



TNO latest developments and planned contributions to OPM Flow

Artur Castiel

Eduardo Barros, Negar Khoshnevis, Paul Egberts, Peter Verveer

TNO innovation
for life

April 23, 2026

Recent Contributions

Grid-independent Wells

Developing OPM Adjoint Architecture

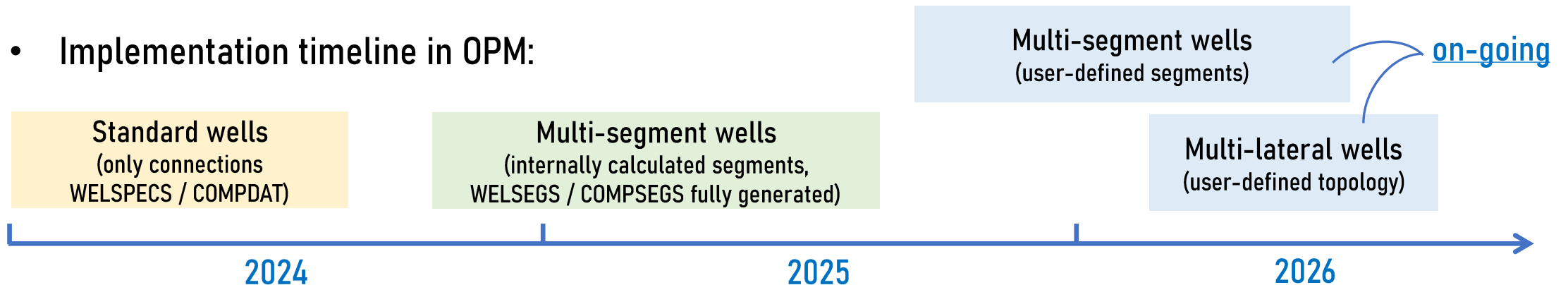
Thermal Brine

Local Grid Refinement

Grid-independent Wells

- Grid-dependent wells: defined by user-input well connections to reservoir grid (I, J, K)
- Grid-independent wells: defined by user-input well coordinates (x, y, z)
- Typical workflow for grid-dependent wells:
 - Load reservoir model in interactive geological modelling software (Petrel / RMS / ResInsight)
 - Load well paths (x, y, TVD, MD information available in imported .dev / well survey files)
 - Export connections, etc. (INCLUDE files containing WELSPECS / COMPDAT)
- Why grid-independent wells?
 - Better suited for **automated workflows**: computer-assisted history matching / uncertainty quantification / optimization

Implementation timeline in OPM:



Grid Independent Wells

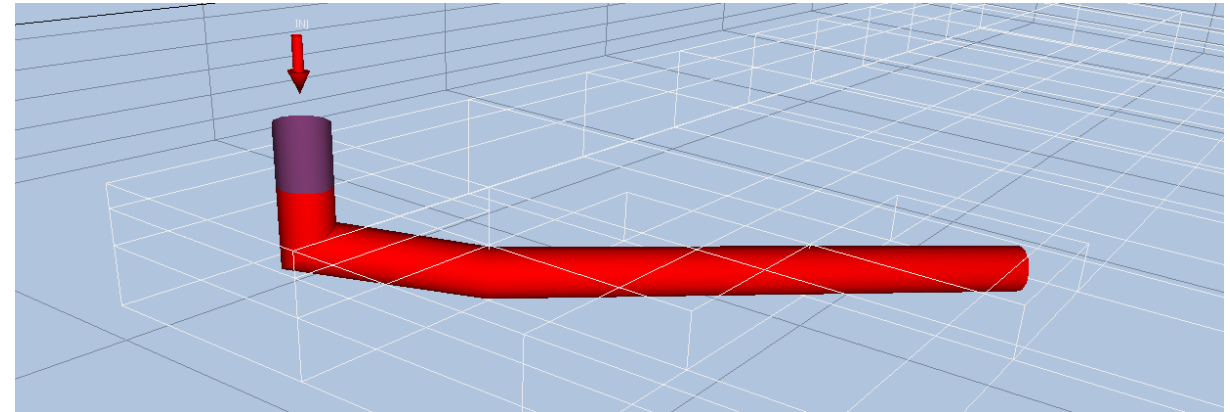
WELTRAJ/COMPTRAJ keywords

WELTRAJ

```
-- WELNAME  IBRANCH  XCORD  YCORD  TVD  MD
'INJ'      1*      10400  11600  8325.0  8325.0  /
'INJ'      1*      10400  11600  8425.0  8425.0  /
'PROD'     1*      3400.00  4500.00  8325.00  8325.00  /
'PROD'     1*      3858.19  4688.67  8374.60  8825.00  /
'PROD'     1*      5326.76  6005.95  8403.97  10825.00  /
'PROD'     1*      6500.00  7300.00  8410.00  12571.74  /
/
```

COMPTRAJ

```
-- WELNAME  IBRANCH  TOP  BOT  REF  ICOMP  STATUS  SATNUM  CONFACT  DW  KH  SKIN  D_FACT
'INJ'      1*      8325.0  8425.0  1*  1*  1*  1*  1*  0.5  1*  0.1  1*  /
'PROD'     1*      8325.00  12571.74  1*  1*  1*  1*  1*  0.5  1*  0.1  1*  /
/
```



- To enable multi-segment well (i.e., pressure drop along wellbore) more information is needed:
 - Segment information is typically prescribed via WELSEGS and COMPSEGS keywords
 - COMPSEGS keyword requires users to input well connections → **grid-dependent**
 - **How to support grid-independent multi-segment wells?**

Grid Independent Wells

Multi-Segment Wells generated by [ResInsight](#) (one segment per cell) can now be replicated in [OPM Flow](#):

```

WELSEGS
-- Name      Dep 1      Tlen 1      Vol 1      Len&Dep      PresDrop
|  PROD      8325.00000  8325.00000  1*         ABS          'HF-' /
-- First Seg  Last Seg  Branch Num  Outlet Seg  Length      Depth Change  Diam      Rough
|  2          2          1           1           8425.80645  8335.00000    0.50000  0.0000000 /
|  3          3          1           2           8689.42579  8361.15104    0.50000  0.0000000 /
|  4          4          1           3           8935.18266  8376.21803    0.50000  0.0000000 /
|  5          5          1           4           9157.90639  8379.48873    0.50000  0.0000000 /
|  6          6          1           5           9838.84091  8389.48825    0.50000  0.0000000 /
|  7          7          1           6           10597.98095 8400.63623    0.50000  0.0000000 /
|  8          8          1           7           11321.64879 8405.68450    0.50000  0.0000000 /
|  9          9          1           8           11997.06205 8408.01613    0.50000  0.0000000 /
| 10         10         1           9           12369.26637 8409.30103    0.50000  0.0000000 /
/

COMPSEGS
-- Name
|  PROD /
-- I      J      K      Branch no  Start Length  End Length  Dir Pen  End Range  Connection Depth
|  2      2      1      1           8325.00000    8526.61290  /
|  2      2      2      1           8526.61290    8852.23868  /
|  2      2      3      1           8852.23868    9018.12665  /
|  3      2      3      1           9018.12665    9297.68614  /
|  3      3      3      1           9297.68614    10379.99568 /
|  4      3      3      1           10379.99568   10815.96623 /
|  4      4      3      1           10815.96623   11827.33135 /
|  5      4      3      1           11827.33135   12166.79274 /
|  5      5      3      1           12166.79274   12571.74000 /
/
    
```

TNO latest developments and planned contributions to OPM Flow



Grid Independent Wells

Multi-Segment Wells generated by **ResInsight** (one segment per cell) can now be replicated in **OPM Flow**:

OPM Flow input: blank WELSEGS with first segment required
(without WELSEGS, OPM Flow will handle wells as standard wells)

```

WELSEGS
| INJ 8325.00000 8325.00000 1* ABS
/
WELSEGS
| PROD 8325.00000 8325.00000 1* ABS
/
WELTRAJ
-- WELNAME IBRANCH XCORD YCORD TVD
'INJ' 1* 10400 11600 832
'INJ' 1* 10400 11600 842
'PROD' 1* 3400.00 4500.00 832
'PROD' 1* 3858.19 4688.67 837
'PROD' 1* 5326.76 6005.95 840
'PROD' 1* 6500.00 7300.00 841
/
COMPTRAJ
-- WELNAME IBRANCH TOP BOT REF
'INJ' 1* 8325.0 8425.0 1*
'PROD' 1* 8325.00 12571.74 1*
    
```

WELLNAME	SEG	BRN	MAIN	INLET	OUTLET	SEGMENT	TOT LENGTH	DEPTH	T.V. DEPTH	DIA OR F	VFP TAB OR	AREA	VOLUME	P DROP
:SEG TYPE:	AND NO	NO	NO	NO	NO	LENGTH	TO END	CHANGE	AT END	SCALING	ABS ROUGHN	X-SECTN	MULT	
:	:	:	:	:	:	METRES	METRES	METRES	METRES	METRES	METRES	M**2	M3	FACTOR 1
:PROD	: 1	: 1	: 2	: 0	: 8325.0	: 8325.0	: 8325.0	: 8325.0	: 8325.0	: 0	: 0	: 0	: 0.000	: 1.000
:	: HF-	: 2	: 3	: 1	: 100.80	: 8425.8	: 10.000	: 8335.0	: 0.5000	: 0.000000	: 0.01824	: 3.525	: 1.000	
:	:	: 3	: 4	: 2	: 263.61	: 8689.4	: 26.151	: 8361.1	: 0.5000	: 0.000000	: 0.01824	: 9.219	: 1.000	
:	:	: 4	: 5	: 3	: 245.75	: 8935.1	: 15.066	: 8376.2	: 0.5000	: 0.000000	: 0.01824	: 8.594	: 1.000	
:	:	: 5	: 6	: 4	: 222.72	: 9157.9	: 3.2706	: 8379.4	: 0.5000	: 0.000000	: 0.01824	: 7.788	: 1.000	
:	:	: 6	: 7	: 5	: 680.93	: 9838.8	: 9.9995	: 8389.4	: 0.5000	: 0.000000	: 0.01824	: 23.81	: 1.000	
:	:	: 7	: 8	: 6	: 759.14	: 10597.	: 11.147	: 8400.6	: 0.5000	: 0.000000	: 0.01824	: 26.54	: 1.000	
:	:	: 8	: 9	: 7	: 723.66	: 11321.	: 5.0482	: 8405.6	: 0.5000	: 0.000000	: 0.01824	: 25.30	: 1.000	
:	:	: 9	: 10	: 8	: 675.41	: 11997.	: 2.3316	: 8408.0	: 0.5000	: 0.000000	: 0.01824	: 23.62	: 1.000	
:	:	: 10	: 0	: 9	: 372.20	: 12369.	: 1.2849	: 8409.3	: 0.5000	: 0.000000	: 0.01824	: 13.01	: 1.000	

1: The pressure drop multiplier is not implemented in opm/flow and will always show the default value 1.0.

MULTI-SEGMENT WELL: CONNECTION DATA

WELL NAME	CONNECTION	SEGMENT NUMBER	BRANCH ID	TUB LENGTH START	TUB LENGTH END	TUB LENGTH CENTR	TUB LENGTH END SEGMT	CONNECTION DEPTH	SEGMENT DEPTH	GRID BLOCK DEPTH
:	:	:	:	PERFS METRES	PERFS METRES	PERFS METRES	PERFS METRES	METRES	METRES	METRES
: PROD	: 2,	2,	1:	2	: 1	: 8325.0	: 8526.6	: 8425.8	: 8425.8	: 8335.0
:	: 2,	2,	2:	3	: 1	: 8526.6	: 8852.2	: 8689.4	: 8689.4	: 8361.1
:	: 2,	2,	3:	4	: 1	: 8852.2	: 9018.1	: 8935.1	: 8935.1	: 8376.2
:	: 3,	2,	3:	5	: 1	: 9018.1	: 9297.6	: 9157.9	: 9157.9	: 8379.4
:	: 3,	3,	3:	6	: 1	: 9297.6	: 10379.	: 9838.8	: 9838.8	: 8389.4
:	: 4,	3,	3:	7	: 1	: 10379.	: 10815.	: 10597.	: 10597.	: 8400.6
:	: 4,	4,	3:	8	: 1	: 10815.	: 11827.	: 11321.	: 11321.	: 8405.6
:	: 5,	4,	3:	9	: 1	: 11827.	: 12166.	: 11997.	: 11997.	: 8408.0
:	: 5,	5,	3:	10	: 1	: 12166.	: 12571.	: 12369.	: 12369.	: 8409.3

Grid Independent Wells

Multi-Segment Wells generated by WELTRAJ/COMPTRAJ:

Status:

- Activate multi-segments if the WELSEGS keyword is present
- Auto-generate a single segment for each cell traversed by the well
- Segments are assumed to be on the main branch
- User-defined segments not allowed yet (work in progress)

Next steps:

- Fully support WELSEGS keyword:
 - User-defined segments
 - Multi-lateral wells
 - Multiple branches
- Support multi-lateral wells with auto-generated segments and multiple branches
- Export COMPDAT/WELSEGS/COMPSEGS keywords for comparison with other compatible simulators

Grid-independent wells with:

- User-defined segments
- User-defined topology (multi-lateral wells)

From blank WELSEGS to user-defined WELSEGS

```

WELSEGS
| INJ  8325.00000  8325.00000  1*  ABS  'HF-'  /
|
|
WELSEGS
| PROD 8325.00000  8325.00000  1*  ABS  'HF-'  /
|
|

WELTRAJ
-- WELNAME  IBRANCH  XCORD  YCORD  TVD  MD
| 'INJ'      1*      10400  11600  8325.0  8325.0  /
| 'INJ'      1*      10400  11600  8425.0  8425.0  /
| 'PROD'     1*      3400.00  4500.00  8325.00  8325.00  /
| 'PROD'     1*      3858.19  4688.67  8374.60  8825.00  /
| 'PROD'     1*      5326.76  6005.95  8403.97  10825.00  /
| 'PROD'     1*      6500.00  7300.00  8410.00  12571.74  /
|

COMPTRAJ
-- WELNAME  IBRANCH  TOP  BOT  REF  ICOMP  STATUS  SATNUM  CONFACT  DW  KH  SKIN  D_FACT
| 'INJ'      1*      8325.0  8425.0  1*  1*  1*  1*  1*  0.5  1*  0.1  1*  /
| 'PROD'     1*      8325.00  12571.74  1*  1*  1*  1*  1*  0.5  1*  0.1  1*  /
    
```

Recent Contributions

Grid-independent Wells

Developing OPM Adjoint Architecture

Thermal Brine

Local Grid Refinement

Developing OPM Adjoint Architecture

- Applications of adjoint method:
 - Life-cycle optimization (focus in this presentation)
 - History matching
- Life-cycle optimization:
 - **how to operate wells so to maximise a lifecycle value J like NPV**
- Adjoint based lifecycle optimisation: gradient based optimization of NPV with gradients efficiently calculated by the adjoint method
- First thoughts on implementation of adjoint in OPM
 - **22-page technical report** describing conceptual implementation approach has been produced
It can be shared for upcoming discussions in recently started **makeSENSE project**
 - **In this presentation: a brief walkthrough the main ingredients of this report**

Developing OPM Adjoint Architecture

Review on Adjoint

- Reservoir simulator: $F(x, u) = 0$ (1)
 - x : state vector (grid block pressures, saturations, ...)
 - u : control vector (BHP, rate, ...)
- Solving Eq. (1) gives $x = x(u)$
- Objective function $J = J(x, u) = J(x(u), u)$
- Optimization problem: $\max_u J(u)$
 - Gradient based optimization requires dJ/du
 - $\frac{dJ}{du} = \frac{dJ(x(u), u)}{du} = J_x \frac{dx}{du} + J_u$
 - Eq. (1): $F_x \frac{dx}{du} + F_u = 0$ so that $\frac{dx}{du} = -F_x^{-1} F_u$
- Adjoint method: solve λ from $F_x^T \lambda^T = J_x^T$ and calculate $\frac{dJ}{du} = -\lambda F_u + J_u$

$$\left. \begin{array}{l} \\ \\ \\ \end{array} \right\} \begin{array}{l} \text{Forward method: expensive} \\ \frac{dJ}{du} = -J_x \underbrace{F_x^{-1} F_u} + J_u \\ \text{Adjoint method: cheap} \end{array}$$

Developing OPM Adjoint Architecture

Forward vs Adjoint Method

Forward (Direct) Method

$$F_x S = -F_u \Rightarrow S = -F_x^{-1} F_u$$

→ requires m linear solves
(one per control variable)

⚠ Prohibitive when $n_w \times N$ is large

Adjoint Method ✓

$$F_x^T \lambda^T = J_x^T$$

→ only 1 linear solve
independent of m

✓ Gradient cost ≈ 1 forward simulation

Key Result:

$$\frac{dJ}{du} = -\lambda F_u + J_u$$

Developing OPM Adjoint Architecture

General Mathematical Model

General conservation equation: mass, energy, or momentum. For simplicity, we use mass conservation

Conservation Equation (per phase α):

$$\left[\frac{\partial(\phi \rho_{\alpha} S_{\alpha})}{\partial t} \right]^{accum} + [\nabla \cdot (\rho_{\alpha} \vec{v}_{\alpha})]^{flux} = [\rho_{\alpha} q_{\alpha}]^{source}$$

Symbol	Meaning
ϕ	Porosity
$\rho_{\alpha}, S_{\alpha}$	Density & saturation, phase α
v_{α}	Darcy flux velocity
λ_{α}	Phase mobility $k_{r\alpha} / \mu_{\alpha}$
K	Permeability tensor
ψ_{α}	Phase potential

Darcy's Law

$$\vec{v}_{\alpha} = -\lambda_{\alpha} \bar{K} \nabla \psi_{\alpha}, \quad \lambda_{\alpha} = \frac{k_{r\alpha}}{\mu_{\alpha}}$$

Potential

$$\psi_{\alpha} = p_{\alpha} - \rho_{\alpha} g z$$

Saturation Closure

$$\sum_{\alpha} S_{\alpha} = 1$$

Developing OPM Adjoint Architecture

Discrete Equations

Reservoir Residual

$$F_{R,k,\alpha}^{n+1} = |\Omega_k| \rho_\alpha q_\alpha - \phi_k |\Omega_k| \frac{(\rho_\alpha S_\alpha)^{n+1} - (\rho_\alpha S_\alpha)^n}{\Delta t} - \sum_{\Gamma_j} [\rho_\alpha (\vec{v} \cdot \vec{N})]_{\Gamma_j} = 0$$

Peaceman Source Term

$$q_{\alpha,k}^{\text{well}} = \lambda_{\alpha,k} WI_k (p_w - p_k)$$

x_w = wellbore pressure p_w

x_r = cell pressure & saturation

WI_k = well index (Peaceman)

Well Equations F_w

BHP control:

$$F_{W,w} = p_w - u_w = 0$$

→ $A_{ww} = I$

Rate control:

$$F_{W,w} = \sum_{\alpha} Q_{\alpha,w} - u_w = 0$$

→ A_{ww} diagonal

Developing OPM Adjoint Architecture

Block Jacobian Structure

Current-step Jacobian $A^k = \partial f^k / \partial x^k$

$$A^k = \begin{pmatrix} A_{RR} & A_{RW} \\ A_{WR} & A_{WW} \end{pmatrix}^k$$

Previous-step $B^k = \partial f^k / \partial x^{k-1}$

$$B^k = \begin{pmatrix} B_{RR} & 0 \\ 0 & 0 \end{pmatrix}^k$$

Control derivative D^k

$$D^k = \begin{pmatrix} 0 \\ -I_{n_w} \end{pmatrix}$$

- A_{RR} Reservoir Jacobian: accumulation + flux + Peaceman x_r -dependence
- A_{RW} $\frac{\partial F_R}{\partial x_W}$: reservoir coupling to p_w — nonzero at perforated cells
- A_{WR} $\frac{\partial F_W}{\partial x_R}$: zero (BHP) | nonzero via $\frac{\partial q}{\partial p}, \frac{\partial q}{\partial S_\alpha}$ (rate ctrl)
- A_{WW} = I (BHP ctrl) | diagonal $\Sigma \lambda \alpha W I$ (rate ctrl)
- B_{RR} $\partial F_R / \partial x^{k-1}$: block-diagonal, links step k to $k-1$
- D_{RW} $\frac{\partial F_W}{\partial u} = -I$ regardless of control mode

Developing OPM Adjoint Architecture

How does the adjoint method look like for a reservoir simulator?

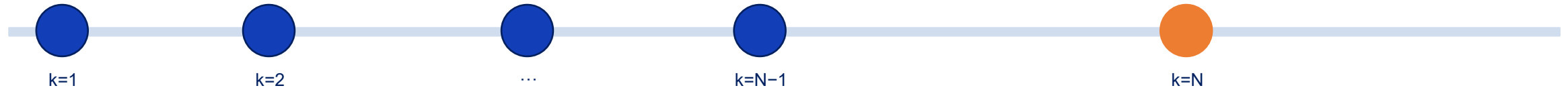
- Simulation time stepping
 - time stepping: $t^0, \dots, t^{k-1}, t^k, \dots, t^N$
 - start with initial state x^0 at t^0
 - time step $t^{k-1} \rightarrow t^k$ gives state transition $x^{k-1} \rightarrow x^k$
 - This requires specification of (well) control u^k for this timestep
 - discretized equations: $f^k(x^{k-1}, x^k, u^k) = 0, k = 1, \dots, N$
- Adjoint equation:
 - $\lambda^k A^k + \lambda^{k+1} B^{k+1} = \frac{\partial J}{\partial x^k}, k = 1, \dots, N - 1$
 - $\lambda^N A^N = \frac{\partial J}{\partial x^N}$
- Gradient: $\frac{dJ}{du^k} = -\lambda^k D^k + J_{u^k}$

$$A^k = \frac{\partial f^k}{\partial x^k}, B^k = \frac{\partial f^k}{\partial x^{k-1}}, D^k = \frac{\partial f^k}{\partial u^k}$$

Developing OPM Adjoint Architecture

Adjoint Backward Sweep

← *Backward adjoint sweep*



Terminal step $k = N$

$$\lambda^N A^N = \frac{\partial J}{\partial x^N}$$

Interior steps $k < N$

$$\lambda^k A^k = -\lambda^{k+1} B^{k+1} \frac{\partial J}{\partial x^k}$$

Per-step Gradient:

$$\frac{dJ}{du^k} = \lambda^k + J_{u^k}$$

Developing OPM Adjoint Architecture

Objective Functions:

Lifecycle optimisation: J is a time integral of weighted injection/production rates

General form:

$$J = \sum_{k=1}^N \theta^k(t_k) \Delta t_k$$

with

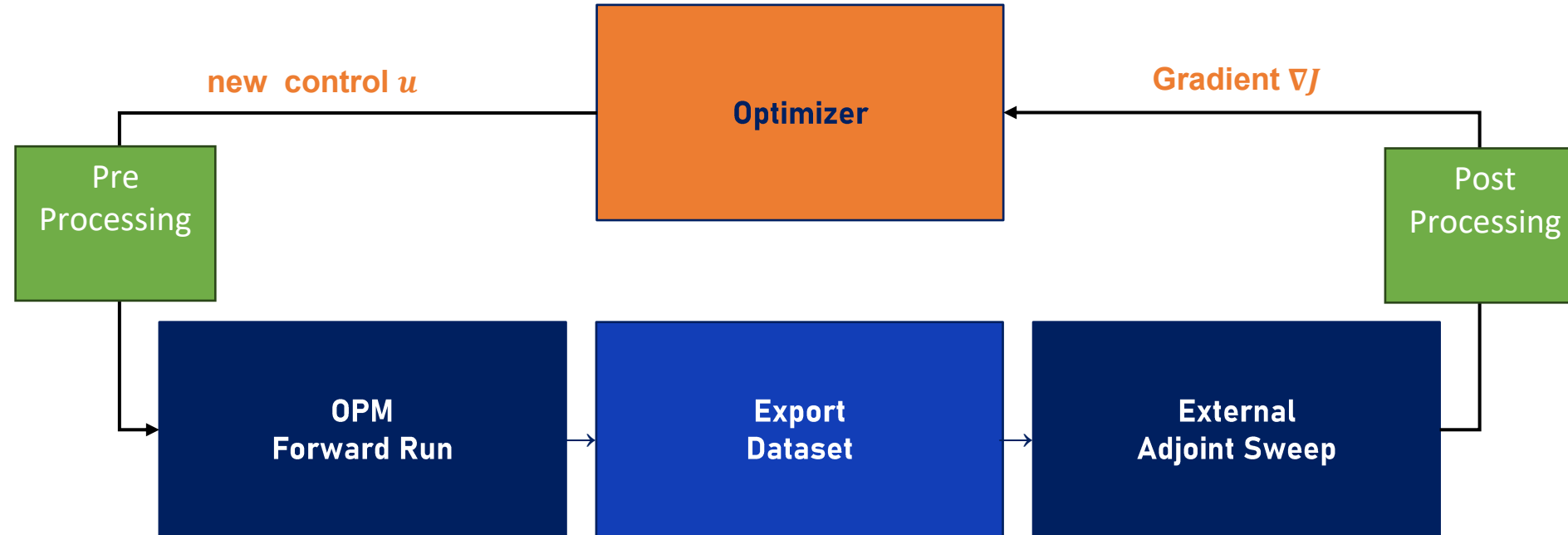
$$\theta^k = \sum_{well, phase} \omega_{phase}^{well}(t_k) Q_{phase}^{well}(t_k)$$

For appropriate choices of weight factors ω_{phase}^{well} the objective function J can be

- (discounted) NPV,
- Cumulative oil produced or
- Recovery factor (RF)

Developing OPM Adjoint Architecture

Possible OPM Architecture



Developing OPM Adjoint Architecture

Implementation Plan

1

Map forward data flow

Trace Newton loop, Jacobian assembly, WellState storage locations

2

Instrument OPM for export

Write A^k , states, control mode, u^k to HDF5 at each accepted step

3

Extract B_{RR}^k

Re-evaluate AD accumulation at old-level x_{Ri} ; only new OPM computation

4

Implement objective module

θ^k and three partial derivatives via AD types; verify vs. FD

5

Build external adjoint sweep

Backward loop: $RHS \rightarrow (A^k)^T$ solve \rightarrow accumulate gradient

6

Validate vs. finite differences

Centered FD on all controls; relative accuracy $< 10^{-4}$

7

Connect to optimizer

*Callable $u \rightarrow (J, \nabla J)$; for example *scipy L-BFGS-B* for first end-to-end run*

8

Test on realistic case

SPE9 or small field model — correctness at scale

Developing OPM Adjoint Architecture

Proposed initial setup for well control



WELCONTR

Initial control (guess)

Well	d	BHP	Liq(m³)
MY_PROD	0	100	50
MY_PROD	100	200	60
MY_PROD	200	100	50

WELLOWER

Lower bounds

Well	d	BHP	Liq(m³)
MY_PROD	0	10	1
MY_PROD	100	10	1
MY_PROD	200	10	1

WELUPPER

Upper bounds

Well	d	BHP	Liq(m³)
MY_PROD	0	300	200
MY_PROD	100	300	200
MY_PROD	200	300	200

WELGRAD

Gradient output (adjoint)

Well	d	$\frac{\partial J}{\partial BHP}$	$\frac{\partial J}{\partial Liq}$
MY_PROD	0	3.5	0.0
MY_PROD	100	6.0	4.2
MY_PROD	200	8.6	0.0

Not yet syntax proposal for objective function

Zero gradient entries → inactive constraint at that control interval (perturbing it has no effect)

Recent Contributions

Grid-independent Wells

Developing OPM Adjoint Architecture

Thermal Brine

Local Grid Refinement

Thermal Brine

- Why?
 - BRINE option to enable salinity-dependent water properties (viscosity, density) already available
 - THERMAL option already available to enable non-isothermal studies
 - Important in geothermal and CO₂ storage applications
 - **How to account for both salinity- and temperature-dependent effects?**
- Allow combination of keywords BRINE and THERMAL
- Water viscosity μ_w depending on temperature T , pressure P and salt concentration c_s

$$\mu_w(T, P, c_s) = \mu_T(T) \frac{\mu_p(P, c_s)}{\mu_p(P_{ref}, c_{s,ref})}$$

- Data coming from keywords
 - VISCREF
 - WATVISCT
 - PVTWSALT

BRINE + THERMAL

Thermal Brine

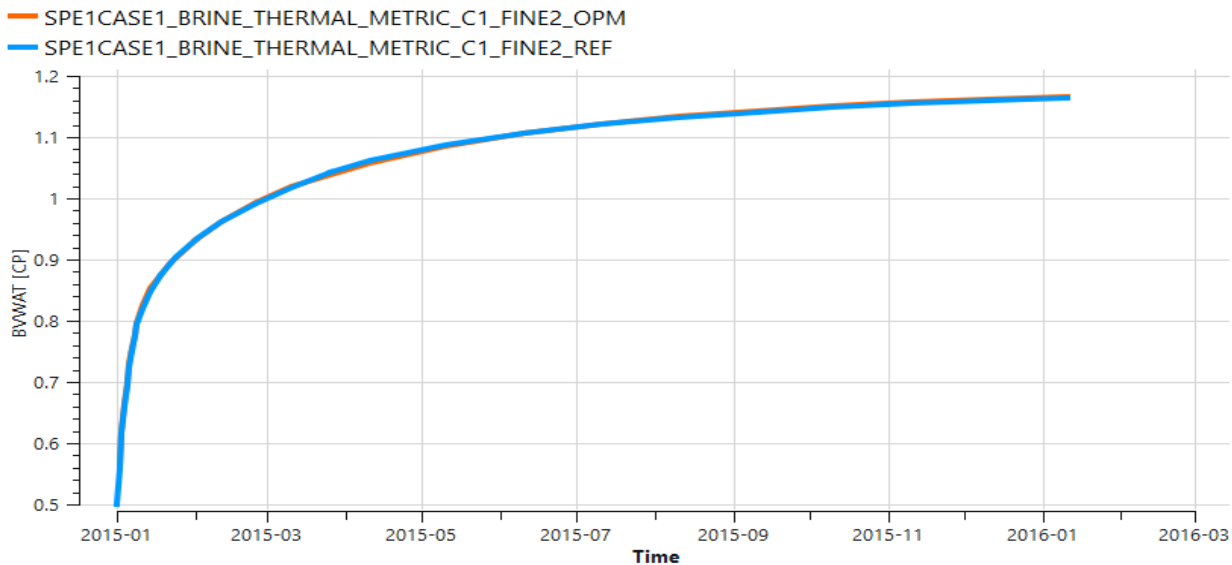
TEST CASE:

- Injection of cold water (30° C) with higher salinity (10 kg/m3) than in-situ brine conditions (80° C and salinity 9 kg/m3)

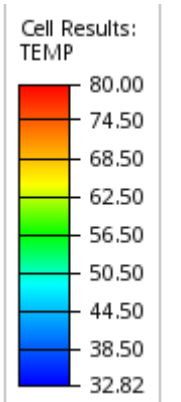
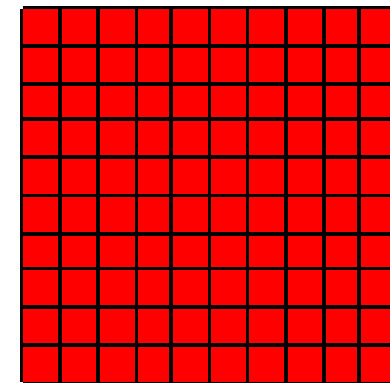
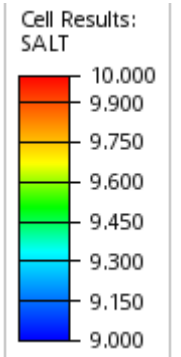
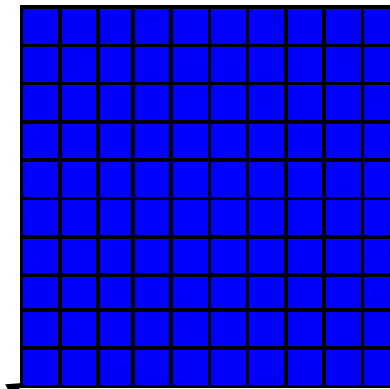
Comparison with reference simulator

Development of the viscosity in grid block [1,1,1]

Block : 1,1,1, BVWAT



Grid block [1,1,1]



Thermal Brine

- Next steps:
 - Harmonize BRINE + THERMAL options with salt precipitation models
 - Support and test of temperature effects on salt precipitation
 - Extend salt solubility model to multiple salts (see Tor Harald's talk on SIMCO2DEPL project)