

Computing derivative information of sequentially coupled subsurface models

de Moraes, Rafael J.; Rodrigues, José R.P.; Hajibeygi, Hadi; Jansen, Jan Dirk

DOI

[10.1007/s10596-018-9772-2](https://doi.org/10.1007/s10596-018-9772-2)

Publication date

2018

Document Version

Final published version

Published in

Computational Geosciences

Citation (APA)

de Moraes, R. J., Rodrigues, J. R. P., Hajibeygi, H., & Jansen, J. D. (2018). Computing derivative information of sequentially coupled subsurface models. *Computational Geosciences*, 22(6), 1527–1541. <https://doi.org/10.1007/s10596-018-9772-2>

Important note

To cite this publication, please use the final published version (if applicable).
Please check the document version above.

Copyright

Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

Takedown policy

Please contact us and provide details if you believe this document breaches copyrights.
We will remove access to the work immediately and investigate your claim.



Computing derivative information of sequentially coupled subsurface models

Rafael J. de Moraes^{1,2} · José R. P. Rodrigues² · Hadi Hajibeygi¹ · Jan Dirk Jansen¹

Received: 6 March 2018 / Accepted: 28 August 2018
© The Author(s) 2018

Abstract

A generic framework for the computation of derivative information required for gradient-based optimization using sequentially coupled subsurface simulation models is presented. The proposed approach allows for the computation of any derivative information with no modification of the mathematical framework. It only requires the forward model Jacobians and the objective function to be appropriately defined. The flexibility of the framework is demonstrated by its application in different reservoir management studies. The performance of the gradient computation strategy is demonstrated in a synthetic water-flooding model, where the forward model is constructed based on a sequentially coupled flow-transport system. The methodology is illustrated for a synthetic model, with different types of applications of data assimilation and life-cycle optimization. Results are compared with the classical fully coupled (FIM) forward simulation. Based on the presented numerical examples, it is demonstrated how, without any modifications of the basic framework, the solution of gradient-based optimization models can be obtained for any given set of coupled equations. The sequential derivative computation methods deliver similar results compared to FIM methods, while being computationally more efficient.

Keywords Sequential coupling · Gradient-based optimization · Data assimilation · Life-cycle optimization · Adjoint method · Direct method

1 Introduction

The exploitation of subsurface resources frequently involves complex physics and geology. Thermal, geomechanical, and chemical processes are just a few phenomena that sometimes must be accounted for, while the domain is often governed by parameters that typically change several orders

of magnitude over a wide range of spatial scales. Numerical simulation of such complex processes can be done with fully implicit methods (FIMs) and sequentially coupled approaches. Even though FIMs provide the most stable simulation platform [2], many efforts have successfully lead to stable and efficient sequential simulations (e.g., [21]). Sequential simulation is often specially attractive for coupled processes of different physical natures, which often operate on different time scales or have different spatial support (e.g., local versus global effects). It is worth to be mentioned that multiscale methods [12, 14, 18, 26] and model reduction techniques [3, 7] have been mainly developed for globally acting processes, and thus function optimally when they are used in sequential frameworks. Note that such a framework would also benefit some preconditioning methods that use sequential strategies (operator splitting) for the solution of linear system of equations. This holds for FIMs simulations using constrained pressure restriction (CPR) preconditioning [6, 13, 35], where a pressure-like system is being extracted from the FIM Jacobian in order to enhance the convergence.

Ultimately, numerical simulation will support reservoir management studies which are often based on optimization

✉ Rafael J. de Moraes
r.moraes@tudelft.nl
José R. P. Rodrigues
jrprodrigues@petrobras.com.br
Hadi Hajibeygi
h.hajibeygi@tudelft.nl
Jan Dirk Jansen
j.d.jansen@tudelft.nl

¹ Department of Geoscience and Engineering, Faculty of Civil Engineering and Geosciences, Delft University of Technology, P.O. Box 5048, 2600, Delft, The Netherlands

² Petrobras Research and Development Center – CENPES, Av. Horácio Macedo 950, Cidade Universitária, Rio de Janeiro, RJ 21941-915, Brazil

techniques. It has been shown that gradient-based optimization techniques are the most efficient ones when applied, for instance, to life-cycle optimization [15, 31] and history matching [4, 22, 28] studies. Moreover, it is well known that the most efficient/accurate gradient computation technique is the adjoint method [15]. Even though a large body of the literature has been dedicated to this topic, most of it discusses the adjoint model for FIMs systems. In this case, the adjoint model is obtained by transposing the forward model system of equations [22, 31]. Also, even though the mathematical framework presented by [30] and [19] does not limit the derivation of the adjoint equations to any particular solution strategy, no explicit discussion on how it can be applied to sequentially coupled system of equations was presented. A multiscale adjoint method applied to life-cycle optimization is presented by [20], in which a sequential solution of flow and transport is employed, such that, consequentially, the adjoint model also follows a sequential solution strategy. However, in that work, the discussion is focused on the promising computational savings provided by multiscale simulation and not so much detail is given as to what extent the gradient computation itself can impose challenges.

The present work presents a general gradient computation formulation for sequentially coupled models. An implicit differentiation strategy [19, 30] is extended to coupled systems of equations. The algorithms for the derivative computation of simulator responses neither depend on the objective function type, nature of the parameters, nor on any specific model coupling. Instead, it is shown how derivative information can be computed based on any coupling strategy. Using a chain-rule formalism, we firstly introduce a generic framework capable of computing the specific derivative information required by any given optimization algorithm. Next, it is shown how such computation is done for sequentially coupled flow and transport. Thereafter, numerical examples including both data assimilation and life-cycle optimization are presented.

2 Mathematical framework for the computation of gradient information of coupled system of equations

We consider a system of discrete-in-time non-linear model equations in implicit (residual) form:

$$\begin{cases} \mathbf{g}_{\mathbf{x}_1}^n(\mathbf{x}_1^{n-1}, \dots, \mathbf{x}_{N_c}^{n-1}, \mathbf{x}_1^n, \dots, \mathbf{x}_{N_c}^n, \theta) = 0 \\ \vdots \\ \mathbf{g}_{\mathbf{x}_{N_c}}^n(\mathbf{x}_1^{n-1}, \dots, \mathbf{x}_{N_c}^{n-1}, \mathbf{x}_1^n, \dots, \mathbf{x}_{N_c}^n, \theta) = 0, \end{cases} \quad (1)$$

where $\mathbf{x}_c^n \in \mathbf{R}^{N_{x_c}^n}$, $c \in \{1, \dots, N_c\}$, is the set of primary variables associated with the model equations $\mathbf{g}_{\mathbf{x}_c}^n$, and N_c

the total number of coupled equations. The superscript n denotes the time-step index and $\theta \in \mathbf{R}^{N_\theta}$ is the vector of parameters with respect to which we aim to compute derivative information. There are $N_X^n = \sum_{c=1}^{N_c} N_{x_c}^n$ primary variables at time-step n and N_θ parameters. Note that the initial conditions are assumed to be

$$\begin{cases} \mathbf{g}_{\mathbf{x}_1}^0(\mathbf{x}_1^0, \dots, \mathbf{x}_{N_c}^0, \theta) = 0 \\ \vdots \\ \mathbf{g}_{\mathbf{x}_{N_c}}^0(\mathbf{x}_1^0, \dots, \mathbf{x}_{N_c}^0, \theta) = 0. \end{cases} \quad (2)$$

The functions defining the set of observable responses for a time-step are described as

$$\mathbf{y}^n = \mathbf{h}^n(\mathbf{x}_1^{n-1}, \dots, \mathbf{x}_{N_c}^{n-1}, \mathbf{x}_1^n, \dots, \mathbf{x}_{N_c}^n, \theta), \quad (3)$$

where \mathbf{h}^n represents the output equations [17]. There are N_Y^n observations in time-step n .

Let

$$\mathbf{g}^n(\mathbf{x}^n, \mathbf{x}^{n-1}, \theta) = \begin{pmatrix} \mathbf{g}_{\mathbf{x}_1}^n \\ \vdots \\ \mathbf{g}_{\mathbf{x}_{N_c}}^n \end{pmatrix}, \quad (4)$$

be the set of model equations, where $\mathbf{g}^n : \mathbf{R}^{N_X^{n-1} \times N_X^n \times N_\theta} \rightarrow \mathbf{R}^{N_Y^n}$,

$$\mathbf{x}^n = \begin{pmatrix} \mathbf{x}_1^n \\ \vdots \\ \mathbf{x}_{N_c}^n \end{pmatrix}, \quad (5)$$

be the state vector, where $\mathbf{x}^n \in \mathbf{R}^{N_X^n}$, and Eq. 3 be redefined as

$$\mathbf{y}^n = \mathbf{h}^n(\mathbf{x}^{n-1}, \mathbf{x}^n, \theta), \quad (6)$$

where $\mathbf{h}^n : \mathbf{R}^{N_X^{n-1} \times N_X^n \times N_\theta} \rightarrow \mathbf{R}^{N_Y^n}$.

A “super-vector” notation [19, 30] is used to capture the evolution in time. All instances of \mathbf{g}^n as defined in Eq. 4 for all time-steps, can be collated in a function $\mathbf{g} : \mathbf{R}^{N_X \times N_\theta} \rightarrow \mathbf{R}^{N_X}$, where $N_X = \sum_{n=0}^N N_X^n$ is the total number of primary variables for all time-steps, such that the system of non-linear equations is represented as

$$\mathbf{g}(\mathbf{x}(\theta), \theta) = 0. \quad (7)$$

Note that we use bold italic font to indicate super vectors and just bold to indicate ordinary vectors.

Equation 7 indicates the dependency of the forward model equations on both the primary variables and the model parameters, even though the model equations are only solved for \mathbf{x} and the dependency on θ has to be taken into account for the implicit differentiation strategy that will be employed later on.

Similarly, all instances of \mathbf{y}^n as defined in Eq. 6 for all time-steps can be collated in a function $\mathbf{h} : \mathbf{R}^{N_X \times N_\theta} \rightarrow$

\mathbf{R}^{N_Y} , where $N_Y = \sum_{n=0}^N N_Y^n$ represents the total number of responses for all time-steps, so that

$$\mathbf{y} = \mathbf{h}(\mathbf{x}(\theta), \theta). \quad (8)$$

Following the same implicit differentiation strategy as in [25, 30], the sensitivity matrix \mathbf{G} (i.e., sensitivity of the responses with respect to the parameters) can be computed by deriving Eq. 8 with respect to θ , i.e.,

$$\mathbf{G} = \frac{d\mathbf{h}}{d\theta} = \frac{\partial \mathbf{h}}{\partial \mathbf{x}} \frac{d\mathbf{x}}{d\theta} + \frac{\partial \mathbf{h}}{\partial \theta}. \quad (9)$$

In order to find a relationship that defines $\frac{d\mathbf{x}}{d\theta}$, Eq. 7 is differentiated with respect to θ

$$\frac{\partial \mathbf{g}}{\partial \mathbf{x}} \frac{d\mathbf{x}}{d\theta} + \frac{\partial \mathbf{g}}{\partial \theta} = \mathbf{0}, \quad (10)$$

so that

$$\frac{d\mathbf{x}}{d\theta} = -\left(\frac{\partial \mathbf{g}}{\partial \mathbf{x}}\right)^{-1} \frac{\partial \mathbf{g}}{\partial \theta}. \quad (11)$$

Substituting Eq. 11 in Eq. 9 gives

$$\mathbf{G} = -\frac{\partial \mathbf{h}}{\partial \mathbf{x}} \left(\frac{\partial \mathbf{g}}{\partial \mathbf{x}}\right)^{-1} \frac{\partial \mathbf{g}}{\partial \theta} + \frac{\partial \mathbf{h}}{\partial \theta}. \quad (12)$$

In order to keep the framework general (in terms of which type of derivative information can be computed), the sensitivity matrix is pre- and post-multiplied by arbitrary matrices \mathbf{V} (of size $N_\theta \times p$) and \mathbf{W} (of size $m \times N_Y$)

$$\mathbf{WGV} = -\mathbf{W} \frac{\partial \mathbf{h}}{\partial \mathbf{x}} \left(\frac{\partial \mathbf{g}}{\partial \mathbf{x}}\right)^{-1} \frac{\partial \mathbf{g}}{\partial \theta} \mathbf{V} + \mathbf{W} \frac{\partial \mathbf{h}}{\partial \theta} \mathbf{V}. \quad (13)$$

The key aspect that defines the computational performance of the gradient computation is the order of the operations involving $\left(\frac{\partial \mathbf{g}}{\partial \mathbf{x}}\right)^{-1}$. Based on that, both the direct [1] and adjoint [4] analytical methods to compute the necessary derivative information can be defined.

If \mathbf{W} is factored out in Eq. 13, it can be re-written as

$$\mathbf{GV} = \frac{\partial \mathbf{h}}{\partial \mathbf{x}} \mathbf{Z} + \frac{\partial \mathbf{h}}{\partial \theta} \mathbf{V}, \quad (14)$$

where

$$\mathbf{Z} = -\left(\frac{\partial \mathbf{g}}{\partial \mathbf{x}}\right)^{-1} \frac{\partial \mathbf{g}}{\partial \theta} \mathbf{V}, \quad (15)$$

is solved from

$$\left(\frac{\partial \mathbf{g}}{\partial \mathbf{x}}\right) \mathbf{Z} = -\frac{\partial \mathbf{g}}{\partial \theta} \mathbf{V}. \quad (16)$$

The linear system described in Eq. 16 can be re-written in a block-wise form for each time-step n :

$$\begin{pmatrix} \frac{\partial \mathbf{g}^0}{\partial \mathbf{x}^0} & & & \\ \frac{\partial \mathbf{g}^1}{\partial \mathbf{x}^0} & \frac{\partial \mathbf{g}^1}{\partial \mathbf{x}^1} & & \\ \frac{\partial \mathbf{g}^2}{\partial \mathbf{x}^0} & \frac{\partial \mathbf{g}^2}{\partial \mathbf{x}^1} & \ddots & \\ & \ddots & \ddots & \frac{\partial \mathbf{g}^N}{\partial \mathbf{x}^{N-1}} \end{pmatrix} \begin{pmatrix} \mathbf{Z}^0 \\ \mathbf{Z}^1 \\ \vdots \\ \mathbf{Z}^N \end{pmatrix} = -\begin{pmatrix} \frac{\partial \mathbf{g}^0}{\partial \theta} \mathbf{V} \\ \frac{\partial \mathbf{g}^1}{\partial \theta} \mathbf{V} \\ \vdots \\ \frac{\partial \mathbf{g}^N}{\partial \theta} \mathbf{V} \end{pmatrix}, \quad (17)$$

where, from Eqs. 4 and 5 one can write

$$\frac{\partial \mathbf{g}^n}{\partial \mathbf{x}^n} = \begin{pmatrix} \frac{\partial \mathbf{g}_{\mathbf{x}_1}^n}{\partial \mathbf{x}_1^n} & \cdots & \frac{\partial \mathbf{g}_{\mathbf{x}_{N_c}}^n}{\partial \mathbf{x}_{N_c}^n} \\ \vdots & \ddots & \vdots \\ \frac{\partial \mathbf{g}_{\mathbf{x}_{N_c}}^n}{\partial \mathbf{x}_1^n} & \cdots & \frac{\partial \mathbf{g}_{\mathbf{x}_{N_c}}^n}{\partial \mathbf{x}_{N_c}^n} \end{pmatrix}, \quad (18)$$

and

$$\frac{\partial \mathbf{g}^n}{\partial \mathbf{x}^{n-1}} = \begin{pmatrix} \frac{\partial \mathbf{g}_{\mathbf{x}_1}^n}{\partial \mathbf{x}_1^{n-1}} & \cdots & \frac{\partial \mathbf{g}_{\mathbf{x}_{N_c}}^n}{\partial \mathbf{x}_{N_c}^{n-1}} \\ \vdots & \ddots & \vdots \\ \frac{\partial \mathbf{g}_{\mathbf{x}_{N_c}}^n}{\partial \mathbf{x}_1^{n-1}} & \cdots & \frac{\partial \mathbf{g}_{\mathbf{x}_{N_c}}^n}{\partial \mathbf{x}_{N_c}^{n-1}} \end{pmatrix}. \quad (19)$$

Here N is the total number of time-steps and the partitioning lines indicate which matrix and vector terms belong to each time-step. Also, from Eqs. 6 and 5

$$\frac{\partial \mathbf{h}^n}{\partial \mathbf{x}^n} = \left(\frac{\partial \mathbf{h}^n}{\partial \mathbf{x}_1^n} \cdots \frac{\partial \mathbf{h}^n}{\partial \mathbf{x}_{N_c}^n} \right), \quad (20)$$

and

$$\frac{\partial \mathbf{h}^n}{\partial \mathbf{x}^{n-1}} = \left(\frac{\partial \mathbf{h}^n}{\partial \mathbf{x}_1^{n-1}} \cdots \frac{\partial \mathbf{h}^n}{\partial \mathbf{x}_{N_c}^{n-1}} \right). \quad (21)$$

This solution strategy is known in the literature as the *Forward Method* [30], *Gradient Simulator* [1], or *Direct Method* [28]. Note that auxiliary matrix \mathbf{Z} has dimensions of $N_X \times p$ and, therefore, it requires $N \times p$ linear systems to be solved. Hence, the cost of computing \mathbf{GV} is proportional to the number of columns in \mathbf{V} , i.e., p .

Now, if \mathbf{V} is factored out in Eq. 13, the equation can be re-written as

$$\mathbf{WG} = \mathbf{Z} \frac{\partial \mathbf{g}}{\partial \theta} + \mathbf{W} \frac{\partial \mathbf{h}}{\partial \theta}, \quad (22)$$

where

$$\mathbf{Z} = -\mathbf{W} \frac{\partial \mathbf{h}}{\partial \mathbf{x}} \left(\frac{\partial \mathbf{g}}{\partial \mathbf{x}}\right)^{-1} \quad (23)$$

is solved from

$$\mathbf{Z} \left(\frac{\partial \mathbf{g}}{\partial \mathbf{x}} \right) = -\mathbf{W} \frac{\partial \mathbf{h}}{\partial \mathbf{x}}. \quad (24)$$

The linear system described in Eq. 24 can be re-written in a block-wise form for each time-step n as

$$\begin{aligned} (\mathbf{Z}^0 | \mathbf{Z}^1 | \dots | \mathbf{Z}^N) \times & \begin{pmatrix} \frac{\partial \mathbf{g}^0}{\partial \mathbf{x}^0} & & & \\ \frac{\partial \mathbf{g}^1}{\partial \mathbf{x}^0} & \frac{\partial \mathbf{g}^1}{\partial \mathbf{x}^1} & & \\ \frac{\partial \mathbf{g}^2}{\partial \mathbf{x}^0} & \frac{\partial \mathbf{g}^2}{\partial \mathbf{x}^1} & & \\ & \ddots & \ddots & \\ & & \frac{\partial \mathbf{g}^N}{\partial \mathbf{x}^{N-1}} & \frac{\partial \mathbf{g}^N}{\partial \mathbf{x}^N} \end{pmatrix} \\ = -(\mathbf{W}^0 | \mathbf{W}^1 | \dots | \mathbf{W}^N) & \\ \times \begin{pmatrix} \frac{\partial \mathbf{h}^0}{\partial \mathbf{x}^0} & & & \\ \frac{\partial \mathbf{h}^1}{\partial \mathbf{x}^0} & \frac{\partial \mathbf{h}^1}{\partial \mathbf{x}^1} & & \\ \frac{\partial \mathbf{h}^2}{\partial \mathbf{x}^0} & \frac{\partial \mathbf{h}^2}{\partial \mathbf{x}^1} & & \\ & \ddots & \ddots & \\ & & \frac{\partial \mathbf{h}^N}{\partial \mathbf{x}^{N-1}} & \frac{\partial \mathbf{h}^N}{\partial \mathbf{x}^N} \end{pmatrix} \end{aligned} \quad (25)$$

One should note that Eq. 25 is solved backward in time. Now, by taking the transpose of Eq. 25, the linear system of equations that must be solved for each time-step for the adjoint method reads

$$\begin{aligned} (\mathbf{Z}^n)^T = & \left(\frac{\partial \mathbf{g}^n}{\partial \mathbf{x}^n} \right)^{-T} \times \left(-\left(\mathbf{W}^n \frac{\partial \mathbf{h}^n}{\partial \mathbf{x}^n} \right)^T \right. \\ & \left. - \left(\mathbf{W}^{n+1} \frac{\partial \mathbf{h}^{n+1}}{\partial \mathbf{x}^n} \right)^T - \left(\frac{\partial \mathbf{g}^{n+1}}{\partial \mathbf{x}^n} \right)^T (\mathbf{Z}^{n+1})^T \right). \end{aligned} \quad (26)$$

This solution strategy is known in the literature as the *Adjoint* (or *Backward Method*) (Chavent, 1975). Note that now \mathbf{Z} has dimensions of $N_X \times m$, hence it requires $N \times m$ linear systems to be solved. As such, the cost of computing $\mathbf{W}\mathbf{G}$ is proportional to the number of rows in \mathbf{W} , i.e., m .

Although the derivation as presented so far is general, in order to properly formulate the actual method to analytically compute the derivative information, the structure of the partial derivative matrices involved in the computations must be taken into account. This is only possible if the specific coupling strategy and the proper dependencies of the model equations and primary variables are taken into account. Therefore, in the rest for the paper we focus our studies on sequentially coupled multiphase flow simulations.

2.1 Remarks about the framework

The appropriate selection of the arbitrary matrices \mathbf{W} and \mathbf{V} allows one single framework to compute any derivative

information and avoids the expensive computation of \mathbf{G} . For instance, in case of quasi-Newton methods [27], the gradient of the objective function $O = O(\mathbf{y}(\boldsymbol{\theta}))$ is directly required. Via the chain-rule, one can write

$$\nabla_{\boldsymbol{\theta}} O = \left(\frac{dO}{d\boldsymbol{\theta}} \right)^T = \left(\frac{dO}{d\mathbf{h}} \frac{d\mathbf{h}}{d\boldsymbol{\theta}} \right)^T = \mathbf{G}^T \nabla_{\mathbf{h}} O. \quad (27)$$

The operation $(\mathbf{W}\mathbf{G})^T = \mathbf{G}^T \mathbf{W}^T$ gives the product of \mathbf{G}^T with the (column) vector $\mathbf{W}^T = \nabla_{\mathbf{h}} O$. Hence, the adjoint method can be efficiently employed to compute the objective function gradient with respect to the parameters, as described in Eq. 27.

Now, in case of conjugate gradient methods [27], products of \mathbf{G} and \mathbf{G}^T with arbitrary vectors are required. The product $\mathbf{G}\mathbf{V}$, with $n = 1$ can be efficiently computed by the direct method while, the product $\mathbf{G}^T \mathbf{W}^T$, with $m = 1$ can be efficiently computed using the adjoint method.

Another factor that maintains the flexibility of the framework is the formal partitioning of \mathbf{g} and \mathbf{x} according to the coupling of the equations. The computation of the auxiliary matrix \mathbf{Z} in Eqs. 15 and 26 will follow the partition of the \mathbf{g} . Once \mathbf{Z} is fully determined, the sensitivity matrix products (14) and (22) remain unchanged. Hence, the framework requires the Jacobians of \mathbf{g} w.r.t. \mathbf{x} and $\boldsymbol{\theta}$ to be determined from the coupled forward model equations.

We highlight that linear system solutions involving $\frac{\partial \mathbf{g}}{\partial \mathbf{x}}$ are required on both direct and adjoint methods (see Eqs. 16 and 24) in order to determine the auxiliary matrix \mathbf{Z} . More specifically, the unique derivative information computation requires $\frac{\partial \mathbf{g}^n}{\partial \mathbf{x}^n}$ to be full-rank. This is true in most of the cases given that this is the same partial derivative matrix required by the forward simulation. For instance, this matrix represents the Jacobian used by Newton-Raphson non-linear solvers, typically employed in the forward simulation.

The importance of the implementation separation between the forward model and the adjoint model was previously highlighted by [31], who also presented a discussion about memory requirements related to the storage of the partial derivatives (or states required to re-evaluate them during the backward runs). In that work, the computational aspects were discussed in the context of an optimal control problem using FIM. Note that, as shown in the previous section, the framework presented in our paper is readily applicable to different coupling strategies or derivative computation problems.

Also, both the direct and adjoint methods are treated in the same framework. The direct method is usually directly associated with the forward simulation. All the derivative information related computation is usually presented as part of the forward time-stepping process. Here, it is shown that it can also be achieved in complete separation from the

forward simulation. The requirement is the same as for the adjoint method: the required Jacobians must be stored/re-evaluated for the derivative information required at a later stage. However, on the one hand, the separation from the forward simulation reduces the code intrusion; on the other hand, this strategy requires the storage of the partial derivative matrices also for the direct method. Even though it has implications from a memory usage perspective, the computational efficiency of the direct method remains the same considering an efficient strategy to dump/load the partial derivative matrices from the hard-disk or their reconstruction from the primary variables states (similar concerns are associated with the adjoint method).

3 Applications of the framework: life-cycle optimization and assisted history matching of sequentially coupled flow and transport forward model

3.1 Algebraic description of forward model equations

The computation of derivative information for sequentially coupled systems is illustrated in the context of flow and transport in heterogeneous porous media. More specifically, two-phase, immiscible, incompressible flow is considered, with no gravity and capillary effects. The total mass balance (flow) equation is given by

$$-\nabla \cdot (\lambda \mathbf{K} \cdot \nabla p) = \nabla \cdot (\mathbf{u}) = q, \quad (28)$$

where \mathbf{u} is the total velocity, \mathbf{K} the absolute permeability tensor and p is the pressure [2]. The total mobility is given by $\lambda = \lambda_o + \lambda_w$, with the subscripts o and w standing for, respectively, oil and water, and the total source term is given by $q = q_o + q_w$.

The transport equation for a given phase α can be written as

$$\phi \frac{\partial S_\alpha}{\partial t} + \nabla \cdot (f_\alpha \mathbf{u}) = q_\alpha, \quad (29)$$

where S_α and f_α are, respectively, the saturation and fractional flow of phase α . The system is closed via the saturation constraint

$$\sum_{\alpha=o,w} S_\alpha = 1. \quad (30)$$

The discrete form of Eq. 28 reads

$$\mathbf{g}_p^n = \mathbf{A}^{n-1} \mathbf{p}^n - \mathbf{q}^{n-1} = \mathbf{0}, \quad (31)$$

where $\mathbf{p}^n \in \mathbf{R}^{N_b}$ and $\mathbf{q}^{n-1} \in \mathbf{R}^{N_b}$ are vectors of pressure and source terms, respectively, N_b is the number of grid blocks, and $\mathbf{A}^{n-1} \in \mathbf{R}^{N_b \times N_b}$ is the system matrix. Interfacial rock properties are computed by means of harmonic averages for the absolute permeabilities,

whereas an upwind scheme is employed for interfacial fluid properties (i.e., mobilities). The dependency of the fluid mobilities on the saturation is treated lagged in time because of the sequential solution strategy.

The discrete form of Eq. 29 reads

$$\mathbf{g}_s^n = \mathbf{V} (\mathbf{s}^n - \mathbf{s}^{n-1}) + \mathbf{F}^t \mathbf{u}^n - \mathbf{q}_\alpha^t = \mathbf{0}, \quad (32)$$

where $\mathbf{s} \in \mathbf{R}^{N_b}$, $\mathbf{F}^t \in \mathbf{R}^{N_b \times N_I}$, and $\mathbf{u}^n \in \mathbf{R}^{N_I}$ are, respectively, the saturation vector, the upwind fractional flow matrix and the vector containing the normal to grid interfaces velocity components, with N_I being the number of grid interfaces,

$$\mathbf{V} = \mathbf{I} \frac{\mathbf{V}_\phi}{\Delta t}, \quad (33)$$

where $\mathbf{V} \in \mathbf{R}^{N_b \times N_b}$, and \mathbf{u}^n is computed from

$$\mathbf{g}_u^n = \mathbf{u}^n - \Lambda^{n-1} \mathbf{p}^n = \mathbf{0}, \quad (34)$$

where $\Lambda^{n-1} \in \mathbf{R}^{N_I \times N_b}$ is the transmissibility matrix. Furthermore, we highlight that, in our implementation, α is considered to be water, and hence, water saturation is a primary variable. Therefore, all references to saturation found from here on are w.r.t. water saturation. Additionally, Δt is the time-step size, $\mathbf{V}_\phi \in \mathbf{R}^{N_b}$ is the vector containing the grid block pore-volumes, and \mathbf{I} is the identity matrix.

The de-coupling of Eqs. 31 and 32 allows the system to be solved sequentially, with no dependency of Eq. 31 on \mathbf{s}^n . If $t = n - 1$, the fractional flow and source terms are evaluated at the previous time-step. This is the so-called implicit-pressure explicit-saturation (IMPES) discretization in time [2]. However, to avoid time-step size limitations [5], the so-called sequential implicit strategy (IMPSAT) can be defined by making $t = n$. Although (32) now has a non-linear dependency on \mathbf{s}^n , this scheme allows for larger time-steps.

3.2 Gradient computation

From the discrete forward simulation (31), (32), and (34), Eq. 1 can be specialized as

$$\begin{cases} \mathbf{g}_p^n(\mathbf{p}^n, \mathbf{s}^{n-1}, \theta) = \mathbf{0} \\ \mathbf{g}_u^n(\mathbf{p}^n, \mathbf{u}^n, \mathbf{s}^{n-1}, \theta) = \mathbf{0} \\ \mathbf{g}_s^n(\mathbf{p}^n, \mathbf{u}^n, \mathbf{s}^{n-1}, \mathbf{s}^n, \theta) = \mathbf{0}, \end{cases} \quad (35)$$

where \mathbf{g}_p^n , \mathbf{g}_u^n and \mathbf{g}_s^n are, respectively, the vector-valued equations describing flow (pressure) and transport (saturation) at time-step n . The equations that determine the initial conditions are assumed to be

$$\begin{cases} \mathbf{g}_p^0(\mathbf{p}^0, \theta) = \mathbf{0} \\ \mathbf{g}_u^0(\mathbf{p}^0, \mathbf{u}^0, \theta) = \mathbf{0} \\ \mathbf{g}_s^0(\mathbf{p}^0, \mathbf{u}^0, \mathbf{s}^0, \theta) = \mathbf{0}. \end{cases} \quad (36)$$

From Eq. 35, let

$$\mathbf{g}^n = \begin{pmatrix} \mathbf{g}_p^n \\ \mathbf{g}_u^n \\ \mathbf{g}_s^n \end{pmatrix}. \quad (37)$$

Also, based on the corresponding primary variables associated to Eqs. 35 and 5 can be redefined as

$$\mathbf{x}^n = \begin{pmatrix} \mathbf{p}^n \\ \mathbf{u}^n \\ \mathbf{s}^n \end{pmatrix}. \quad (38)$$

The functions defining the set of observable outputs at time-step n will be assumed to be functions of both \mathbf{p}^n and \mathbf{s}^t , i.e.,

$$\mathbf{y}^n = \mathbf{h}^n(\mathbf{p}^n, \mathbf{s}^t, \theta), \quad (39)$$

which, in the case of IMPES reads

$$\mathbf{y}^n = \mathbf{h}^n(\mathbf{p}^n, \mathbf{s}^{n-1}, \theta), \quad (40)$$

and for IMPSAT reads

$$\mathbf{y}^n = \mathbf{h}^n(\mathbf{p}^n, \mathbf{s}^n, \theta). \quad (41)$$

From Eqs. 38 and 39 can be re-written as

$$\mathbf{y}^n = \mathbf{h}^n(\mathbf{x}^{n-1}, \mathbf{x}^n, \theta). \quad (42)$$

Based on Eqs. 35, 38, 18 and 19 can be redefined, now taking into account the appropriate dependencies of equations and variables for the flow-transport coupling, as

$$\frac{\partial \mathbf{g}^n}{\partial \mathbf{x}^n} = \begin{pmatrix} \frac{\partial \mathbf{g}_p^n}{\partial \mathbf{p}^n} & \mathbf{0} & \mathbf{0} \\ \frac{\partial \mathbf{g}_u^n}{\partial \mathbf{p}^n} & \frac{\partial \mathbf{g}_u^n}{\partial \mathbf{u}^n} & \mathbf{0} \\ \frac{\partial \mathbf{g}_s^n}{\partial \mathbf{p}^n} & \frac{\partial \mathbf{g}_s^n}{\partial \mathbf{u}^n} & \frac{\partial \mathbf{g}_s^n}{\partial \mathbf{s}^n} \end{pmatrix} = \begin{pmatrix} \mathbf{A}^{n-1} & \mathbf{0} & \mathbf{0} \\ -\Lambda^{n-1} & \mathbf{I} & \mathbf{0} \\ -\frac{\partial \mathbf{q}_\alpha^n}{\partial \mathbf{p}^n} & \mathbf{F}^t & \frac{\partial \mathbf{g}_s^n}{\partial \mathbf{s}^n} \end{pmatrix}, \quad (43)$$

and

$$\frac{\partial \mathbf{g}^n}{\partial \mathbf{x}^{n-1}} = \begin{pmatrix} \mathbf{0} & \mathbf{0} & \frac{\partial \mathbf{g}_p^n}{\partial \mathbf{s}^{n-1}} \\ \mathbf{0} & \mathbf{0} & \frac{\partial \mathbf{g}_u^n}{\partial \mathbf{s}^{n-1}} \\ \mathbf{0} & \mathbf{0} & \frac{\partial \mathbf{g}_s^n}{\partial \mathbf{s}^{n-1}} \end{pmatrix}. \quad (44)$$

Furthermore, based on Eqs. 38 and 42, it follows that

$$\frac{\partial \mathbf{h}^n}{\partial \mathbf{x}^n} = \begin{pmatrix} \frac{\partial \mathbf{h}^n}{\partial \mathbf{p}^n} & \mathbf{0} & \frac{\partial \mathbf{h}^n}{\partial \mathbf{s}^n} \end{pmatrix}, \quad (45)$$

and

$$\frac{\partial \mathbf{h}^n}{\partial \mathbf{x}^{n-1}} = \begin{pmatrix} \mathbf{0} & \mathbf{0} & \frac{\partial \mathbf{h}^n}{\partial \mathbf{s}^{n-1}} \end{pmatrix}. \quad (46)$$

Note that $\frac{\partial \mathbf{h}^n}{\partial \mathbf{s}^n} = \mathbf{0}$ in Eq. 45 if a sequential explicit method is used. On the other hand, $\frac{\partial \mathbf{h}^n}{\partial \mathbf{s}^{n-1}} = \mathbf{0}$ in Eq. 46 if a sequential implicit method is used.

Also, one should note that $\frac{\partial \mathbf{g}_s^n}{\partial \mathbf{s}^n}$ in Eq. 43 becomes diagonal if IMPES is used and non-diagonal if IMPSAT is used. On the other hand, $\frac{\partial \mathbf{g}_s^n}{\partial \mathbf{s}^{n-1}}$ becomes diagonal in Eq. 44 if IMPSAT is used and non-diagonal if IMPES is used.

3.2.1 The direct method

If Eqs. 43, 44, 45, and 46 are used in Eq. 17, the algorithm to compute the required gradient information using the direct method can be defined for the flow-transport coupling. The linear systems that must be solved for the flow equation in the direct method, for every time-step n , are given by

$$\mathbf{Z}_p^n = \left(\frac{\partial \mathbf{g}_p^n}{\partial \mathbf{p}^n} \right)^{-1} \left(\frac{\partial \mathbf{g}_p^n}{\partial \theta} \mathbf{V} - \frac{\partial \mathbf{g}_p^n}{\partial \mathbf{s}^{n-1}} \mathbf{Z}_s^{n-1} \right), \quad (47)$$

for the pressure equation,

$$\mathbf{Z}_u^n = \left(\frac{\partial \mathbf{g}_u^n}{\partial \mathbf{u}^n} \right)^{-1} \left(\frac{\partial \mathbf{g}_u^n}{\partial \theta} \mathbf{V} - \frac{\partial \mathbf{g}_u^n}{\partial \mathbf{p}^n} \mathbf{Z}_p^n - \frac{\partial \mathbf{g}_u^n}{\partial \mathbf{s}^{n-1}} \mathbf{Z}_s^{n-1} \right) \quad (48)$$

for the velocity equation, and

$$\mathbf{Z}_s^n = \left(\frac{\partial \mathbf{g}_s^n}{\partial \mathbf{s}^n} \right)^{-1} \left(\frac{\partial \mathbf{g}_s^n}{\partial \theta} \mathbf{V} - \frac{\partial \mathbf{g}_s^n}{\partial \mathbf{p}^n} \mathbf{Z}_p^n - \frac{\partial \mathbf{g}_s^n}{\partial \mathbf{u}^n} \mathbf{Z}_u^n - \frac{\partial \mathbf{g}_s^n}{\partial \mathbf{s}^{n-1}} \mathbf{Z}_s^{n-1} \right) \quad (49)$$

for the transport equation.

From Eqs. 47 and 49, Eq. 14 can be redefined based on the partitioning

$$\mathbf{Z}^n = \begin{pmatrix} \mathbf{Z}_p^n \\ \mathbf{Z}_u^n \\ \mathbf{Z}_s^n \end{pmatrix}, \quad (50)$$

so that the computation of the product $\mathbf{G}\mathbf{V}$ at time-step n is given by

$$(\mathbf{G}\mathbf{V})^n = \mathbf{G}^n \mathbf{V} = \frac{\partial \mathbf{h}^n}{\partial \theta} \mathbf{V} - \frac{\partial \mathbf{h}^n}{\partial \mathbf{p}^n} \mathbf{Z}_p^n - \frac{\partial \mathbf{h}^n}{\partial \mathbf{s}^n} \mathbf{Z}_s^n - \frac{\partial \mathbf{h}^n}{\partial \mathbf{s}^{n-1}} \mathbf{Z}_s^{n-1}. \quad (51)$$

Now, the direct method algorithm can be defined and is depicted in Algorithm 1.

3.2.2 The adjoint method

By transposing Eqs. 43, 44, 45 and 46 and replacing them in Eq. 26, the linear system that must be solved for the flow equation, for every time-step n , now reads

$$\begin{pmatrix} \mathbf{Z}_p^n \end{pmatrix}^T = \left(\frac{\partial \mathbf{g}_p^n}{\partial \mathbf{p}^n} \right)^{-T} \times \left(- \left(\frac{\partial \mathbf{g}_u^n}{\partial \mathbf{p}^n} \right)^T (\mathbf{Z}_u^n)^T - \left(\frac{\partial \mathbf{g}_s^n}{\partial \mathbf{p}^n} \right)^T (\mathbf{Z}_s^n)^T - \left(\mathbf{W}^n \frac{\partial \mathbf{h}^n}{\partial \mathbf{p}^n} \right)^T \right) \quad (52)$$

Algorithm 1: Right multiplying the sensitivity matrix by an arbitrary matrix via the direct method.

Input : Partial derivative matrices of \mathbf{g}_p , \mathbf{g}_u and \mathbf{g}_s w.r.t. \mathbf{x} and θ , \mathbf{V}

Output: \mathbf{GV}

```

1 foreach  $n = 0, 1, 2, \dots, N$  do
2   foreach  $j = 1, 2, \dots, p$  do
3     Solve for the  $j$ -th column of  $\mathbf{Z}_p^n$  using Eq. 47.
4     Solve for the  $j$ -th column of  $\mathbf{Z}_u^n$  using Eq. 48.
5     Solve for the  $j$ -th column of  $\mathbf{Z}_s^n$  using Eq. 49.
6   If there are responses at  $n$ , compute  $(\mathbf{GV})^n$  using Eq. 51

```

for the velocity equation

$$(\mathbf{Z}_u^n)^T = - \left(\frac{\partial \mathbf{g}_u^n}{\partial \mathbf{u}^n} \right)^{-T} \left(\frac{\partial \mathbf{g}_s^n}{\partial \mathbf{u}^n} \right)^T (\mathbf{Z}_s^n)^T \quad (53)$$

and for the transport equation

$$\begin{aligned}
(\mathbf{Z}_s^n)^T = & \left(\frac{\partial \mathbf{g}_s^n}{\partial \mathbf{s}^n} \right)^{-T} \left(- \left(\mathbf{W}^n \frac{\partial \mathbf{h}^n}{\partial \mathbf{s}^n} \right)^T - \left(\mathbf{W}^{n+1} \frac{\partial \mathbf{h}^{n+1}}{\partial \mathbf{s}^n} \right)^T \right. \\
& - \left(\frac{\partial \mathbf{g}_p^{n+1}}{\partial \mathbf{s}^n} \right)^T (\mathbf{Z}_p^{n+1})^T - \left(\frac{\partial \mathbf{g}_u^{n+1}}{\partial \mathbf{s}^n} \right)^T (\mathbf{Z}_u^{n+1})^T \\
& \left. - \left(\frac{\partial \mathbf{g}_s^{n+1}}{\partial \mathbf{s}^n} \right)^T (\mathbf{Z}_s^{n+1})^T \right). \quad (54)
\end{aligned}$$

By blocking Eq. 22 in time we have

$$\mathbf{WG} = \sum_{n=0}^N \left(\mathbf{Z}_p^n \frac{\partial \mathbf{g}_p^n}{\partial \theta} + \mathbf{Z}_u^n \frac{\partial \mathbf{g}_u^n}{\partial \theta} + \mathbf{Z}_s^n \frac{\partial \mathbf{g}_s^n}{\partial \theta} + \mathbf{W}^n \frac{\partial \mathbf{h}^n}{\partial \theta} \right). \quad (55)$$

The adjoint algorithm for the sequential coupling is described in Algorithm 2. The gradient computation does not only involve a backward simulation, but the solution of pressure and transport-related terms in the backward run is reversed when compared to the order in which the equations are solved in the forward simulation.

For the backward simulations, we precisely follow the time-stepping strategy taken in the forward simulation. This results in stable simulations, something that is related to the fact we are using the transpose matrices of the ones employed in the forward simulation. More studies related to the stability of adjoint simulations can be found in [32, 34].

3.3 Gradient computation and optimization for data assimilation

In data assimilation studies, one is interested to incorporate responses (or observations) from the real system into the

Algorithm 2: Left multiplying the sensitivity matrix by an arbitrary matrix via the adjoint method.

Input : Partial derivative matrices of \mathbf{g}_p , \mathbf{g}_u and \mathbf{g}_s w.r.t. \mathbf{x} and θ , \mathbf{W}

Output: \mathbf{WG}

```

1 foreach  $n = N, \dots, 2, 1, 0$  do
2   foreach  $i = 1, 2, \dots, m$  do
3     Solve for the  $i$ -th column of  $(\mathbf{Z}_s^n)^T$  using Eq. 54.
4     Solve for the  $i$ -th column of  $(\mathbf{Z}_u^n)^T$  using Eq. 53.
5     Solve for the  $i$ -th column of  $(\mathbf{Z}_p^n)^T$  using Eq. 52.
6   Update  $(\mathbf{WG})$  using Eq. 55.

```

numerical model by updating the (uncertain) model parameters so that the model's response reproduces the system observations. From a mathematical point of view, this exercise can be approached as an optimization problem

$$\begin{aligned}
& \underset{\theta}{\text{minimize}} && O(\mathbf{h}(\mathbf{x}, \theta)) \\
& \text{subject to} && \mathbf{g}(\mathbf{x}, \theta) = \mathbf{0}, \\
& && \theta \in [\theta_{\min}, \theta_{\max}], \quad (56)
\end{aligned}$$

where O is usually an objective function that represents the misfit between observed data and model responses. In data assimilation problems, θ represents the uncertain parameters, which are usually bounded between the upper and lower bounds θ_{\min} and θ_{\max} . A commonly used misfit objective function [28], with a regularization term, is given by

$$\begin{aligned}
O(\mathbf{y}, \theta) = & \frac{1}{2} (\theta - \theta_{\text{prior}})^T \mathbf{C}_\theta^{-1} (\theta - \theta_{\text{prior}}) \\
& + \frac{1}{2} (\mathbf{h}(\mathbf{x}, \theta) - \mathbf{d}_{\text{obs}})^T \mathbf{C}_D^{-1} (\mathbf{h}(\mathbf{x}, \theta) - \mathbf{d}_{\text{obs}}), \quad (57)
\end{aligned}$$

where \mathbf{C}_θ is the parameter covariance matrix, θ_{prior} is the vector containing a prior estimate of the uncertain parameters, \mathbf{d}_{obs} the observed data one desires to match, and \mathbf{C}_D the data covariance matrix. The gradient of Eq. 57 is given by

$$\begin{aligned}
\nabla_\theta O = & \mathbf{C}_\theta^{-1} (\theta - \theta_{\text{prior}}) + \left(\frac{d\mathbf{h}}{d\theta} \right)^T \mathbf{C}_D^{-1} (\mathbf{h}(\mathbf{x}, \theta) - \mathbf{d}_{\text{obs}}) \\
= & \mathbf{C}_\theta^{-1} (\theta - \theta_{\text{prior}}) + \mathbf{G}^T \mathbf{m}. \quad (58)
\end{aligned}$$

Since calculating the gradient using the adjoint method requires computational cost proportional to one extra simulation, while the direct method requires cost proportional to

N_θ extra simulations, the adjoint method is computationally the most efficient one. Note that

$$\mathbf{m} = \mathbf{C}_D^{-1} (\mathbf{h}(\mathbf{x}, \theta) - \mathbf{d}_{obs}), \quad (59)$$

where \mathbf{m} is an auxiliary vector, so the gradient of O can be written as $\nabla_\theta O = (\mathbf{m}^T \mathbf{G})^T$. Moreover, Algorithm 2, with $\mathbf{W} = \mathbf{m}^T$, calculates $\nabla_\theta O$ with a cost proportional to one extra simulation, instead of proportional to the number of parameters as in the direct method. For this reason, in the data assimilation studies shown here, the adjoint method is used when evaluating Eq. 58.

3.4 Gradient computation and optimization for life-cycle optimization

Life-cycle optimization aims to find the optimal set of control input parameters that maximizes an economic objective (e.g., the recovery factor or the net present value). This problem can also be represented as an optimization problem

$$\begin{aligned} & \underset{\theta}{\text{maximize}} && O(\mathbf{h}(\mathbf{x}, \theta)) \\ & \text{subject to} && \mathbf{g}(\mathbf{x}, \theta) = \mathbf{0}, \\ & && \mathbf{c}(\mathbf{x}, \theta) = \mathbf{0}, \\ & && \mathbf{d}(\mathbf{x}, \theta) < \mathbf{0}, \\ & && \theta \in [\theta_{min}, \theta_{max}], \end{aligned} \quad (60)$$

where \mathbf{c} and \mathbf{d} represent, respectively, equality and inequality operational constraints (e.g., maximum injection pressure). Now, θ represent the control parameters (e.g., well bottom-hole pressures or rates).

Here, let us assume the economical objective function $O = J$ to be the net present value, which is given in a simplified way by [15]

$$J = \sum_{n=1}^N \frac{[(q_{o,n}) \cdot r_o - (q_{wp,n}) \cdot r_{wp} - (q_{wi,n}) \cdot r_{wi}] \cdot \Delta t_n}{(1+b)^{\frac{t_n}{\tau_t}}}. \quad (61)$$

In Eq. 61, $q_{o,n}$ represents the oil production rate in m^3/day , $q_{wp,n}$ is the water production rate in m^3/day , $q_{wi,n}$ is the water injection rate in m^3/day , r_o is the price of oil produced in $\$/\text{m}^3$, r_{wp} is the cost of produced water in $\$/\text{m}^3$, r_{wi} is the cost of injected water in $\$/\text{m}^3$, Δt_n is the difference between consecutive time-steps in days, b is the discount factor expressed as a fraction per year, t_n is the cumulative time in days corresponding to time-step n , and τ_t is the reference time period for discounting, typically 1 year.

The well rates are computed via the Peaceman [29] formulation as

$$q(\mathbf{x}, \theta) = T \lambda_\alpha (p_b - p_w), \quad (62)$$

where p_b is the grid-block pressure, p_w is the wellbore pressure, T is a connectivity index, and λ_α is the mobility of phase α .

Equation 61 can be re-written in vectorial form as

$$J = \mathbf{r}_o^T \mathbf{q}_o - \mathbf{r}_{wp}^T \mathbf{q}_{wp} - \mathbf{r}_{wi}^T \mathbf{q}_{wi}, \quad (63)$$

where $\mathbf{q}_o \in \mathbf{R}^N$, $\mathbf{q}_{wp} \in \mathbf{R}^N$, $\mathbf{q}_{wi} \in \mathbf{R}^N$, and

$$\mathbf{r}_o = \left[\frac{r_o \Delta t_1}{(1+b)^{\frac{t_1}{\tau_t}}} \cdots \frac{r_o \Delta t_N}{(1+b)^{\frac{t_N}{\tau_t}}} \right]^T,$$

$$\mathbf{r}_{wp} = \left[\frac{r_{wp} \Delta t_1}{(1+b)^{\frac{t_1}{\tau_t}}} \cdots \frac{r_{wp} \Delta t_N}{(1+b)^{\frac{t_N}{\tau_t}}} \right]^T,$$

$$\mathbf{r}_{wi} = \left[\frac{r_{wi} \Delta t_1}{(1+b)^{\frac{t_1}{\tau_t}}} \cdots \frac{r_{wi} \Delta t_N}{(1+b)^{\frac{t_N}{\tau_t}}} \right]^T.$$

Furthermore, Eq. 63 can be re-written as

$$J = \mathbf{r}^T \mathbf{h}, \quad (64)$$

where

$$\mathbf{h} = [\mathbf{q}_o^T \quad -\mathbf{q}_{wp}^T \quad -\mathbf{q}_{wi}^T]^T, \quad \mathbf{r} = [\mathbf{r}_o^T \quad \mathbf{r}_{wp}^T \quad \mathbf{r}_{wi}^T]^T.$$

This allows us to write the gradient of Eq. 64 as

$$\nabla_\theta J = \mathbf{r}^T \frac{d\mathbf{h}}{d\theta} = \mathbf{r}^T \mathbf{G}. \quad (65)$$

Equation 64 allows the adjoint method to be employed in the computation of $\nabla_\theta J$ by making $\mathbf{W} = \mathbf{r}^T$. Just like in the data assimilation case, $\nabla_\theta J$ is efficiently computed with cost proportional to one backward simulation using the adjoint method.

3.5 Algorithm complexity analysis

As already mentioned, sequential methods can lead to efficient simulation strategies. Because the direct and adjoint derivative computation methods are tightly related to the numerical method employed in the forward simulation, a computational efficiency gain is also observed in these derivative computation methods.

The computational efficiency of the methods is assessed via an asymptotic analysis. In the analysis, only the most computationally intensive operations involved in the algorithms are considered. Hence, because the cost of solving linear system of equations overwhelms the cost of the matrix-vector products, only the former is considered over the latter. The cost associated to the solution of a linear system is considered to be $\mathcal{O}(\alpha N^\beta)$, where α and β are constants dependent of the linear solvers employed, and N is the size of the system.

Let us consider the computational cost associated to solve the derivative information for each time-step performed in the forward simulation for the different methods (FIM, IMPSAT, and IMPES). In the FIM case, for each column of \mathbf{V} for the direct method, or each row of \mathbf{W} for the adjoint method, a linear system of size $2 \times N_b$ must be solved, leading to a complexity $\mathcal{O}_{FIM}(\alpha(2 \times N_b)^\beta)$. In the IMPSAT case, a linear system must be solved for the flow and transport equations, leading to a complexity $\mathcal{O}_{IMPSAT}(\alpha N_b^\beta + \alpha N_b^\beta)$. Now, in the IMPES case, the saturation can be obtained via a negligible matrix-vector multiplication, which requires the solution of only one linear system of size N_b , leading to a complexity of $\mathcal{O}_{IMPES}(\alpha N_b^\beta)$. Figure 1 illustrates the cost ratios $\mathcal{O}_{IMPSAT}/\mathcal{O}_{FIM}$ and $\mathcal{O}_{IMPES}/\mathcal{O}_{FIM}$ for different values of β . It is considered that the linear solver employed in the solution of the different coupling strategies' systems are equally efficient (i.e., same β).

It is possible to see that it is always more or equally efficient to solve the resulting linear system(s) of equations in a sequential manner than using a FIM. Another aspect that is not captured in our analysis is that once we have the system de-coupling, it is possible to employ more efficient solution strategies based on the underlying physics and on the resulting system of equations' properties.

However, the cost per time-step associated to the sequential gradient computation methods are smaller or equal to the FIM gradient computation method. Due to numerical instabilities, sequential methods (mainly IMPES) usually require more time-steps than FIM methods due to the limitations imposed by the CFL condition. Therefore, there is a trade-off between number of time-steps and time-step cost,

$$\left(\frac{\mathcal{O}_{Seq}}{\mathcal{O}_{FIM}}\right)_{Total} = \frac{N_{Seq}^{TS}}{N_{FIM}^{TS}} \frac{\mathcal{O}_{Seq}}{\mathcal{O}_{FIM}},$$

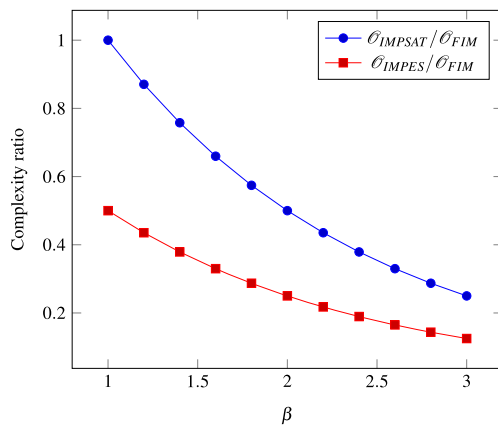


Fig. 1 Computational complexity ratio between IMPES and FIM (red) and IMPSAT and FIM (blue) for different values of β for one time-step

where N_{Seq}^{TS} is the total number of time-steps taken in the sequential (either IMPES or IMPSAT) simulation, N_{FIM}^{TS} is the total number of time-steps taken in the FIM simulation, and \mathcal{O}_{Seq} is the cost associated to the sequential simulation (either IMPES or IMPSAT).

Furthermore, both superior efficiency and stability could be achieved if an adaptive implicit sequential coupling [10, 33] is employed. The framework here presented could be directly employed by properly accounting for the explicit/implicit cells in the computation of the partial derivative matrices. The implementation of an AIM derivative computation method in a fully featured simulator has been used in the literature [30].

Also, we highlight that, although not captured in the above computational asymptotic analysis, it is important to note that the more time-steps are taken by the forward simulation, the more extra information (partial derivative matrices) must be computed and assembled, as well as stored/re-evaluated at each time-step to be later used in the backward simulation.

4 Numerical experiments

A synthetic model is considered as proof of concept (see Fig. 2). It is a 2D inverted five-spot model, consisting of a 21×21 equidistant Cartesian mesh with grid block dimensions of $33.3 \times 33.3 \times 2$ m. The reservoir porosity is constant and equal to 0.3. The fluid properties are described in Table 1.

The uncertainty around the absolute permeability distribution is represented by an ensemble of different permeability realizations. The ensemble is generated via the decomposition of a reference permeability “image” using principal component analysis parametrization [16]. Figure 3 illustrates four different permeability realizations from the ensemble of 1,000 members.

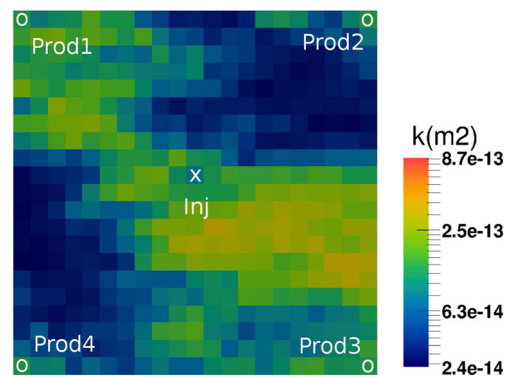


Fig. 2 The synthetic inverted five-spot model used in the numerical experiments. One of the 1,000 permeability realizations is shown

Table 1 Fluid properties for five-spot model

Property	Value	Unit
Oil dynamic viscosity (μ_o)	0.5×10^{-3}	Pa s
Water dynamic viscosity (μ_w)	1.0×10^{-3}	Pa s
End-point relative permeability, oil (k_{row})	0.9	–
End-point relative permeability, water (k_{rw})	0.6	–
Corey exponent, oil (N_o)	2.0	–
Corey exponent, water (N_w)	2.0	–
Residual-oil saturation ($S_{or} = 0.2$)	0.2	–
Connate-water saturation (S_{wc})	0.0	–

4.1 Gradient accuracy

In order to quantify how much the gradients computed by the presented sequential methods deviate from those computed using a FIM method, we calculate the angle between the gradient given by the FIM method and the gradients given by the IMPES and IMPSAT sequential methods. The angle between the gradient vectors can be computed as

$$\alpha = \cos^{-1} \left(\nabla_{\theta}^T \hat{O}_{FIM} \nabla_{\theta} \hat{O}_{Seq} \right), \quad (66)$$

where,

$$\nabla_{\theta} \hat{O}_{FIM} = \frac{\nabla_{\theta} O_{FIM}}{\|\nabla_{\theta} O_{FIM}\|_2} \quad (67)$$

and

$$\nabla_{\theta} \hat{O}_{Seq} = \frac{\nabla_{\theta} O_{Seq}}{\|\nabla_{\theta} O_{Seq}\|_2}. \quad (68)$$

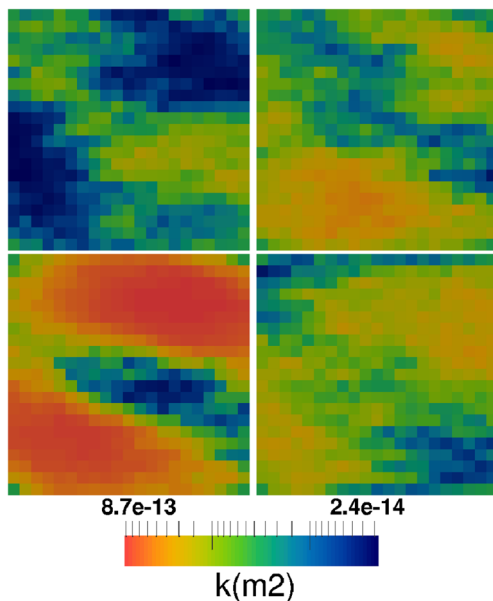


Fig. 3 Four different permeability realizations from the ensemble of 1,000 members used in the data assimilation study

Also, $\nabla_{\theta} O_{FS}$ and $\nabla_{\theta} O_{Seq}$ denote the FIM and the sequential (IMPES and FIM) analytical gradients, respectively. As a minimum requirement, acceptable MS gradients are obtained if α is much smaller than 90° [9].

The error metric has been computed for both the direct (Algorithm 1) and adjoint (Algorithm 2) methods. The metric is assessed for the gradient of the misfit objective function (58) and the life-cycle optimization function (65), which experiments setup are described, respectively, in Sections 4.2.1 and 4.3 Table 2. Also, the metric is evaluated considering the gradient computed at the initial parameter values.

It can be observed that the angles for both direct and adjoint methods are equally accurate. This is an expected result giving that the difference between the two algorithms is the order in which the operations are evaluated. Also, the angles indicate that algorithms here presented provide gradients that are consistent with the FIM derivative calculation method. That is an indication that, when the gradient computed via the sequential derivative computation algorithms are utilized by a gradient-based algorithm, the optimization solution path should not be too different from an optimization performed utilizing gradients computed by a FIM derivative calculation algorithm. This will be illustrated next, when the gradients are employed in different optimization exercises.

4.2 Water-flooding data assimilation

In the data assimilation studies shown here, we run the minimization problem defined by Eq. 56 by setting the objective function to be Eq. 57 and defining the vector of parameters as the natural logarithm of the permeability in each grid cell

$$\theta = [\ln k_1 \ln k_2 \dots \ln k_{N_b}]^T. \quad (69)$$

The covariance matrix C_{θ} is computed from the ensemble of realizations as

$$C_{\theta} = \frac{1}{N_e - 1} (\Theta - \mu e^T) (\Theta - \mu e^T)^T \quad (70)$$

Table 2 Angle (in degrees) between gradient vectors computed via the FIM method and the IMPES and IMPSAT methods for the synthetic inverted five-spot test case

	$\nabla_{\theta} O$		$\nabla_{\theta} J$	
	Direct	Adjoint	Direct	Adjoint
$\alpha_{FIM-IMPES}(^{\circ})$	5.5845	5.5845	0.3427	0.3427
$\alpha_{FIM-IMPESAT}(^{\circ})$	5.2232	5.2232	0.5508	0.5508

where Θ is the $N_b \times N_e$ matrix whose j -th column is given by the member of the ensemble θ_j , $j \in \{1, \dots, N_e\}$,

$$\mu = \frac{1}{N_e} \sum_{j=1}^{N_e} \theta_j \quad (71)$$

is the ensemble mean, and $\mathbf{e} = [\mathbf{1}, \dots, \mathbf{1}]^T$ is a vector of ones of size $N_e \times 1$. In Eq. 57, the prior is taken to be the ensemble mean

$$\theta_{prior} = \mu. \quad (72)$$

Additionally, \mathbf{C}_D is a diagonal matrix [28] given by

$$\mathbf{C}_D = \sigma^2 \mathbf{I}, \quad (73)$$

where σ^2 is the variance of the data measurement error.

The optimization utilizes a limited-memory Broyden Fletcher Goldfarb Shanno (LBFGS) implementation [27]. The LBFGS algorithm requires the objective function gradient. The misfit objective function gradient given by Eq. 58 can be computed via the adjoint method (Algorithm 2) with a cost proportional to one simulation backward in time. The optimization stopping criterion is determined by the minimum objective function (OF) value that is possible to satisfy a given noise level [28]. Following the definition of the model parameters in Eq. 69, next, we show how the framework can be employed with no modifications by defining different model responses.

4.2.1 Permeability estimation from well production data assimilation

In this exercise, we make the responses to be the water rates at the production wells at certain observation times

$$\mathbf{h}(\mathbf{x}, \theta) = \left[\mathbf{q}_{w_{obs}}^{\text{Prod1}T} \quad \mathbf{q}_{w_{obs}}^{\text{Prod2}T} \quad \mathbf{q}_{w_{obs}}^{\text{Prod3}T} \quad \mathbf{q}_{w_{obs}}^{\text{Prod4}T} \right]^T. \quad (74)$$

The observed data is generated using a twin experiment. One realization of the permeability ensemble was randomly chosen to be considered the “truth”. The water rates resulting from the simulation of 10 years of the model, with a 5% white noise level to represent the measurement error, were considered to be the observed data. The water rates are considered to be observed at every 6 months.

The water well rate matches for the FIM, IMPES, and IMPSAT methods, as obtained from the optimizations, are presented, respectively, in Figs. 4, 5, and 6. It can be noted that the gradients computed from the three different forward simulations are successfully employed in the optimization algorithm, leading to matched responses that accurately reproduce the observed data.

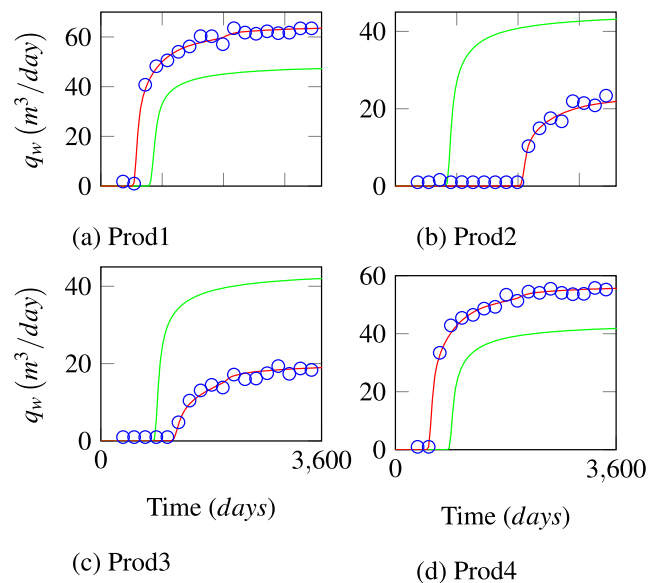


Fig. 4 Model responses, i.e., well rates, for the well data assimilation exercise utilizing the FIM method. In the figures, the green line represents the initial well rates, the blue circles the observed rates, and the red lines the rates after matching

The matched permeability fields are shown in Fig. 7. The resulting permeability fields are also in good agreement with the reference “truth” model, in particular the permeability orientation.

It can be noted from Fig. 8 that the optimizations for the three different forward model coupling strategies follow a similar iteration path. That was expected based on the angles

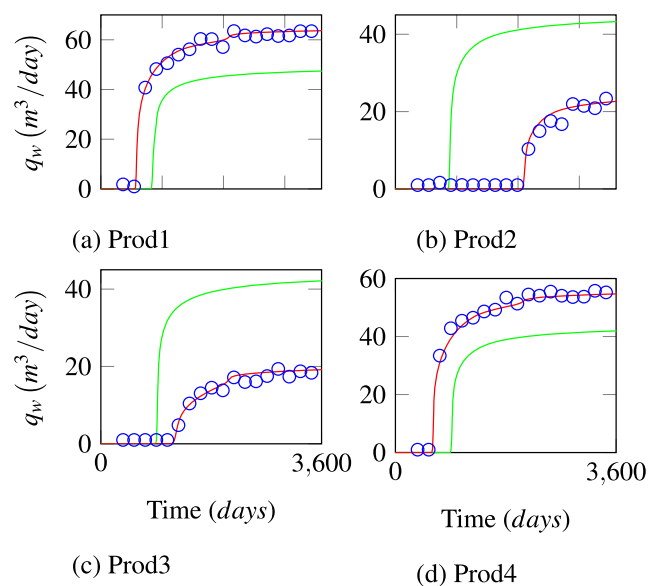


Fig. 5 Model responses, i.e., well rates, for the well data assimilation exercise utilizing the IMPES method. In the figures, the green line represents the initial well rates, the blue circles the observed rates, and the red lines the rates after matching

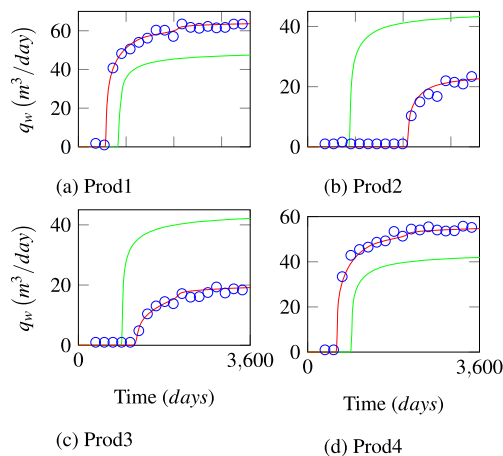


Fig. 6 Model responses, i.e., well rates, for the well data assimilation exercise utilizing the FIM method. In the figures, the green line represents the initial well rates, the blue circles the observed rates, and the red lines the rates after matching

shown in Table 2. Since the IMPES and IMPSAT gradients almost do not deviate from the FIM gradient, providing a similar search path, similar minima are found.

4.2.2 Seismic data assimilation

In this exercise, it is aimed to demonstrated how the framework can seamlessly accommodate different types of model responses. The observed data is now considered to be a spatially distributed response. More specifically, we consider the reservoir pressure distribution to be the observed data. Such data can be obtained e.g., from a seismic survey. The reservoir pressure can be attained from the seismic images if

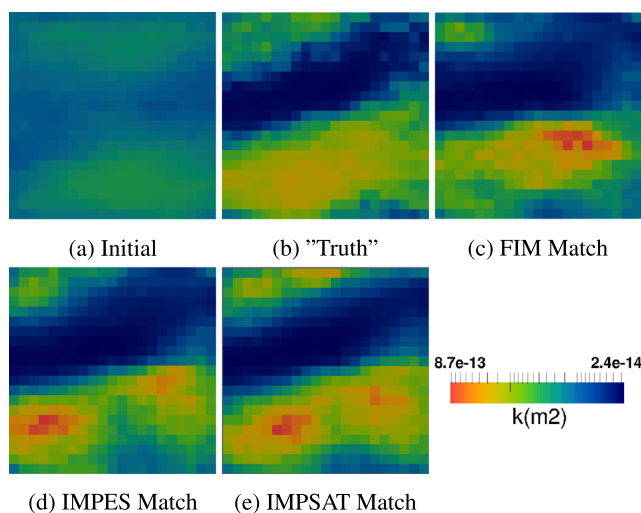


Fig. 7 Permeability field update for the well data assimilation exercise. Initial permeability field (a), permeability field from “Truth” (b) and permeability field after match utilizing the FIM (c), the IMPES (d), and the IMPSAT (e) methods

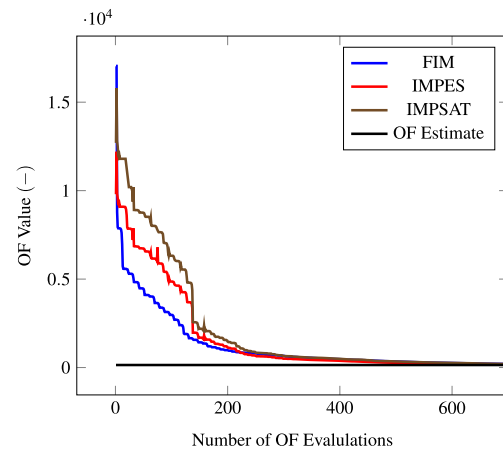


Fig. 8 Optimization performance for the well production data assimilation exercises with gradients computed from FIM (blue), IMPES (red), and IMPSAT (brown) forward simulations. The OF Estimate line (black) indicates the minimum OF value that is possible to satisfy the observed data noise level [28]

techniques like the ones presented in [24] and [23] are used. In this case, the derivative computation framework can be applied just like in the previous exercise by making

$$\mathbf{h}(\mathbf{x}, \boldsymbol{\theta}) = [p_1 \ p_2 \ \dots \ p_N]^T. \quad (75)$$

The observed pressure values are taken from the same twin experiment used in the well data history matching shown previously. Also, in the forward simulation, the flow and transport equations are sequentially coupled using the IMPES strategy.

We note that the framework is still applicable if any other spatially distributed property is considered as an observation, e.g., the more widely used impedances, provided that the necessary Jacobians of \mathbf{h} (e.g., via the derivatives of the petro-elastic equations) are available [8, 11].

In this experiment, perfect observed data is considered, with measurement error in the range of those usually employed in synthetic studies (see e.g., [28]), is employed.

The pressure match is illustrated in Fig. 9, while the resulting permeability field after the data assimilation is presented in Fig. 10.

Here, the IMPES method is employed in the forward simulation.

Once again, the data assimilation process results in a matched model that recovers the twin experiment response. Just like in the previous example, the resulting permeability field is in good agreement with the reference.

4.3 Water-flooding life-cycle optimization

In the life-cycle optimization studies shown here, we run the maximization problem defined by Eq. 60 by setting the

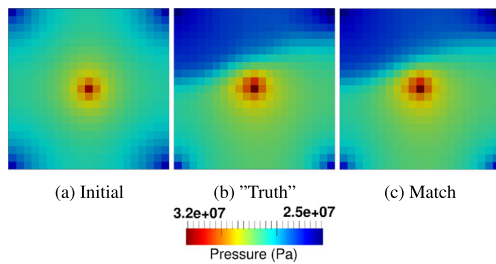


Fig. 9 Model responses, i.e., pressure distribution, for the seismic data assimilation exercise. Initial pressure distribution (a), response from the “truth” (b), and pressure distribution after the match (c)

objective function as in Eq. 64 and defining the vector of parameters to be the well bottom-hole pressures at some given control times

$$\theta = \left[p_{w_1}^1 \cdots p_{w_{N_w}}^1 \cdots p_{w_1}^{K_C} \cdots p_{w_{N_w}}^{K_C} \right]^T \quad (76)$$

where N_w is the total number of controlled wells and K_C is the total number of timesteps when a control change occurs. The economical parameters for oil production are defined in Table 3.

By allowing all well bottom-hole pressure values (5 in total) to change every six control time steps of 720 days gives a total number of control parameters equal to 30. The values of the bottom-hole pressures are bounded for the production wells between a minimum value of 28 MPa and a maximum value of 30 MPa. The injection well pressures are bounded between a minimum value of 30 MPa and maximum value of 32 MPa. The initial starting strategy is one wherein the injector well operates at a constant BHP of 31 MPa and the production wells at a constant BHP of 29 MPa.

In this exercise, the optimization utilizes the steepest ascent algorithm [27]. The line-search step length is reduced by half if the newly proposed controls given by the gradient do not lead to an increase of the objective function. This backtracking is allowed to be repeated five times. The control parameters are normalized with respect to the difference between the

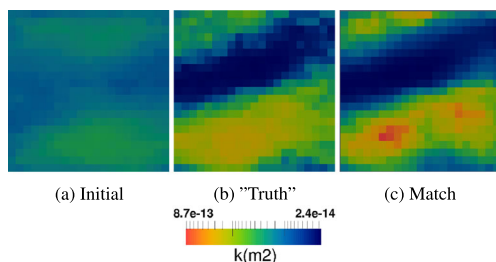


Fig. 10 Permeability field update for the seismic data assimilation exercise. Initial permeability field (a), permeability field from “Truth” (b), and permeability field after match (c)

Table 3 Economic parameters associated with oil production

	Value	Unit
Oil price	252	\$/m ³
Cost of injected water	60	\$/m ³
Cost of produced water	30	\$/m ³

bounds, and a normalized gradient is considered in the line-search direction computation.

Here, we run three different optimizations with different forward model coupling strategies: FIM, IMPES, and IMPSAT.

The optimal control parameters found by the optimization of the different coupling strategies can be found in Fig. 11 and the optimization performance is illustrated in Fig. 12.

It can be noted that, for the three couplings considered, the framework provides consistent objective function gradients that provide similar search directions. Similar optimized NPVs are also achieved, with an NPV increase of approximately 20%. Furthermore, except for small deviations in

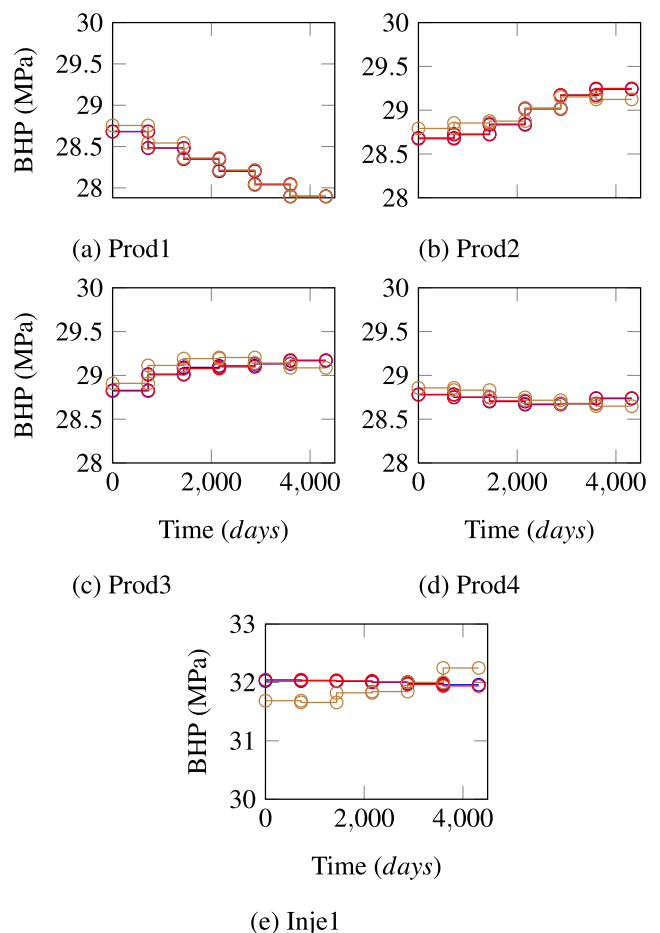


Fig. 11 Optimal controls (bottom-hole pressures) resulting from the optimization exercises with gradients computed from FIM (blue), IMPES (red), and IMPSAT (brown) forward simulations

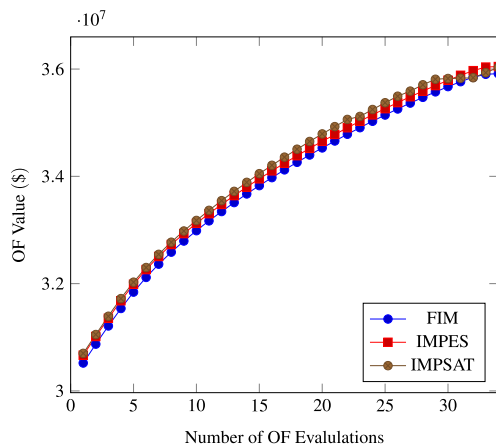


Fig. 12 Optimization performance for the optimization exercises with gradients computed from FIM (blue), IMPES (red), and IMPES (brown) forward simulations

the injection well, the optimal control strategies are nearly identical between the different optimization runs.

5 Concluding remarks

An efficient, general framework that addresses the derivative information computation of sequentially coupled system of equations is presented. The flexibility of the framework is illustrated in small data assimilation and life-cycle optimization studies in which the forward model's flow and transport equations are sequentially coupled. In the applications, it is shown how different objective functions (i.e. NPV and least-squares misfit), parameters (i.e. BHPs and grid-block permeabilities), and responses (i.e. well rates and grid-block pressures) can be accounted for in the computation without any change in the framework. Numerical results of a simple synthetic model demonstrates that the framework can be successfully employed to optimization studies. It is shown that the sequential derivative computation methods deliver similar results compared to the classical FIM methods. Furthermore, the computational asymptotic analysis of the presented algorithms shows that the sequential derivative computation methods are more computationally efficient when compared to FIM methods.

Acknowledgements This work was carried out as part of the first author PhD, made possible by the Petrobras training program.

Open Access This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

References

- Anterion, F., Eymard, R., Karcher, B.: Use of parameter gradients for reservoir history matching. In: SPE Symposium on Reservoir Simulation. Society of Petroleum Engineers (1989). <https://doi.org/10.2118/18433-MS>
- Aziz, K., Settari, A.: Petroleum reservoir simulation. Applied Science Publishers Ltd., Barking (1979). ISBN:0-85334-787-5
- Cardoso, M., Durlofsky, L., Sarma, P.: Development and application of reduced-order modeling procedures for subsurface flow simulation. *International Journal for Numerical Methods in Engineering* **77**(9), 1322–1350 (2009)
- Chavent, G., Dupuy, M., Lemmonier, P.: History matching by use of optimal theory. *Soc. Pet. Eng. J.* **15**(01), 74–86 (1975)
- Coats, K.: Impes stability: selection of stable timesteps. *SPE J.* **8**(02), 181–187 (2003)
- Cusini, M., van Kruijsdijk, C., Hajibeygi, H.: Algebraic dynamic multilevel (adm) method for fully implicit simulations of multiphase flow in porous media. *J. Comput. Phys.* **314**, 60–79 (2016)
- van Doren, J.F., Markovinović, R., Jansen, J.D.: Reduced-order optimal control of water flooding using proper orthogonal decomposition. *Comput. Geosci.* **10**(1), 137–158 (2006). <https://doi.org/10.1007/s10596-005-9014-2>
- Emerick, A.A., Moraes, R., Rodrigues, J.: History matching 4D seismic data with efficient gradient based methods. In: EUROPEC/EAGE Conference and Exhibition. Society of Petroleum Engineers (2007)
- Fonseca, R.M., Kahrobaei, S.S., Van Gastel, L.J.T., Leeuwenburgh, O., Jansen, J.D.: Quantification of the impact of ensemble size on the quality of an ensemble gradient using principles of hypothesis testing. In: SPE Reservoir Simulation Symposium. Society of Petroleum Engineers (2015). <http://doi.org/10.2118/173236-MS>
- Forsyth, P., Jr., Sammon, P.: Practical considerations for adaptive implicit methods in reservoir simulation. *J. Comput. Phys.* **62**(2), 265–281 (1986)
- Gosselin, O., Aanonsen, S., Aavatsmark, I., Cominelli, A., Gonard, R., Kolasinski, M., Ferdinandi, F., Kovacic, L., Neylon, K.: History matching using time-lapse seismic (Huts). In: SPE Annual Technical Conference and Exhibition. Society of Petroleum Engineers (2003)
- Hajibeygi, H., Jenny, P.: Adaptive iterative multiscale finite volume method. *J. Comput. Phys.* **230**(3), 628–643 (2011)
- Han, C., Wallis, J., Sarma, P., Li, G., Schrader, M., Chen, W.H.: Adaptation of the cpr preconditioner for efficient solution of the adjoint equation. *SPE J.* **18**(02), 207–213 (2013)
- Hou, T.Y., Wu, X.H.: A multiscale finite element method for elliptic problems in composite materials and porous media. *J. Comput. Phys.* **134**(1), 169–189 (1997)
- Jansen, J.D.: Adjoint-based optimization of multi-phase flow through porous media—a review. *Comput. Fluids* **46**(1), 40–51 (2011)
- Jansen, J.D.: A simple algorithm to generate small geostatistical ensembles for subsurface flow simulation. Research note. Dept. of Geoscience and Engineering, Delft University of Technology, The Netherlands. <http://resolver.tudelft.nl/uuid:6000459e-a0cb-40d1-843b-81650053e093> (2013)
- Jansen, J.D.: Gradient-based optimization of flow through porous media, v.3 (2016)
- Jenny, P., Lee, S., Tchelepi, H.: Adaptive multiscale finite-volume method for multiphase flow and transport in porous media. *Multiscale Model. Simul.* **3**(1), 50–64 (2005). <https://doi.org/10.1137/030600795>

19. Kraaijevanger, J.F.B.M., Egberts, P.J.P., Valstar, J.R., Buurman, H.W.: Optimal waterflood design using the adjoint method. In: SPE Reservoir Simulation Symposium. Society of Petroleum Engineers (2007). <https://doi.org/10.2118/105764-MS>
20. Krogstad, S., Hauge, V.L., Gulbransen, A.: Adjoint multiscale mixed finite elements. *SPE J.* **16**(01), 162–171 (2011)
21. Lee, S., Efendiev, Y., Tchelepi, H.: Hybrid upwind discretization of nonlinear two-phase flow with gravity. *Adv. Water Resour.* **82**, 27–38 (2015)
22. Li, R., Reynolds, A., Oliver, D.: History matching of three-phase flow production data. *SPEJ* **8**(04), 328–340 (2003)
23. Lumley, D., Meadows, M., Cole, S., Adams, D.: Estimation of reservoir pressure and saturations by crossplot inversion of 4D seismic attributes. In: SEG Technical Program Expanded Abstracts 2003, pp. 1513–1516. Society of Exploration Geophysicists (2003)
24. MacBeth, C., Floricich, M., Soldo, J.: Going quantitative with 4d seismic analysis. *Geophys. Prospect.* **54**(3), 303–317 (2006)
25. de Moraes, R.J., Rodrigues, J.R., Hajibeygi, H., Jansen, J.D.: Multiscale gradient computation for flow in heterogeneous porous media. *J. Comput. Phys.* **336**, 644–663 (2017)
26. Møyner, O., Lie, K.A.: A multiscale restriction-smoothed basis method for high contrast porous media represented on unstructured grids. *J. Comput. Phys.* **304**, 46–71 (2016)
27. Nocedal, J., Wright, S.: Numerical optimization. Springer Science & Business Media, New York (2006)
28. Oliver, D.S., Reynolds, A.C., Liu, N.: Inverse theory for petroleum reservoir characterization and history matching. Cambridge University Press, Cambridge (2008)
29. Peaceman, D.W.: Interpretation of well-block pressures in numerical reservoir simulation with nonsquare grid blocks and anisotropic permeability. *Soc. Pet. Eng. J.* **23**(03), 531–543 (1983)
30. Rodrigues, J.R.P.: Calculating derivatives for automatic history matching. *Comput. Geosci.* **10**(1), 119–136 (2006)
31. Sarma, P., Aziz, K., Durlafsky, L.J.: Implementation of adjoint solution for optimal control of smart wells. In: SPE Reservoir Simulation Symposium. Society of Petroleum Engineers (2005)
32. Serban, R., Petzold, L.R.: Efficient computation of sensitivities for ordinary differential equation boundary value problems. *SIAM J. Numer. Anal.* **40**(1), 220–232 (2002)
33. Thomas, G., Thurnau, D.: Reservoir simulation using an adaptive implicit method. *Soc. Pet. Eng. J.* **23**(05), 759–768 (1983)
34. Ulbrich, S.: Adjoint-based derivative computations for the optimal control of discontinuous solutions of hyperbolic conservation laws. *Syst. Control Lett.* **48**(3–4), 313–328 (2003)
35. Wallis, J.R., Kendall, R., Little, T.: Constrained residual acceleration of conjugate residual methods. In: SPE Reservoir Simulation Symposium. Society of Petroleum Engineers (1985)