

Multiscale Gradient Computation for Subsurface Flow Models

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

DOI

[10.3997/2214-4609.201601891](https://doi.org/10.3997/2214-4609.201601891)

Publication date

2016

Document Version

Final published version

Published in

Proceedings of the 15th European Conference on the Mathematics of Oil Recovery

Citation (APA)

Jesus de Moraes, R., Rodrigues, J. R. P., Hajibeygi, H., & Jansen, J. D. (2016). Multiscale Gradient Computation for Subsurface Flow Models. In *Proceedings of the 15th European Conference on the Mathematics of Oil Recovery: Amsterdam, Netherlands* (pp. 1-17). EAGE. <https://doi.org/10.3997/2214-4609.201601891>

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.

Green Open Access added to TU Delft Institutional Repository

'You share, we take care!' – Taverne project

<https://www.openaccess.nl/en/you-share-we-take-care>

Otherwise as indicated in the copyright section: the publisher is the copyright holder of this work and the author uses the Dutch legislation to make this work public.

We Gra 05

Multiscale Gradient Computation for Subsurface Flow Models

R. Moraes* (Delft University of Technology), J.R.P. Rodrigues (Petrobras),
H. Hajibeygi (Delft University of Technology) & J.D. Jansen (Delft
University of Technology)

SUMMARY

We present an efficient multiscale (MS) gradient computation that is suitable for reservoir management studies involving optimization techniques for, e.g., computer-assisted history matching or life-cycle production optimization. The general, algebraic framework allows for the calculation of gradients using both the Direct and Adjoint derivative methods. The framework also allows for the utilization of any MS formulation in the forward reservoir simulation that can be algebraically expressed in terms of a restriction and a prolongation operator. In the implementation, extra partial derivative information required by the gradient methods is computed via automatic differentiation. Numerical experiments demonstrate the accuracy of the method compared against those based on fine-scale simulation (industry standard).

Introduction

Model-based reservoir management techniques typically rely on the information provided by derivatives. For instance, in sensitivity analysis studies, derivatives can be directly used to identify the most influential parameters in the reservoir responses. Also, derivative information can be utilized in history matching (Oliver and Chen 2011) and control optimization (Jansen 2011) studies, where gradient-based optimization techniques are employed in the minimization/maximization of an objective function.

Such types of model-based reservoir management studies are computationally demanding exercises. They require multiple evaluations of the reservoir model in order to compute its response under the influence of different inputs. Reduced-order modelling (ROM) techniques have been employed to reduce the computational cost of the reservoir response evaluation. On the sensitivity analysis realm, response surface models and design of experiments are often used with this aim (see e.g. Yeten et al. 2005). In history matching and optimization studies, techniques like upscaling (Durlofsky 2005), streamline simulation (Datta-Gupta and King 2007), and proper orthogonal decomposition (Jansen and Durlofsky 2016) are employed to create reservoir models that are faster to evaluate. However, ROM and upscaling methods usually do not provide accurate enough system responses due to excessive simplifications of the fluid-rock physics and heterogeneous geological properties.

Multiscale (MS) methods (Hou and Wu 1997; Jenny et al. 2003; Efendiev and Hou 2009), however, solve a coarser simulation model, thus increasing the computational speed up, while still utilizing features of the fine scale heterogeneous nonlinear simulation model. This provides an efficient and accurate solution of the reservoir model. MS methods are accurate because they honor the high-resolution data using locally computed and adaptively updated basis functions. Important is that the coarse-scale solutions are mapped into the original high-resolution fine scale, using the basis functions. As such, errors can be calculated against the fine-scale reference systems (and not upscaled averaged ones). This allows for development of convergent iterative MS procedures (Hajibeygi et al. 2008, Zhou and Tchelepi 2012). Recent developments include MS formulations for fractured media (Tene et al. 2016) with compositional effects (Hajibeygi and Tchelepi 2014, Kozlova et al. 2015) and complex well configurations (Jenny and Lunati 2009). In addition, algebraic formulation of the method has made it convenient to integrate it within existing simulators using structured (Wang et al. 2014) and unstructured (Shah et al. 2016) grids. The method has been also extended to fully-implicit formulations where all unknowns cross multiple dynamically-defined scales (Cusini et al. 2016). Although these developments are found efficient, they are mainly limited to forward simulation modeling. Of high importance is to integrate the MS methods within reservoir management workflows.

From the derivative calculation techniques perspective, the most traditional approaches are either computationally expensive or inaccurate. For instance, numerical differentiation (see, e.g., Oliver et al. 2008 and Heath 2002) suffers from discretization approximations and truncation errors, and is impractical when the number of parameters is large. Alternatively, analytical methods – *Direct Methods* and *Adjoint Methods* (Chen et al. 1974, Chavent et al. 1975) – can provide accurate and efficient derivatives under the appropriate conditions (to be further discussed in the *Derivation of the Multiscale Gradient Computation Mathematical Framework* section). However, the use of Adjoint Methods has not been extensively adopted mainly because they are code-intrusive and require a substantial implementation effort. Younis and Aziz (2007) discuss how automatic differentiation can partly alleviate the burden of computing derivative information. However, most commercial simulators do not provide analytical derivative capabilities nor do they provide access to extend their functionality in this direction. Partially due to these drawbacks, ensemble methods such as the Ensemble Kalman Filter (EnKF) have become very popular in the data assimilation community

(Aanonsen et al. 2009). Similarly, stochastic approximate gradient techniques such as ensemble optimization (EnOpt) and the stochastic simplex approximate gradient (StoSAG) method are increasingly being used for life cycle optimization (Chen et al. 2009, Fonseca et al. 2016). These methods, however, by design, provide an approximation of the gradient.

The multiscale gradient computation has been studied in the past. In the work of Fu et al. (2010) a Multiscale Adjoint (MSADJ) method is applied to sensitivity computation, where a partitioning of the fine-scale problem is employed, using the same concepts that support the Multiscale Finite Volume forward simulation techniques. The global adjoint problem is solved via a set of coupled sub-grid problems described at a coarser scale. The coarse-scale sensitivities are then interpolated to the local fine grid by reconstructing the local variability of the model parameters with the aid of solving embedded adjoint sub-problems. In a follow up paper, Fu et al. (2011) show that their MSADJ method can be efficiently applied to inverse problems of single-phase flow in heterogeneous porous media. Krogstad et al. (2011) developed an adjoint model for a multiphase flow solver based on a Multiscale Mixed Finite Element formulation. Both pressure and saturations are solved at coarse scale. They demonstrate their method for optimization of the controls settings. As opposed to the work of Fu et al. (2011), the method presented by Krogstad et al. (2011) does not require fine grid quantities. Very good computational performance is reported in their paper for water flooding optimization of relatively big models.

The present development introduces a mathematical framework to compute sensitivities/gradients in a multiscale strategy. The framework enables the same computational efficiency as the methods proposed by Fu et al. (2011) and Krogstad et al. (2011). However, its formulation allows for full flexibility with respect to the types of gradient information that are required by the different model-based reservoir management studies. This is achieved via a self-contained elementary derivation performed via an implicit differentiation strategy, as opposed to the more traditional Lagrange multiplier formulation. Also, the formulation naturally provides not only the Adjoint Method, but also the Direct Method. It is important to note that, although in this work the multiscale finite volume (MSFV) is being studied, the proposed MS-gradient method is not restricted to a specific MS method. Instead, it can be utilized in combination with any MS (and multi-level) strategy that is expressed in terms of restriction and prolongation operators.

This paper is structured as follows. First, the multiscale gradient computation method is derived based on the MS reservoir model equations and the respective model responses. The computation of the required prolongation (matrix of basis functions) operator derivatives is developed within the Multiscale Finite Volume (MSFV) formulation. Thereafter, the Numerical Experiments section describes a systematic investigation of the validation, robustness, and accuracy of the MS-gradient method for test cases of increasing complexity. Because the proposed method is quite fundamental, the experiments are aimed to evaluate the gradient computation itself, rather than any specific application.

Derivation of the Multiscale Gradient Computation Mathematical Framework

Forward Simulation Model

We start the derivation of the forward simulation model's response with respect to the parameters by firstly describing its equations in a generic, purely algebraic form. We restrict the analysis to time-independent (quasi steady-state) problems. In that case, the set of equations that describes the forward simulation at the fine scale can be expressed, without any assumption regarding the underlying physical model, as

$$\mathbf{g}_F(\mathbf{x}, \boldsymbol{\theta}) = \mathbf{0}, \quad (1)$$

where $\mathbf{g}_F : \mathbb{R}^{N_F} \times \mathbb{R}^{N_\theta} \rightarrow \mathbb{R}^{N_F}$ represents the set of algebraic simulator equations, $\mathbf{x} \in \mathbb{R}^{N_F}$ is the state vector (which, for single-phase flow, contains the grid block pressures), $\boldsymbol{\theta} \in \mathbb{R}^{N_\theta}$ is the vector of parameters (which typically contains grid block permeabilities and porosities but possibly also other parameters such as fault transmissibility multipliers or structural parameters), and the subscript F refers to ‘fine scale’. There are N_F fine-scale cells and N_θ parameters. Eq. (1) implicitly assumes a dependency of the state vector \mathbf{x} on the parameters $\boldsymbol{\theta}$, i.e.,

$$\mathbf{x} = \mathbf{x}(\boldsymbol{\theta}). \quad (2)$$

Once the model state is determined, the observable responses of the forward model are computed. The forward model responses (typically wellbore pressures and/or rates) may not only depend on the model state, but also on the parameters themselves, and can be expressed as

$$\mathbf{y}_F = \mathbf{h}_F(\mathbf{x}, \boldsymbol{\theta}), \quad (3)$$

where $\mathbf{h}_F : \mathbb{R}^{N_F} \times \mathbb{R}^{N_\theta} \rightarrow \mathbb{R}^{N_y}$ represents the output equations (Jansen 2013a). It is assumed that \mathbf{g}_F can be described as

$$\mathbf{g}_F(\mathbf{x}, \boldsymbol{\theta}) = \mathbf{A}(\boldsymbol{\theta})\mathbf{x} - \mathbf{q}(\boldsymbol{\theta}), \quad (4)$$

where $\mathbf{A}(\boldsymbol{\theta})$ is an $N_F \times N_F$ matrix and $\mathbf{q}(\boldsymbol{\theta})$ is an N_F vector.

Algebraic Multiscale Formulation of Subsurface Flow in Heterogeneous Porous Media

MS methods provide accurate and efficient solutions to the flow equations by incorporating the fine-scale heterogeneities in a coarse-scale operator (Hajibeygi 2011). This is achieved by basis functions, which are local solutions of the governing equations without right-hand-side (RHS) terms, subject to assumed local boundary conditions. These local basis functions construct the prolongation (interpolation) operator, $\mathbf{P} = \mathbf{P}(\boldsymbol{\theta})$, which is an $N_F \times N_C$ matrix responsible to map (interpolate) the coarse-scale solution to the fine-scale resolution (Zhou and Tchelepi 2008). For the purpose of the development here presented, the required basic knowledge about the multiscale strategy is that a coarse-scale system can be algebraically described once the restriction operator, $\mathbf{R} = \mathbf{R}(\boldsymbol{\theta})$, is defined as an $N_C \times N_F$ matrix which maps the fine scale to the coarse scale (more information can be found in Wang et al. 2014). Let $\tilde{\mathbf{x}} \in \mathbb{R}^{N_C}$ be the coarse scale solution ($N_C \ll N_F$), $\mathbf{R} = \mathbf{R}(\boldsymbol{\theta})$ be an $N_C \times N_F$ matrix mapping from the fine scale to the coarse scale and $\mathbf{P} = \mathbf{P}(\boldsymbol{\theta})$ be an $N_F \times N_C$ matrix mapping from the coarse scale to the fine scale. \mathbf{R} and \mathbf{P} will be referred to as the restriction and prolongation operators, respectively. $\tilde{\mathbf{x}}$ is obtained by solving

$$\tilde{\mathbf{g}} = (\mathbf{R}\mathbf{A}\mathbf{P})\tilde{\mathbf{x}} - (\mathbf{R}\mathbf{q}) = \tilde{\mathbf{A}}\tilde{\mathbf{x}} - \tilde{\mathbf{q}} = \tilde{\mathbf{0}}. \quad (5)$$

Finally, the approximated fine-scale solution \mathbf{x}' is obtained by interpolating the coarse scale solution $\tilde{\mathbf{x}}$, i.e.,

$$\mathbf{x}' = \mathbf{P}\tilde{\mathbf{x}}. \quad (6)$$

Derivative Calculation of Simulator Responses

In order to derive an expression to compute the derivatives of the responses obtained by the multiscale method with respect to the parameters, we follow a formulation similar to the one that has been used by Rodrigues (2006) and Kraaijevanger et al. (2007). This formulation can be interpreted as a form of implicit differentiation and is equivalent to the standard Lagrange multiplier formulation (Jansen 2016). However, unlike the Lagrange multiplier formulation, this approach does not depend on a particular objective function and naturally provides both the Direct and the Adjoint Methods, as well as algorithms for multiplying the sensitivity matrix and its transpose with arbitrary vectors.

We define \mathbf{g}' , the function that describes the approximate primary variables solution at the fine scale from Eq. (6) as

$$\mathbf{g}' = \mathbf{x}' - \mathbf{P}\tilde{\mathbf{x}} = \mathbf{0}. \quad (7)$$

The state vector can now be described as the combination of both sets of primary variables in the fine and coarse scales, i.e.,

$$\mathbf{x} = \begin{bmatrix} \tilde{\mathbf{x}} \\ \mathbf{x}' \end{bmatrix}, \quad (8)$$

and, similarly, the model equations being the combination of the equations at both scales, i.e.,

$$\mathbf{g}(\mathbf{x}, \boldsymbol{\theta}) = \begin{bmatrix} \tilde{\mathbf{g}} \\ \mathbf{g}' \end{bmatrix} = \mathbf{0}. \quad (9)$$

By defining the state vector as in Eq. (8), we allow for the ability to describe the state not only at the fine scale, but at the coarse scale as well. This definition is a key aspect of the formulation. The simulator responses \mathbf{y} obtained from the multiscale method are represented as

$$\mathbf{y} = \mathbf{h}(\mathbf{x}, \boldsymbol{\theta}). \quad (10)$$

The sensitivity matrix \mathbf{G} is then computed by obtaining the derivative of Eq.(10) with respect to $\boldsymbol{\theta}$ as

$$\mathbf{G} = \frac{d\mathbf{h}}{d\boldsymbol{\theta}} = \frac{\partial \mathbf{h}}{\partial \mathbf{x}} \frac{d\mathbf{x}}{d\boldsymbol{\theta}} + \frac{\partial \mathbf{h}}{\partial \boldsymbol{\theta}}. \quad (11)$$

In order to find a relationship that defines the derivative of the state vector with respect to the parameters, we differentiate Eq. (9) with respect to $\boldsymbol{\theta}$

$$\frac{\partial \mathbf{g}}{\partial \mathbf{x}} \frac{d\mathbf{x}}{d\boldsymbol{\theta}} + \frac{\partial \mathbf{g}}{\partial \boldsymbol{\theta}} = \mathbf{0}, \quad (12)$$

so that

$$\frac{d\mathbf{x}}{d\boldsymbol{\theta}} = - \left(\frac{\partial \mathbf{g}}{\partial \mathbf{x}} \right)^{-1} \frac{\partial \mathbf{g}}{\partial \boldsymbol{\theta}}. \quad (13)$$

Substituting Eq. (13) in Eq. (11) gives

$$\mathbf{G} = - \frac{\partial \mathbf{h}}{\partial \mathbf{x}} \left(\frac{\partial \mathbf{g}}{\partial \mathbf{x}} \right)^{-1} \frac{\partial \mathbf{g}}{\partial \boldsymbol{\theta}} + \frac{\partial \mathbf{h}}{\partial \boldsymbol{\theta}}. \quad (14)$$

From Eqs. (5), (7), (8) and (9), it follows that

$$\frac{\partial \mathbf{g}}{\partial \mathbf{x}} = \begin{bmatrix} \mathbf{RAP} & \mathbf{0} \\ -\mathbf{P} & \mathbf{I} \end{bmatrix}. \quad (15)$$

Note that

$$\left(\frac{\partial \mathbf{g}}{\partial \mathbf{x}} \right)^{-1} = \begin{bmatrix} (\mathbf{RAP})^{-1} & \mathbf{0} \\ \mathbf{P}(\mathbf{RAP})^{-1} & \mathbf{I} \end{bmatrix} \quad (16)$$

holds. Thus, Eq. (14) can be restated as

$$\mathbf{G} = \left(\frac{\partial \mathbf{h}}{\partial \tilde{\mathbf{x}}} + \frac{\partial \mathbf{h}}{\partial \mathbf{x}'} \mathbf{P} \right) (\mathbf{RAP})^{-1} \frac{\partial \tilde{\mathbf{g}}}{\partial \boldsymbol{\theta}} + \frac{\partial \mathbf{h}}{\partial \mathbf{x}'} \frac{\partial \mathbf{g}'}{\partial \boldsymbol{\theta}} + \frac{\partial \mathbf{h}}{\partial \boldsymbol{\theta}}. \quad (17)$$

By deriving Eq. (5) with respect to θ one finds

$$\frac{\partial \tilde{\mathbf{g}}}{\partial \theta} = \left[\frac{\partial \mathbf{R}}{\partial \theta} (\mathbf{A}\mathbf{P}) + \mathbf{R} \frac{\partial \mathbf{A}}{\partial \theta} \mathbf{P} + (\mathbf{R}\mathbf{A}) \frac{\partial \mathbf{P}}{\partial \theta} \right] \tilde{\mathbf{x}} - \frac{\partial \mathbf{R}}{\partial \theta} \mathbf{q} - \mathbf{R} \frac{\partial \mathbf{q}}{\partial \theta} \quad (18)$$

and also, from Eq. (7),

$$\frac{\partial \mathbf{g}'}{\partial \theta} = -\frac{\partial \mathbf{P}}{\partial \theta} \tilde{\mathbf{x}}. \quad (19)$$

Note that the partial derivatives of the matrices \mathbf{A} , \mathbf{R} and \mathbf{P} with respect to the vector of parameters θ result in (third order) tensors. The tensors (and the operations with them) can be fully expressed using the Index notation. The substitution of Eqs. (18) and (19) in Eq. (17) leads to the expression that will serve as the basis for the multiscale gradient computation, i.e.,

$$\mathbf{G} = \left(\frac{\partial \mathbf{h}}{\partial \tilde{\mathbf{x}}} + \frac{\partial \mathbf{h}}{\partial \mathbf{x}'} \mathbf{P} \right) (\mathbf{R}\mathbf{A}\mathbf{P})^{-1} \left\{ \left[\frac{\partial \mathbf{R}}{\partial \theta} (\mathbf{A}\mathbf{P}) + \mathbf{R} \frac{\partial \mathbf{A}}{\partial \theta} \mathbf{P} + (\mathbf{R}\mathbf{A}) \frac{\partial \mathbf{P}}{\partial \theta} \right] \tilde{\mathbf{x}} - \frac{\partial \mathbf{R}}{\partial \theta} \mathbf{q} - \mathbf{R} \frac{\partial \mathbf{q}}{\partial \theta} \right\} - \frac{\partial \mathbf{h}}{\partial \mathbf{x}'} \frac{\partial \mathbf{P}}{\partial \theta} \tilde{\mathbf{x}} + \frac{\partial \mathbf{h}}{\partial \theta} \quad (20)$$

Eq. (20) is quite general, i.e., it can be employed to compute gradients for any MS (and multi-level) method, given that the partial derivatives of \mathbf{R} and \mathbf{P} are provided.

For the sake of simplicity and to be coherent with the numerical experiments herewith presented, from now on we neglect the dependency of the restriction operator \mathbf{R} and the right-hand-side vector \mathbf{q} on the parameters. This is consistent with the MSFV method (where the restriction operator is based on a finite volume integration operator at coarse scale (see e.g. Wang et al. 2014)). After these simplifications, Eq. (20) can be rewritten as

$$\mathbf{G} = \left(\frac{\partial \mathbf{h}}{\partial \tilde{\mathbf{x}}} + \frac{\partial \mathbf{h}}{\partial \mathbf{x}'} \mathbf{P} \right) (\mathbf{R}\mathbf{A}\mathbf{P})^{-1} \mathbf{R} \left(\frac{\partial \mathbf{A}}{\partial \theta} \mathbf{P} + \mathbf{A} \frac{\partial \mathbf{P}}{\partial \theta} \right) \tilde{\mathbf{x}} - \frac{\partial \mathbf{h}}{\partial \mathbf{x}'} \frac{\partial \mathbf{P}}{\partial \theta} \tilde{\mathbf{x}} + \frac{\partial \mathbf{h}}{\partial \theta}. \quad (21)$$

The order of the operations involved in $(\mathbf{R}\mathbf{A}\mathbf{P})^{-1}$ term is crucial to determine the computational performance of the sensitivity matrix computation. This order defines two different algorithms known in the literature as the Direct and Adjoint Methods. The following sections discuss how the two methods are derived for the MS scenario defined by Eq. (21), as well as the (dis)advantages of utilizing each of them.

Computation of Gradient Information: Generalization of the Framework

According to the type of study one wants to perform, different derivative information has to be provided. For instance, for optimization methods, Quasi-Newton methods require the gradient of the objective function (and possibly constraints). In history matching algorithms, as well as in sensitivity analysis studies, the sensitivity matrix \mathbf{G} plays an important role. In Gauss-Newton methods, matrix $\mathbf{G}^T \mathbf{G}$ is directly used, while conjugate gradient methods require products of \mathbf{G} and \mathbf{G}^T with arbitrary vectors. Nocedal and Wright (2006) provide more detailed information on the different optimization algorithms.

To preserve general applicability of our method, we consider the general problem of multiplying \mathbf{G} by arbitrary matrices \mathbf{W} (of order $m \times N_y$) and (\mathbf{V}) (of order $N_\theta \times n$) from left and right, respectively. When $n=1$, $\mathbf{G}\mathbf{V}$ is simply the product of \mathbf{G} with a vector, whereas, when $m=1$, $(\mathbf{W}\mathbf{G})^T = \mathbf{G}^T \mathbf{W}^T$ gives the product of \mathbf{G}^T with the (column) vector \mathbf{W}^T . Those examples show how, by defining algorithms to calculate $\mathbf{G}\mathbf{V}$ and $\mathbf{W}\mathbf{G}$ for arbitrary \mathbf{V} and \mathbf{W} , different types of derivative information can be accommodated in a single framework.

Direct Method

We start by considering the calculation of $\mathbf{G}\mathbf{V}$ for a given $N_\theta \times n$ matrix \mathbf{V} . From Eq. (21), by defining

$$\mathbf{Z} = \left[(\mathbf{RAP})^{-1} \mathbf{R} \left(\frac{\partial \mathbf{A}}{\partial \boldsymbol{\theta}} \mathbf{P} + \mathbf{A} \frac{\partial \mathbf{P}}{\partial \boldsymbol{\theta}} \right) \tilde{\mathbf{x}} \right] \mathbf{V}, \quad (22)$$

the product $\mathbf{G}\mathbf{V}$ can be rewritten as

$$\mathbf{G}\mathbf{V} = \left(\frac{\partial \mathbf{h}}{\partial \tilde{\mathbf{x}}} + \frac{\partial \mathbf{h}}{\partial \mathbf{x}'} \mathbf{P} \right) \mathbf{Z} - \left(\frac{\partial \mathbf{h}}{\partial \mathbf{x}'} \frac{\partial \mathbf{P}}{\partial \boldsymbol{\theta}} \tilde{\mathbf{x}} \right) \mathbf{V} + \frac{\partial \mathbf{h}}{\partial \boldsymbol{\theta}} \mathbf{V} \quad (23)$$

Here, \mathbf{Z} is obtained as the solution to the following linear system of equations

$$(\mathbf{RAP})\mathbf{Z} = \left[\mathbf{R} \left(\frac{\partial \mathbf{A}}{\partial \boldsymbol{\theta}} \mathbf{P} \tilde{\mathbf{x}} + \mathbf{A} \frac{\partial \mathbf{P}}{\partial \boldsymbol{\theta}} \tilde{\mathbf{x}} \right) \right] \mathbf{V}. \quad (24)$$

This is known as the *Forward Method* (Rodrigues 2006) or *Direct Method* (Oliver et al. 2008). Note that \mathbf{Z} has dimensions of $N_C \times n$ and, therefore, it requires n linear systems to be solved. Hence, the cost of computing $\mathbf{G}\mathbf{V}$ is proportional to the number of columns in \mathbf{V} . In particular, to obtain the full sensitivity matrix \mathbf{G} with the Direct Method, in which case one has to set \mathbf{V} equal to the identity matrix of order N_θ , the cost will be proportional to the number of parameters.

Adjoint Method

Next we consider the calculation of $\mathbf{W}\mathbf{G}$ for a given $m \times N_Y$ matrix \mathbf{W} . From Eq. (21), defining

$$\mathbf{Z}^T = \mathbf{W} \left(\frac{\partial \mathbf{h}}{\partial \tilde{\mathbf{x}}} + \frac{\partial \mathbf{h}}{\partial \mathbf{x}'} \mathbf{P} \right) (\mathbf{RAP})^{-1}, \quad (25)$$

one obtains

$$\mathbf{W}\mathbf{G} = \mathbf{Z}^T \mathbf{R} \left(\frac{\partial \mathbf{A}}{\partial \boldsymbol{\theta}} \mathbf{P} + \mathbf{A} \frac{\partial \mathbf{P}}{\partial \boldsymbol{\theta}} \right) \tilde{\mathbf{x}} - \mathbf{W} \frac{\partial \mathbf{h}}{\partial \mathbf{x}'} \frac{\partial \mathbf{P}}{\partial \boldsymbol{\theta}} \tilde{\mathbf{x}} + \mathbf{W} \frac{\partial \mathbf{h}}{\partial \boldsymbol{\theta}}, \quad (26)$$

which can be rearranged as

$$\mathbf{W}\mathbf{G} = \left(\mathbf{Z}^T \mathbf{R} \frac{\partial \mathbf{A}}{\partial \boldsymbol{\theta}} \mathbf{P} \tilde{\mathbf{x}} + \left(\mathbf{Z}^T \mathbf{R} \mathbf{A} - \mathbf{W} \frac{\partial \mathbf{h}}{\partial \mathbf{x}'} \right) \frac{\partial \mathbf{P}}{\partial \boldsymbol{\theta}} \tilde{\mathbf{x}} \right) + \mathbf{W} \frac{\partial \mathbf{h}}{\partial \boldsymbol{\theta}}. \quad (27)$$

Multiplying Eq. (25) by \mathbf{RAP} from the right and transposing, \mathbf{Z} is obtained as the solution to the following linear system of equations

$$(\mathbf{RAP})^T \mathbf{Z} = \left(\frac{\partial \mathbf{h}}{\partial \tilde{\mathbf{x}}} + \frac{\partial \mathbf{h}}{\partial \mathbf{x}'} \mathbf{P} \right)^T \mathbf{W}^T. \quad (28)$$

This is known in the literature as the *Adjoint* (or *Backward*) *Method* (Chavent et al. 1975). Note that now \mathbf{Z} has dimensions of $N_C \times m$, hence it requires 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} . In particular, to obtain the full sensitivity matrix \mathbf{G} with the Adjoint Method, in which case one has to set \mathbf{W} equal to the identity matrix of order N_Y , the cost will be proportional to the number of responses.

Partial Derivative Computation and Automatic Differentiation

In the computation of analytical gradients, a significant part of the overall implementation effort is associated with the computation of partial derivative matrices. This computation step often requires access to the source code because not all partial derivatives are readily available, as opposed to the

state variable derivatives via the Jacobian matrix. To calculate the partial derivative matrices an Automatic Differentiation (AD) approach is used, because of its flexibility and accuracy. A review of different AD approaches is provided by Younis et al. (2007). In our work, an Operator Overloading AD technique (Bendtsen and Stauning 1996) based on Expression Templates (Vandevorde and Josuttis 2003) is applied. This facilitates the computation and assemblage of all the partial derivative matrices because they are obtained at the same time the system matrix \mathbf{A} and the simulator response \mathbf{y} are computed.

Partial Derivative of MSFV Prolongation Operator with Respect to the Parameters

A particular aspect of the methodology is the computation of partial derivatives of the prolongation operator with respect to the parameters, i.e., $(\partial\mathbf{P}/\partial\boldsymbol{\theta})$ in Eq. (21), and operations with this tensor. This is particularly challenging because the computation of \mathbf{P} might involve complex operations, e.g., the solution of linear systems in the case of MSFV methods.

In this work, we consider the MSFV method; extension to other MS methodologies can be obtained along the same lines. In this case, \mathbf{P} is the assembly of all basis functions obtained via the solution of local flow problems, i.e.,

$$-\nabla \cdot (\lambda \cdot \nabla \varphi) = 0 \quad (29)$$

where λ is the mobility and φ is the basis function value. The dual-grid sub domains where the basis functions are computed are defined as follows. A primal coarse grid (on which the conservative coarse-scale system is constructed) and a dual coarse grid, which is obtained by connecting coarse nodes, are defined on a given fine-scale grid. A coarse node is a fine cell inside (typically at the centre of) each coarse cell. Basis functions are solved locally on these dual coarse cells. Such overlapping coarse and dual coarse grids are crucial for conservative solutions in MSFV. An illustration of the MSFV grids is provided in Figure 1. The prolongation operator can be expressed in terms of the basis functions corresponding to each coarse cell $j \in \{1, \dots, N_C\}$ as

$$\mathbf{P} = [\boldsymbol{\varphi}_1 \quad \boldsymbol{\varphi}_2 \quad \cdots \quad \boldsymbol{\varphi}_{N_C-1} \quad \boldsymbol{\varphi}_{N_C}], \quad (30)$$

where the basis function belonging to cell j , $\boldsymbol{\varphi}_j$, is at its column j , being a vector of dimension N_F (Wang et al. 2014).

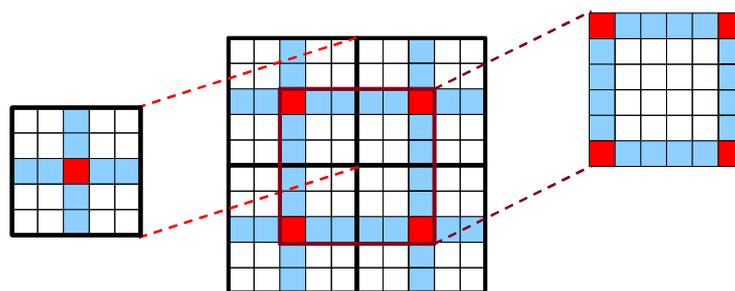


Figure 1 Illustration of MSFV coarse grids for a 2D domain. Given a fine-scale grid (shown in light solid black lines), the coarse grid (shown in solid bold black) is imposed as a non-overlapping partition of the computational domain. The coarse nodes (vertices) are then selected (filled in red cells). Connecting coarse nodes constructs the dual-coarse grid (highlighted in blue) where basis functions are solved.

Construction of the basis functions $\boldsymbol{\varphi}_j$ is based on the Finite Volume (FV) discretization of Eq. (29) on the dual coarse grid cells. In order to compute basis functions, reduced dimension problems are first solved at the edge cells to provide Dirichlet boundary conditions for internal cells. For the sake of

clarity, let us assume a Cartesian two-dimensional solution domain (although extension to 3D is straightforward) More specifically, let Λ_j be the set of all edges emanating from coarse node j (in 2D, $\#\Lambda_j = 4$). For each edge $e \in \Lambda_j$, let

$$\mathbf{A}_{e,j}^E \boldsymbol{\varphi}_{e,j}^E - \mathbf{b}_{e,j}^E = \mathbf{0} \quad (31)$$

be the reduced-dimension discrete form of Eq. (29) along the edge e , where $\boldsymbol{\varphi}_{e,j}^E$ is the basis function at that edge, and $\mathbf{b}_{e,j}^E$ is the RHS which results from specifying corner values of $\varphi_{e,j}^E = 1$ at the vertex j and $\varphi_{e,j}^E = 0$ at the opposite vertex of e . If Γ_j is the set of all faces which have coarse node j as a common vertex ($\#\Gamma_j = 4$ in 2D), for each face $f \in \Gamma_j$, according to Eq. (29), one can solve

$$\mathbf{A}_{f,j}^F \boldsymbol{\varphi}_{f,j}^F - \sum_{e \in \Lambda_j} \mathbf{E}_{e,f}^F \boldsymbol{\varphi}_{e,j}^E = \mathbf{0}, \quad (32)$$

for $\boldsymbol{\varphi}_{f,j}^F$, which is the basis function at the internal (face) cells. Also, $\mathbf{E}_{e,f}^F$ is a transformation matrix that appropriately assembles a vector represented in the edge topology of e into the face topology of f . Finally, $\boldsymbol{\varphi}_j$ is the result of assembling the contribution of each $\boldsymbol{\varphi}_{f,j}^F$, $f \in \Gamma_j$, into the overall fine mesh topology:

$$\boldsymbol{\varphi}_j = \sum_{f \in \Gamma_j} \mathbf{E}_f \boldsymbol{\varphi}_{f,j}^F, \quad (33)$$

where \mathbf{E}_f is a transformation matrix that assembles a vector represented in the face topology of f into a size N_F vector following the fine grid topology.

The derivative of $\boldsymbol{\varphi}_j$ w.r.t. the parameters is obtained from Eq. (33), i.e.,

$$\frac{\partial \boldsymbol{\varphi}_j}{\partial \boldsymbol{\theta}} = \sum_{f \in \Gamma_j} \mathbf{E}_f \frac{\partial \boldsymbol{\varphi}_{f,j}^F}{\partial \boldsymbol{\theta}}. \quad (34)$$

Differentiating Eq. (32), one can write

$$\frac{\partial \boldsymbol{\varphi}_{f,j}^F}{\partial \boldsymbol{\theta}} = (\mathbf{A}_{f,j}^F)^{-1} \left(-\frac{\partial \mathbf{A}_{f,j}^F}{\partial \boldsymbol{\theta}} \boldsymbol{\varphi}_{f,j}^F + \sum_{e \in \Lambda_j} \mathbf{E}_{e,f}^F \frac{\partial \boldsymbol{\varphi}_{e,j}^E}{\partial \boldsymbol{\theta}} \right). \quad (35)$$

Similarly, from Eq. (31),

$$\frac{\partial \boldsymbol{\varphi}_{e,j}^E}{\partial \boldsymbol{\theta}} = -(\mathbf{A}_{e,j}^E)^{-1} \frac{\partial \mathbf{A}_{e,j}^E}{\partial \boldsymbol{\theta}} \boldsymbol{\varphi}_{e,j}^E \quad (36)$$

holds. Combining Eqs. (35) and (36) into (34) results in

$$\frac{\partial \boldsymbol{\varphi}_j}{\partial \boldsymbol{\theta}} = \sum_{f \in \Gamma_j} \mathbf{E}_f (\mathbf{A}_{f,j}^F)^{-1} \left(-\frac{\partial \mathbf{A}_{f,j}^F}{\partial \boldsymbol{\theta}} \boldsymbol{\varphi}_{f,j}^F - \sum_{e \in \Lambda_j} \mathbf{E}_{e,f}^F (\mathbf{A}_{e,j}^E)^{-1} \frac{\partial \mathbf{A}_{e,j}^E}{\partial \boldsymbol{\theta}} \boldsymbol{\varphi}_{e,j}^E \right). \quad (37)$$

The third order tensor defining the derivative of \mathbf{P} w.r.t. the parameters is obtained by grouping all $\partial \boldsymbol{\varphi}_j / \partial \boldsymbol{\theta}$ at its slices,

$$\frac{\partial \mathbf{P}}{\partial \boldsymbol{\theta}} = \left[\begin{array}{cccc} \frac{\partial \boldsymbol{\varphi}_1}{\partial \boldsymbol{\theta}} & \frac{\partial \boldsymbol{\varphi}_2}{\partial \boldsymbol{\theta}} & \dots & \frac{\partial \boldsymbol{\varphi}_{N_C-1}}{\partial \boldsymbol{\theta}} & \frac{\partial \boldsymbol{\varphi}_{N_C}}{\partial \boldsymbol{\theta}} \end{array} \right]_{N_F \times N_C \times N_\theta}. \quad (38)$$

$\partial \mathbf{P} / \partial \boldsymbol{\theta}$ is a $N_F \times N_C \times N_\theta$ third order tensor that can be interpreted as a stack of N_θ matrices of partial derivatives $\partial \boldsymbol{\varphi}_j / \partial \boldsymbol{\theta}$, $j \in \{1, \dots, N_C\}$, each of dimension $N_F \times N_C$. As part of the computation of the Direct Method the product $(\partial \mathbf{P} / \partial \boldsymbol{\theta} \bar{\mathbf{x}}) \mathbf{m}$ is required, where $\bar{\mathbf{x}} \in \mathbb{R}^{N_C}$ and $\mathbf{m} \in \mathbb{R}^{N_\theta}$. Since

$$\frac{\partial \mathbf{P}}{\partial \boldsymbol{\theta}} \bar{\mathbf{x}} = \sum_{j=1}^{N_c} \frac{\partial \boldsymbol{\varphi}_j}{\partial \boldsymbol{\theta}} \bar{\mathbf{x}}_j, \quad (39)$$

where $\bar{\mathbf{x}}_j$ denotes the j -th entry of vector $\bar{\mathbf{x}}$, it follows, from (37), that

$$\left(\frac{\partial \mathbf{P}}{\partial \boldsymbol{\theta}} \bar{\mathbf{x}} \right) \mathbf{m} = \sum_{j=1}^{N_c} \bar{\mathbf{x}}_j \sum_{f \in \Gamma_j} \mathbf{E}_f (\mathbf{A}_{f,j}^F)^{-1} \left(- \left(\frac{\partial \mathbf{A}_{f,j}^F}{\partial \boldsymbol{\theta}} \boldsymbol{\varphi}_{f,j}^F \right) \mathbf{m} - \sum_{e \in \Lambda_j} \mathbf{E}_{e,f}^F (\mathbf{A}_{e,j}^E)^{-1} \left(\frac{\partial \mathbf{A}_{e,j}^E}{\partial \boldsymbol{\theta}} \boldsymbol{\varphi}_{e,j}^E \right) \mathbf{m} \right). \quad (40)$$

In a similar manner, as part of the computation of the Adjoint Method, the product $\mathbf{m}^T (\partial \mathbf{P} / \partial \boldsymbol{\theta} \bar{\mathbf{x}})$ is required, where $\bar{\mathbf{x}} \in \mathbb{R}^{N_c}$ and $\mathbf{m} \in \mathbb{R}^{N_e}$. From Eqs. (37) and (39), one obtains

$$\mathbf{m}^T \left(\frac{\partial \mathbf{P}}{\partial \boldsymbol{\theta}} \bar{\mathbf{x}} \right) = \sum_{j=1}^{N_c} \bar{\mathbf{x}}_j \sum_{f \in \Gamma_j} \mathbf{m}^T \mathbf{E}_f (\mathbf{A}_{f,j}^F)^{-1} \left(- \frac{\partial \mathbf{A}_{f,j}^F}{\partial \boldsymbol{\theta}} \boldsymbol{\varphi}_{f,j}^F - \sum_{e \in \Lambda_j} \mathbf{E}_{e,f}^F (\mathbf{A}_{e,j}^E)^{-1} \left(\frac{\partial \mathbf{A}_{e,j}^E}{\partial \boldsymbol{\theta}} \boldsymbol{\varphi}_{e,j}^E \right) \right). \quad (41)$$

Numerical Experiments

In this section, a systematic performance study for the MS-gradient method is presented for single-phase incompressible flow in heterogeneous porous media. Because this is the scenario for which the MS methods were originally developed, it provides a simple, yet effective, way to validate the derivative calculation methodology proposed in this work.

The following numerical experiments are presented to first validate and then assess the accuracy of the gradient information computed by the method. For this purpose, a misfit objective function with no regularization term

$$O(\boldsymbol{\theta}) = \frac{1}{2} (\mathbf{h}(\mathbf{x}, \boldsymbol{\theta}) - \mathbf{d}_{obs})^T \mathbf{C}_D^{-1} (\mathbf{h}(\mathbf{x}, \boldsymbol{\theta}) - \mathbf{d}_{obs}), \quad (42)$$

with a gradient

$$\nabla_{\boldsymbol{\theta}} O = \mathbf{G}^T \mathbf{C}_D^{-1} (\mathbf{h}(\mathbf{x}, \boldsymbol{\theta}) - \mathbf{d}_{obs}), \quad (43)$$

is considered (Oliver et al. 2008). In all experiments, the fitting parameters are cell-centered permeabilities and the observed quantity is the fine scale pressure at the location of observation wells. Therefore there is no explicit dependency of the response on the parameters, i.e.,

$$\frac{\partial \mathbf{h}}{\partial \mathbf{x}'} = \mathbf{I}, \quad (44)$$

and

$$\frac{\partial \mathbf{h}}{\partial \bar{\mathbf{x}}} = \mathbf{0}. \quad (45)$$

The tensor $\partial \mathbf{A} / \partial \boldsymbol{\theta}$ represents partial derivatives of the system (transmissibility) matrix with respect to permeability.

Wells and other source terms are considered by supplementary well basis functions (Jenny and Lunati 2009; Tene et al. 2016). Well functions are local solutions of the Eq. (29) considering a unit solution at the well location and zero values at the coarse nodes. Hence, considering well functions, the prolongation operator (for porous rock) is enriched and reads

$$\mathbf{P} = [\boldsymbol{\varphi}_1 \quad \cdots \quad \boldsymbol{\varphi}_{N_c} \mid \boldsymbol{\psi}_1 \quad \cdots \quad \boldsymbol{\psi}_{N_w}], \quad (46)$$

where each well function $\boldsymbol{\psi}_w$ adds a column vector to the porous rock prolongation operator. Note that there are N_w wells in the domain. For rate-constrained wells, one has to add additional rows to

the prolongation (and consequently, the coarse-scale system) in order to solve for the well pressures as additional unknowns (Jenny and Lunati 2009 and Tene et al. 2016). The derivative of Eq. (46) w.r.t. the parameters is given by

$$\frac{\partial \mathbf{P}}{\partial \boldsymbol{\theta}} = \begin{bmatrix} \frac{\partial \boldsymbol{\phi}_1}{\partial \boldsymbol{\theta}} & \dots & \frac{\partial \boldsymbol{\phi}_{N_c}}{\partial \boldsymbol{\theta}} & \frac{\partial \boldsymbol{\psi}_1}{\partial \boldsymbol{\theta}} & \dots & \frac{\partial \boldsymbol{\psi}_{N_w}}{\partial \boldsymbol{\theta}} \end{bmatrix}. \quad (47)$$

The computation of well basis function partial derivatives follows the solution strategy discussed for Eq. (37). In all the following test cases, well basis functions are included.

Validation Experiments

The MS-gradient method is validated against the numerical differentiation method for a higher-order, two-sided Taylor approximation of

$$\nabla_{\boldsymbol{\theta}} O_i = \frac{1}{2\delta\theta_i} \left(O(\theta_i, \dots, \theta_{i-1}, \theta_i + \delta\theta_i, \theta_{i+1}, \dots, \theta_{N_\theta}) - O(\theta_i, \dots, \theta_{i-1}, \theta_i - \delta\theta_i, \theta_{i+1}, \dots, \theta_{N_\theta}) \right), \quad (48)$$

where δ is a multiplicative parameter perturbation. The relative error can be defined as

$$\varepsilon = \frac{\|\nabla_{\boldsymbol{\theta}} O_{FD} - \nabla_{\boldsymbol{\theta}} O_{AN}\|_2}{\|\nabla_{\boldsymbol{\theta}} O_{AN}\|_2}, \quad (49)$$

where $\nabla_{\boldsymbol{\theta}} O_{FD}$ is obtained by performing the appropriate number of multiscale reservoir simulations required to evaluate Eq. (48) and $\nabla_{\boldsymbol{\theta}} O_{AN}$ is obtained by either employing the Direct or Adjoint Method to evaluate Eq. (43). Note that

$$\mathbf{m} = \mathbf{C}_D^{-1} (\mathbf{h}(\mathbf{x}, \boldsymbol{\theta}) - \mathbf{d}_{obs}), \quad (50)$$

where \mathbf{m} is an auxiliary vector, so the gradient of O can be written as $\nabla_{\boldsymbol{\theta}} O = (\mathbf{m}^T \mathbf{G})^T$ and $\nabla_{\boldsymbol{\theta}} O$ can be calculated with a cost proportional to one extra MS simulation, making the Adjoint Method a much more efficient way to calculate the gradient than the Direct Method. Moreover, it is assumed, for simplicity, $\mathbf{C}_D = \mathbf{I}$.

In order to validate the proposed derivative calculation methods, as well as their implementation, we investigate the linear decrease of the error ε by decreasing the perturbation value δ (see chapter 8 of Heath 2002) from 10^{-1} to 10^{-4} (the range within which we observe discretization errors only).

The investigation is carried out in two 2D test cases, one homogeneous and another heterogeneous. In both cases the fine grid size is 21x21, while the coarse grid size is 3x3 (coarsening ratio of 7x7). The primal- and dual-coarse grids are illustrated in Figure 2a.

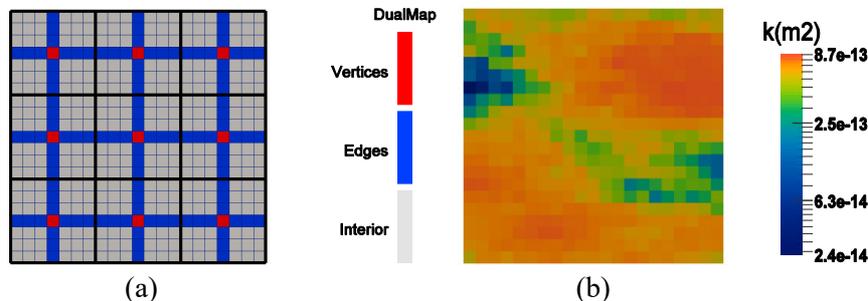


Figure 2 (a) Fine, primal and dual coarse grids for 2D validation test cases and (b) reference permeability field.

The permeability field is extracted from 1,000 geological realizations, as shown in Figure 2b, and serves to compute the observed pressure. For the heterogeneous case, another geological realization is

chosen from the ensemble. The ensemble is generated via the decomposition of a reference permeability “image” using Principal Component Analysis parameterization. Figure 3 illustrates 4 different permeability realizations from the ensemble. See Jansen (2013b) for more details.

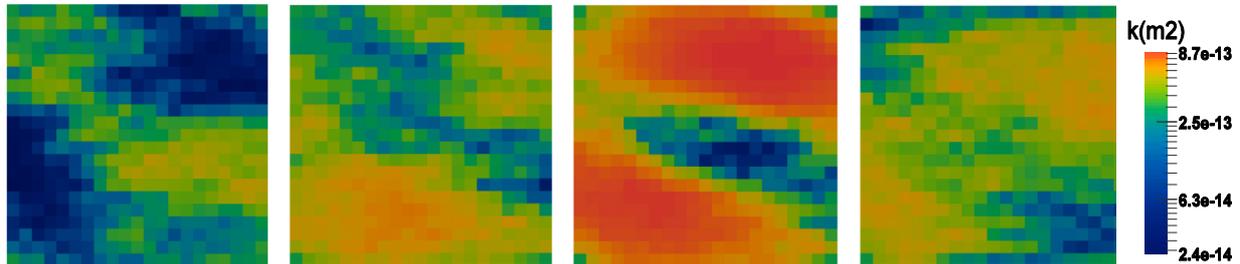


Figure 3 Four different permeability realizations from the ensemble of 1,000 members used in the 2D numerical experiments.

A quarter five-spot well configuration is considered, with two observation wells close to the operating wells. The well positions and operating pressures are described in Table 1.

Table 1 Well configuration for the homogeneous, two-dimensional case.

Well	Fine scale position (I, J)	Well type	Prescribe pressure [Pa]
INJE	(1, 1)	Injection	2
PROD	(21, 21)	Production	1
OBSWELL1	(3, 3)	Observation	-
OBSWELL2	(19, 19)	Observation	-

Figure 4 shows that the MS-gradient and fine-scale methods have exactly the same order of accuracy with respect to the perturbation δ , for all cases. The expected behaviour of linearly decreasing error values as the perturbation size decreases is observed in all experiments. One can also note that the Direct and Adjoint Methods are equally accurate, i.e., they both provide the analytical gradient at the same accuracy.

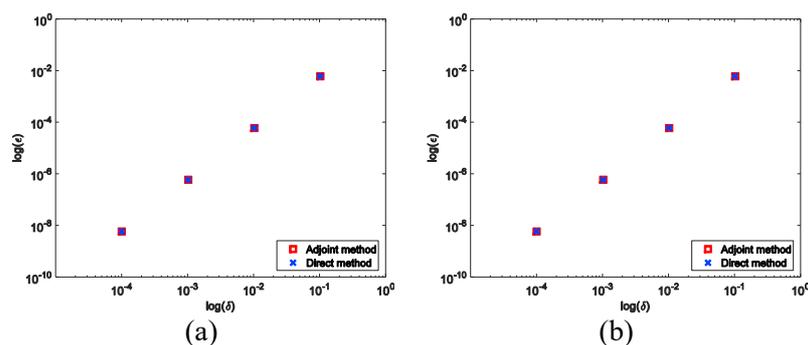


Figure 4 Validation of MS gradient computation method via comparison with numerical differentiation: two-dimensional homogenous (a) and heterogeneous (b) steady-state flow test cases.

Gradient Accuracy

In order to assess the quality of the MS gradient, the angle between fine-scale and MS normalized gradients, i.e.,

$$\alpha = \cos^{-1} \left(\nabla_{\theta}^T \hat{O}_{FS} \nabla_{\theta} \hat{O}_{MS} \right), \quad (51)$$

is measured. Here,

$$\nabla_{\theta} \hat{O}_{FS} = \frac{\nabla_{\theta} O_{FS}}{\|\nabla_{\theta} O_{FS}\|_2} \quad \text{and} \quad \nabla_{\theta} \hat{O}_{MS} = \frac{\nabla_{\theta} O_{MS}}{\|\nabla_{\theta} O_{MS}\|_2}. \quad (52)$$

Also, $\nabla_{\theta} O_{FS}$ and $\nabla_{\theta} O_{MS}$ denote the fine-scale and MS analytical gradients, respectively. As a minimum requirement, acceptable MS gradients are obtained if α is much smaller than 90° (Fonseca et al. 2015).

For the 2D homogenous case, the fine scale and MS gradients result in $\alpha = 10.97^\circ$, using the same setup depicted in Figure 2a. Although the gradients are practically pointing in the same direction, a deviation between the two is observed. To better investigate this difference, Figure 5a and Figure 5b present the fine-scale and MS pressure solutions, respectively, and Figure 5c shows the difference between these two pressure solutions. Figure 5d and Figure 5e present the absolute, normalized gradient of the OF computed via fine-scale and MS gradient methods, respectively. Figure 5f shows the difference between the absolute, normalized gradient computed by the two methods.

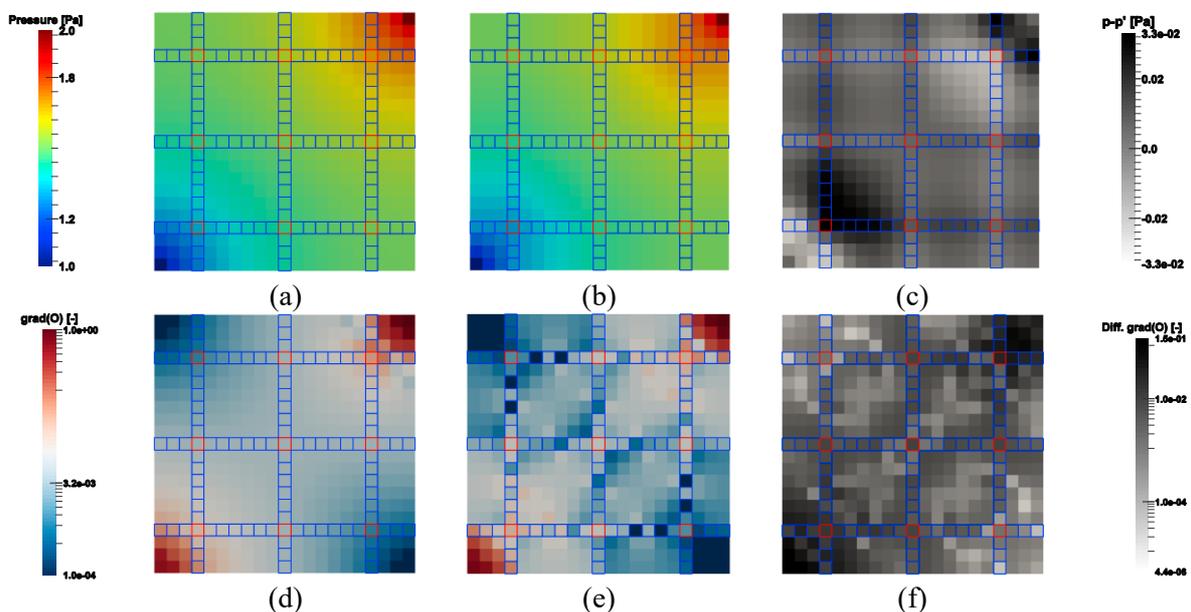


Figure 5 Fine scale (p) pressure solution (a) and MSFV (p') pressure solution (b). Difference between fine-scale and MSFV pressure solutions (c). Absolute, normalized fine scale gradient (d) and MSFV (e). Difference between fine-scale and MS absolute, normalized gradients (f). In both figures the dual-coarse grid is shown. Also shown, in the red boxes, are the coarse nodes.

The difference between the MS and fine-scale methods is due to the localization assumption in the calculation of the basis functions (Hajibeygi et al. 2008). Note that thanks to the well functions, the multiscale solutions are accurately capturing the wells.

Effect of Heterogeneity Distribution

In addition to localization assumptions, basis functions are also affected by heterogeneity distribution. In order to explore this point, four sets of 20 equiprobable geostatistical realizations of permeability fields, generated using sequential Gaussian simulations (Remy et al. 2009), are considered. For each set, variance and mean of $\ln(k)$ are 2.0 and 3.0, respectively, where k is the grid block permeability. As depicted in Figure 6, for the realizations with a long correlation length, the angles between the permeability layers and the horizontal axis are 0° , 15° , and 45° . A patchy (small correlation length) pattern is also considered (Figure 6d). Compared with the previous set, the permeability contrast is much higher in this case.

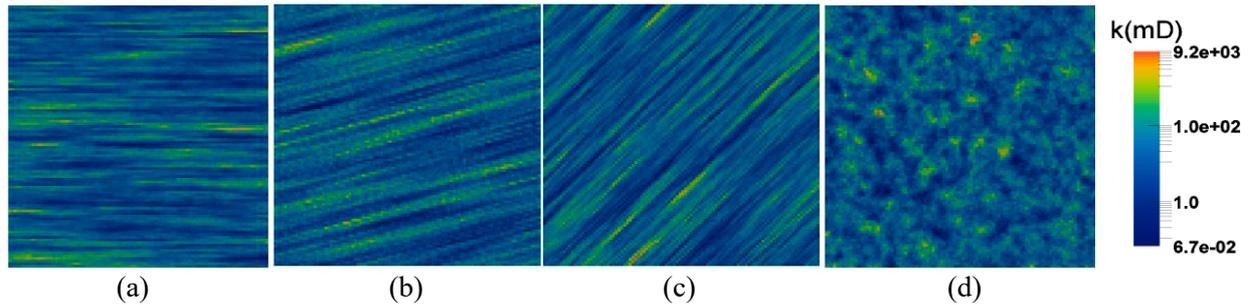


Figure 6 Permeability distribution of four different realizations taken from the sets of 20 geostatistically equiprobable permeability fields with 0° (a), 15° (b), and 45° (c) correlation angles. Also, a patchy field (d) with a small correlation length is considered.

The fine-scale and coarse grids contain 100×100 and 20×20 cells, respectively. The well configuration utilized in this numerical experiment is depicted in Table 2.

Table 2 Well configuration for Case 2.

Well	Fine scale position (I, J)	Well type	Prescribed pressure [Pa]
INJE	(1, 1)	Injection	2
PROD	(100, 100)	Production	1
OBSWELL1	(3, 3)	Observation	-
OBSWELL2	(98, 98)	Observation	-

From Figure 7, one can observe that all cases with $\alpha > 50^\circ$ are associated with small gradient norms and that, for all geological sets, the MS gradient provides gradient directions that are very accurate, compared with the fine scale solution, when the gradient norms are large.

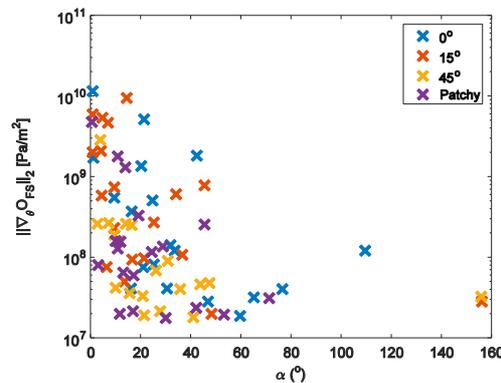


Figure 7 Cross-plot of fine scale gradient norm (log scale) vs. α for the four sets of 20 equiprobable permeability realizations.

Note also that, in the case of a small gradient norm, the OF has a weak dependency on the parameters in the vicinity of the point where the gradient is being calculated. As such, the overall performance of the optimization algorithm is not expected to be affected by replacing the fine-scale gradient calculation with the MS version. In general, an iterative multiscale procedure (Hajibeygi et al. 2008) would guarantee high quality MS solutions –to any desired level– for all challenging scenarios. We intend this to be subject of our future research.

Conclusions

We developed a Multiscale gradient computation strategy using an algebraic framework that does not depend on a particular objective function. This is possible by recasting the derivative calculation problem as the general problem of multiplying a sensitivity matrix and its transpose with arbitrary matrices. Hence the framework is capable of providing different types of gradient information. Such flexibility allows the employment of the framework to any reservoir management study that requires gradient information. Also, the formulation naturally provides both Direct and Adjoint Methods. The choice between one or the other will depend on the type of derivative information requested, i.e. on the ratio between the number of parameters and the number of derivatives required. Since most of the derivations are expressed in terms of general restriction and prolongation operators, the presented strategy can be applied to all MS (and multilevel) methodologies.

The accuracy of the developed MS gradient computation strategy is studied for a set of examples with increasing complexity. The investigations show that the sources of inaccuracies in the MS solution (e.g. localization assumptions) also result in inaccuracies in the gradient computation. However, fortunately, strategies to improve the MS solution (e.g. well functions) also improve the MS gradient accuracy. Another important observation is the fact that greater angles between MS and fine scale gradient vectors are associated with small values of the gradient norms. Because small gradient norms indicate a weak association between parameters and responses, such differences are typically not very relevant in optimization. All this indicates that the method presented allows for accurate, yet less computationally expensive, gradient computation information that can be successfully utilized in reservoir management studies.

The model calibration examples presented show how the MS gradient computation framework can be employed when the parameters are expressed in the fine scale. The framework allows the computation of fine scale derivative information when the prolongation operator depends on the parameters. By employing the MSFV method as forward simulation technique, partial derivatives of the prolongation operator can be computed via the solution of local problems, similarly to the basis function computation. This allows the demanding computations to be performed only at the coarse scale and in the reduced domains associated with the basis functions computation. Although not shown here, the restriction and prolongation operators may not depend on the parameters (e.g. the parameters being the well controls in a life-cycle optimization study). In this case, the computation of partial derivatives of these operators is not necessary. Therefore, for these scenarios, the utilization of MS gradients is even more advantageous from a computational performance perspective.

The indication that strategies to improve the state variables solution also improve the gradient computation suggests that an iterative MS strategy (Hajibeygi et al. 2008) can be an interesting alternative to obtain more accurate gradients. The algebraic framework here introduced allows for the incorporation of a smoothing step in the gradient computation, to resolve high-frequency errors. We intend this extension to be a subject of our future research.

Finally, the multiscale nature of the method may further benefit data assimilation studies. Various types of data are naturally linked to different spatial scales. Hence, it is not uncommon to upscale/downscale quantities to appropriately fit them into the reservoir simulation model. A multiscale data assimilation strategy, in contrast, allows for the appropriate handling of such types of data with minimum loss of information, due to avoiding the excessively upscaled quantities. This will also be a subject of our future research.

Acknowledgements

Rafael Moraes was sponsored by Petrobras S.A. The authors would like to thank the Delft Advanced Reservoir Simulation (DARSim) research team for fruitful discussions during the development of the MS-gradient method.

References

- Aanonsen, S. I., Nævdal, G., Oliver, D. S., Reynolds, A. C., and Vallès, B. [2009]. The ensemble Kalman filter in reservoir engineering - a review. *SPE Journal*, **14**(03), 393-412.
- Bendtsen C. and Stauning, O. [1996]. FADBAD, a Flexible C++ Package for Automatic Differentiation, Using the Forward and Backward Methods (Technical Report IMM-REP-1996-17, Department of Mathematical Modelling, Technical University of Denmark).
- Chavent, G., Dupuy, M., and Lemmonier, P. [1975]. History matching by use of optimal theory. *SPE Journal*, **15** (01), 74-86.
- Chen, W.H., Gavalas, G.R. and Wasserman, M.L. [1974]. A new algorithm for automatic history matching. *SPE Journal* **14**(6) 593-608.
- Chen, Y., Oliver, D.S. and Zhang, D. [2009]. Efficient ensemble-based closed-loop production optimization. *SPE Journal*; **14**(4) 634-645.
- Cusini M., van Kruijsdijk, C., Hajibeygi, H. [2016]. Algebraic dynamic multilevel (ADM) method for fully implicit simulations of multiphase flow in porous media, *Journal of Computational Physics*, **314**, 60-79.
- Datta-Gupta, A. and King, M.J. [2007]. *Streamline simulation: Theory and Practice*, SPE Textbook Series, **11**, SPE, Richardson.
- Durlofsky, L. J. [2005]. Upscaling and gridding of fine scale geological models for flow simulation. In *8th International Forum on Reservoir Simulation Iles Borromees*, Stresa, Italy.
- Efendiev, Y., Hou, T. Y. [2009]. Multiscale Finite Element Methods: Theory and Applications, (Surveys and Tutorials in the Applied Mathematical Sciences, Vol. 4), *Springer*.
- Fonseca, R.M., Chen, B., Jansen, J.D. and Reynolds, A.C. [2016]. A stochastic simplex approximate gradient (StoSAG) for optimization under uncertainty. Submitted to *International Journal for Numerical Methods in Engineering*.
- Fonseca, R.M., Kahrobaei, S., Van Gastel, L.J.T., Leeuwenburgh, O., and Jansen, J.D. [2015]. Quantification of the impact of ensemble size on the quality of an ensemble gradient using principles of hypothesis testing. Paper 173236-MS presented at the 2015 SPE Reservoir Simulation Symposium, Houston, USA, 22-25 February.
- Fu, J., Tchelepi, H. A., and Caers, J. [2010]. A multiscale adjoint method to compute sensitivity coefficients for flow in heterogeneous porous media. *Advances in water resources*, **33**(6), 698-709.
- Fu, J., Caers, J., and Tchelepi, H. A. [2011]. A multiscale method for subsurface inverse modeling: Single-phase transient flow. *Advances in Water Resources*, **34**(8), 967-979.
- Hajibeygi, H. [2011]. Iterative multiscale finite volume method for multiphase flow in porous media with complex physics (Doctoral dissertation, Diss., *Eidgenössische Technische Hochschule ETH Zürich*, Nr. 19872, 2011).
- Hajibeygi, H., Bonfigli, G., Hesse, M. A., and Jenny, P. [2008]. Iterative multiscale finite-volume method. *Journal of Computational Physics*, **227**(19), 8604-8621.
- Hajibeygi, H., Jenny, P. [2011]. Adaptive iterative multiscale finite volume method, *Journal of Computational Physics*, **230**(6), 628-643.
- Hajibeygi, H., Tchelepi, H.A. [2014]. Compositional multiscale finite-volume formulation, *SPE Journal* **19** (2), 316-326.
- Heath, M.T. [2002]. *Scientific Computing: An Introductory Survey*, 2nd edition, McGraw-Hill.
- Hou, T. Y., and Wu, X. H. [1997]. A multiscale finite element method for elliptic problems in composite materials and porous media. *Journal of Computational Physics*, **134**(1), 169-189.
- Jansen, J. D. [2011]. Adjoint-based optimization of multi-phase flow through porous media—a review. *Computers & Fluids*, **46**(1), 40-51.
- Jansen, J.D. [2013a]. *A systems description of flow through porous media*. SpringerBriefs in Earth Sciences, Springer.

- Jansen, J. D. [2013b]. A simple algorithm to generate small geostatistical ensembles for subsurface flow simulation. (Research Note, *TU Delft, Delft University of Technology*).
- Jansen, J.D. [2016]. *Gradient-based optimization of flow through porous media, v.3*. Course notes AES1490. Delft University of Technology.
- Jansen, J.D. and Durlafsky, L., [2016]. Use of reduced-order models in well control optimization. *Optimization and Engineering*. Published online. DOI: 10.1007/s11081-016-9313-6.
- Jenny, P., Lee, S. H., and Tchelepi, H. A. [2003]. Multi-scale finite-volume method for elliptic problems in subsurface flow simulation. *Journal of Computational Physics*, **187**(1), 47-67.
- Jenny, P., & Lunati, I. [2009]. Modeling complex wells with the multi-scale finite-volume method. *Journal of Computational Physics*, **228**(3), 687-702.
- Kozlova, A., Li, Z., Natvig, J., Watanabe, S., Zhou, Y., Bratvedt, K. and Lee, S. [2015]. A Real-Field Multiscale Black-Oil Reservoir Simulator. Paper 173226-MS presented at *SPE Reservoir Simulation Symposium*, 23-25 February, Houston, USA.
- Kraaijevanger, J. F. B. M., Egberts, P. J. P., Valstar, J. R., & Buurman, H. W. [2007]. Optimal waterflood design using the adjoint method. Paper 105764-MS presented at the *SPE Reservoir Simulation Symposium*, 26-28 February, Houston, USA.
- Krogstad, S., Hauge, V. L., and Gulbransen, A. [2011]. Adjoint multiscale mixed finite elements. *SPE Journal*, **16**(01), 162-171.
- Nocedal, J., & Wright, S. [2006]. *Numerical optimization*. Springer Science & Business Media.
- Oliver, D. S., Reynolds, A. C., and Liu, N. [2008]. *Inverse theory for petroleum reservoir characterization and history matching*. Cambridge University Press.
- Oliver, D. S., and Chen, Y. [2011]. Recent progress on reservoir history matching: a review. *Computational Geosciences*, **15**(1), 185-221.
- Remy, N., Boucher, A., and Wu, J. [2009]. *Applied Geostatistics with SGeMS: A User's Guide*, Cambridge University Press, New York.
- Rodrigues, J. R. P. [2006]. Calculating derivatives for automatic history matching. *Computational Geosciences*, **10**(1), 119-136.
- Sarma, P., Aziz, K., and Durlafsky, L. J. [2005]. Implementation of adjoint solution for optimal control of smart wells. Paper 92864-MS presented at the *SPE Reservoir Simulation Symposium*, 31 January-2 February, The Woodlands, USA.
- Shah, S., Moyner, O., Tene, M., Lie, K-A., Hajibeygi, H [2016]. The multiscale restriction smoothed basis method for fractured porous media (F-MsRSB), *Journal of Computational Physics*, 318, 36-57.
- Tene, M., Al Kobaisi, M.S., Hajibeygi, H. [2016]. Algebraic multiscale method for flow in heterogeneous porous media with embedded discrete fractures (F-AMS), *Journal of Computational Physics*, in press, DOI: 10.1016/j.jcp.2016.06.012.
- Vandevoorde, D. and Josuttis, N. M. [2003]. C++ Templates: The Complete Guide. *Addison-Wesley Professional*.
- Wang, Y., Hajibeygi, H., and Tchelepi, H. A. [2014]. Algebraic multiscale solver for flow in heterogeneous porous media. *Journal of Computational Physics*, **259**, 284-303.
- Yeten, B., Castellini, A., Guyaguler, B., and Chen, W. H. [2005]. A comparison study on experimental design and response surface methodologies. Paper 93347-MS presented at the *SPE Reservoir Simulation Symposium*, 31 January-2 February, The Woodlands, USA.
- Younis, R., and Aziz, K. [2007]. Parallel automatically differentiable data-types for next-generation simulator development. Paper 106493-MS presented at the *SPE Reservoir Simulation Symposium*, 26-28 February, Houston, USA.
- Zhou, H. and Tchelepi, H.A. [2008]. Operator-based multiscale method for compressible flow. *SPE Journal*, **13**(2), 523-539.
- Zhou, H. and Tchelepi, H.A. [2012]. Two-stage algebraic multiscale linear solver for highly heterogeneous reservoir models. *SPE Journal*, **17**(2), 523-539.