Inversion, history matching, clustering and linear algebra

Andrew R. Conn  
arconn@us.ibm.com

IBM T.J. Watson Research Center

January, 2016, 4-8, Tenth U.S.-Mexico Workshop on Optimization and its Applications,  

Merida, Mexico
Automatic History Matching

\[ m = \arg\min \| F(m, y) - d(y) \|_2^2 + R(m) \]

s.t. \[ d(y) = F(m, y) + \hat{U} + \mu \]

- **History matching**
- **Static model parameters**
  \[ m_s \equiv \{ \kappa, \phi, K \} \]
- **Dynamic model state**
  \[ m_d \equiv \{ p, S, K \} \]
- **Observed data**
  \[ d = \{ p^w, S^w, q^{p,w}, K \} + \hat{U} \]
- **Simulation (observation)**
  \[ F(m) = \{ p, S, q^p, K \} \]
Mathematician’s point of view

- Typically an undetermined least-squares problem
- “Classical” fit based on local well data. Many good fits.
- The vast null space means the problem is intrinsically ill-posed
- Our purpose is to predict the future based upon (past) data.
- Very few of the fits will do this successfully
- Need to make the problem less underdetermined
  
  Two obvious things one should do

- Listen to and incorporate what the geologists can tell us
- Use more global data than just the well-logs
  
  If relevant and possible

- Use smart linear algebra and updating
Integration of 4D seismic data into reservoir models

Use more global data than just the well-logs

<table>
<thead>
<tr>
<th></th>
<th>spatial resolution</th>
<th>temporal resolution</th>
<th>alignment</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>areal</td>
<td>vertical</td>
<td></td>
</tr>
<tr>
<td>production data</td>
<td>low</td>
<td>high</td>
<td>high</td>
</tr>
<tr>
<td>seismic data</td>
<td>High</td>
<td>low</td>
<td>low</td>
</tr>
</tbody>
</table>

Compensate for the spatial sparsity of the production data via seismic information.

Furthermore we can exploit existing adjoint functionality of modern simulators by transforming the seismic data to “equivalent” pseudo-wells.

IMPORTANT FOR THE OPTIMIZATION

First Trick: change into something you can solve.
Advanced industrial simulators offer adjoint /derivative computation capability for wells

- Idea: Use virtual wells that mimic the (interpreted) saturation measurement of seismic information. So we have adjoints.

Impact on simulated fluid flows can be marginalized by:

- Volumetric sample of insignificant size – does not interfere with fluids flow simulation
- Using a very short time-step when a saturation ‘measurement’ is conducted
- Shutting in virtual-wells when no measurement is taken
FIELD RESULTS

COMBINED PRODUCTION DATA & 4D SEISMIC

- Combined History Matching of production and 4D seismic leads to significant improvement in model performance (x10 improved match)
- Highly efficient workflow (hours replacing months)
- Understanding of boundaries and reservoir connectivity

4D seismic

1998 2001 2004

Water

Oil

Prediction from initial geological model

Combined production - 4D History Match
The Practical effect of an underdetermined problem

DECISION RELEVANT PRIOR SAMPLING – THE FUTURE

- ... with very different predictions and predicted Net Present Value
For each realization mass flux vector fields is computed $\dot{F}_i(x,y,z;t)$.

- Fluxes capture chief characteristics of dynamics, yet, 4D vector fields are of a large dimension (3 x grid cells x time steps).
- Clustering in such large dimensional space is intractable.
- Instead, reduced order representation of each flux is considered.
Defining measures of similarity

Mass Flux Representation in Reduced Space

- Singular value decomposition of vector fields from all realizations enables reduced order representation

\[ U\Sigma V^T = \left[ \dot{F}_1(x, y, z; t), \dot{F}_2(x, y, z; t), \ldots, \dot{F}_n(x, y, z; t) \right] \]
Assessing Clustering Results

DENDOGRAM OF WATER+OIL FLUX
(short simulations Low Perm)

Flux clustering pick up complete spectrum of training rock models

Big question! do these clusters provide different production scenarios?
Assessing Clustering Results

Dendograms based on OIL+WAT fluxes (Low Perm)

Very short simulations are used to generate fluxes.

A very narrow window of time (well modulations are key) is used and still we’re able to pick up long-term trends in production data.

Representatives can now be extracted.

In-between flow-distinct scenarios are nicely discriminated by clustering.
- **Size of initial ensemble** can readily be reduced by orders of magnitude

- **Each representative** can be regarded as a sample from a **density function**

- This **density function** can further be used for History Matching, model maturation

- **Practicality is not compromised as no full simulations are performed**
Let $A$ be an augmentation of the matrices $A_1 \in \mathbb{R}^{m_1 \times n}$, $A_2 \in \mathbb{R}^{m_2 \times n}$

$$A = \begin{bmatrix} A_1 \\ A_2 \end{bmatrix} \in \mathbb{R}^{(m_1 + m_2) \times n}$$

Let the SVD decomposition of these matrices be given by:

$$A_1 = U_1 S_1 V_1^*$$
$$A_2 = U_2 S_2 V_2^*$$

with $U_i \in \mathbb{R}^{m_i \times n}$, $S_i \in \mathbb{R}^{n \times n}$, $V_i \in \mathbb{R}^{n \times n}$

We seek the decomposition $A = USV^*$ of the augmented matrix $A$
By definition

\[
A^*_K AV = (VS^*U^*)(USV^*)V = VS^2
\]

Starting with

\[
A^*_K AV = \begin{bmatrix} A_1^* & A_2^* \end{bmatrix} \begin{bmatrix} A_1 \\ A_2 \end{bmatrix} V = \begin{bmatrix} A_1^*A_1 + A_2^*A_2 \\ M_1 \\ M_2 \end{bmatrix} V = VS^2
\]

Then solve the (relatively small) eigen-problem \( KV = VS^2 \)

\( U \) is then be given by

\[
U = AVS^{-1} = (USV^*)VS^{-1}
\]

In our context we can ignore the reduction in stability
Note that here, we can save some computation by utilizing the small $n \times n$ (number of columns) product

$$A_1^* A_1 = M_1$$

from the previous run, and therefore, we retain the product

$$A_2^* A_2 = M_2$$

for future use.

This process can be repeated further giving $M_1, M_2, \ldots, M_k$

$$[M_1 + M_2 + \ldots + M_k]V = VS^2$$
Let us assume that a set of \( n < m \) model realizations \( A \in \mathbb{C}^{m \times n} \).

Further assume their effective rank \( k \) is relatively small \( k = n \)

\[
\| A - U^{(k)} S^{(k)} V^{(k)*} \|_2 \leq \delta_k
\]

Partition \( A \) into \( s \) subsets for which we can effectively compute their SVD

\[
A = [A_1, A_2, \ldots, A_s]
\]

SVD of each can be computed in parallel

\[
U_1 S_1 V_1^* = A_1, \quad U_2 S_2 V_2^* = A_2, \quad \ldots, \quad U_s S_s V_s^* = A_s
\]

Given singular values, we select the top singular entries

\[
\sum_i k_i = k_s \geq k
\]
Re-orthogonalize the union of the selected SVs

\[
\begin{bmatrix}
U_1^{(k_1)} S_1^{(k_1)} V_1^{(k_1)}, & U_2^{(k_2)} S_2^{(k_2)} V_2^{(k_2)}, & \ldots, & U_s^{(k_s)} S_s^{(k_s)} V_s^{(k_s)}
\end{bmatrix}
\]

2nd truncation can be performed now

The output would be \( k_T \leq k_s \) ordered spanning vectors

If needed, randomly mix the remaining vectors for further distributed processing

The process is repeated until a sufficiently small set is obtained

Finding a spanning set is a key problem for a broad range of numerical algorithms but for large scale matrices it is computationally intensive [of the order of \( \min(mn^2, m^2n) \) for an \( m \times n \) matrix] or even unattainable.
A set is constructed of 50x20 random vectors

Variability of additional 20 entries is simulated via noisy linear combination of the 50x20 set

50x40 set was split into two 50x20 sets

More than 10 SVDs were retained from each set
Following independent SVD and composition
Four Fundamental Unproved Theorems:

Asymptotics are rarely seen in practise but the best methods in theory are the best in practise.

A sensible person normally gives up on determining global optima. (So a sensible person doesn’t try to solve MINLPs ???????)

It is always better to obtain and use derivatives if you can.

Simulated Annealing, Genetic Algorithms etc are usually for the ignorant or the desperate.
Asymptotics are rarely seen in practise but the best methods in theory are the best in practise.

A sensible person normally gives up on determining global optima. (So a sensible person doesn’t try to solve MINLPs ?????)

It is always better to obtain and use derivatives if you can.

Simulated Annealing, Genetic Algorithms etc are usually for the ignorant or the desperate.
Four Fundamental Unproved Theorems:

Asymptotics are rarely seen in practise but the best methods in theory are the best in practise.

A sensible person normally gives up on determining global optima. (So a sensible person doesn’t try to solve MINLPs ????????)

It is always better to obtain and use derivatives if you can.

Simulated Annealing, Genetic Algorithms etc are usually for the ignorant or the desperate.
Four Fundamental Unproved Theorems:

Asymptotics are rarely seen in practise but the best methods in theory are the best in practise.

A sensible person normally gives up on determining global optima. (So a sensible person doesn’t try to solve MINLPs ???????)

It is always better to obtain and use derivatives if you can.

Simulated Annealing, Genetic Algorithms etc are usually for the ignorant or the desperate.
Four Fundamental Unproved Theorems:

Asymptotics are rarely seen in practise but the best methods in theory are the best in practise.

A sensible person normally gives up on determining global optima. (So a sensible person doesn’t try to solve MINLPs ????????)

It is always better to obtain and use derivatives if you can.

Simulated Annealing, Genetic Algorithms etc are usually for the ignorant or the desperate.
Numerical Results: History Matching

50 layers of 2′ with 60 × 220 cells 20′ × 10′
Up-scaled to 30 × 110 × 25 cells of 80′ × 40′ × 4′
10 yrs production: 1 injector well, 1 – 4 producers.

Optimize the number of wells and their locations to maximize the NPV of the field.
Number of variables being set is 14 continuous and 4 binary variables

![Graph showing NPV along iterations and Nb of producers along iterations.](image)

NPV along iterations - NPV\text{max} = 247M$

- Nomad
- TR basic
- TR no good cut
- TR local branching
- no good cut

Nb of producers along iterations

Simulations

NPV (M$)
Run with 3 different tunings. The initial configuration is displayed at top left.

- **Initial well config**
  - NPV = 91M$

- **NOMAD**
  - Best well config (simu 22)
  - NPV = 197M$

- **TR basic**
  - Best well config (simu 11)
  - NPV = 241M$

- **TR no good cut constraint**
  - Best well config (simu 20)
  - NPV = 241M

- **TR local branching + no good cut constraint**
  - Best well config (simu 20)
  - NPV = 247M$
Numerical Results (continued)

Number of variables being set is 4 continuous and 8 binary variables.
Numerical Results (continued)

Number of variables being set is 4 continuous and 8 binary variables.