Model-based optimization of oil and gas production

Jan Dirk Jansen
Delft University of Technology
j.d.jansen@tudelft.nl
Closed-loop reservoir management

Noise -> Input -> System (reservoir, wells & facilities) -> Output -> Noise

Controllable input -> Optimization algorithms

System models

Data assimilation algorithms

Geology, seismics, well logs, well tests, fluid properties, etc.

Input and output noise are managed through optimization algorithms, ensuring closed-loop reservoir management.
Notation: time-discretized equations

System eqs: \( g_k \left( u_k, x_{k-1}, x_k, m \right) = 0 \)

States: \( x = \begin{bmatrix} p^T & s^T \end{bmatrix}^T \) pressures, saturations

Parameters: \( m = \begin{bmatrix} k^T & \phi^T \end{bmatrix}^T \) permeabilities, porosities

Inputs: \( u = \begin{bmatrix} \tilde{p}_{well}^T & \tilde{q}_{well}^T \end{bmatrix}^T \) well pressures/rates

Initial conditions: \( x_0 = \tilde{x}_0 \)

Time interval: \( k = 1, 2, \ldots, K \)
Production optimization: objective function

• Simple Net Present Value (NPV)
• $N_{inj}$ injectors, $N_{prod}$ producers

$$ J = \sum_{k=1}^{K} \left\{ \sum_{j=1}^{N_{prod}} \left[ r_o \times (q_{o,j})_k - r_{wp} \times (q_{wp,j})_k \right] - \sum_{i=1}^{N_{inj}} r_{wi} \times (q_{wi,i})_k \right\} \Delta t_k $$

• $r = \text{unit price or cost}$, $b = \text{discount factor}$, $\tau = 365 \text{ days}$
• Flow rates $q_k$ functions of inputs $u_k$ or outputs (states) $x_k$
Production optimization: maximization problem

• Problem statement: \( \max \mathcal{J}(\mathbf{u}_{1:K}) \) subject to \( \mathbf{u}_{1:K} \)

• System equations: \( \mathbf{g}_k(\mathbf{u}_k, \mathbf{x}_{k-1}, \mathbf{x}_k) = 0 \)

• Initial conditions: \( \mathbf{x}_0 = \bar{\mathbf{x}}_0 \)

• Equality constraints: \( \mathbf{c}_k(\mathbf{u}_k, \mathbf{x}_k) = 0 \)

• Inequality constraints: \( \mathbf{d}_k(\mathbf{u}_k, \mathbf{x}_k) < 0 \)
1) “Open-loop” flooding optimization

System (reservoir, wells & facilities)

Optimization algorithms

System model

Data assimilation algorithms

Geology, seismics, well logs, well tests, fluid properties, etc.

Sensors

Noise

Input

Output

Controllable input

Predicted output

Measured output
12-well example (1)

- 3D reservoir
- High-permeability channels
- 8 injectors, rate-controlled
- 4 producers, BHP-controlled
- Production period of 10 years
- 12 wells x 10 x 12 time steps
  => 1440 optimization parameters
- Bound constraints on controls
- Optimization of monetary value (oil revenues minus water costs)
12-well example (2)

**Reactive Control**

**Cumulative Data**
- Oil Production: $0.00 \times 10^6$ bbl
- Water Production: $0.00 \times 10^6$ bbl
- Water Injection: $0.00 \times 10^6$ bbl
- Revenue: $0.0 \text{ M}$

**Optimal Control**

**Cumulative Data**
- Oil Production: $0.00 \times 10^6$ bbl
- Water Production: $0.00 \times 10^6$ bbl
- Water Injection: $0.00 \times 10^6$ bbl
- Revenue: $0.0 \text{ M}$

*time = 0.00 year*
12-well example (3)

**Cumulative Data**

- **Oil Production:** $2.65 \times 10^8$ bbl
- **Water Production:** $1.31 \times 10^8$ bbl
- **Water Injection:** $3.96 \times 10^6$ bbl

**Revenue:** $45.1 \text{ M}$

**Cumulative Data**

- **Oil Production:** $2.69 \times 10^8$ bbl
- **Water Production:** $0.63 \times 10^8$ bbl
- **Water Injection:** $3.31 \times 10^6$ bbl

**Revenue:** $48.5 \text{ M}$

+8%
Why this wouldn’t work

- Real wells are sparse and far apart
- Real wells have more complicated constraints
- Field management is usually production-focused
- Long-term optimization may jeopardize short-term profit
- Production engineers don’t trust reservoir models anyway

- We do not know the reservoir!
2) “Robust” open-loop flooding optimization
Robust optimization example

- 100 realizations
- Optimize expectation of objective function

\[
\max_{\mathbf{u}_{1:K}} \frac{1}{N_r} \sum_{i=1}^{N_r} J^i \left( \mathbf{u}_{1:K}, \mathbf{m}_i \right)
\]
Robust optimization results

3 control strategies applied to set of 100 realizations:
reactive control, nominal optimization, robust optimization

Van Essen et al., 2006
3) Closed-loop flooding optimization

Noise → Input → System (reservoir, wells & facilities) → Output → Noise

Controllable input

Optimization algorithms

System models

Data assimilation algorithms

Geology, seismics, well logs, well tests, fluid properties, etc.

Predicted output → Measured output

Sensors
"Truth"

Noise → Input

System (reservoir, wells & facilities) → Output
Noise

Optimization algorithms

Controllable input

Sensors

Geology, seismics, well logs, well tests, fluid properties, etc.

System models

Data assimilation algorithms

Predicted output → Measured output
Closed-loop optimization
NPV and contributions from water & oil production

1 month 1 year 2 years 4 years

open-loop
reactive
Optimization techniques

- Global versus local
- Gradient-based versus gradient-free
- Constrained versus non-constrained
- ‘Classical’ versus ‘non-classical’ (simulated annealing, particle swarms, etc.)
- We use ‘optimal control theory’ or ‘adjoint-based’ optimization
Optimal control theory, summary

- Gradient based optimization technique – local optimum
- Gradients of objective function with respect to controls obtained from ‘adjoint’ equation
- Gradients can be used with steepest ascent, quasi Newton, or trust-region methods
- Results in dynamic control strategy, i.e. controls change over time
- Computational effort independent of number of controls
- Output constraints not trivial; various techniques used
- Implementation is code-intrusive
Adjoint-Based Optimization

Part 1 - Theory
Unconstrained optimization (1D)

\[ J(u) = 2u^2 \]

\[ \frac{\partial J}{\partial u} \equiv 4u = 0 \Rightarrow \begin{cases} u = 0 \\ J = 0 \end{cases} \]

\[ \frac{\partial^2 J}{\partial u^2} = 4 > 0 \Rightarrow \text{minimum} \]
Unconstrained optimization (2D)

\[ J(u) = 2(u_1^2 + u_2^2) \]
\[ u = [u_1, u_2]^T \]

\[ \frac{\partial J}{\partial u} = \begin{bmatrix} 4u_1 & 4u_2 \end{bmatrix} = 0^T \Rightarrow \begin{cases} u_1 = 0 \\ u_2 = 0 \\ J = 0 \end{cases} \]

\[ \frac{\partial^2 J}{\partial u^2} = \begin{bmatrix} 4 & 0 \\ 0 & 4 \end{bmatrix} > 0 \Rightarrow \text{minimum} \]
Constrained optimization (elimination)

\[ \mathcal{J}(\mathbf{u}) = 2\left(u_1^2 + u_2^2\right) \quad \text{s.t.} \]

\[ c(\mathbf{u}) \equiv u_1 + u_2 - 0.6 = 0 \]

\[ u_2 = 0.6 - u_1 \]

\[ \mathcal{J} = 4u_1^2 - 2.4u_1 + 0.72 \]

\[ \frac{\partial \mathcal{J}}{\partial u_1} \equiv 8u_1 - 2.4 = 0 \Rightarrow \begin{cases} u_1 = 0.3 \\ u_2 = 0.3 \\ \mathcal{J} = 0.36 \end{cases} \]

\[ \frac{\partial^2 \mathcal{J}}{\partial u_1^2} = 8 > 0 \Rightarrow \text{minimum} \]
Constrained optimization (Lagrange multipliers)

\[ J(u) = 2(u_1^2 + u_2^2) \quad \text{s.t.} \]

\[ c(u) \equiv u_1 + u_2 - 0.6 = 0 \]

\[ \bar{J}(\bar{u}) = J(u) + \lambda c(u) \]

\[ = 2(u_1^2 + u_2^2) + \lambda (u_1 + u_2 - 0.6) \]

\[ \frac{\partial \bar{J}}{\partial \bar{u}} \equiv \begin{bmatrix} 4u_1 + \lambda & 4u_2 + \lambda & u_1 + u_2 - 0.6 \end{bmatrix} = 0^T \Rightarrow \begin{cases} u_1 = 0.3 \\ u_2 = 0.3 \\ \lambda = -1.2 \end{cases} \]

second-order conditions more complex
Lagrange multipliers – interpretation (a)

Recall elimination:

\[ \mathcal{J}(\mathbf{u}) = 2(u_1^2 + u_2^2) \quad \text{s.t.} \quad u_2 = 0.6 - u_1 \]

\[ c(\mathbf{u}) \equiv u_1 + u_2 - 0.6 = 0 \]

\[ \mathcal{J}(u_1) = 4u_1^2 - 2.4u_1 + 0.72 \]

What if \( u_2 \) cannot be expressed in \( u_1 \) or v.v.?

Consider the total differential:

\[ \frac{d\mathcal{J}}{du_1} = \left( \frac{\partial \mathcal{J}}{\partial u_1} + \frac{\partial \mathcal{J}}{\partial u_2} \frac{\partial u_2}{\partial u_1} \right) \]

But how do we compute \( \partial u_2 / \partial u_1 \)?
Lagrange multipliers – interpretation (b)

Consider constraint \( c(u_1, u_2) = 0 \)

Expressed in differential form:

\[
\frac{\partial c}{\partial u_1} \frac{\partial u_1}{\partial u_1} + \frac{\partial c}{\partial u_2} \frac{\partial u_2}{\partial u_2} = 0
\]

Can be rewritten as

\[
\frac{\partial u_2}{\partial u_1} = -\left( \frac{\partial c}{\partial u_2} \right)^{-1} \frac{\partial c}{\partial u_1}
\]

Implicit differentiation!
Lagrange multipliers – interpretation (c)

Given \[ \frac{d\mathcal{J}}{du_1} = \left( \frac{\partial\mathcal{J}}{\partial u_1} + \frac{\partial\mathcal{J}}{\partial u_2} \frac{\partial u_2}{\partial u_1} \right) \]

and \[ \frac{\partial u_2}{\partial u_1} = -\left( \frac{\partial c}{\partial u_2} \right)^{-1} \frac{\partial c}{\partial u_1} \]

we can now write

\[ \frac{d\mathcal{J}}{du_1} = \frac{\partial\mathcal{J}}{\partial u_1} - \frac{\partial\mathcal{J}}{\partial u_2} \left( \frac{\partial c}{\partial u_2} \right)^{-1} \frac{\partial c}{\partial u_1} \]

which, in an optimum, can also be written as

\[ \frac{d\mathcal{J}}{du_1} = 0 \]

\[ \Rightarrow \quad \frac{\partial\mathcal{J}}{\partial u_1} + \frac{\partial\mathcal{J}}{\partial u_2} \left( \frac{\partial c}{\partial u_2} \right)^{-1} \frac{\partial c}{\partial u_1} = 0 \]

\[ \lambda_1 \]
Lagrange multipliers – interpretation (d)

If we have

$$\frac{\partial I}{\partial u_1} + - \frac{\partial I}{\partial u_2} \left( \frac{\partial c}{\partial u_2} \right)^{-1} \frac{\partial c}{\partial u_1} = 0$$

we can also derive that

$$\frac{\partial I}{\partial u_2} + - \frac{\partial I}{\partial u_1} \left( \frac{\partial c}{\partial u_1} \right)^{-1} \frac{\partial c}{\partial u_2} = 0$$

which, if $\lambda_1 = \lambda_2$, can be combined into

$$\frac{\partial I}{\partial u} + \lambda \frac{\partial c}{\partial u} = 0^T$$

$\lambda_1 = \lambda_2$ implies:

$$\frac{\partial I}{\partial u_1} \left( \frac{\partial c}{\partial u_1} \right)^{-1} = \frac{\partial I}{\partial u_2} \left( \frac{\partial c}{\partial u_2} \right)^{-1}$$

OK in optimum

Use of Lagrange multipliers = implicit differentiation
Back to the real thing: Production optimization

• Problem statement: \( \max \mathcal{J}(u_{1:K}) \) subject to \( u_{1:K} \)

• System equations: \( g_k(u_k, x_{k-1}, x_k) = 0 \)

• Initial conditions: \( x_0 = \bar{x}_0 \)

• Equality constraints: \( c_k(u_k, x_k) = 0 \)

• Inequality constraints: \( d_k(u_k, x_k) < 0 \)

• As a first step: disregard constraints \( c_k \) and \( d_k \)
Gradient with implicit differentiation?

What we are looking for:

$$\frac{dJ}{du_k} = \frac{\partial J_k}{\partial u_k} + \sum_{j=k}^{K} \frac{\partial J_j}{\partial x_j} \frac{\partial x_j}{\partial u_k}$$

Effect of $u_k$ on all subsequent time steps

Contributions from time steps $k...K$

$$\frac{\partial x_j}{\partial u_k} = \frac{\partial x_j}{\partial x_{j-1}} \frac{\partial x_{j-1}}{\partial x_{j-2}} ... \frac{\partial x_{k+2}}{\partial x_{k+1}} \frac{\partial x_{k+1}}{\partial x_k} \frac{\partial x_k}{\partial u_k}$$

Requires a lot of implicit differentiation...
Gradient with Lagrange multipliers

• “Adjoin” constraints to objective function:

\[ \mathcal{J} (u_{1:K}, x_{0:K}, \lambda_{0:K}) = \sum_{k=1}^{K} \left[ \mathcal{J}_k (u_k, x_k) + \lambda_0^T (x_0 - \bar{x}_0) \delta_{k-1} + \lambda_k^T g_k (u_k, x_{k-1}, x_k) \right] \]

‘Modified objective function’

• Proceed as before: take first derivatives w.r.t. all independent variables and equate them to zero (i.e. force optimality conditions)

• Note that we can write: \( \frac{\partial g_k}{\partial x_{k-1}} = \frac{\partial g_{k+1}}{\partial x_k} \) (index shift)
Optimality conditions (1)  \[ \bar{J}(u_{1:K}, x_{0:K}, \lambda_{0:K}) \triangleq \sum_{k=1}^{K} \left[ J_k(u_k, y_k) + \lambda_0^T(x_0 - \bar{x}_0)\delta_{k-1} + \lambda_k^T g_k(u_k, x_{k-1}, x_k) \right] \]

\[ \frac{\partial \bar{J}}{\partial u_k} \equiv \frac{\partial J_k}{\partial u_k} + \lambda_k^T \frac{\partial g_k}{\partial u_k} = 0^T \quad k = 1, 2, \ldots, K \]

\[ \frac{\partial \bar{J}}{\partial x_0} \equiv \lambda_1^T \frac{\partial g_1}{\partial x_0} + \lambda_0^T = 0^T \]

\[ \frac{\partial \bar{J}}{\partial x_k} \equiv \frac{\partial J_k}{\partial x_k} + \lambda_{k+1}^T \frac{\partial g_{k+1}}{\partial x_k} + \lambda_k^T \frac{\partial g_k}{\partial x_k} = 0^T \quad k = 1, 2, \ldots, K - 1 \]

\[ \frac{\partial \bar{J}}{\partial x_K} \equiv \frac{\partial J_K}{\partial x_K} + \lambda_K^T \frac{\partial g_K}{\partial x_K} = 0^T \]
Optimality conditions (2) 

\[ \mathcal{J} (u_{1:K}, x_{0:K}, \lambda_{0:K}) \triangleq \sum_{k=1}^{K} \left[ \mathcal{J}_k (u_k, y_k) + \lambda_0^T (x_0 - \tilde{x}_0) \delta_{k-1} + \lambda_k^T g_k (u_k, x_{k-1}, x_k) \right] \]

\[ \frac{\partial \mathcal{J}}{\partial \lambda_0} \equiv (x_0 - \tilde{x}_0)^T = 0^T \]

\[ \frac{\partial \mathcal{J}}{\partial \lambda_k} \equiv g_k^T (u_k, x_{k-1}, x_k) = 0^T \quad k = 1, 2, \ldots, K \]

(Just recovers the initial conditions and system equations)

- The optimality conditions form a joint set of equations for the unknowns \( u_{1:K}, x_{0:K}, \lambda_{0:K} \)
- Can in theory be solved simultaneously (Wathen et al.) but are usually treated sequentially.
Solving the resulting equations (1)

\[ \frac{\partial J}{\partial \lambda_0} \Rightarrow (x_0 - \tilde{x}_0)^T = 0^T \implies x_0 \]

\[ \frac{\partial J}{\partial \lambda_k} \Rightarrow g_k^T(u_k, x_{k-1}, x_k) = 0^T \implies x_{1:K} \]

Running the simulator. (Requires \( \frac{\partial g_k}{\partial x_k} \))

Initial guess!
Solving the resulting equations (1)

\[ \frac{\partial \mathcal{J}}{\partial \lambda_0} \quad \left( x_0 - \tilde{x}_0 \right)^T = 0^T \quad \Rightarrow \quad x_0 \]

\[ \frac{\partial \mathcal{J}}{\partial \lambda_k} \quad g_k^T \left( u_k, x_{k-1}, x_k \right) = 0^T \quad \Rightarrow \quad x_{1:K} \]

\[ \frac{\partial \mathcal{J}}{\partial x_K} \quad \frac{\partial J_K}{\partial x_K} + \lambda_K^T \frac{\partial g_K}{\partial x_K} = 0^T \]

Running
the simulator.
(Requires \( \frac{\partial g_k}{\partial x_k} \))
Solving the resulting equations (1)

\[ \frac{\partial \bar{J}}{\partial \lambda_0} \quad \begin{pmatrix} x_0 - \hat{x}_0 \end{pmatrix}^T = \mathbf{0}^T \implies x_0 \]

\[ \frac{\partial \bar{J}}{\partial \lambda_k} \quad g_k^T \left( u_k, x_{k-1}, x_k \right) = \mathbf{0}^T \implies x_{1:K} \]

\[ \frac{\partial \bar{J}}{\partial x_K} \quad \left( \frac{\partial g_K}{\partial x_K} \right)^T \lambda_K = - \left( \frac{\partial \bar{J}_K}{\partial x_K} \right)^T \implies \lambda_K \]

\[ \frac{\partial \bar{J}}{\partial x_k} \quad \left( \frac{\partial g_k}{\partial x_k} \right)^T \lambda_k = - \left( \frac{\partial g_{k+1}}{\partial x_k} \right)^T \lambda_{k+1} \implies \lambda_{K-1:1} \]

\[ \frac{\partial \bar{J}}{\partial x_0} \quad \lambda_0 = \left( \frac{\partial g_1}{\partial x_0} \right)^T \lambda_1 \implies \lambda_0 \]

Running the simulator. (Requires \( \frac{\partial g_k}{\partial x_k} \))

‘Final condition’

‘Backward’ integration (linear)
Solving the resulting equations (2)

\[ \frac{\partial J_k}{\partial u_k} + \lambda_k^T \frac{\partial g_k}{\partial u_k} \neq 0^T \]

Usually not!
Solving the resulting equations (2)

\[
\frac{\partial \bar{J}}{\partial u_k} = \frac{\partial J_k}{\partial u_k} + \lambda_k^T \frac{\partial g_k}{\partial u_k}
\]

Recall

\[
\frac{d J}{d u_k} = \frac{\partial J_k}{\partial u_k} + \sum_{j=k}^{K} \frac{\partial J_j}{\partial y_j} \frac{\partial x_j}{\partial u_k}
\]

\[
\frac{\partial \bar{J}}{\partial u_k} = \frac{d J}{d u_k} \quad !!! \quad \text{Just what we need}
\]

Can now be used, e.g., in steepest ascent:

\[
u_k^{i+1} = \mathbf{u}_k^i + \alpha \left( \frac{d J}{d u_k^i} \right)^T
\]
Summary adjoint-based optimization

- Adjoint ~ implicit differentiation
- Computational effort independent of number of controls
- Gradient-based optimization – local optimum
- Constraint handling: GRG, lumping, SQP, augmented Lagrangian, … ; not trivial
- Beautiful, but code-intrusive and requires lots of programming => automatic differentiation
- Available in Eclipse (limited functionality), AD-GPRS, MRST, proprietary simulators
- Alternatives: ensemble methods (EnOpt, StoSAG), streamline-based methods, ‘non classical methods’ (particle swarm, etc.; often in combination with ‘proxies’ to reduce computational effort)
Adjoint-Based Optimization

Part 2 - Examples
Classic example; smart horizontal wells

- 45 x 45 grid blocks
- 45 inj. & prod. segments
- $p_{wf}, q_{t}$ at segments known
- 1 PV injected, $q_{inj} = q_{prod}$
- oil price $r_o = 80 \ $/m³
- water costs $r_w = 20 \ $/m³
- discount rate $b = 0\%$

Brouwer and Jansen, 2004, SPEJ
Results; conventional production

Equal pressures in all injector/producer segments
Results; rate-constrained (1)

Conventional (equal pressure in all segments, no control)

Best possible (identical total rates, no pressure constraints)
Results; rate-constrained (2)

NPV
+60%

Production
+ 41% cum oil
- 45% cum wat
Pressure-constrained operation

- Limited energy available
- Total injection/production rate dependent on number of active wells
Results: pressure-constrained

Improvement in NPV
+53%

Production
+16% cum oil
-77% cum water

Injection
-32% cum water
Optimum valve-settings (1)

- Bang-bang (on-off) solution
- Necessary condition: linear controls, linear constraints
Optimum valve-settings (2)

All the action is around the heterogeneities
Optimum valve settings (3)

Streaks act as well extensions

Presence of heterogeneities essential for optimization
Optimum valve-settings (4)

3 valves in injector

4 valves in producer

No need for 45 segments per well
St. Joseph field re-development case

Objective: to determine the value of down-hole control in planned water injectors, in terms of incremental cumulative oil production

- Maximum number of ICVs: 5
- Water injection rate: 10,000 bbl/d per well
- Trajectory of water injector fixed
- Optimum number of ICVs?
- Optimum configuration of perforation zones?
- Optimum operation of the ICVs?

Van Essen et al., 2010, SPEREE
Pilot study on sector model

- Strongly layered structure
- Very limited vertical communication
- Dips approximately 20°
- 21,909 active grid blocks
- Dimensions 1600m x 500m x 450m
- No aquifer support
- 1 gas injection well
- 1 (planned) water injection well
- 7 production wells in sector
Smart water injection well

Properties
• Fixed flow rate of 10,000 bbl/d
• Fixed location and trajectory
• Horizontal section perforated
• Lift table captures pressure drop

Variables
• Number of ICVs
• Length of the perforation zones
• Operation of ICVs

• Controls: kdh multipliers
Base case

• No control
  – All kdh multipliers in 102 layers equal to 1
• Water injection into each layer result of permeability, pressure difference, etc.
  – Performance quantified in terms of cumulative oil production
• Also water injection rate into each zone is determined
  – Zones B, C, D and E
  – No injection in A
Base case results

- Cumulative oil production: 11.47 MMstb

Van Essen et al., 2010, SPEREE
Full 102 zone control (‘technical limit’)

- Cumulative oil production: 12.82 MMstb
- Increase of 11.7% (1.35 MMstb)
Standard 4-group control (geological insight)

- Cumulative oil production: 12.40 MMstb
- Increase of 8.1% (0.93 MMstb)
Alternative 4-group control (optimal grouping)

- Cumulative oil production: 12.62 MMstb
- Increase of 10.0% (1.15 MMstb)
Link with short-term optimization

Noise → Input → System (reservoir, wells & facilities) → Output → Noise

Controllable input

Optimization algorithms

System models

Data assimilation algorithms

Sensors

Geology, seismics, well logs, well tests, fluid properties, etc.

Predicted output → Measured output
Life-cycle optimization vs. reactive control (1)
Life-cycle optimization vs. reactive control (2)

Net Present Value

Reactive Control

Optimal Control

Inj1, Prod1, Prod2, Prod3, Prod4, Prod5

Inj1, Inj2, Inj3, Inj4, Inj5

Inj6

Prod1, Prod2, Prod3, Prod4

time = 10.00 year
Life-cycle optimization vs. reactive control (3)

- Life-cycle optimization attractive for reservoir engineers
  - Increased NPV due to improved sweep efficiency

- Not so attractive from production engineering point of view
  - Decreased short term production
  - Erratic behavior of optimal operational strategy

Van Essen et al., 2011, SPEJ

IPAM 2017 - Computational Issues in Oil Field Applications 61
Hierarchical optimization

• Take production objectives into account by incorporating them as additional optimization criteria:

• Formal solution:
  – Order objectives according to importance
  – Optimize objectives sequentially
  – Optimality of upper objective constrains optimization of lower one

• Only possible if there are redundant degrees of freedom in input parameters after meeting primary objective
Objective function with ridges
Example: Hierarchical optimization using null-space approach (1)

- 3D reservoir
- 8 injection / 4 production wells
- Period of 10 years
- Producers at constant BHP
- Rates in injectors optimized

- **Primary objective**: undiscounted NPV over the life of the field
- **Secondary objective**: NPV with very high discount factor (25%) to emphasize importance of short term production
Example: Hierarchical optimization using null-space approach (2)

Optimization of secondary objective function - constrained to null-space of primary objective

Optimization of secondary objective function - unconstrained

Van Essen et al., 2011, SPEJ
Example: Hierarchical optimization using null-space approach (3)
Controlability of a dynamic system is the ability to influence the states through manipulation of the inputs.

Observability of a dynamic system is the ability to determine the states through observation of the outputs.

Identifiability of a dynamic system is the ability to determine the parameters from the input-output behavior.

All very limited for reservoir simulation models!

Zandvliet, M. et al., 2008: *Computational Geosciences* 12 (4) 808-822.

Model based optimization – conclusions

‘Well control’ optimization:
- Adjoint-based techniques work well; constraints, regularization, storage, efficiency, still to be improved
- Alternatives: gradient-free, particle swarms, EnOpt, StoSAG
- Controllability very limited. Increased by heterogeneities

Well location optimization (not discussed):
- Gradient-free seems to work best
- Combination with well control optimization

Field implementation:
- Well control optimization: none reported
- Acceptance will require combi with short-term optimization
- Computer-assisted history matching: thriving!
- Well location/trajectory optimization: up and coming!
- Advisory mode – tools for discussion
References adjoint-based optimization (1)

Review paper (with additional references)

Early use in history matching

Early use in flooding optimization
References adjoint-based optimization (2)


**TU Delft series**


References adjoint-based optimization (3)


References adjoint-based optimization (4)

Computational aspects

Han, C., Wallis, J., Sarma, P. et al., 2013: Adaptation of the CPR preconditioner for efficient solution of the adjoint equation. SPE Journal 18 (2) 207-213. DOI: org/10.2118/141300-PA.

Algebraic formulation


Constraint handling
References adjoint-based optimization (5)


**Closed-loop reservoir management**


References adjoint-based optimization (6)


Acknowledgments

• Colleagues and students of
  – TU Delft – Department of Geoscience and Engineering
  – TU Eindhoven (TUE) – Department of Electrical Engineering
  – TU Delft – Delft Institute for Applied Mathematics
  – TNO – Built Environment and Geosciences

• Especially for the optimization results presented in this tutorial: Prof. Paul Van den Hof (TUE), Prof. Arnold Heemink (TUD), and (former) PhD students Roald Brouwer, Maarten Zandvliet and Gijs van Essen

• Sponsors: Shell (Recovery Factory program), ENI, Petrobras, Statoil (ISAPP program)