

DELFT UNIVERSITY OF TECHNOLOGY

REPORT 07-05

AN EFFICIENT GRADIENT-BASED PARAMETER ESTIMATION ALGORITHM
USING REPRESENTER EXPANSIONS

J.R. ROMMELSE, J.D. JANSEN AND A.W. HEEMINK

ISSN 1389-6520

Reports of the Department of Applied Mathematical Analysis

Delft 2007

Copyright © 2007 by Department of Applied Mathematical Analysis, Delft, The Netherlands.

No part of the Journal may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission from Department of Applied Mathematical Analysis, Delft University of Technology, The Netherlands.

An efficient gradient-based parameter estimation algorithm using representer expansions

Joris Rommelse ^{*} Jan Dirk Jansen [†] Arnold Heemink [‡]

March 12, 2007

Abstract

The discrepancy between observed measurements and their model predicted antitheses can be used to improve either the model output alone or both the model output and the parameters that underlie the model. In case of parameter estimation, methods exist that can efficiently calculate the gradient of the discrepancy to changes in the parameters, assuming that there are no uncertainties in addition to the unknown parameters. Usually many different parameter sets exist that locally minimize the discrepancy, so the gradient must be regularized before it can be used by gradient-based minimization algorithms. This article proposes a method for calculating a gradient in the presence of additional model errors, through the use of representer expansions. The representers are data-driven basis functions that perform the regularization. All available data can be used during every iteration of the minimization scheme, as is the case in the classical Representer Method (RM). However, the method also allows adaptive selection of different portions of the data during different iterations to reduce computation time; the engineer now has the freedom to choose the number of basis functions and revise this choice at every iteration. The method also differs from the classic RM by the introduction of measurement representers in addition to state, adjoint and parameter representers and by the fact that no correction terms need to be calculated. Unlike the classic RM, where the minimization scheme is prescribed, the new RM provides a gradient that can be used in any minimization algorithm.

The new RM is first explained and then its applicability to an advection-diffusion problem is illustrated.

^{*}Delft University of Technology, Department of Geotechnology, E-mail: J.R.Rommelse@tudelft.nl

[†]Delft University of Technology, Department of Geotechnology & Shell Intl. E&P, Exploratory Research, E-mail: J.D.Jansen@tudelft.nl

[‡]Delft University of Technology, Department of Applied mathematics, E-mail: A.W.Heemink@tudelft.nl

1 Introduction

1.1 Gradient-based parameter estimation

Data assimilation methods aim to improve numerical models by comparing actual measurements of a physical system with the numerical model predictions of these measurements. As the parameters of the numerical model are changed, so do the predicted state variables and the predicted measurements. The discrepancy between the "measured measurements" and the "predicted measurements" can be used to update only the state variables (state estimation) or also the parameters (parameter estimation) in order to decrease this discrepancy. When only the state of the model is predicted, the model itself is not corrected. Alternatively, the model parameters, and hence the model itself, may be changed until the predicted output lie satisfactorily close to the measurements. Parameter estimation aims at improving the predictive ability of the model, whereas state estimation does not. Estimating initial states falls in the category parameter estimation, estimating all other states is state estimation. When the output of the model is used to make decisions, state estimation is appropriate for time scales on which the error in the model's predictive ability can be neglected. When the model is used for making long-term decisions, parameter estimation algorithms must be used.

This article focusses on gradient-based parameter estimation algorithms. More precisely, it proposes a method for calculating the gradient of the discrepancy with respect to changes in the parameters, in the presence of model errors.

1.2 Model errors; strong and weak constraints

Often, the discrepancy between measured measurements and predicted measurements is formulated using the Euclidean norm. The objective of the data assimilation is then to minimize the square of this norm with respect to the model parameters while the numerical model is used as a (strong) constraint.

However, there is an additional phenomenon that may cause the discrepancy; the model is an approximation, so even if the parameters were known, the model might still produce incorrect output. These errors can be modelled as extra parameters, which are also added to the objective function using the 2-norm. The model is then used as a weak constraint in the minimization problem.

1.3 Notation

The state variables at time t_i are denoted by $\mathbf{x}_i, i \in \{0, \dots, n\}$. Running the model is denoted by

$$\mathbf{g}(\mathbf{x}_i, \mathbf{x}_{i-1}, \mathbf{a}, \varepsilon_i) = \mathbf{0}$$

where the model parameters are collected in the vector \mathbf{a} and the model errors on interval $[t_{i-1}, t_i]$ are contained in the vector ε_i . The initial states may be part of the parameter estimation process, so $\mathbf{x}_0 = \mathbf{x}_0(\mathbf{a})$. The minimization relies on the availability of the first

(mean) and second (covariance) order statistics of the model parameters and model errors. These are denoted by \mathbf{a}^{prior} , \mathbf{P}_a , $\varepsilon_i^{prior} = \mathbf{0}$ and $\mathbf{P}_{\varepsilon_i}$.

In the strong constraint case, the model errors are explicitly set to zero. The objective function that has to be minimized is

$$J = \frac{1}{2} (\mathbf{h}(\mathbf{x}_{\{0,\dots,n\}}) - \mathbf{m})^T P_{\mathbf{h}}^{-1} (\mathbf{h}(\mathbf{x}_{\{0,\dots,n\}}) - \mathbf{m}) + \frac{1}{2} (\mathbf{a} - \mathbf{a}^{prior})^T \mathbf{P}_a^{-1} (\mathbf{a} - \mathbf{a}^{prior}) + \sum_{i=1}^n \lambda_i^T \mathbf{g}(\mathbf{x}_i, \mathbf{x}_{i-1}, \mathbf{a}, \mathbf{0}) \quad (1)$$

where \mathbf{m} contains the actual physical measurements, possibly taken at different times, and \mathbf{h} is the measurement operator that operates on all state variables at all time steps. $P_{\mathbf{h}}$ represents the uncertainty in the measurements in the form of an error covariance matrix. The last term of Eq. (1) represents the system equations \mathbf{g} that have been adjoined to the objective function with the aid of Lagrange multipliers λ_i in the usual fashion [Bennett, 2006].

When model errors ε_i are taken into account, they are no longer explicitly set to zero, but they become additional parameters in the minimization process. They are assumed to be zero-mean, so the objective becomes

$$J = \frac{1}{2} (\mathbf{h}(\mathbf{x}_{\{0,\dots,n\}}) - \mathbf{m})^T P_{\mathbf{h}}^{-1} (\mathbf{h}(\mathbf{x}_{\{0,\dots,n\}}) - \mathbf{m}) + \frac{1}{2} (\mathbf{a} - \mathbf{a}^{prior})^T \mathbf{P}_a^{-1} (\mathbf{a} - \mathbf{a}^{prior}) + \frac{1}{2} \sum_{i=1}^n \varepsilon_i^T P_{\varepsilon_i}^{-1} \varepsilon_i + \sum_{i=1}^n \lambda_i^T \mathbf{g}(\mathbf{x}_i, \mathbf{x}_{i-1}, \mathbf{a}, \varepsilon_i) \quad (2)$$

1.4 Representer Method

In literature, the Representer Method (RM) [Bennett and McIntosh, 1982] [Eknes and Evensen, 1997, Bennett, 2002, Valstar et al., 2004, Baird and Dawson, 2005] [Janssen et al., 2006, Przybysz et al., 2007] is usually derived as an iterative method that solves the weak constraint least-squares minimization problem Eq. (2). Simultaneously it decomposes the deviation of the estimated parameters from the prior parameters into the isolated effects of every measurement. This regularizes the minimization problem and it also gives information that can be used to quantify the usefulness of every single measurement.

In this article, a new Representer Method is derived as a postprocessor that evaluates the effect of the measurements on the solution of the weak constraint minimization problem. It is then reformulated to produce a regularized gradient that can be used by any gradient-based minimization algorithm in order to find the solution of the weak constraint minimization problem. The method allows decomposition of the parameter vector into a general number of basis functions rather than a number that necessarily needs to be

equal to the number of measurements as in the classical method. This makes the method computationally more attractive for applications where many measurements are available. Due to a new linearization, no correction terms have to be calculated, as was the case in the earlier versions of the RM as applied to non-linear problems. In our opinion, this new formulation of the non-linear RM is less complex and therefore easier to understand.

2 Gradient of the strong constraint minimization problem

2.1 Obtaining a gradient

The derivatives of Eq. (2) with respect to λ_i , ε_i , \mathbf{x}_i and \mathbf{a} are

$$\left(\frac{\partial J}{\partial \lambda_i}\right)^T = \mathbf{g}(\mathbf{x}_i, \mathbf{x}_{i-1}, \mathbf{a}, \varepsilon_i) \quad (3)$$

$$\left(\frac{\partial J}{\partial \varepsilon_i}\right)^T = P_{\varepsilon_i}^{-1} \varepsilon_i + \left(\frac{\partial \mathbf{g}(\mathbf{x}_i, \mathbf{x}_{i-1}, \mathbf{a}, \varepsilon_i)}{\partial \varepsilon_i}\right)^T \lambda_i \quad (4)$$

$$\begin{aligned} \left(\frac{\partial J}{\partial \mathbf{x}_i}\right)^T &= \left(\frac{\partial \mathbf{h}}{\partial \mathbf{x}_i}\right)^T P_{\mathbf{h}}^{-1} (\mathbf{h}(\mathbf{x}_{\{0, \dots, n\}}) - \mathbf{m}) + \\ &+ \left(\frac{\partial \mathbf{g}(\mathbf{x}_i, \mathbf{x}_{i-1}, \mathbf{a}, \varepsilon_i)}{\partial \mathbf{x}_i}\right)^T \lambda_i + \left(\frac{\partial \mathbf{g}(\mathbf{x}_{i+1}, \mathbf{x}_i, \mathbf{a}, \varepsilon_{i+1})}{\partial \mathbf{x}_i}\right)^T \lambda_{i+1} \end{aligned} \quad (5)$$

$$\left(\frac{\partial J}{\partial \mathbf{a}}\right)^T = \mathbf{P}_{\mathbf{a}}^{-1} (\mathbf{a} - \mathbf{a}^{prior}) + \sum_{i=1}^n \left(\frac{\partial \mathbf{g}(\mathbf{x}_i, \mathbf{x}_{i-1}, \mathbf{a}, \varepsilon_i)}{\partial \mathbf{a}}\right)^T \lambda_i \quad (6)$$

For t_n , the term including λ_{i+1} is missing from Eq. (5). This can be done by introducing $\lambda_{n+1} = \mathbf{0}$. For t_0 , the term including λ_i is missing. In case the initial states \mathbf{x}_0 are part of the parameter estimation process, the term

$$\left(\frac{\partial \mathbf{x}_0(\mathbf{a})}{\partial \mathbf{a}}\right)^T \left(\frac{\partial \mathbf{g}(\mathbf{x}_1, \mathbf{x}_0, \mathbf{a}, \varepsilon_1)}{\partial \mathbf{x}_0}\right)^T \lambda_1$$

should be added to Eq. (6). For the strong constraint case, where ε_i is explicitly set to $\mathbf{0}$, the gradient of the objective function with respect to the model parameters, $\left(\frac{\partial J}{\partial \mathbf{a}}\right)^T$, can be calculated using Eq. (6), where the model states \mathbf{x}_i and adjoint states λ_i follow from sequentially solving Eq. (3) and Eq. (5) with the left-hand sides, $\left(\frac{\partial J}{\partial \lambda_i}\right)^T$ and $\left(\frac{\partial J}{\partial \mathbf{x}_i}\right)^T$, set to zero. Eq. (4) does not need to be used; instead $\varepsilon_i = \mathbf{0}$ is used.

2.2 Using the gradient

To solve the strong constraint minimization problem, a numerical routine must be implemented that evaluates J , Eq. (1), and $\left(\frac{\partial J}{\partial \mathbf{a}}\right)^T$, as given in section 2.1. This routine can then be passed to any gradient-based minimization software package, together with a set of initial parameters (usually $\mathbf{a}^{init} = \mathbf{a}^{prior}$) and some appropriate minimization options that are algorithm-dependent.

Often the objective function has multiple local minima and the minimization process needs to be regularized. If low-order parameters \mathbf{b} are introduced such that $\mathbf{a} = \mathbf{a}^{prior} + \mathbf{Q}\mathbf{b}$, with $\mathbf{Q}^T\mathbf{Q} = \mathbf{I}$, then a regularized gradient can be found as $\mathbf{Q}\mathbf{Q}^T \left(\frac{\partial J}{\partial \mathbf{a}}\right)^T$. The orthogonal matrix \mathbf{Q} can for example be obtained by selecting several left-singular vectors (section 3.7) of a square root \mathbf{L} of the covariance matrix $\mathbf{P}_a = \mathbf{L}\mathbf{L}^T$.

3 Gradient of the weak constraint minimization problem

3.1 Local minimizer

In a stationary point (denoted by superscript s) of Eq. (2), all gradients are equal to zero, so

$$\mathbf{g}(\mathbf{x}_i^s, \mathbf{x}_{i-1}^s, \mathbf{a}^s, \varepsilon_i^s) = \mathbf{0} \quad (7)$$

$$\varepsilon_i^s = -P_{\varepsilon_i} \left(\frac{\partial \mathbf{g}(\mathbf{x}_i, \mathbf{x}_{i-1}, \mathbf{a}, \varepsilon_i)}{\partial \varepsilon_i} \right)^T \lambda_i^s \quad (8)$$

$$\begin{aligned} & \left(\frac{\partial \mathbf{h}(\mathbf{x}_{\{0, \dots, n\}}^s)}{\partial \mathbf{x}_i^s} \right)^T P_{\mathbf{h}}^{-1}(\mathbf{m} - \mathbf{h}(\mathbf{x}_{\{0, \dots, n\}}^s)) = \\ & = \left(\frac{\partial \mathbf{g}(\mathbf{x}_i^s, \mathbf{x}_{i-1}^s, \mathbf{a}^s, \varepsilon_i^s)}{\partial \mathbf{x}_i^s} \right)^T \lambda_i^s + \left(\frac{\partial \mathbf{g}(\mathbf{x}_{i+1}^s, \mathbf{x}_i^s, \mathbf{a}^s, \varepsilon_{i+1}^s)}{\partial \mathbf{x}_i^s} \right)^T \lambda_{i+1}^s \end{aligned} \quad (9)$$

$$\mathbf{P}_a^{-1}(\mathbf{a}^{prior} - \mathbf{a}^s) = \sum_{i=1}^n \left(\frac{\partial \mathbf{g}(\mathbf{x}_i^s, \mathbf{x}_{i-1}^s, \mathbf{a}^s, \varepsilon_i^s)}{\partial \mathbf{a}^s} \right)^T \lambda_i^s \quad (10)$$

Unlike section 2.1, the forward equations Eq. (7) and the adjoint equations Eq. (9) are now coupled because the model errors ε_i^s are no longer equal to zero; they are related to the adjoint states λ_i^s by Eq. (8).

3.2 Basis functions

The minimization algorithm is started with $\mathbf{a}^{start} = \mathbf{a}^{prior}$ and $\varepsilon_i^{start} = \lambda_i^{start} = \mathbf{0}$. Applying these prior conditions to Eq. (7) gives the prior system states \mathbf{x}_i^{prior} . Given the prior states,

also prior measurements can be predicted, $\mathbf{h}(\mathbf{x}_{\{0,\dots,n\}}^{prior})$. The causes why the variables move away from their prior are parameterized by \mathbf{b} . In the classic Representer Method there is a 1-1-relationship between one such cause and an isolated measurement in space and time. In this article, this assumption is abandoned. Moreover, for computational purposes, it is interesting to assume that the number of parameters in the vector \mathbf{b} is (much) smaller than the number of measurements.

The deviations from the priors are now decomposed as

$$\mathbf{x}_i^s - \mathbf{x}_i^{prior} = \mathbf{A}_i \mathbf{b} \quad (11)$$

$$\lambda_i^s = \mathbf{B}_i \mathbf{b} \quad (12)$$

$$\mathbf{a}^s - \mathbf{a}^{prior} = \mathbf{C} \mathbf{b} \quad (13)$$

$$\mathbf{h}(\mathbf{x}_{\{0,\dots,n\}}^s) - \mathbf{h}(\mathbf{x}_{\{0,\dots,n\}}^{prior}) = \mathbf{D} \mathbf{b} \quad (14)$$

The columns of \mathbf{A}_i , \mathbf{B}_i , \mathbf{C} and \mathbf{D} contain the state representers, the adjoint representers, the parameter representers and the measurement representers respectively. When the measurement operator \mathbf{h} is linear, the measurement representers \mathbf{D} can be constructed by applying \mathbf{h} to the matrix that is obtained by concatenating the state representers \mathbf{A}_i as row blocks. Alternatively, the RM can be formulated in terms of state representers without defining measurement representers, as is done in the classic RM. Only when the measurement operator is non-linear, the introduction of measurement representers has added value. Theoretically, it is also possible to introduce error representers

$$\varepsilon_i^s = \mathbf{E}_i \mathbf{b}$$

However, looking at Eq. (8), these are nothing more than modified adjoint representers

$$\mathbf{E}_i = -P_{\varepsilon_i} \left(\frac{\partial \mathbf{g}(\mathbf{x}_i, \mathbf{x}_{i-1}, \mathbf{a}, \varepsilon_i^s)}{\partial \varepsilon_i} \right)^T \mathbf{B}_i$$

and they have no practical application.

3.3 Representer equations

Substitution of Eq. (11), Eq. (12) and Eq. (13) into Eq. (9) and Eq. (10) results in

$$\begin{aligned} & \left(\frac{\partial \mathbf{h}(\mathbf{x}_{\{0,\dots,n\}}^s)}{\partial \mathbf{x}_i^s} \right)^T P_{\mathbf{h}}^{-1} (\mathbf{m} - \mathbf{h}(\mathbf{x}_{\{0,\dots,n\}}^s)) = \\ & = \left(\frac{\partial \mathbf{g}(\mathbf{x}_i^s, \mathbf{x}_{i-1}^s, \mathbf{a}^s, \varepsilon_i^s)}{\partial \mathbf{x}_i^s} \right)^T \mathbf{B}_i \mathbf{b} + \left(\frac{\partial \mathbf{g}(\mathbf{x}_{i+1}^s, \mathbf{x}_i^s, \mathbf{a}^s, \varepsilon_i^s)}{\partial \mathbf{x}_i^s} \right)^T \mathbf{B}_{i+1} \mathbf{b} \end{aligned} \quad (15)$$

$$\mathbf{C}\mathbf{b} = -\mathbf{P}_a \sum_{i=1}^n \left(\frac{\partial \mathbf{g}(\mathbf{x}_i^s, \mathbf{x}_{i-1}^s, \mathbf{a}^s, \varepsilon_i^s)}{\partial \mathbf{a}^s} \right)^T \mathbf{B}_i \mathbf{b} \quad (16)$$

Eq. (15) can be simplified by requiring

$$P_h^{-1}(\mathbf{m} - \mathbf{h}(\mathbf{x}_{\{0, \dots, n\}}^s)) = \mathbf{Q}\mathbf{b} \quad (17)$$

The matrix \mathbf{Q} is explained in section 3.7. Using this requirement provides a means to calculate the *adjoint representers* \mathbf{B}_i :

$$\begin{aligned} & \left(\frac{\partial \mathbf{g}(\mathbf{x}_i^s, \mathbf{x}_{i-1}^s, \mathbf{a}^s, \varepsilon_i^s)}{\partial \mathbf{x}_i^s} \right)^T \mathbf{B}_i = \\ & = \left(\frac{\partial \mathbf{h}(\mathbf{x}_{\{0, \dots, n\}}^s)}{\partial \mathbf{x}_i^s} \right)^T \mathbf{Q} - \left(\frac{\partial \mathbf{g}(\mathbf{x}_{i+1}^s, \mathbf{x}_i^s, \mathbf{a}^s, \varepsilon_i^s)}{\partial \mathbf{x}_i^s} \right)^T \mathbf{B}_{i+1} \end{aligned} \quad (18)$$

The *parameter representers* \mathbf{C} follow by removing \mathbf{b} from Eq. (16)

$$\mathbf{C} = -\mathbf{P}_a \sum_{i=1}^n \left(\frac{\partial \mathbf{g}(\mathbf{x}_i^s, \mathbf{x}_{i-1}^s, \mathbf{a}^s, \varepsilon_i^s)}{\partial \mathbf{a}^s} \right)^T \mathbf{B}_i \quad (19)$$

Optionally, the term

$$\left(\frac{\partial \mathbf{x}_0(\mathbf{a}^s)}{\partial \mathbf{a}^s} \right)^T \left(\frac{\partial \mathbf{g}(\mathbf{x}_1^s, \mathbf{x}_0^s, \mathbf{a}^s, \varepsilon_1^s)}{\partial \mathbf{x}_0^s} \right)^T \mathbf{B}_1$$

has to be added to Eq. (19) for estimating the initial states. The *state representers* \mathbf{A}_i are obtained by differentiating Eq. (7) with respect to the representer coefficients \mathbf{b}

$$\begin{aligned} & \frac{\partial \mathbf{g}(\mathbf{x}_i^s, \mathbf{x}_{i-1}^s, \mathbf{a}^s, \varepsilon_i^s)}{\partial \mathbf{x}_i^s} \mathbf{A}_i + \frac{\partial \mathbf{g}(\mathbf{x}_i^s, \mathbf{x}_{i-1}^s, \mathbf{a}^s, \varepsilon_i^s)}{\partial \mathbf{x}_{i-1}^s} \mathbf{A}_{i-1} + \frac{\partial \mathbf{g}(\mathbf{x}_i^s, \mathbf{x}_{i-1}^s, \mathbf{a}^s, \varepsilon_i^s)}{\partial \mathbf{a}^s} \mathbf{C} = \\ & = \frac{\partial \mathbf{g}(\mathbf{x}_i^s, \mathbf{x}_{i-1}^s, \mathbf{a}^s, \varepsilon_i^s)}{\partial \varepsilon_i^s} P_{\varepsilon_i} \left(\frac{\partial \mathbf{g}(\mathbf{x}_i, \mathbf{x}_{i-1}, \mathbf{a}, \varepsilon_i)}{\partial \varepsilon_i^s} \right)^T \mathbf{B}_i \end{aligned} \quad (20)$$

and the same is done with Eq. (14) to obtain the *measurement representers* \mathbf{D}

$$\mathbf{D} = \sum_{i=0}^n \frac{\partial \mathbf{h}(\mathbf{x}_{\{0, \dots, n\}}^s)}{\partial \mathbf{x}_i^s} \mathbf{A}_i \quad (21)$$

Substitution of Eq. (14) into Eq. (17) indicates that the *representer coefficients* \mathbf{b} should be obtained as the least-squares solution of

$$(\mathbf{D} + P_h \mathbf{Q}) \mathbf{b} = \mathbf{m} - \mathbf{h}(\mathbf{x}_{\{0, \dots, n\}}^{prior}) \quad (22)$$

3.4 Representer Method as iterative minimizer

The representer method can be used as post-processor after a local minimum of Eq. (2) has been found by another method. Only equations Eq. (18) and Eq. (19) need to be calculated. If no such local minimum has yet been found, the representer method can also be used to iteratively approach a minimum. The steps that need to be taken then are (superscript s now stands for estimate, rather than stationary point):

1. Initialize the parameter estimate \mathbf{a}^s equal to the parameter prior \mathbf{a}^{prior} .
2. Initialize the adjoint states λ_i^s and model errors ε_i^s equal to zero.
3. Run the non-linear model Eq. (7).
4. Choose \mathbf{Q} (section 3.7).
5. Calculate the adjoint representers Eq. (18).
6. Calculate the parameter representers Eq. (19).
7. Calculate the state representers Eq. (20).
8. Calculate the measurement representers Eq. (21).
9. Calculate new representer coefficients Eq. (22).
10. Calculate new adjoint states Eq. (12).
11. Calculate the model errors Eq. (8).
12. Calculate new parameters. Eq. (13) can be used, or a line search can be included

$$\mathbf{a}_{new}^s = (1 - \alpha) \mathbf{a}_{old}^s + \alpha (\mathbf{a}^{prior} + \mathbf{Cb})$$

13. Go to 3 if stopping criterion has not been fulfilled.

3.5 Obtaining a gradient

After step 3 of section 3.4 has finished, the measurements can be predicted. Together with the input parameters (step 1) and the model errors (step 11), the objective Eq. (2) can be evaluated. Instead of step 12, a direction that decreases the objective can be calculated as

$$\mathbf{d}_{decrease} = \mathbf{a}^{prior} + \mathbf{Cb} - \mathbf{a}_{old}^s$$

so a (approximate) gradient is given by

$$\left(\frac{\partial J}{\partial \mathbf{a}^s} \right)^T = \mathbf{a}^s - \mathbf{a}^{prior} - \mathbf{Cb}$$

3.6 Using the gradient

The objective and its gradient cannot directly be used in standard gradient-based minimization algorithms. A routine that evaluates the objective and the gradient also modifies the values of the adjoint states and model errors. However, the interface of the minimization algorithm is not equipped to handle these variables. Moreover, during a line search of the minimization algorithm, the gradient routine modifies the model errors. When the line search method rejects a step, the model errors must be reset to the last accepted values.

Modifications to the minimization algorithm:

- Before the gradient routine is called, the global boolean variables "bFirst" and "bStep" are set to "true" and "false" respectively.
- After every successful line search, "bStep" is set to "true".

Modifications to the objective and gradient routine:

- The values of the model errors must be "remembered" from a previous call to the routine ("static" variable in C++ or "persistent" in Matlab).
- At the beginning of the routine, the model errors must be initialized by zero if "bFirst" is equal to "true", after which "bFirst" is set to "false".
- All changes to the model errors are stored in temporary variables, which are "remembered" between successive calls to the routine.
- At the beginning of the routine, the model errors are overwritten by the temporary values if "bStep" equals "true", after which "bStep" is set to "false".

Especially because of the second modification to the minimizer, the method is not suited to interface with third party software without the availability of the source code. However, if the source code is available, making the modifications is almost trivial.

Unlike the gradient of the strong constraint problem that needs regularization, section 2.2, the gradient of the weak constraint problem, section 3.5, is already regularized by the representer expansions.

3.7 Choosing Q

Eq. (18) states that \mathbf{Q} must be chosen such that several rows of $\frac{\partial \mathbf{h}(\mathbf{x}_{\{0, \dots, n\}}^s)}{\partial \mathbf{x}_i^s}$ are removed by the multiplication

$$\mathbf{Q}^T \frac{\partial \mathbf{h}(\mathbf{x}_{\{0, \dots, n\}}^s)}{\partial \mathbf{x}_i^s}$$

Here, the terms $\frac{\partial \mathbf{h}}{\partial \mathbf{x}_i^s}$ are column blocks of the full measurement sensitivity matrix

$$\mathbf{J}_h = \frac{\partial \mathbf{h}(\mathbf{x}_{\{0, \dots, n\}}^s)}{\partial \mathbf{x}_{\{0, \dots, n\}}^s} = \left[\frac{\partial \mathbf{h}(\mathbf{x}_{\{0, \dots, n\}}^s)}{\partial \mathbf{x}_0^s} \quad \dots \quad \frac{\partial \mathbf{h}(\mathbf{x}_{\{0, \dots, n\}}^s)}{\partial \mathbf{x}_n^s} \right] \quad (23)$$

It makes sense to remove rows of \mathbf{J}_h rather than of the individual blocks. In other words, the same linear transformation is used to remove rows of all individual blocks, or \mathbf{Q} is the same for all time steps. Eq. (18) allows \mathbf{Q} to be different for different time steps, but then the operation in Eq. (22) would no longer be well defined.

For example, a singular value decomposition of \mathbf{J}_h can be used, so $\mathbf{J}_h = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$, where $\mathbf{\Sigma}$ has the form

$$\mathbf{\Sigma} = \begin{bmatrix} \sigma_1 & & 0 & 0 \\ & \ddots & 0 & 0 \\ & & \sigma_m & 0 & 0 \end{bmatrix}$$

and the bottom part of \mathbf{V}^T is filled with zeros. Extra zero-rows can be created in the product $\mathbf{\Sigma}\mathbf{V}^T = \mathbf{U}^T\mathbf{J}_h$ by setting the smallest singular values to zero, which is equivalent to removing rows of \mathbf{U}^T . Therefore the \mathbf{Q} matrix that is proposed in this article is

$$\mathbf{Q} = \mathbf{U}_{[:,1:k]}$$

which means that \mathbf{Q} is formed by calculating the left-singular vectors of \mathbf{J}_h in a matrix \mathbf{U} and then only k columns are kept. The singular values in $\mathbf{\Sigma}$ can even help to make a decision on the number of representer functions k , based on a preservation of energy principle.

4 Numerical experiments

4.1 Twin experiment: advection-diffusion

Experiments were done on a 2D advection-diffusion problem originating from petroleum reservoir engineering. One liquid phase (water) is injected into a petroleum reservoir to displace the other liquid phase (oil) towards the production wells. The state of the reservoir is described by pressure and water saturation in all 21x21x1 grid blocks of 10x10x20 m . The saturation of a liquid phase is the volume fraction of that phase within the total liquid volume. The evolution of the pressure with time is described by a diffusion equation and the evolution of saturation is described by an advection equation. Due to capillary forces in the pores of the reservoir rock, the pressure of the water and oil phase will not be equal. However, this effect is ignored, as well as gravity effects. Pressure differences can be translated into a velocity field which drives the advection equation. The diffusion coefficient of the diffusion equation is linearly dependent on the absolute permeability, which is a measure for how easy a liquid can flow through the pores of the reservoir rock. The diffusion coefficient is also weakly non-linear dependent on the pressure itself,

since the fluid properties like density and viscosity are usually prescribed by empirical functions of pressure. The saturation from the advection equation non-linearly affects the diffusion equation, since the reduction of the permeability of one phase is highly non-linearly influenced by the presence of the other phase. This effect is usually dealt with by empirical functions of saturation in the form of relative permeability curves.

In the experiments, synthetic data is generated by injecting water in the middle of the reservoir and producing oil (and unfortunately also water) at the corners. Water is injected at a rate of 1 pore volume per year. A pore volume is the total reservoir volume that is not occupied by rock, so the volume that is accessible to liquids. The pressures in the wells are measured after 100 and 200 days, resulting in 10 measurements. A database with 1000 possible permeability fields was available. One realization, Fig. (1) was used as "the truth". The others are used to construct a covariance matrix that is used in the objective function that has to be minimized. They are also used to construct basis functions that regularize the gradient of the strong constraint problem. In the twin experiment, the synthetic measurements are used to reconstruct the permeability field.

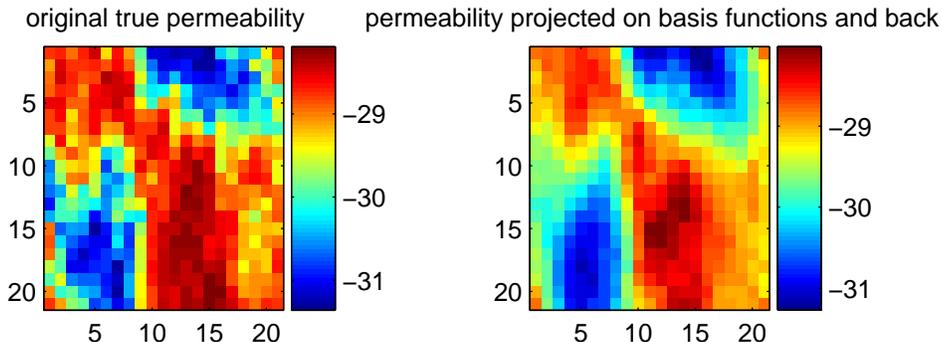


Figure 1: True permeability data $[\ln(m^2)]$ used to synthesize pressure data in the wells and best possible reconstruction using 25 basis functions.

4.2 Reservoir simulator in weak or stochastic mode

The 2-phase reservoir simulator can be written as

$$\frac{d}{dt}(\mathbf{f}_1(\mathbf{x})) = \mathbf{f}_2(\mathbf{x}, \mathbf{a})$$

where \mathbf{x} contains the water saturation and water pressure (equal to oil pressure) for every grid block and \mathbf{a} contains the permeabilities of all grid blocks. \mathbf{f}_1 describes the presence of water and oil mass in the grid blocks and \mathbf{f}_2 models the flow through the grid block interfaces. Injection/production is modelled as sources/sinks, which are included in \mathbf{f}_2 .

A fully implicit time discretization is used

$$\tilde{\mathbf{g}}(\mathbf{x}_i, \mathbf{x}_{i-1}, \mathbf{a}) = \mathbf{f}_1(\mathbf{x}_i) - (t_i - t_{i-1}) \mathbf{f}_2(\mathbf{x}_i, \mathbf{a}) - \mathbf{f}_1(\mathbf{x}_{i-1}) = \mathbf{0} \quad (24)$$

The model errors are introduced as additional sources/sinks in all grid blocks. In other words; after \mathbf{x}_i has been solved from Eq. (24), the water and oil masses in the grid blocks have not correctly been predicted and must still be modified. The prediction gets worse as the time step $(t_i - t_{i-1})$ gets larger. Therefore the correction is modelled proportional to $(t_i - t_{i-1})$. If the additional sources become too strong, then unrealistically high pressures will be observed. If the additional sinks become too strong, then saturations outside $[0, 1]$ will occur. In this article, the additional sinks are non-linearly constrained by \mathbf{f}_1 . The stochastic reservoir simulator has the form

$$\begin{aligned} \mathbf{g}(\mathbf{x}_i, \mathbf{x}_{i-1}, \mathbf{a}, \varepsilon_i) &= \\ = \mathbf{f}_1(\mathbf{x}_i) - (t_i - t_{i-1}) \mathbf{f}_2(\mathbf{x}_i, \mathbf{a}) - \mathbf{f}_1(\mathbf{x}_{i-1}) + \min\{\mathbf{f}_1(\mathbf{x}_i), (t_i - t_{i-1}) \varepsilon_i\} &= \mathbf{0} \end{aligned} \quad (25)$$

For the synthetic truth, ε_i is generated as white noise, Fig. (2). Applying this stochastic forcing to the reservoir simulator results in wiggly pressure and saturation responses in the wells, Fig (3). Although the pressure is not smooth in time, it is still smooth in space, Fig (4). This is not the case for the water saturation.

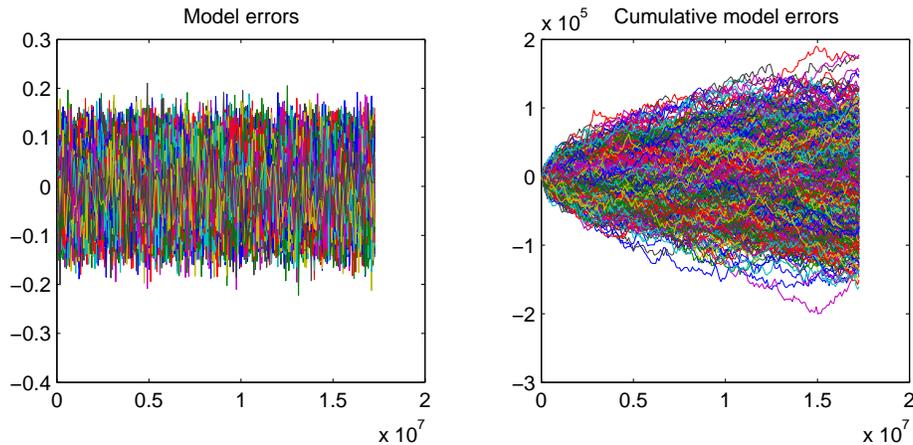


Figure 2: Model errors as functions of time

4.3 Permeability reconstructed

Four cases were examined; zero / low / middle / high model errors with standard deviations of 0 , $5 \cdot 10^{-4}$, $5 \cdot 10^{-3}$ and $5 \cdot 10^{-2} [kg s^{-1}]$ respectively. For the first case, the gradient of the strong constraint problem was used and regularized by the leading 25 left-singular vectors of \mathbf{P}_a . The best possible permeability that can be reconstructed using these basis functions is shown in Fig. (1). The gradient of the weak constraint problem was obtained and regularized by the Representer Expansions for the other cases. The figures from section 3.3 were obtained from the case with the highest model errors. The gradients were used in a steepest descent scheme with constant step size, and in the LBFGS

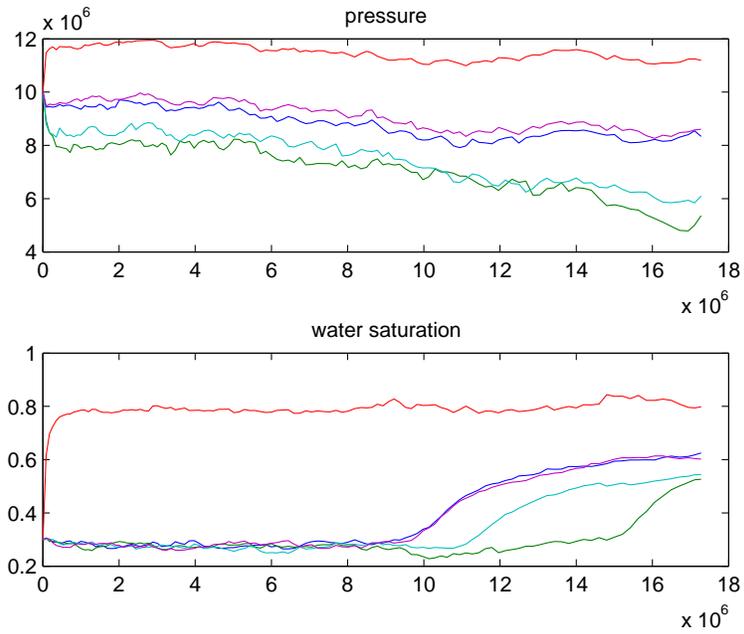


Figure 3: Pressure and saturation response in the wells when simulating with large model errors

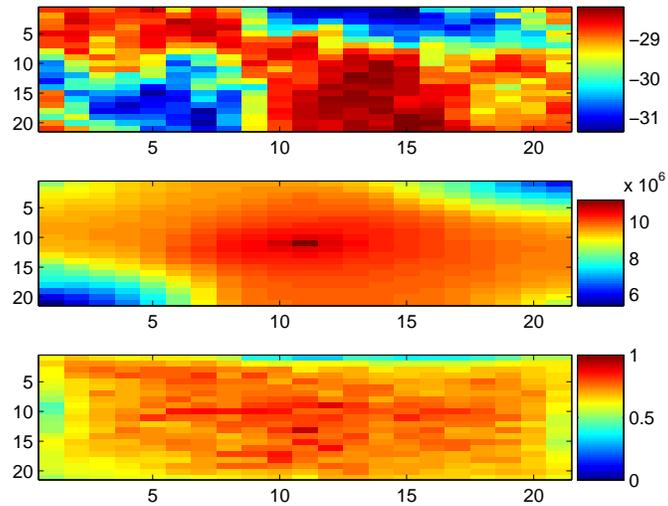


Figure 4: Spatial plots of permeability, pressure and water saturation at 200 days

[Gao and Reynolds, 2006, Ulbrich, 2002] algorithm with a line search based on Wolfe conditions [Nocedal and Wright, 1999]. Fig (5) shows the decrease of the objective function as function of iteration number for both minimization algorithms and all four cases.

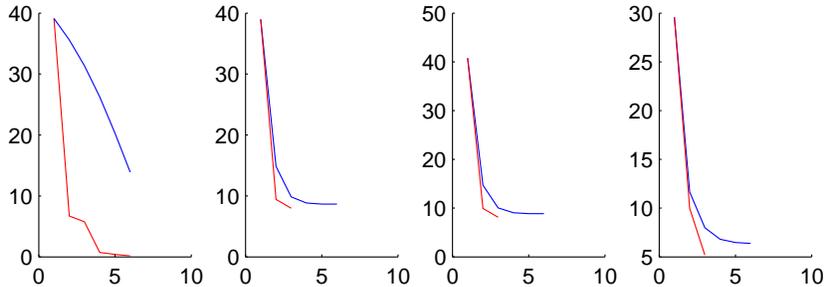


Figure 5: Objective function as function of iteration number. From left to right: zero/low/middle/high model errors. In red: steepest descent; in blue: LBFSGS

The prior permeability as well as the final reconstructions for the cases with zero / low / middle / high model errors using LBFSGS are shown in Fig. (6). In the strong constraint case, the gradient was regularized using 25 basis functions that were obtained as left-singular vectors of the permeability covariance matrix. The effect of different basis functions is shown in Fig. (7). The last picture is obtained using Representer Expansions; steps 2, 10 and 11 of section 3.4 can be ignored when the RM is used for solving a strong constraint problem.

4.4 Additional output from minimization process

Besides reconstructing parameters, the Order-reduced Non-linear Representer Method gives additional information. Fig. (8) shows the model errors that were reconstructed by LBFSGS for the case where the truth was synthesized using high model errors. Compared to the original, Fig. (2), they are underestimated, smoothed and biased. Fig. (9) shows the reconstructed pressure and saturation responses in the wells. These are smoothed as well. The parameter representer multiplied by their representer coefficients are plotted in Fig. (10). Notice the different scales; some measurements have a larger impact on the final permeability estimate than others, both in space (different columns) and time (different rows).

4.5 Order reduction

Fig. (6) was obtained without any order reduction (\mathbf{Q} equal to identity matrix). Fig. (11) was created using an order reduction by a factor two; \mathbf{Q} is obtained by a permutation of the columns of the identity matrix and then adding the right most columns to the left most columns. The resulting columns are then normalized. The permutation is different

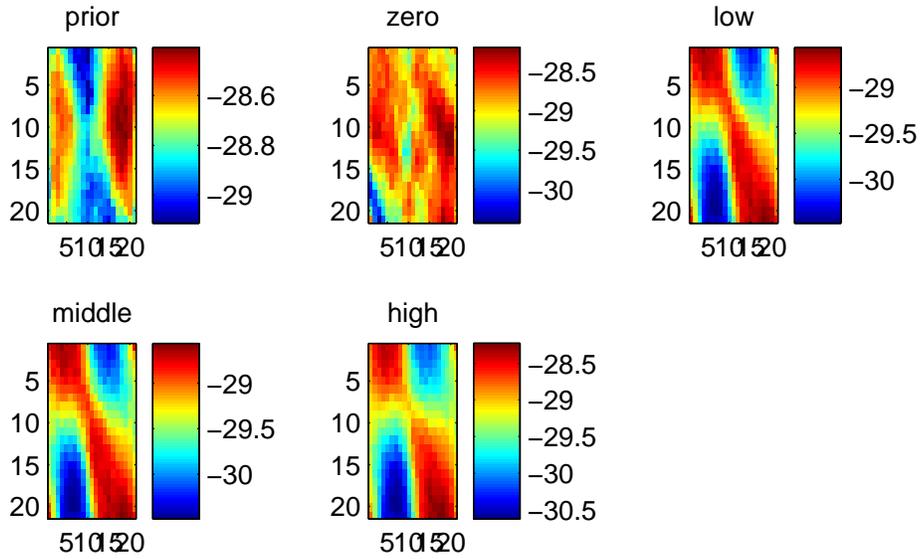


Figure 6: Prior permeability and final estimates in the cases with zero/low/middle/high model errors

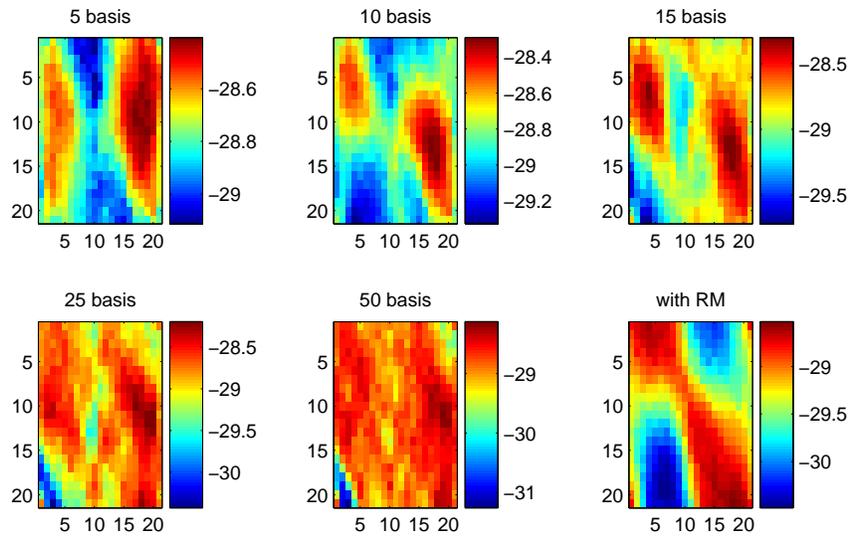


Figure 7: Final permeability estimates for the strong constraint case using different basis functions to regularize the gradient.

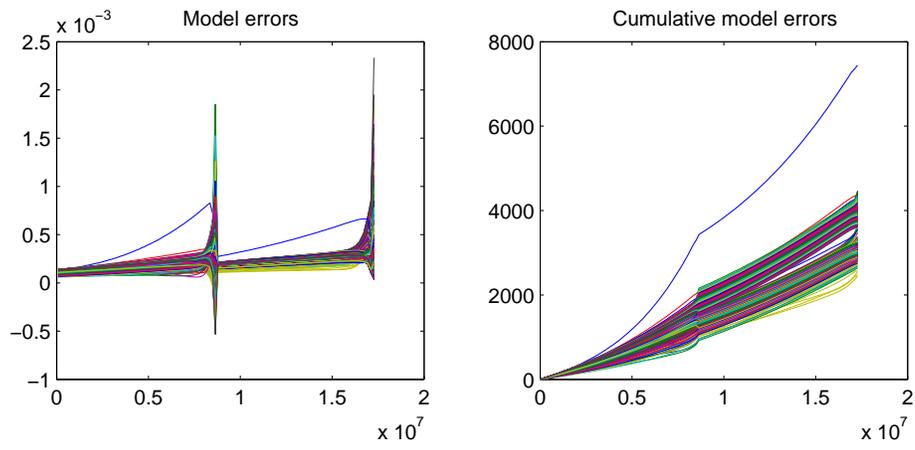


Figure 8: Model errors reconstructed by LBFSGS

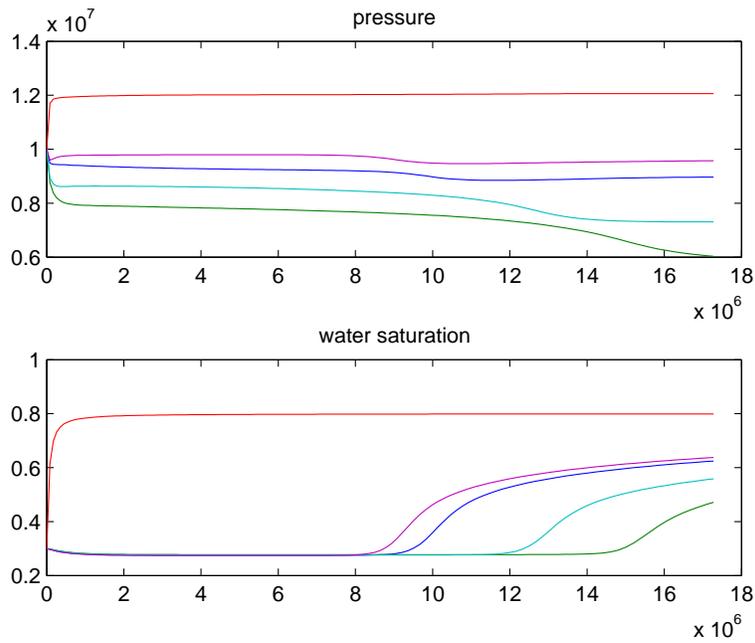


Figure 9: Pressure and saturation response in the wells reconstructed by LBFSGS

and random for every evaluation of the gradient. The stopping criterion is already reached after one iteration. If the columns of \mathbf{Q} are lumped even more, there comes a point where LBFGS reaches the stopping criterion after zero iterations.

5 Discussion

5.1 Strong constraint solver and the RM as post-processor

In theory, the weak constraint minimization problem can easily be turned into a strong constraint minimization problem by treating the model errors as additional model parameters

$$\tilde{\mathbf{a}} = [\mathbf{a}^T \quad \varepsilon_1^T \quad \cdots \quad \varepsilon_n^T]^T \quad (26)$$

Now any strong constraint solver can be used to solve the weak constraint problem. However, strong constraint solvers highly depend on regularization techniques or methods to reduce the order of the parameter space. Usually basis functions are chosen and used during the entire minimization process. The result is then accepted as the solution of the minimization problem. The RM discussed in this article can be used as a post-processor to evaluate the outcome of the strong constraint solver and to update or overwrite the user-defined basis function to initialize a new strong constraint solve.

5.2 Variable time steps

Most modern simulators are equipped with a time stepping mechanism that detects instabilities or unphysical values for the state variables and decreases the time step. Whenever possible the time step is increased again to reduce computation time. In an iterative method the length and the number of time steps therefore varies.

Building a strong constraint minimization problem out of a weak constraint one, as described in section 5.1, is not possible when successive iterations use different time steps, because the parameter vector Eq. (26) is only defined for one iteration. However, the Order-reduced Non-linear Representer Method can still be used. The model errors ε_i^s that were calculated in the old iteration must be interpolated to run the model in the new iteration Eq. (7). Here an integral average

$$\varepsilon_i^j = \frac{1}{t_i^j - t_{i-1}^j} \int_{t_{i-1}^j}^{t_i^j} \varepsilon^{j-1} dt$$

is used, where ε^{j-1} is the step function that is defined by

$$\{t_0^{j-1}, \dots, t_n^{j-1}\} \times \{\varepsilon_1^{j-1}, \dots, \varepsilon_n^{j-1}\}$$

from the old iteration.

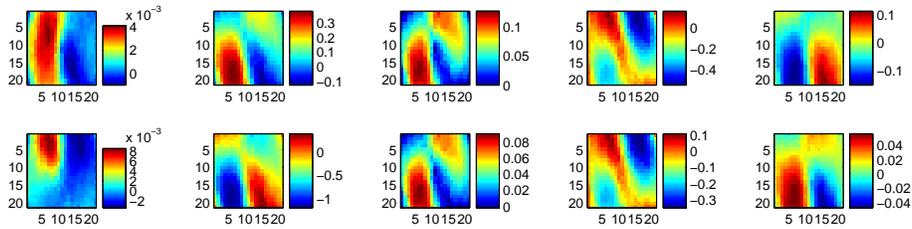


Figure 10: Parameter representers. Top row: pressure measurements after 100 days; bottom row: after 200 days; middle column: measurements obtained from injection well; other columns: north-west (NW), SW, NE and SE producers

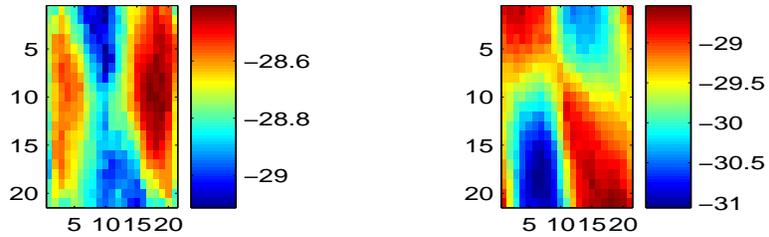


Figure 11: Prior permeability and final estimate for the case with high model errors using LBFGS and order reduction by a factor 2

An improvement in the parameters may cause the simulator to use more time steps, which may cause the term $\frac{1}{2}\varepsilon^T P_\varepsilon^{-1}\varepsilon$ with $\varepsilon^T = [\varepsilon_1 \ \cdots \ \varepsilon_n]^T$ in Eq. (2) to increase disproportionately. Normalization factors can be added, so

$$\begin{aligned}
J = & \frac{1}{2|\mathbf{m}|} (\mathbf{h}(\mathbf{x}_{\{0,\dots,n\}}) - \mathbf{m})^T P_{\mathbf{h}}^{-1} (\mathbf{h}(\mathbf{x}_{\{0,\dots,n\}}) - \mathbf{m}) + \\
& + \frac{1}{2|\mathbf{a}|} (\mathbf{a} - \mathbf{a}^{prior})^T \mathbf{P}_{\mathbf{a}}^{-1} (\mathbf{a} - \mathbf{a}^{prior}) + \frac{1}{2|\varepsilon|} \varepsilon^T P_\varepsilon^{-1} \varepsilon + \\
& + \sum_{i=1}^n \lambda_i^T \mathbf{g}(\mathbf{x}_i, \mathbf{x}_{i-1}, \mathbf{a}, \varepsilon_i)
\end{aligned} \tag{27}$$

where $|\cdot|$ stands for counting the number of elements in a vector.

5.3 Measure of success

Variational data assimilation methods are designed to minimize some data-misfit objective, Eq. (2). Their success can be measured by which (local) minimum they can find and how fast they can find it. However, different performance measures can be explored as well. For example in closed-loop reservoir management [Jansen et al., 2005], figures Fig. (3) and Fig. (9) can be compared. When water breaks through in production wells, they become financially less profitable and eventually have to be shut in. The goal is to predict water breakthrough long before the water actually arrives at the production wells, so different control strategies can be applied to postpone the water breakthrough. An alternative measure of success for a data assimilation algorithm can be how well the saturation profiles in the wells are reconstructed. The difference between Fig. (3) and Fig. (9) must therefore be quantified somehow. [Cheng et al., 2005] proposes to shift the curves in time to find a best fit; the shift quantifies how well the water breakthrough is estimated in time, the fit quantifies how well the behavior of the water during the breakthrough is estimated.

5.4 Use of parameter representers to quantify the impact of measurements

Fig. (10) shows the effect of every measurement on the final parameter estimate. Even when the new RM is used in order-reduced mode, then one extra iteration can be made after convergence of the method to produce all the parameter representers by running in full mode. Using the parameter representers, the usefulness of measurements must somehow be quantified, preferably as a monetary value. Care must be taken when interpreting these quantities. For example, a measurement can give a better understanding of the subsurface, but it might also indicate that oil production will be lower than prognosed. This does not mean that the impact of the measurement should be quantified with a negative number. Research in this area is ongoing. Once the effect of measurements can be quantified, this quantification technique can help in designing measurement strategies; it may for example help in seismic acquisition.

5.5 Data selection

Attempts have been made to preprocess the data and discard the data with the most uncertainty to reduce the number of representer functions and reduce the computation time. [Schwaighofer and Tresp, 2003] mentions the Random and the Sparse Greedy Matrix Approximation (SGMA) versions of the Subset of Representers Method (SRM). Such preprocessing is based on the measurement uncertainty matrix $P_{\mathbf{h}}$ and remains unchanged during the minimization process. This article proposes to choose a different preprocessing of the data at every iteration based on the measurement sensitivity matrix $\mathbf{J}_{\mathbf{h}} = \frac{\partial \mathbf{h}(\mathbf{x}_{\{0, \dots, n\}}^s)}{\partial \mathbf{x}_{\{0, \dots, n\}}^s}$.

Since the preprocessing itself also costs computation time, the selection from the full dataset can also be used for more than one iteration. These two criteria can also be combined. In that case $P_{\mathbf{h}}$ moves from Eq. (22) to Eq. (18) and the choice of \mathbf{Q} is based on $P_{\mathbf{h}}^{-1} \mathbf{J}_{\mathbf{h}}$, which looks like a good compromise between how much the engineer trusts the measured data ($P_{\mathbf{h}}$) and how sensitive the forecasted measurements are to changes in the state variables. These criteria are also used in the Greedy Posterior Approximation version of SRM [Schwaighofer and Tresp, 2003], although the number of basis functions is fixed.

5.6 Regularization

One might get the idea that solving the representer coefficients \mathbf{b} from Eq. (17) instead of Eq. (22) makes the state representer, Eq. (20), and measurement representer, Eq. (21), obsolete. This might be true if extra, user-defined, regularization is applied to the resulting gradient, since the state- and measurement representer functions are part of the regularization.

6 Conclusion

6.1 Summary

This article introduces a new formulation of the representer method. Since it can handle non-linear dynamics, non-linear measurement operators and non-linear model errors, it can handle situations that are more realistic than previous implementations [Valstar et al., 2004, Rommelse et al., 2006, Przybysz et al., 2007]. The derivation is explained and then the method is illustrated by estimating the permeability of a petroleum reservoir using a 2-phase 5-spot waterflood. Experiments were done comparing a strong constraint with weak constraints of different magnitude. The use of gradients of the strong and weak constraint problems in steepest descent and LBFGS minimization schemes is shown. An example is shown where the number of representer is reduced by a factor two, without degrading the quality of the final permeability estimate.

6.2 Classic RM and Order-reduced Non-linear RM

Things that are new in this formulation of the Representer Method:

- The new RM method does not solve the weak constraint minimization problem directly; it produces a regularized gradient that can be used by any gradient-based minimization algorithm (after minor modifications). Solving the minimization problem is then left to this algorithm.
- In the classic RM, the number of representer coefficients is equal to the number of measurements, meaning that \mathbf{D} in Eq. (22) is a square matrix. P_h is also square and can therefore be added to \mathbf{D} without modifications; \mathbf{Q} is thus chosen equal to the identity matrix, $\mathbf{Q} = \mathbf{I}$. In applications where there are many measurements, computational feasibility is created by choosing the number of representer coefficients (much) smaller than the number of measurements. \mathbf{D} has then more rows than columns, and \mathbf{Q} must be used to reduce the number of columns in $P_h\mathbf{Q}$ so it can be added to \mathbf{D} in Eq. (22).
- In previous non-linear versions of RM [Valstar et al., 2004], the state variables are decomposed around the results of the last simulation, rather than the prior. To correct for this, an extra correction term is introduced and an additional equation is needed to compute this correction term. The correction term is used to force a decoupling of the representer equations in the classical method. In section 3.3 the representer equations are decoupled and can be used in the same order as in the classical method, without using any correction terms. In the classic method, the state variables are essentially also decomposed around the prior, although the prior is never explicitly available in computer memory. If it were available, the correction term could easily be calculated as the difference between the prior and the last available forecast of the state variables. Alternatively the equations can be rewritten in terms of a forecast and a prior rather than a forecast and a correction, as is done in the new RM. One might argue that a linearization around the prior is only valid as long as variables stay close to the prior. This argument applies to both the classic and the new RM. In both methods, the representers are calculated by evaluating the Jacobians using the last predicted state variables. Therefore both methods use the same linearization. An extra iteration loop can be used, where a new prior is chosen equal to the final estimate after the method has converged. The new prior is then used to initiate a new minimization.
- In this article the term "measurement representers" is first introduced. The classic RM is derived without these basis functions, probably because in the case of a linear measurement operator, the measurement representers are constructed by concatenating the state representers of different time steps as row blocks into one large matrix.

6.3 Conclusions

- The permeability estimates were better using weak constraints than strong constraints. In the strong constraint case, the regularization is prescribed by the user, giving him/her an extra opportunity to introduce his/her judgement (prejudice) into the final estimate. In the weak constraint case, the regularization is data-driven, but can also be assisted with extra user-defined regularization or data preprocessing. Representer Expansions can also be used in the strong constraint case, giving better estimates at the expense of extra computation time.
- LBFGS decreases the objective function in much fewer iterations than steepest descent, especially in the strong constraint case. However, due to the line search, one iteration of LBFGS typically costs twice as much computation time than one iteration of steepest descent. Still LBFGS is faster.
- Order reduction can be used to reduce computation time without loss of quality of the estimated parameters. When the order is reduced to the extreme, the minimization algorithm reaches the stopping criterion before performing any iterations.

References

- [Baird and Dawson, 2005] Baird, J. and Dawson, C. (2005). The representer method for data assimilation in single-phase darcy flow in porous media. *Computational Geosciences*, 9(4):247–271.
- [Bennett, 2002] Bennett, A. F. (2002). *Inverse modeling of the ocean and atmosphere*. Cambridge University Press.
- [Bennett, 2006] Bennett, A. F. (2006). *Lagrangian Fluid Dynamics*. Cambridge Monographs on Mechanics. Cambridge University Press, New York.
- [Bennett and McIntosh, 1982] Bennett, A. F. and McIntosh, P. C. (1982). Open ocean modeling as an inverse problem; tidal theory. *Journal of Physical Oceanography*, 12(10):1004–1018.
- [Cheng et al., 2005] Cheng, H., Datta-Gupta, A., and Zhong, H. (2005). A comparison of travel time and amplitude inversion for production data integration into geologic models: Sensitivity, non-linearity and practical implications. *SPE Journal*, 10(1):75–90.
- [Eknes and Evensen, 1997] Eknes, M. and Evensen, G. (1997). Parameter estimation solving a weak constraint formulation for an ekman flow model. *Journal of Geophysical Research*, 102(6C):12479–12491.

- [Gao and Reynolds, 2006] Gao, G. and Reynolds, A. C. (2006). An improved implementation of the lbfgs algorithm for automatic history matching. *SPE Journal*, 11(1):5 – 17.
- [Jansen et al., 2005] Jansen, J. D., Brouwer, D. R., Naevdal, G., and van Kruijsdijk, C. P. J. W. (2005). Closed-loop reservoir management. *First Break*, 23:43–48.
- [Janssen et al., 2006] Janssen, G. M. C. M., Valstar, J. R., and van der Zee, S. E. A. T. M. (2006). Inverse modeling of multimodal conductivity distributions. *Water Resources Research*, 42(3):Art. No. W03410.
- [Nocedal and Wright, 1999] Nocedal, J. and Wright, S. J. (1999). *Numerical Optimization*. Springer Verlag, New York.
- [Przybysz et al., 2007] Przybysz, J. K., Hanea, R. G., Jansen, J. D., and Heemink, A. W. (2007). Application of the representer method for parameter estimation in numerical reservoir models. *Computational Geosciences*.
- [Rommelse et al., 2006] Rommelse, J. R., Kleptsova, O., Jansen, J. D., and Heemink, A. W. (2006). Data assimilation in reservoir management using the representer method and the ensemble kalman filter. In *Proceedings of the 10th European Conference on the Mathematics of Oil Recovery*, Amsterdam, Netherlands.
- [Schwaighofer and Tresp, 2003] Schwaighofer, A. and Tresp, V. (2003). Transductive and inductive methods for approximate gaussian process regression. *Advances in Neural Information Processing Systems*, 15:953 – 960.
- [Ulbrich, 2002] Ulbrich, S. (2002). Limited-memory bfgs-verfahren mit powell-wolfe schrittweitenregel.
- [Valstar et al., 2004] Valstar, J. R., McLaughlin, D. B., te Stroet, C. B. M., and van Geer, F. C. (2004). A representer-based inverse method for groundwater flow and transport applications. *Water Resources Research*, 40.