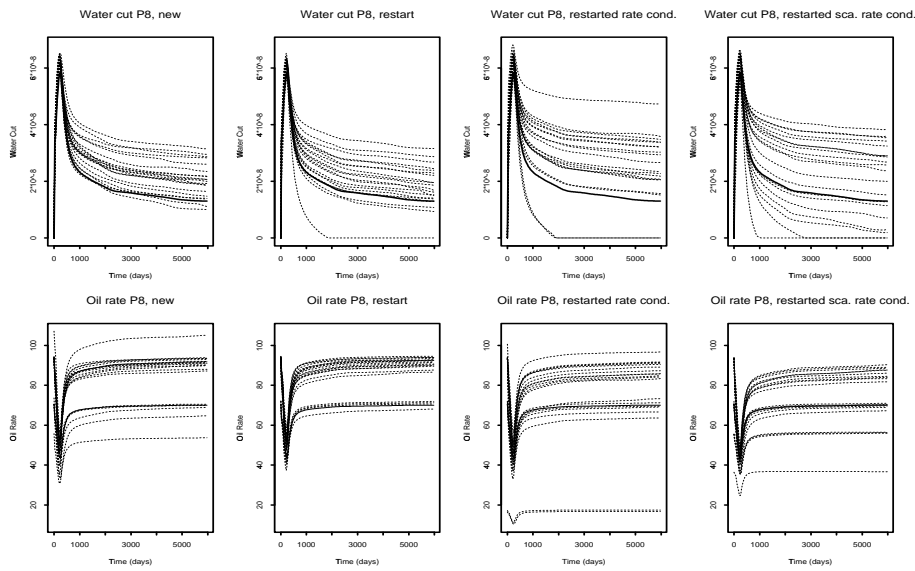


Figure 12: First reservoir: Production data for well P8.

## B Production data plots — Second Reservoir

Common for all plots are that the production data in the well are shown with the water cut (top) and oil rates (bottom) for approach A, B and C with automatic rate factor computation (left to right). The bold lines indicates  $r_0$  while the dotted lines indicate the results of the 20 simulations of each approach.

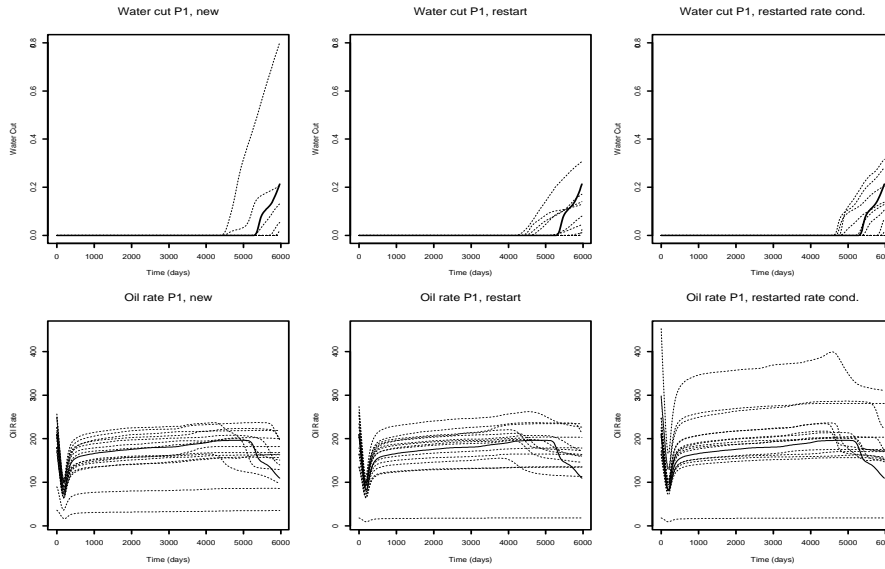


Figure 13: Second reservoir: Production data for well P1.

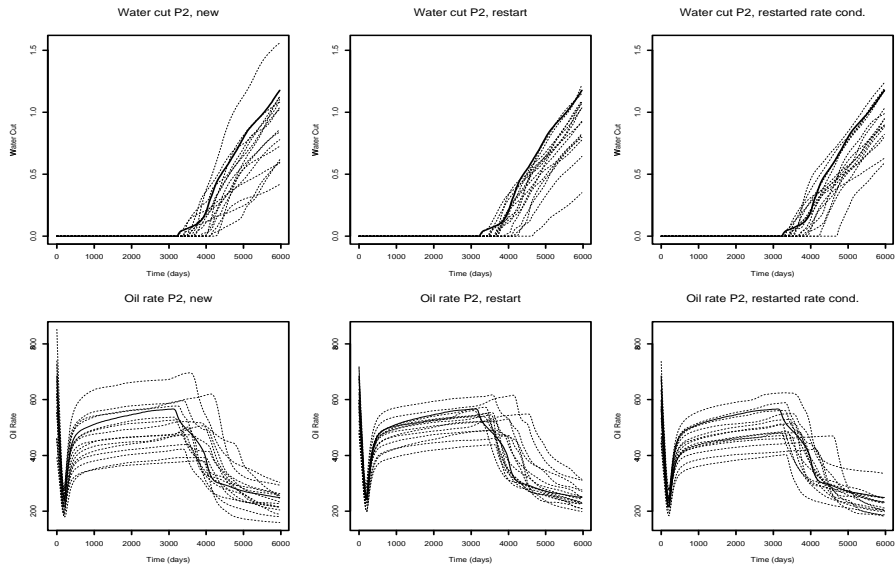


Figure 14: Second reservoir: Production data for well P2.

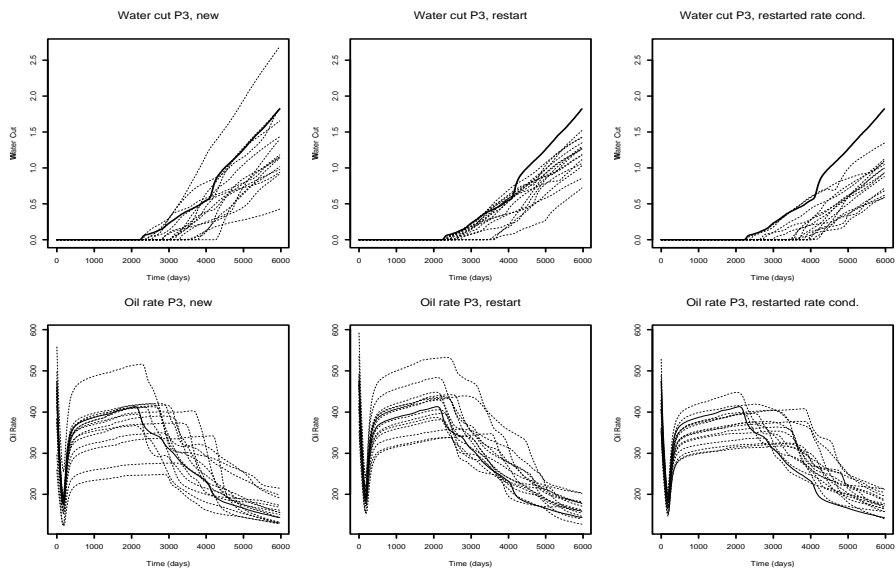


Figure 15: Second reservoir: Production data for well P3.

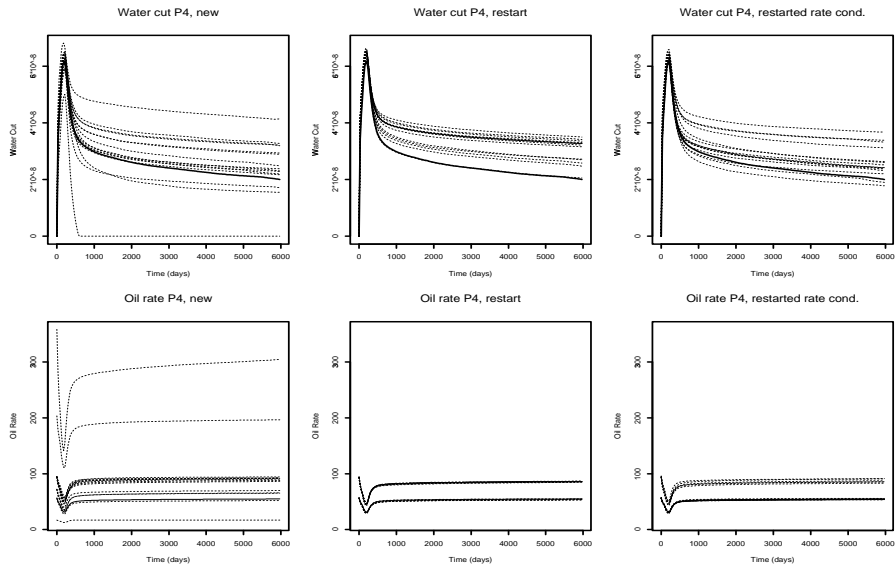


Figure 16: Second reservoir: Production data for well P4.

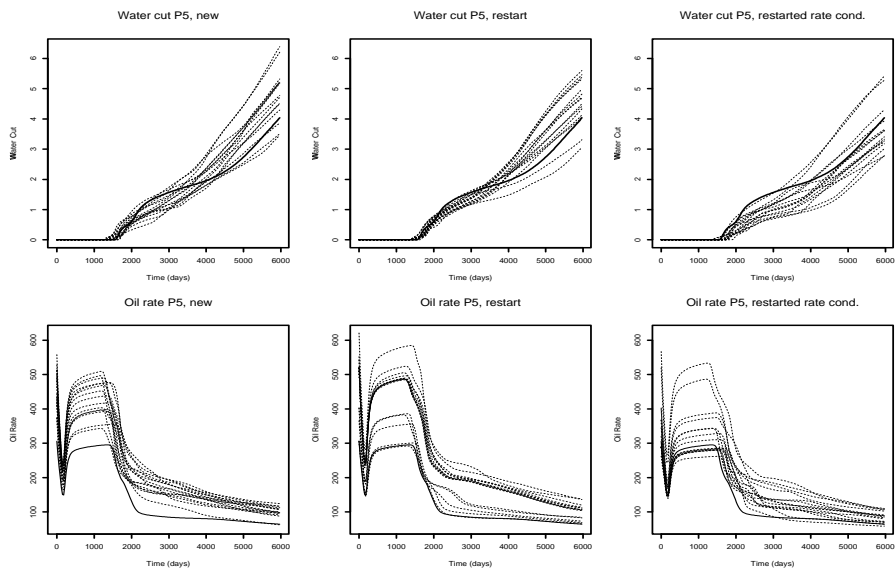


Figure 17: Second reservoir: Production data for well P5.



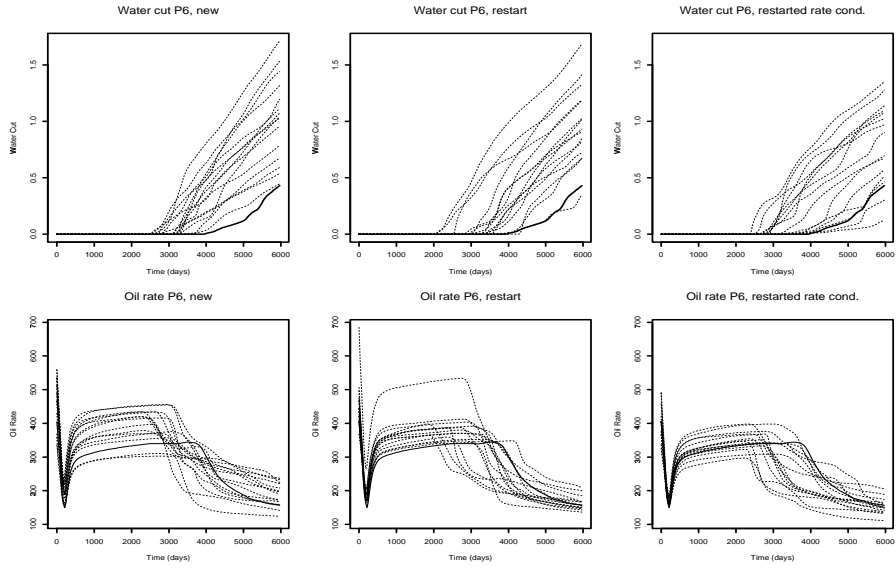


Figure 18: Second reservoir: Production data for well P6.

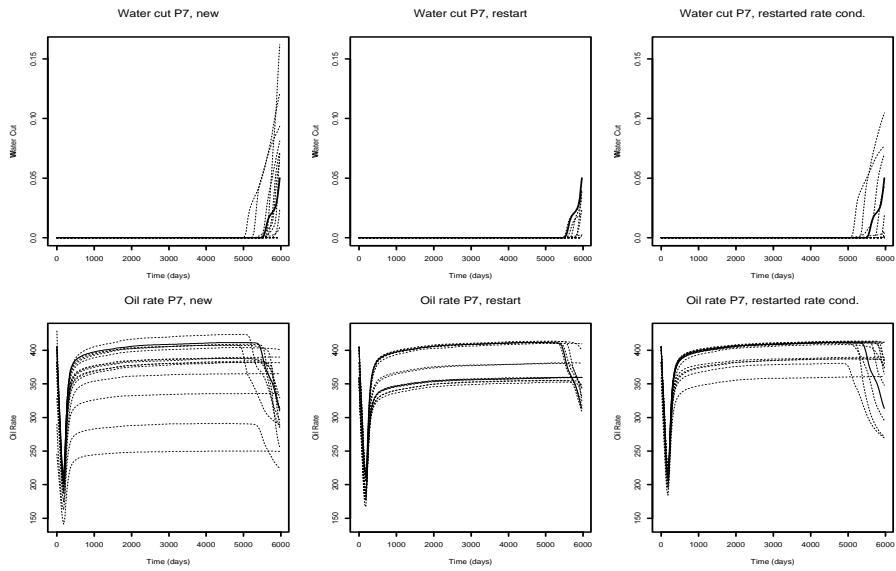


Figure 19: Second reservoir: Production data for well P7.

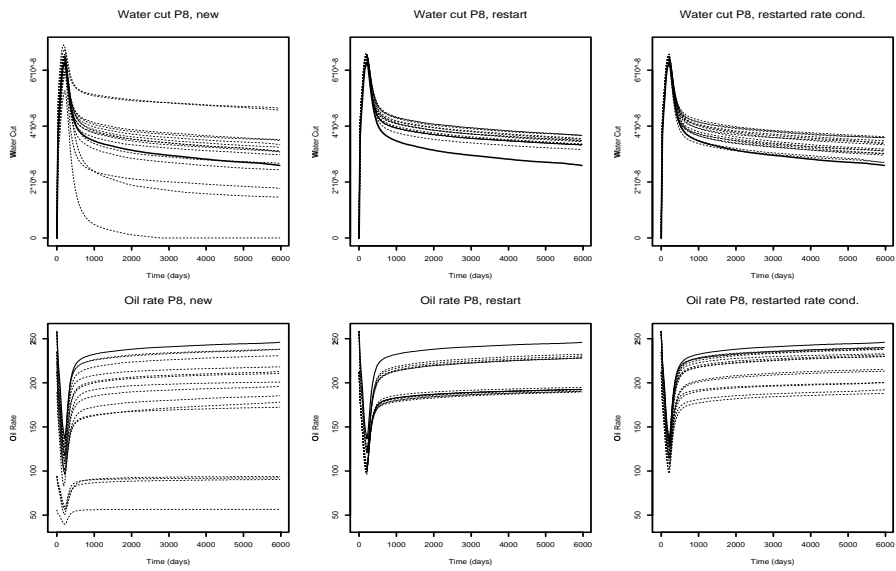


Figure 20: Second reservoir: Production data for well P8.

## C Model file extensions

The project has resulted in two new key-words for the model file. An example of their use is given below:

```
RESTART zoneA.fett ;
```

```
WELLRATE <well-pressure-file(string)> wellpress.dat
          [chn-perm(real)]           1.0
          [crev-perm(real)]          0.50
          [back-perm(real)]          0.01
          [ps-precision(real)]       0.0000003
          [rate-variance(real)]      1
          [burn-in-time(int)]        1
          ;
```

The pressure file `wellpress.dat` for 8 wells  $w_1, \dots, w_8$  could be defined as shown below:

```
w1 0.0 -1000.0
w2 0.0 -1000.0
w3 0.0 -1000.0
w4 0.0 -1000.0
w5 0.0 -1000.0
w6 1.0 1500.0
w7 1.0 1750.0
w8 1.0 1750.0
```

This defines three injectors ( $w_6, w_7$  and  $w_8$ ) and five producers. Note that the sum of the rates equals zero in a steady-state reservoir.

## D STORM REALISATION FORMAT

```

parametric_realisation
storm_<module> ! module is either fluvial or gmpp.
<version> <ascii>
<modelfile name> <zone number>
<Nf = number of facies>
<facies 1 name> <facies 1 number> <facies 2 name> ...<facies Nf
number>
! Note that this also define the erosion rule. <facies 1> erodes <facies
2> etc.
<xmin> <dx> <ymin> <dy> <top surface> <bottom surface> <top erosion>
<bottom erosion> <simulation box thickness> <angle from E-W>! vol-
ume data

Nw ! the number of well files
<well1> <well2> ...<wellNw> ! project relative paths

! Module spesific data:
<simulation-area-file>
<use2DFields>
<griddist>
<res-cosanghor> <res-cosangver> <res-sinanghor> <res-singangver>
<Nchn>
<xmin> <xmax> <ymin> <ymax> <zmin> <zmax>
<refpoint-x> <refpoint-y> <refpoint-z> ! Simulation box
<nWellFacies> <barrierFacies> ! #facies in wells, number for barrier
facies in well
<origFacNumber[1]> ...<origFacNumber[nWellFacies]> ! facies num-
bers in well
<combinedFacNumber[1]> ...<combinedFacNumber[nWellFacies]> ! fa-
cies numbers in Fluvial

<text> ! FAMILIES
<Nfam>
! for(i = 1; i <= Nfam; i++)
! {
<number> <gridstart> <gridend>
<line-A> <line-B> <line-C> <line-D>

```

```

<zmean>
<exphd[gridstart]> ...<exphd[gridend]>
<minchn> <maxchn>
<width[0]> <width[1]>
<thick> <axiscenter>
<sdhor> <sdver>
<expMeanTop[0]> <expMeanTop[1]>
<expMeanBot[0]> <expMeanBot[1]>
<covMean[0][0]> <covMean[0][1]> ...<covMean[1][1]>
<covMeanUD[0][0]> <covMeanUD[0][1]> ...<covMeanUD[1][1]>
<nchn> <nobschn>
<potential>
! }

```

```
<text>! MODULE_END
```

```

NO ! The total number of objects
! for(i = 1; i <= NO; i++)
! {
<object no i type>! This could be either axial or bitmap
<facies number (relates to facies list above)> <identification number>

```

```
! Module specific data:
```

```
! If facies == channel {
```

```

<nCrev[0]> <nCrev[1]> <nBarr>! LEFT RIGHT
<mean[0]> <mean[1]> <mean[2]> <mean[3]>!HD, VD, HW, VT
<sd[0]> <sd[1]> <sd[2]> <sd[3]>!HD, VD, HW, VT
<meancrev[0]> <meancrev[1]> <meancrev[2]>!WIDTH VERT THICK
<sdcrev[0]> <sdcrev[1]> <sdcrev[2]>!WIDTH VERT THICK
<minval[0]> <minval[1]> <minval[2]> <minval[3]>!HD, VD, HW, VT
<maxval[0]> <maxval[1]> <maxval[2]> <maxval[3]>!HD, VD, HW, VT
<number> <ncondpt>!number is internal channel id; currently not used.
<meanvd>
<condPtIndex[0]> ...<condPtIndex[ncondpt-1]>
<condpt[0]> ...<condpt[ncondpt-1]> (4 numbers per condpt)
<nWells>
<nTop>
<sim2DPointsTop[0]> ...<sim2DPointsTop[nTop]>
<nBot>
<sim2DPointsBot[0]> ...<sim2DPointsBot[nBot]>

```

```

<nQuilt>
!The quilts follow now, see details for quilt format below. !An empty line
shows the end of the last quilt.
<hor> <ver>
<distWell[0]> ...<distWell[nWells-1]>
! }
! else if facies == crevasse
<ncondpt> ! cond points for WIDTH VERT THICK:
<condPt [0] [0]> <condPt [0] [1]> <condPt [0] [2]>
:
<condPt [ncondpt-1] [0]> <condPt [ncondpt-1] [1]> <condPt [ncondpt-1] [2]>
!if use 2D-fields
! {
<x-length> <y-length> <x-increment> <y-increment>
! write 1D channel fields:
<NX> <1>
<MidpointY 1> ...<MidpointY NX>
<MidpointZ 1> <MidpointZ 2> ...<MidpointZ 1,NX>
<Width 1> ...<Width NX>
<Thickness 1> <Thickness 2> ...<Thickness NX>
! }

!If channel: observations in internal GW format
! {
Nc ! The number of conditioned well data used for this object.
<First well number (relates to well list above)> <facies code> <object
number (within this zone and facies)>
:
<Nc'th well number (relates to well list above)> <facies code> <object
number (within this zone and facies)>
! }

<text> ! MODULE_END

Nc ! The number of conditioned well data used for this object.
<First well number (relates to well list above)> <facies code> <object
number (within this zone and facies)>
:
<Nc'th well number (relates to well list above)> <facies code> <object

```

number (within this zone and facies)>

! If axial:

<xstart> <ystart> <zstart> <xend> <yend> <zend> ! Relative to simulation box

*NX* ! The number of parameterization sections along the object

*NY* ! The number of parameterization sections across the object

<MidpointY 1> ...<MidpointY *NX*>

<MidpointZ 1,1> <MidpointZ 1,2> ...<MidpointZ 1,*NY*> ...<MidpointZ *NX,NY*>

<Width 1> ...<Width *NX*>

<Thickness 1,1> <Thickness 1,2> ...<Thickness 1,*NY*> ...<Thickness *NX,NY*>

!Else if ellipse:

<centerpt.x> <centerpt.y> <centerpt.z> <axis.x> <axis.y> <axis.z>

! } next object

For realisations of the Fluvial module, the objects are written to file in the order: CHANNEL, BARRIER, CREVASSE. For each channel, its corresponding crevasses and barriers are given before a new channel object.

!Quilt format: (All rectangles are given yMin, yMax, zMin, zMax)

<nLists> <nCrevEdges> <xCoord>

<legalAreaLL> <legalAreaUR>

<forbiddenAreas[0].nItems>

<forbiddenAreas[0][0]> ...forbiddenAreas[0][*n*]>

:

<forbiddenAreas[*nLists*].nItems>

<forbiddenAreas[*nLists*][0]> ...forbiddenAreas[*nLists*][*n*]>

<edgeLines.nItems>

<edgeLines[0]> ...<edgeLines[*n*]>

<nCrevEdges>

<crevEdges[0][0]> <crevEdges[0][1]> ...<crevEdges[*n*][0]> <crevEdges[*n*][1]>