

Iterative multiscale gradient computation for heterogeneous subsurface flow

Jesus de Moraes, Rafael; de Zeeuw, Wessel; R. P. Rodrigues, José; Hajibeygi, Hadi; Jansen, Jan Dirk

DOI

[10.1016/j.advwatres.2019.05.016](https://doi.org/10.1016/j.advwatres.2019.05.016)

Publication date

2019

Document Version

Final published version

Published in

Advances in Water Resources

Citation (APA)

Jesus de Moraes, R., de Zeeuw, W., R. P. Rodrigues, J., Hajibeygi, H., & Jansen, J. D. (2019). Iterative multiscale gradient computation for heterogeneous subsurface flow. *Advances in Water Resources*, 129, 210-221. <https://doi.org/10.1016/j.advwatres.2019.05.016>

Important note

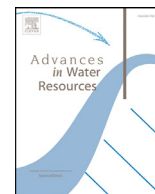
To cite this publication, please use the final published version (if applicable). Please check the document version above.

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.



Iterative multiscale gradient computation for heterogeneous subsurface flow

Rafael J. de Moraes^{a,b,*}, Wessel de Zeeuw^a, José Roberto P. Rodrigues^b, Hadi Hajibeygi^a, Jan Dirk Jansen^a

^a Department of Geoscience and Engineering, Faculty of Civil Engineering and Geosciences, Delft University of Technology, P.O. Box 5048, Delft 2600, the Netherlands

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

ARTICLE INFO

Keywords:

Subsurface flow
Gradient computation
Multiscale methods
Iterative multiscale finite volume
Direct method
Adjoint method

ABSTRACT

We introduce a semi-analytical iterative multiscale derivative computation methodology that allows for error control and reduction to any desired accuracy, up to fine-scale precision. The model responses are computed by the multiscale forward simulation of flow in heterogeneous porous media. The derivative computation method is based on the augmentation of the model equation and state vectors with the smoothing stage defined by the iterative multiscale method. In the formulation, we avoid additional complexity involved in computing partial derivatives associated to the smoothing step. We account for it as an approximate derivative computation stage. The numerical experiments illustrate how the newly introduced derivative method computes misfit objective function gradients that converge to fine-scale one as the iterative multiscale residual converges. The robustness of the methodology is investigated for test cases with high contrast permeability fields. The iterative multiscale gradient method casts a promising approach, with minimal accuracy-efficiency tradeoff, for large-scale heterogeneous porous media optimization problems.

1. Introduction

Derivative computation is an important aspect of gradient-based optimization algorithms. When the objective (or cost) function evaluation involves the numerical simulation of discretized partial derivative equations (PDE), efficient gradient computation is of utmost importance. It is well documented in the literature that the most efficient and accurate way of computing derivatives are the analytical direct (Anterion et al., 1989; Rodrigues, 2006; Oliver et al., 2008) (when the number of cost functionals is greater than the number of parameters) and Adjoint (Chavent et al., 1975; Li et al., 2003; Oliver et al., 2008; Kraaijevanger et al., 2007; Rodrigues, 2006; Jansen, 2011) (if the number of parameters is greater than the number cost functionals) methods. However, even when considering efficient gradient methods, due to the necessity to evaluate the forward model and the derivative information many times, up until the optimality conditions are met, techniques to reduce the forward model simulation cost have been proposed (Jansen, 2011; Cardoso et al., 2009; van Doren et al., 2006).

Multiscale (MS) simulation methods (Jenny et al., 2003; Hou and Wu, 1997) have been increasingly employed for the efficient solution

of elliptic (Zhou and Tchelepi, 2008) and parabolic (Tene et al., 2015) equations, more specifically subsurface flow problems in highly heterogeneous porous media. Also, many developments to extend its applicability to extended physics have been observed in the recent years (Lie et al., 2017).

Moreover, MS derivative computation strategies, based on MS forward simulation models, has also been subject of study. MS adjoint formulation (MS-ADJ) for single-phase subsurface flow have been presented in Fu et al. (2010, 2011). MS adjoint computation methods for multiphase flow have also been developed (Krogstad et al., 2011; Moraes et al., 2017). More recently, a mathematical framework for MS computation of derivative information has been developed (de Moraes et al., 2017). It has been highlighted in de Moraes et al. (2017); Fu et al. (2010) that inaccurate MS gradient computations could lead to inaccurate gradient directions. However, as it is indicated in de Moraes et al. (2017), strategies that improve the MS forward simulation solution (e.g. refinement of the MS coarse grid) result in better gradient estimates. Moreover, in Fu et al. (2010) it is suggested that an iterative MS gradient computation strategy could resolve the multiscale gradient inaccuracies.

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

E-mail addresses: r.moraes@tudelft.nl (R.J. de Moraes), wesseldezeeuw@hotmail.com (W. de Zeeuw), jrprodrigues@petrobras.com (J.R.P. Rodrigues), h.hajibeygi@tudelft.nl (H. Hajibeygi), j.d.jansen@tudelft.nl (J.D. Jansen).

<https://doi.org/10.1016/j.advwatres.2019.05.016>

Received 18 September 2018; Received in revised form 20 May 2019; Accepted 23 May 2019

Available online 23 May 2019

0309-1708/© 2019 The Authors. Published by Elsevier Ltd. This is an open access article under the CC BY license. (<http://creativecommons.org/licenses/by/4.0/>)

In this work, we develop an iterative multiscale gradient computation strategy which converges to the fine-scale gradient solution, thus allowing for error control and reduction for multiscale gradients. The derivative computation method is based on the generic mathematical framework introduced in [de Moraes et al. \(2017\)](#). By augmenting the MS model equation and the state vectors with the i-MSFV smoothing stage, the framework is capable of providing derivative information at any desired accuracy, up to fine-scale precision. The augmentation is addressed by the implicit differentiation strategy. In the formulation, we avoid additional complexity involved in computing partial derivatives associated to the smoothing step by also only approximately solving the derivative state equation associated to the it. Also, the strategy seamlessly accommodates both the Direct and Adjoint methods.

The remaining of this paper is organized as follows. Firstly, we present an algebraic formulation for the i-MSFV method, suitable for the derivation of the derivative computation methods. Next, we derive the Direct and Adjoint methods to compute derivative information, following the i-MSFV framework, when the algorithms are presented. Numerical validation of the method against numerical differentiation is presented. The numerical experiments conducted show that fine-scale gradient can be reproduced via the i-MSFV method if the forward simulation converges to a small enough residual tolerance. We show numerical evidence that there is a relationship between the gradient quality and the i-MSFV solution residual by comparing fine-scale gradient and i-MSFV gradients and relating the difference between the two with the pressure error norm. Concluding remarks are finally presented in the last section.

2. Algebraic and algorithmic description of the multiscale iterative method

We consider the set of equations that algebraically describes the forward simulation at the fine scale, without any assumption regarding the underlying physical model, as ([de Moraes et al., 2017](#))

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

where $\mathbf{g}_F : \mathbb{R}^{N_F} \times \mathbb{R}^{N_\theta} \rightarrow \mathbb{R}^{N_F}$ represents the set of algebraic forward model equations, $\mathbf{x} \in \mathbb{R}^{N_F}$ is the state vector (which, for single-phase flow, contains the grid block pressures), $\theta \in \mathbb{R}^{N_\theta}$ is the vector of 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 θ , i.e.

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

Once the model state is determined, the observable responses of the forward model are computed. The forward model responses may not only depend on the model state, but also on the parameters themselves, and can be expressed as

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

where $\mathbf{h}_F : \mathbb{R}^{N_F} \times \mathbb{R}^{N_\theta} \rightarrow \mathbb{R}^{N_y}$ represents the output equations ([Jansen, 2016](#)). It is assumed that \mathbf{g}_F can be described as

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

where $\mathbf{A}(\theta) \in \mathbb{R}^{N_F} \times \mathbb{R}^{N_F}$ matrix and $\mathbf{q}(\theta) \in \mathbb{R}^{N_F}$.

A two-stage multiscale (MS) solution strategy ([Jenny et al., 2003](#); [Wang et al., 2014](#)) can be devised by firstly computing a coarse scale solution

$$\check{\mathbf{g}}(\check{\mathbf{x}}, \theta) = (\mathbf{R}\mathbf{A}\mathbf{P})\check{\mathbf{x}} - (\mathbf{R}\mathbf{q}) = \check{\mathbf{A}}\check{\mathbf{x}} - \check{\mathbf{q}} = \check{\mathbf{0}}, \quad (5)$$

$\check{\mathbf{g}} : \mathbb{R}^{N_C} \times \mathbb{R}^{N_\theta} \rightarrow \mathbb{R}^{N_C}$, where N_C is the number of coarse grid-blocks, and then an approximated fine-scale solution

$$\mathbf{g}'(\mathbf{x}', \check{\mathbf{x}}, \theta) = \mathbf{x}' - \mathbf{P}\check{\mathbf{x}} = \mathbf{0}, \quad (6)$$

where $\mathbf{g}' : \mathbb{R}^{N_F} \times \mathbb{R}^{N_C} \times \mathbb{R}^{N_\theta} \rightarrow \mathbb{R}^{N_F}$.

The so-called prolongation operator $\mathbf{P} = \mathbf{P}(\theta)$ which is an $N_F \times N_C$ matrix that maps (interpolates) the coarse-scale solution to the fine-scale resolution. The so-called restriction operator $\mathbf{R} = \mathbf{R}(\theta)$ is defined as an $N_C \times N_F$ matrix which maps the fine scale to the coarse scale. More information about how these operators are constructed for the Multiscale Finite Volume (MSFV) method can be found in [Zhou and Tchelepi \(2008\)](#); [Wang et al. \(2014\)](#). Let $\check{\mathbf{x}} \in \mathbb{R}^{N_C}$ be the coarse scale solution ($N_C \ll N_F$), and $\mathbf{x}' \in \mathbb{R}^{N_F}$ the approximated fine-scale solution.

The iterative multiscale strategy ([Hajibeygi et al., 2008](#)) can be devised by considering versions of [Eqs. \(4\)](#) and [\(5\)](#) written in residual form. Let \mathbf{x}^{v-1} be an approximate solution to [Eq. \(4\)](#) at iteration $v - 1$ and

$$\mathbf{r}^{v-1} = \mathbf{q} - \mathbf{A}\mathbf{x}^{v-1} \quad (7)$$

be the corresponding residual. A multiscale improvement to this approximation can be devised by writing [Eq. \(5\)](#) in residual form as

$$\check{\mathbf{g}}^v(\check{\mathbf{x}}^v, \mathbf{x}^{v-1}, \theta) = \check{\mathbf{A}}\check{\mathbf{x}}^v - \check{\mathbf{r}}^{v-1} = \check{\mathbf{0}}, \quad (8)$$

where

$$\check{\mathbf{r}}^{v-1} = \mathbf{R}\mathbf{r}^{v-1}, \quad (9)$$

$\check{\mathbf{r}}^{v-1} \in \mathbb{R}^{N_C}$. Here, $\check{\mathbf{x}}^v$ is redefined as the coarse scale correction. Redefining [Eq. \(6\)](#), we have

$$\mathbf{g}''^v(\mathbf{x}'^v, \check{\mathbf{x}}^v, \theta) = \mathbf{x}'^v - \mathbf{P}\check{\mathbf{x}}^v = \mathbf{0}, \quad (10)$$

such that \mathbf{x}'^v now represents the approximate fine-scale correction at iteration v , i.e.,

$$\mathbf{x}^{v-1/2} = \mathbf{x}^{v-1} + \mathbf{x}'^v \quad (11)$$

is an approximate solution of [Eq. \(4\)](#) augmented with the correction from the coarse-scale calculation.

The approximate solution provided by [Eq. \(11\)](#) can be improved if successive smoothing steps are employed ([Hajibeygi et al., 2008](#)). Let

$$\mathbf{r}_\sigma^{v-1} = \mathbf{q} - \mathbf{A}\mathbf{x}^{v-1/2} = \mathbf{q} - \mathbf{A}(\mathbf{x}^{v-1} + \mathbf{x}'^v), \quad (12)$$

$\mathbf{r}_\sigma^{v-1} \in \mathbb{R}^{N_F}$, be the smoothed residual obtained from the approximation given by [Eq. \(11\)](#) and

$$\mathbf{g}_\sigma^v(\mathbf{x}'^v, \mathbf{x}_\sigma^v, \mathbf{x}^{v-1}, \theta) = \mathbf{A}\mathbf{x}_\sigma^v - \mathbf{r}_\sigma^{v-1} = \mathbf{0}, \quad (13)$$

a version of [Eq. \(4\)](#) written in residual form. Here $\mathbf{x}_\sigma^v \in \mathbb{R}^{N_F}$ is the smoothed fine-scale correction at iteration v . The solution smoothing is obtained by solving [Eq. \(13\)](#) using any iterative solver up to a prescribed (loose) tolerance or (small) maximum number of iterations ([Hajibeygi et al., 2008](#); [Tene et al., 2015](#)). The solution for a given iteration v is, hence, obtained from

$$\mathbf{g}_x^v(\mathbf{x}^v, \mathbf{x}'^v, \mathbf{x}_\sigma^v, \mathbf{x}^{v-1}, \theta) = \mathbf{x}^v - \mathbf{x}^{v-1} - \mathbf{x}'^v - \mathbf{x}_\sigma^v = \mathbf{0}, \quad (14)$$

where $\mathbf{g}_x^v : \mathbb{R}^{N_F} \times \mathbb{R}^{N_F} \times \mathbb{R}^{N_F} \times \mathbb{R}^{N_F} \times \mathbb{R}^{N_\theta} \rightarrow \mathbb{R}^{N_F}$.

The MS iterative strategy is fully depicted in [Algorithm 1](#), where $\|\cdot\|$ represents the 2-norm.

In [Algorithm 1](#), ϵ and ϵ_σ are, respectively, the user-defined tolerances for the outer-loop and smoothing step. That allows to control the smoothing step as a relative improvement starting from the MS approximate solution. An investigation of an optimal relationship between the number of outer loops and the number of smoothing steps is presented in [Tene et al. \(2015\)](#).

3. Iterative multiscale gradient computation

For the developments that will follow in this section, it is convenient to write the set of equations that is solved in every iteration, namely

Algorithm 1: Iterative multiscale method (Hajibeygi et al., 2008).

Input : $\check{\mathbf{A}}, \mathbf{A}, \mathbf{R}, \mathbf{P}, \mathbf{q}, \epsilon, \epsilon_\sigma$
Output: Approximate solution for the linear system $\mathbf{Ax} = \mathbf{q}$

- 1 Set $\mathbf{x}^0 = \mathbf{0}$
- 2 **for** $v = 1, 2, \dots$ **do**
- 3 Compute $\mathbf{r}^{v-1} = \mathbf{q} - \mathbf{Ax}^{v-1}$
- 4 If $\frac{\|\mathbf{r}^{v-1}\|}{\|\mathbf{r}^0\|} < \epsilon$, quit with solution given by $\mathbf{x} = \mathbf{x}^{v-1}$
- 5 Solve $\check{\mathbf{x}}^v = \check{\mathbf{A}}^{-1}(\mathbf{R}\mathbf{r}^{v-1})$
- 6 Compute $\mathbf{x}'^v = \mathbf{P}\check{\mathbf{x}}^v$
- 7 Compute $\mathbf{r}_\sigma^{v-1} = \mathbf{q} - \mathbf{A}(\mathbf{x}^{v-1} + \mathbf{x}'^v)$
- 8 Iteratively solve $\mathbf{x}_\sigma^v = \mathbf{A}^{-1}\mathbf{r}_\sigma^{v-1}$ until $\frac{\|\mathbf{r}_\sigma^{v-1} - \mathbf{Ax}_\sigma^v\|}{\|\mathbf{r}_\sigma^{v-1}\|} < \epsilon_\sigma$
- 9 Update $\mathbf{x}^v = \mathbf{x}^{v-1} + \mathbf{x}'^v + \mathbf{x}_\sigma^v$

Eqs. (8), (10), (13) and (14), in matrix form as

$$\begin{pmatrix} 0 & 0 & 0 & \mathbf{RA} & \check{\mathbf{A}} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -\mathbf{P} & \mathbf{I} & 0 & 0 \\ 0 & 0 & 0 & \mathbf{A} & 0 & \mathbf{A} & \mathbf{A} & 0 \\ 0 & 0 & 0 & -\mathbf{I} & 0 & -\mathbf{I} & -\mathbf{I} & \mathbf{I} \end{pmatrix} \times \begin{pmatrix} \check{\mathbf{x}}^{\nu-1} \\ \mathbf{x}^{\nu-1} \\ \mathbf{x}_\sigma^{\nu-1} \\ \mathbf{x}^{\nu-1} \\ \check{\mathbf{x}}^\nu \\ \mathbf{x}'^\nu \\ \mathbf{x}_\sigma^\nu \\ \mathbf{x}^\nu \end{pmatrix} - \begin{pmatrix} \mathbf{Rq} \\ 0 \\ \mathbf{q} \\ 0 \end{pmatrix} = \mathbf{0} \quad (15)$$

One must note, however, that, in the i-MSFV procedure, Eq. (13) is solved only approximately and, therefore, strictly speaking the equation in the third row of Eq. (15) does not hold. The idea here is to describe the procedure in an algebraic manner, ignoring this approximation, in order to facilitate the presentation of the derivative calculation algorithms in the next section. Once the derivative calculation methods are obtained under the assumption that the algebraic relations in Eq. (15) hold, the same type of smoothing approach employed in the i-MSFV to resolve high frequency errors will be used in the solution of the derivative information. This results in a practical semi-analytical algorithm for derivative calculation in an iterative formulation. Note that a fully analytical procedure would require calculating the derivative of the smoothing operator employed in the i-MSFV (step 8 in Algorithm 1), which can be quite complex, due to its nonlinear character (Frank, 2017). The proposed semi-analytical approach also becomes general and applicable to any iterative procedure used in step 8, regardless of its nature. A truly analytical derivative method would require the implementation of derivative calculation for each iterative procedure used. More details on the implication of this assumption are discussed in 3.2.1.

Considering all equations that must be solved for all iterations $v \in \{1, \dots, N_v\}$, they can be collated in a super-vector (Rodrigues, 2006; Kraaijevanger et al., 2007; Jansen, 2016) fashion as

$$\mathbf{g}(\mathbf{x}, \theta) = \begin{pmatrix} \check{\mathbf{g}}^1(\check{\mathbf{x}}^1, \mathbf{x}'^0, \mathbf{x}_\sigma^0, \theta) \\ \mathbf{g}'^1(\mathbf{x}'^1, \check{\mathbf{x}}^1, \theta) \\ \mathbf{g}_\sigma^1(\mathbf{x}_\sigma^1, \mathbf{x}'^1, \mathbf{x}'^0, \mathbf{x}_\sigma^0, \theta) \\ \mathbf{g}_x^1(\mathbf{x}^1, \mathbf{x}_\sigma^1, \mathbf{x}'^1, \mathbf{x}'^0, \theta) \mathbf{g}^2(\check{\mathbf{x}}^2, \mathbf{x}'^1, \mathbf{x}_\sigma^1, \theta) \\ \mathbf{g}'^2(\mathbf{x}'^2, \check{\mathbf{x}}^2, \theta) \\ \mathbf{g}_\sigma^2(\mathbf{x}_\sigma^2, \mathbf{x}'^2, \mathbf{x}'^1, \mathbf{x}_\sigma^1, \theta) \\ \mathbf{g}_x^2(\mathbf{x}^2, \mathbf{x}_\sigma^2, \mathbf{x}'^2, \mathbf{x}'^1, \theta) \\ \vdots \\ \check{\mathbf{g}}^{N_v}(\check{\mathbf{x}}^{N_v}, \mathbf{x}'^{N_v-1}, \mathbf{x}_\sigma^{N_v-1}, \theta) \\ \mathbf{g}'^{N_v}(\mathbf{x}'^{N_v}, \check{\mathbf{x}}^{N_v}, \theta) \\ \mathbf{g}_\sigma^{N_v}(\mathbf{x}_\sigma^{N_v}, \mathbf{x}'^{N_v}, \mathbf{x}'^{N_v-1}, \mathbf{x}_\sigma^{N_v-1}, \theta) \\ \mathbf{g}_x^{N_v}(\mathbf{x}^{N_v}, \mathbf{x}_\sigma^{N_v}, \mathbf{x}'^{N_v}, \mathbf{x}'^{N_v-1}, \theta) \end{pmatrix} = \mathbf{0}, \quad (16)$$

and the super-state vector defined as

$$\mathbf{x}(\theta) = \begin{pmatrix} \check{\mathbf{x}}^1 \\ \mathbf{x}'^1 \\ \mathbf{x}_\sigma^1 \\ \mathbf{x}^1 \end{pmatrix}^T \dots \begin{pmatrix} \check{\mathbf{x}}^{N_v-1} \\ \mathbf{x}'^{N_v-1} \\ \mathbf{x}_\sigma^{N_v-1} \\ \mathbf{x}^{N_v-1} \end{pmatrix}^T \begin{pmatrix} \check{\mathbf{x}}^{N_v} \\ \mathbf{x}'^{N_v} \\ \mathbf{x}_\sigma^{N_v} \\ \mathbf{x}^{N_v} \end{pmatrix}^T. \quad (17)$$

Note that we use bold-italic in the notation to represent super-vectors.

It is discussed in Rodrigues (2006); de Moraes et al. (2017), de Moraes et al. how any derivative information can be efficiently computed from the pre and post multiplication of the sensitivity matrix \mathbf{G} , $\mathbf{G} \in \mathbb{R}^{N_y \times N_\theta}$, by arbitrary matrices as

$$\mathbf{WGV} = -\mathbf{W} \frac{\partial \mathbf{h}}{\partial \mathbf{x}} \left(\frac{\partial \mathbf{g}}{\partial \mathbf{x}} \right)^{-1} \frac{\partial \mathbf{g}}{\partial \theta} \mathbf{V} + \mathbf{W} \frac{\partial \mathbf{h}}{\partial \theta} \mathbf{V}, \quad (18)$$

where \mathbf{V} (of order $N_\theta \times p$) and \mathbf{W} (of order $m \times N_y$) are arbitrary matrices, defined based on the derivative information one wants to obtain.

Eq. (18) requires the partial derivative of the model equations with respect to the parameters. From Eqs. (5) and (6), as discussed in de Moraes et al. (2017), it follows that

$$\frac{\partial \check{\mathbf{g}}^v}{\partial \theta} = \left[\frac{\partial \mathbf{R}}{\partial \theta} (\mathbf{AP}) + \mathbf{R} \frac{\partial \mathbf{A}}{\partial \theta} \mathbf{P} + (\mathbf{RA}) \frac{\partial \mathbf{P}}{\partial \theta} \right] \check{\mathbf{x}}^v - \frac{\partial \mathbf{R}}{\partial \theta} \mathbf{q} + \mathbf{R} \frac{\partial \mathbf{q}}{\partial \theta} + \left(\frac{\partial \mathbf{R}}{\partial \theta} \mathbf{A} + \mathbf{R} \frac{\partial \mathbf{A}}{\partial \theta} \right) \mathbf{x}^{v-1}, \quad (19)$$

$$\frac{\partial \mathbf{g}'^v}{\partial \theta} = -\frac{\partial \mathbf{P}}{\partial \theta} \check{\mathbf{x}}^v, \quad (20)$$

and deriving Eq. (13) with respect to θ

$$\frac{\partial \mathbf{g}_\sigma^v}{\partial \theta} = \frac{\partial \mathbf{A}}{\partial \theta} \mathbf{x}_\sigma^v + \frac{\partial \mathbf{A}}{\partial \theta} \mathbf{x}'^v - \frac{\partial \mathbf{q}}{\partial \theta} + \frac{\partial \mathbf{A}}{\partial \theta} \mathbf{x}^{v-1}. \quad (21)$$

Also, from Eq. (14), it follows that

$$\frac{\partial \mathbf{g}_x^v}{\partial \theta} = \mathbf{0} \quad (22)$$

For the sake of simplicity and in coherence with the MSFV method used in the numerical experiments, the dependency of \mathbf{R} on θ is neglected.

Now, the order the operations in Eq. (18) are evaluated define the Direct and Adjoint methods. The derivation of both methods will be discussed in the next sections.

3.1. The direct method

If \mathbf{W} is factored out in Eq. (18), it can be rewritten as

$$\mathbf{G}\mathbf{V} = \frac{\partial \mathbf{h}}{\partial \mathbf{x}} \mathbf{Z} + \frac{\partial \mathbf{h}}{\partial \theta} \mathbf{V}, \tag{23}$$

where

$$\mathbf{Z} = -\left(\frac{\partial \mathbf{g}}{\partial \mathbf{x}}\right)^{-1} \frac{\partial \mathbf{g}}{\partial \theta} \mathbf{V}, \tag{24}$$

is solved from

$$\left(\frac{\partial \mathbf{g}}{\partial \mathbf{x}}\right) \mathbf{Z} = -\frac{\partial \mathbf{g}}{\partial \theta} \mathbf{V}. \tag{25}$$

The linear system described in Eq. (25) can be re-written in a block-wise fashion for each iteration ν :

$$\begin{pmatrix} \frac{\partial \mathbf{g}^0}{\partial \mathbf{x}^0} & & & \\ \frac{\partial \mathbf{g}^1}{\partial \mathbf{x}^0} & \frac{\partial \mathbf{g}^1}{\partial \mathbf{x}^1} & & \\ & \ddots & \ddots & \\ & & \frac{\partial \mathbf{g}^{N_\nu}}{\partial \mathbf{x}^{N_\nu-1}} & \frac{\partial \mathbf{g}^{N_\nu}}{\partial \mathbf{x}^{N_\nu}} \end{pmatrix} \begin{pmatrix} \mathbf{Z}^0 \\ \mathbf{Z}^1 \\ \vdots \\ \mathbf{Z}^{N_\nu} \end{pmatrix} = \begin{pmatrix} \frac{\partial \mathbf{g}^0}{\partial \theta} \mathbf{V} \\ \frac{\partial \mathbf{g}^1}{\partial \theta} \mathbf{V} \\ \vdots \\ \frac{\partial \mathbf{g}^{N_\nu}}{\partial \theta} \mathbf{V} \end{pmatrix}, \tag{26}$$

where, from Eqs. (8), (10), (13), and (14)

$$\frac{\partial \mathbf{g}^\nu}{\partial \mathbf{x}^\nu} = \begin{pmatrix} \frac{\partial \check{\mathbf{g}}^\nu}{\partial \mathbf{x}^\nu} \\ \frac{\partial \mathbf{g}^\nu}{\partial \mathbf{g}^\nu} \\ \frac{\partial \mathbf{x}^\nu}{\partial \mathbf{g}_\sigma} \\ \frac{\partial \mathbf{x}^\nu}{\partial \mathbf{g}_x} \end{pmatrix} = \begin{pmatrix} \check{\mathbf{A}} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ -\mathbf{P} & \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{A} & \mathbf{A} & \mathbf{0} \\ \mathbf{0} & -\mathbf{I} & -\mathbf{I} & \mathbf{I} \end{pmatrix}, \tag{27}$$

and

$$\frac{\partial \mathbf{g}^\nu}{\partial \mathbf{x}^{\nu-1}} = \begin{pmatrix} \frac{\partial \check{\mathbf{g}}^\nu}{\partial \mathbf{x}^{\nu-1}} \\ \frac{\partial \mathbf{g}'^\nu}{\partial \mathbf{x}^{\nu-1}} \\ \frac{\partial \mathbf{x}^\nu}{\partial \mathbf{g}_\sigma} \\ \frac{\partial \mathbf{x}^\nu}{\partial \mathbf{g}_x} \end{pmatrix} = \begin{pmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{R}\mathbf{A} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{A} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & -\mathbf{I} \end{pmatrix}. \tag{28}$$

The partitioning lines indicate which matrix and vector terms belong to each iteration.

Substituting Eq. (27) and (28) in Eq. (26), follows that

$$\begin{pmatrix} \frac{\partial \check{\mathbf{g}}^\nu}{\partial \mathbf{x}^\nu} \\ \frac{\partial \mathbf{x}^\nu}{\partial \mathbf{g}^\nu} \\ \frac{\partial \mathbf{x}^\nu}{\partial \mathbf{g}_\sigma} \\ \frac{\partial \mathbf{x}^\nu}{\partial \mathbf{g}_x} \\ \frac{\partial \mathbf{x}^\nu}{\partial \mathbf{x}^{\nu-1}} \\ \frac{\partial \check{\mathbf{g}}^\nu}{\partial \mathbf{x}^{\nu-1}} \\ \frac{\partial \mathbf{x}^{\nu-1}}{\partial \mathbf{g}^{\nu-1}} \\ \frac{\partial \mathbf{x}^{\nu-1}}{\partial \mathbf{g}'^{\nu-1}} \\ \frac{\partial \mathbf{x}^{\nu-1}}{\partial \mathbf{g}_\sigma} \\ \frac{\partial \mathbf{x}^{\nu-1}}{\partial \mathbf{g}_x} \\ \frac{\partial \mathbf{x}^{\nu-1}}{\partial \mathbf{x}^{\nu-2}} \\ \vdots \\ \frac{\partial \check{\mathbf{g}}^\nu}{\partial \mathbf{x}^{\nu-1}} \\ \frac{\partial \mathbf{x}^{\nu-1}}{\partial \mathbf{g}^{\nu-1}} \\ \frac{\partial \mathbf{x}^{\nu-1}}{\partial \mathbf{g}'^{\nu-1}} \\ \frac{\partial \mathbf{x}^{\nu-1}}{\partial \mathbf{g}_\sigma} \\ \frac{\partial \mathbf{x}^{\nu-1}}{\partial \mathbf{g}_x} \end{pmatrix} \begin{pmatrix} \check{\mathbf{Z}}^0 \\ \mathbf{Z}'^0 \\ \mathbf{Z}_\sigma^0 \\ \mathbf{Z}_x^0 \\ \check{\mathbf{Z}}^1 \\ \mathbf{Z}'^1 \\ \mathbf{Z}_\sigma^1 \\ \mathbf{Z}_x^1 \\ \vdots \\ \check{\mathbf{Z}}^{N_\nu} \\ \mathbf{Z}'^{N_\nu} \\ \mathbf{Z}_\sigma^{N_\nu} \\ \mathbf{Z}_x^{N_\nu} \end{pmatrix} = \begin{pmatrix} \frac{\partial \check{\mathbf{g}}^0}{\partial \theta} \mathbf{V} \\ \frac{\partial \mathbf{g}'^0}{\partial \theta} \mathbf{V} \\ \frac{\partial \mathbf{g}_\sigma^0}{\partial \theta} \mathbf{V} \\ \frac{\partial \theta}{\partial \theta} \mathbf{0} \\ \frac{\partial \check{\mathbf{g}}^1}{\partial \theta} \mathbf{V} \\ \frac{\partial \mathbf{g}'^1}{\partial \theta} \mathbf{V} \\ \frac{\partial \mathbf{g}_\sigma^1}{\partial \theta} \mathbf{V} \\ \frac{\partial \theta}{\partial \theta} \mathbf{0} \\ \vdots \\ \frac{\partial \check{\mathbf{g}}^{N_\nu}}{\partial \theta} \mathbf{V} \\ \frac{\partial \mathbf{g}'^{N_\nu}}{\partial \theta} \mathbf{V} \\ \frac{\partial \mathbf{g}_\sigma^{N_\nu}}{\partial \theta} \mathbf{V} \\ \frac{\partial \theta}{\partial \theta} \mathbf{0} \end{pmatrix}, \tag{29}$$

For every iteration, the linear system that must be solved for the coarse-scale equation is

$$\check{\mathbf{Z}}^\nu = \check{\mathbf{A}}^{-1} \left(-\frac{\partial \check{\mathbf{g}}^\nu}{\partial \theta} \mathbf{V} - \mathbf{R}\mathbf{A}\mathbf{Z}_x^{\nu-1} \right), \tag{30}$$

for the fine-scale approximate solution equation

$$\mathbf{Z}'^\nu = -\mathbf{P}\check{\mathbf{Z}}^\nu - \frac{\partial \mathbf{g}'^\nu}{\partial \theta} \mathbf{V}, \tag{31}$$

and for the smoothing equation

$$\mathbf{Z}_\sigma^v = \mathbf{A}^{-1} \left(-\frac{\partial \mathbf{g}_\sigma^v}{\partial \theta} \mathbf{V} - \mathbf{A} \mathbf{Z}'^v - \mathbf{A} \mathbf{Z}_{x'}^{v-1} \right). \quad (32)$$

As pointed out above, it would not be feasible to fully solve Eq. (32). In order to obtain a practical derivative calculation method, the same smoothing procedure used in the i-MSFV is applied here, i.e., \mathbf{Z}_σ^v is obtained by solving Eq. (32) using any iterative solver up to a prescribed (loose) tolerance or a (small) maximum number of iterations.

Finally,

$$\mathbf{Z}_x^v = \mathbf{Z}_x^{v-1} + \mathbf{Z}'^v + \mathbf{Z}_\sigma^v. \quad (33)$$

Observing that \mathbf{h} only depends on $\mathbf{x} = \mathbf{x}^{N_\nu}$, Eq. (23) can be simplified to

$$\mathbf{G} \mathbf{V} = \frac{\partial \mathbf{h}}{\partial \mathbf{x}} \mathbf{Z}_x^{N_\nu} + \frac{\partial \mathbf{h}}{\partial \theta} \mathbf{V}. \quad (34)$$

Now the Direct method for the iterative multiscale gradient computation can be fully determined. It is depicted in Algorithm 2. Note that references to superscript -1 correspond to zero terms.

Algorithm 2: Post multiplication of \mathbf{G} by \mathbf{V} via the direct method.

Input : $\mathbf{R}, \mathbf{A}, \mathbf{P}, \mathbf{x}, \frac{\partial \mathbf{A}}{\partial \theta}, \frac{\partial \mathbf{h}}{\partial \mathbf{x}}, \frac{\partial \mathbf{h}}{\partial \theta}, \mathbf{V}, \epsilon_\sigma$
Output: The $\mathbf{G} \mathbf{V}$ product

- 1 **foreach** $j = 1, 2, \dots, n$ **do**
- 2 **foreach** $\nu = 0, 1, \dots, N_\nu$ **do**
- 3 Compute $\beta = \frac{\partial \mathbf{P}}{\partial \theta} \check{\mathbf{x}}^{\nu} \mathbf{V}_{\cdot j}$; // Algorithm 3, in [19]
 where $\mathbf{m} = \mathbf{V}_{\cdot j}$
- 4 Compute $\alpha = \mathbf{R} \mathbf{A} \beta + \mathbf{R} \left(\frac{\partial \mathbf{A}}{\partial \theta} \mathbf{x}^{\nu-1} - \frac{\partial \mathbf{q}}{\partial \theta} + \frac{\partial \mathbf{A}}{\partial \theta} \mathbf{P} \check{\mathbf{x}}^{\nu} \right) \mathbf{V}_{\cdot j}$
- 5 Solve $\check{\mathbf{Z}}_{\cdot j}^{\nu} = \check{\mathbf{A}}^{-1} \left(-\alpha - \mathbf{R} \mathbf{A} \mathbf{Z}_{x'}^{v-1} \right)$
- 6 Compute $\mathbf{Z}'_{\cdot j}^{\nu} = -\mathbf{P} \check{\mathbf{Z}}_{\cdot j}^{\nu} + \beta$
- 7 Compute $\frac{\partial \mathbf{g}'_{\sigma}{}^{\nu}}{\partial \theta} = \frac{\partial \mathbf{A}}{\partial \theta} \mathbf{x}'_{\sigma}{}^{\nu} + \frac{\partial \mathbf{A}}{\partial \theta} \mathbf{x}'^{\nu} - \frac{\partial \mathbf{q}}{\partial \theta} + \frac{\partial \mathbf{A}}{\partial \theta} \mathbf{x}^{\nu-1}$
- 8 Compute $\delta = \left(-\frac{\partial \mathbf{g}'_{\sigma}{}^{\nu}}{\partial \theta} \mathbf{V}_{\cdot j} - \mathbf{A} \mathbf{Z}'_{\cdot j}{}^{\nu} - \mathbf{A} \mathbf{Z}_{x'}^{v-1} \right)$
- 9 Iteratively solve $\mathbf{Z}_{\sigma \cdot j}^{\nu} = \mathbf{A}^{-1} \delta$ until $\frac{\| \delta - \mathbf{A} \mathbf{Z}_{\sigma \cdot j}^{\nu} \|}{\| \delta \|} < \epsilon_\sigma$
- 10 Compute $\mathbf{Z}_{x \cdot j}^{\nu} = \mathbf{Z}_{x \cdot j}^{v-1} + \mathbf{Z}'_{\cdot j}{}^{\nu} + \mathbf{Z}_{\sigma \cdot j}^{\nu}$
- 11 Compute $(\mathbf{G} \mathbf{V})_{\cdot j} = \frac{\partial \mathbf{h}}{\partial \mathbf{x}} \mathbf{Z}_{x \cdot j}^{N_\nu} + \frac{\partial \mathbf{h}}{\partial \theta} \mathbf{V}_{\cdot j}$

3.2. The adjoint method

Now, if \mathbf{V} is factored out in Eq. (18), it can be rewritten as

$$\mathbf{W} \mathbf{G} = \mathbf{Z} \frac{\partial \mathbf{g}}{\partial \theta} + \mathbf{W} \frac{\partial \mathbf{h}}{\partial \theta}, \quad (35)$$

where

$$\mathbf{Z} = -\mathbf{W} \frac{\partial \mathbf{h}}{\partial \mathbf{x}} \left(\frac{\partial \mathbf{g}}{\partial \mathbf{x}} \right)^{-1} \quad (36)$$

is solved from

$$\mathbf{Z} \left(\frac{\partial \mathbf{g}}{\partial \mathbf{x}} \right) = -\mathbf{W} \frac{\partial \mathbf{h}}{\partial \mathbf{x}}. \quad (37)$$

The linear system described in Eq. (37) can be re-written in a block-wise fashion for each iteration ν as

$$\begin{pmatrix} \mathbf{Z}^0 & \mathbf{Z}^1 & \dots & \mathbf{Z}^{N_\nu} \end{pmatrix} \times \begin{pmatrix} \frac{\partial \mathbf{g}^0}{\partial \mathbf{x}^0} & & & \\ \frac{\partial \mathbf{g}^1}{\partial \mathbf{x}^0} & \frac{\partial \mathbf{g}^1}{\partial \mathbf{x}^1} & & \\ & \ddots & \ddots & \\ & & \frac{\partial \mathbf{g}^{N_\nu}}{\partial \mathbf{x}^{N_\nu-1}} & \frac{\partial \mathbf{g}^{N_\nu}}{\partial \mathbf{x}^{N_\nu}} \end{pmatrix} = -\mathbf{W} \frac{\partial \mathbf{h}}{\partial \mathbf{x}} \quad (38)$$

The structure of Eq. (38) shows that it can be solved via back substitution, i.e. the solution of the gradient information is backward in the iterations.

Substituting Eqs. (27) and (28) in Eq. (38), follows that

$$\begin{pmatrix} \check{\mathbf{Z}}^0 \\ \mathbf{Z}'^0 \\ \mathbf{Z}_\sigma^0 \\ \mathbf{Z}_x^0 \\ \check{\mathbf{Z}}^1 \\ \mathbf{Z}'^1 \\ \mathbf{Z}_\sigma^1 \\ \mathbf{Z}_x^1 \\ \vdots \\ \check{\mathbf{Z}}^{N_\nu} \\ \mathbf{Z}'^{N_\nu} \\ \mathbf{Z}_\sigma^{N_\nu} \\ \mathbf{Z}_x^{N_\nu} \end{pmatrix}^T \begin{pmatrix} \frac{\partial \check{\mathbf{g}}^{\nu}}{\partial \mathbf{x}^{\nu}} & & & \\ \frac{\partial \mathbf{g}'^{\nu}}{\partial \mathbf{x}^{\nu}} & & & \\ \frac{\partial \mathbf{g}'_{\sigma}{}^{\nu}}{\partial \mathbf{x}^{\nu}} & & & \\ \frac{\partial \mathbf{g}'_x{}^{\nu}}{\partial \mathbf{x}^{\nu}} & & & \\ \frac{\partial \check{\mathbf{g}}^{\nu}}{\partial \mathbf{x}^{\nu-1}} & \frac{\partial \check{\mathbf{g}}^{\nu}}{\partial \mathbf{x}^{\nu}} & & \\ \frac{\partial \mathbf{g}'^{\nu}}{\partial \mathbf{x}^{\nu-1}} & \frac{\partial \mathbf{g}'^{\nu}}{\partial \mathbf{x}^{\nu}} & & \\ \frac{\partial \mathbf{g}'_{\sigma}{}^{\nu}}{\partial \mathbf{x}^{\nu-1}} & \frac{\partial \mathbf{g}'_{\sigma}{}^{\nu}}{\partial \mathbf{x}^{\nu}} & & \\ \frac{\partial \mathbf{g}'_x{}^{\nu}}{\partial \mathbf{x}^{\nu-1}} & \frac{\partial \mathbf{g}'_x{}^{\nu}}{\partial \mathbf{x}^{\nu}} & & \\ \frac{\partial \mathbf{g}'^{\nu}}{\partial \mathbf{x}^{\nu-1}} & \frac{\partial \mathbf{g}'^{\nu}}{\partial \mathbf{x}^{\nu}} & & \\ & \ddots & \ddots & \\ & & \frac{\partial \check{\mathbf{g}}^{\nu}}{\partial \mathbf{x}^{\nu-1}} & \frac{\partial \check{\mathbf{g}}^{\nu}}{\partial \mathbf{x}^{\nu}} \\ & & \frac{\partial \mathbf{g}'^{\nu}}{\partial \mathbf{x}^{\nu-1}} & \frac{\partial \mathbf{g}'^{\nu}}{\partial \mathbf{x}^{\nu}} \\ & & \frac{\partial \mathbf{g}'_{\sigma}{}^{\nu}}{\partial \mathbf{x}^{\nu-1}} & \frac{\partial \mathbf{g}'_{\sigma}{}^{\nu}}{\partial \mathbf{x}^{\nu}} \\ & & \frac{\partial \mathbf{g}'_x{}^{\nu}}{\partial \mathbf{x}^{\nu-1}} & \frac{\partial \mathbf{g}'_x{}^{\nu}}{\partial \mathbf{x}^{\nu}} \end{pmatrix} = -\mathbf{W} \frac{\partial \mathbf{h}}{\partial \mathbf{x}} \quad (39)$$

From Eq. (39) and recalling that \mathbf{h} only depends on $\mathbf{x} = \mathbf{x}^{N_\nu}$, the (transposed) linear system that must be solved to compute \mathbf{Z}^{ν} is

given by

$$\begin{pmatrix} \check{\mathbf{A}}^T & -\mathbf{P}^T & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{A}^T & -\mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{A}^T & -\mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{A}^T \mathbf{R}^T & \mathbf{0} & \mathbf{A}^T & -\mathbf{I} \end{pmatrix} \times \begin{pmatrix} (\check{\mathbf{Z}}^\nu)^T \\ (\mathbf{Z}'^\nu)^T \\ (\mathbf{Z}_\sigma^\nu)^T \\ (\mathbf{Z}_x^\nu)^T \\ (\check{\mathbf{Z}}^{\nu+1})^T \\ (\mathbf{Z}'^{\nu+1})^T \\ (\mathbf{Z}_\sigma^{\nu+1})^T \\ (\mathbf{Z}_x^{\nu+1})^T \end{pmatrix} = \mathbf{0}, \quad (40)$$

for $\nu \neq N_\nu$, while \mathbf{Z}^{N_ν} is calculated from

$$\begin{pmatrix} \check{\mathbf{A}}^T & -\mathbf{P}^T & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{A}^T & -\mathbf{I} \\ \mathbf{0} & \mathbf{0} & \mathbf{A}^T & -\mathbf{I} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} \end{pmatrix} \begin{pmatrix} (\check{\mathbf{Z}}^{N_\nu})^T \\ (\mathbf{Z}'^{N_\nu})^T \\ (\mathbf{Z}_\sigma^{N_\nu})^T \\ (\mathbf{Z}_x^{N_\nu})^T \end{pmatrix} = \begin{pmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ -(\frac{\partial \mathbf{h}}{\partial \mathbf{x}})^T \mathbf{W}^T \end{pmatrix}. \quad (41)$$

The equation that computes the "adjoint state" associated with \mathbf{g}_x^ν reads

$$(\mathbf{Z}_x^\nu)^T = (\mathbf{Z}_x^{\nu-1})^T - \mathbf{A}^T \mathbf{R}^T (\check{\mathbf{Z}}^{\nu-1})^T - \mathbf{A}^T (\mathbf{Z}_\sigma^{\nu-1})^T, \nu \neq N_\nu, \quad (42)$$

and

$$(\mathbf{Z}_x^{N_\nu})^T = -(\frac{\partial \mathbf{h}}{\partial \mathbf{x}})^T \mathbf{W}^T. \quad (43)$$

The "adjoint state" for \mathbf{g}_σ^ν is calculated from

$$(\mathbf{Z}_\sigma^\nu)^T = \mathbf{A}^{-T} (\mathbf{Z}_x^\nu)^T. \quad (44)$$

As discussed for the direct method, in the proposed derivative calculation method, Eq. (44) is solved as a smoothing step only, using any iterative solver up to a prescribed tolerance or number of iterations.

For \mathbf{g}^ν , the adjoint state is given by

$$(\mathbf{Z}'^\nu)^T = (\mathbf{Z}_x^\nu)^T - \mathbf{A}^T (\mathbf{Z}_\sigma^\nu)^T, \quad (45)$$

and finally for $\check{\mathbf{g}}^\nu$

$$\check{\mathbf{Z}}^\nu = (\check{\mathbf{A}}^{-T}) (\mathbf{P}^T \mathbf{Z}'^\nu). \quad (46)$$

Eq. (35) can be written as a sum where each term corresponds to the contribution of one iteration

$$\mathbf{W}\mathbf{G} = \sum_{\nu=0}^{N_\nu} \left(\check{\mathbf{Z}}^\nu \frac{\partial \mathbf{g}^\nu}{\partial \theta} + \mathbf{Z}'^\nu \frac{\partial \mathbf{g}^\nu}{\partial \theta} + \mathbf{Z}_\sigma^\nu \frac{\partial \mathbf{g}_\sigma^\nu}{\partial \theta} \right) + \mathbf{W} \frac{\partial \mathbf{h}}{\partial \theta}. \quad (47)$$

The algorithm can now be fully defined and is presented in 3. Note the use of the notation $(\mathbf{W}\mathbf{G})^\nu$ to denote the partial sums in Eq. (47) and that references to superscript $N_\nu + 1$ correspond to zero terms.

Algorithm 3: Pre multiplication of \mathbf{G} by \mathbf{W} via the adjoint method.

```

Input :  $\mathbf{R}, \mathbf{A}, \mathbf{P}, \mathbf{x}, \frac{\partial \mathbf{A}}{\partial \theta}, \frac{\partial \mathbf{h}}{\partial \mathbf{x}}, \frac{\partial \mathbf{h}}{\partial \theta}, \mathbf{W}, \epsilon_\sigma$ 
Output: The  $\mathbf{W}\mathbf{G}$  product
1 foreach  $i = 1, 2, \dots, m$  do
2   foreach  $\nu = N_\nu, \dots, 1, 0$  do
3     Set  $(\mathbf{Z}_x^\nu)^T =$ 
        $\begin{cases} -(\frac{\partial \mathbf{h}}{\partial \mathbf{x}})^T \mathbf{W}^T, & \text{if } \nu = N_\nu \\ (\mathbf{Z}_x^{\nu+1})^T - \mathbf{A}^T (\mathbf{Z}_\sigma^{\nu+1})^T - \mathbf{A}^T \mathbf{R}^T (\check{\mathbf{Z}}^{\nu+1})^T, & \text{otherwise} \end{cases}$ 
4     Iteratively solve  $(\mathbf{Z}_\sigma^\nu)^T = \mathbf{A}^{-T} (\mathbf{Z}_x^\nu)^T$  until
        $\frac{\| \mathbf{Z}_x^\nu - \mathbf{A} \mathbf{Z}_\sigma^{\nu,j} \|}{\| \mathbf{Z}_x^\nu \|} < \epsilon_\sigma$ 
5     Compute  $(\mathbf{Z}'^\nu)^T = (\mathbf{Z}_x^\nu)^T - \mathbf{A}^T (\mathbf{Z}_\sigma^\nu)^T$ 
6     Solve  $\check{\mathbf{Z}}_{i..}^\nu = (\check{\mathbf{A}}^{-T}) (\mathbf{P}^T \mathbf{Z}'_{i..}^{\nu+1})$ 
7     Compute  $\alpha^T = \check{\mathbf{Z}}_{i..}^\nu (\mathbf{R}\mathbf{A}) - \mathbf{Z}'_{i..}^\nu$ 
8     Compute  $\beta = \alpha \frac{\partial \mathbf{P}}{\partial \theta} \check{\mathbf{x}}^\nu$ ; // Algorithm 4 in [19] where
        $\mathbf{m}^T = \alpha^T$ 
9     Compute  $\gamma = \mathbf{R} \frac{\partial \mathbf{A}}{\partial \theta} \mathbf{P} \check{\mathbf{x}}^\nu - \mathbf{R} \frac{\partial \mathbf{q}}{\partial \theta} + \mathbf{R} \frac{\partial \mathbf{A}}{\partial \theta} \mathbf{x}^{\nu-1}$ 
10    Compute  $\frac{\partial \mathbf{g}_\sigma^\nu}{\partial \theta} = \frac{\partial \mathbf{A}}{\partial \theta} \mathbf{x}_\sigma^\nu + \frac{\partial \mathbf{A}}{\partial \theta} \mathbf{x}^{\nu-1} - \frac{\partial \mathbf{q}}{\partial \theta} + \frac{\partial \mathbf{A}}{\partial \theta} \mathbf{x}^{\nu-1}$ 
11    Update  $(\mathbf{W}\mathbf{G})_{i..}^\nu = (\mathbf{W}\mathbf{G})_{i..}^{\nu+1} + \beta + \check{\mathbf{Z}}_{i..}^\nu \gamma + \mathbf{Z}_{\sigma i..}^\nu \frac{\partial \mathbf{g}_\sigma^\nu}{\partial \theta}$ 
12  Update  $\mathbf{W}\mathbf{G} = (\mathbf{W}\mathbf{G})^0 + \mathbf{W} \frac{\partial \mathbf{h}}{\partial \theta}$ 

```

3.2.1. Remarks about the framework

An alternative formulation for the i-MSFV formulation has been previously proposed and investigated in Frank (2017). In that work, both the state and the model equation vectors explicitly account for the smoothing stage. The formulation here proposed is based on two observations. Firstly, the implementation of the aforementioned variant, although offers more control over the gradient quality, relies on the ability of computing partial derivative matrices of smoothing step with respect to the parameters. More specifically, it implies on the knowledge of how the precondition \mathbf{M} of the system matrix \mathbf{A} is built. For simpler iterative strategies, e.g. Jacobi, which construction of \mathbf{M} can be simple, the computation of $\frac{\partial \mathbf{M}}{\partial \theta}$ is relatively simple. However, simpler iterative methods are usually less efficient. Also, the requirement of knowing the construction of \mathbf{M} hampers the utilization of 'black-box' type of preconditioners. Secondly, it has been shown in Tene et al. (2015) that only a limited number of smoothing steps are necessary to result in an efficient i-MSFV solution strategy. Hence, not much extra control would be achieved.

Note that the linear solvers employed in lines 9 and 4 of, respectively, Algorithms 2 and 3, are the same solvers employed in the solution of the forward simulation, using the same prescribed (loose) tolerance. Hence, the algorithms share the same computational advantages by solving for the approximated derivative information arising from the smoothing step.

The backward algorithm requires storing all intermediate states generated during the iterations in the forward run. It also requires solving many systems of equations for each backward time-step. If the iteration process in the forward run goes until machine precision is reached, then essentially the fully coupled system has been solved to fine-scale precision and it might be more beneficial to neglect the iteration history and aim to solve the fine-scale system of adjoint equations in a more efficient way, given that the derivative computation problem is linear. However, we highlight that, as it has been shown in the literature (Fonseca et al., 2015; de Moraes et al., 2017), approximate gradients computed from

approximate solutions are already sufficient to efficiently/successfully lead the optimization process to the optimal solution. How accurate this gradient will need to be is dependent on different aspects, e.g. the optimization algorithm, how early/late the optimization process is in terms of finding the maximum/minimum of a given objective function, among others. The algorithm here proposed has the advantage of controlling the gradient quality, as will be shown in the numerical experiments presented in the next section.

4. Numerical experiments

Given the fundamental nature of the developments presented here, we focus the numerical experiments on the validation of the computation and assessment of the gradient quality provided by the iterative multiscale method.

Our experiments will be based on the evaluation of the gradient of a misfit objective function with no regularization term (Oliver et al., 2008)

$$O(\theta) = \frac{1}{2} (\mathbf{h}(\mathbf{x}, \theta) - \mathbf{d}_{obs})^T \mathbf{C}_D^{-1} (\mathbf{h}(\mathbf{x}, \theta) - \mathbf{d}_{obs}), \quad (48)$$

with a gradient

$$\nabla_{\theta} O = \mathbf{G}^T \mathbf{C}_D^{-1} (\mathbf{h}(\mathbf{x}, \theta) - \mathbf{d}_{obs}), \quad (49)$$

where $\mathbf{C}_D \in \mathbb{R}^{N_Y \times N_Y}$ is the so-called data covariance matrix. Here, \mathbf{C}_D will be considered a diagonal matrix given by Oliver et al. (2008)

$$\mathbf{C}_D = \sigma^2 \mathbf{I}, \quad (50)$$

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

In all experiments, the fitting parameters are cell-centered permeabilities. The observed quantity, \mathbf{d}_{obs} , is the fine scale pressure at the location of (non-flowing) observation wells, therefore

$$\frac{\partial \mathbf{h}}{\partial \mathbf{x}'} = \mathbf{I}, \quad (51)$$

and

$$\frac{\partial \mathbf{h}}{\partial \mathbf{x}} = \mathbf{0}. \quad (52)$$

In all experiments, the standard deviation of the pressure measurement error is $\sigma \approx 0.03$ (note that the measurement error is also non-dimensional). This represents a (very accurate) measurement error in the range of those usually employed in synthetic study cases (see e.g. Oliver et al., 2008).

Note that the wells are controlled by bottom-hole pressure, expressed in terms of a non-dimensional pressure, i.e.,

$$p_D = \frac{p - p_{prod}}{p_{inj} - p_{prod}}, \quad (53)$$

where p_{inj} and p_{prod} are the injection and production pressures, respectively. In all the experiments, $p_{inj} = 1.0$ and $p_{prod} = 0.0$, the grid-block dimensions are $\Delta x = \Delta y = \Delta z = 1$ m and the fluid viscosity is 1.0×10^{-3} Pa s. In addition, in all the following test cases, well basis functions are included.

The metric utilized to assess the i-MSFV gradient quality is the angle between fine-scale and i-MSFV normalized gradients, i.e.,

$$\alpha = \cos^{-1} (\nabla_{\theta}^T \hat{O}_{FS} \nabla_{\theta} \hat{O}_{MS}). \quad (54)$$

Here,

$$\nabla_{\theta} \hat{O}_{FS} = \frac{\nabla_{\theta} O_{FS}}{\|\nabla_{\theta} O_{FS}\|_2} \quad (55)$$

and

$$\nabla_{\theta} \hat{O}_{MS} = \frac{\nabla_{\theta} O_{MS}}{\|\nabla_{\theta} O_{MS}\|_2}. \quad (56)$$

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).

And to prove our hypothesis, we particularly interested in observing the behaviour of the metric as more accurate i-MSFV solutions are computed.

In our i-MSFV implementation, the iterative process is controlled by the outer loop residual ϵ and the pre-conditioner smoother error tolerance ϵ_σ . The Krylov subspace biconjugate gradient stabilized method (BiCGSTAB, Saad, 2003) is employed in the smoothing stage.

4.1. Validation experiments

Before focusing in the validation, in this section we will validate the iMS-gradient method against the numerical differentiation method with a higher order, two-sided Taylor approximation

$$\nabla_{\theta} O_i = \frac{1}{2\delta\theta_i} \left(O(\theta_1, \dots, \theta_{i-1}, \theta_i + \delta\theta_i, \theta_{i+1}, \dots, \theta_{N_\theta}) - O(\theta_1, \dots, \theta_{i-1}, \theta_i - \delta\theta_i, \theta_{i+1}, \dots, \theta_{N_\theta}) \right) \quad (57)$$

where we consider δ to be a multiplicative parameter perturbation. We define the relative error as

$$\epsilon = \frac{\|\nabla_{\theta} O_{NUM} - \nabla_{\theta} O_{iMS}\|_2}{\|\nabla_{\theta} O_{iMS}\|_2} \quad (58)$$

Here $\nabla_{\theta} O_{NUM}$ is obtained by performing the proper amount of iterative multiscale reservoir simulations required to evaluate Eq. [57]. $\nabla_{\theta} O_{iMS}$ is obtained by employing the iterative Direct or Adjoint gradient method.

To evaluate the correctness of the proposed iterative gradient computation methods, as well as their implementation, we investigate the linear decrease of the relative error ϵ by decreasing the parameter perturbation δ from 10^{-1} to 10^{-4} . This investigation is carried out in two distinct examples. Both have a fine grid of 21×21 grid blocks. We employ a 7×7 coarsening ratio, giving a 3×3 coarse grid. The reference twin-experiment is generated with permeability realization number 992. Fig. 1 illustrates the fine-, coarse- and dual-grid cells along with the reference permeability.

Next we determine the well positions. We use the so-called quarter well spot. Here, two observation wells are placed near operating wells. The full specifications can be found in Table 1.

The results of this experiment are found in Fig. 2. Here, we use a single outer iteration. We use a very tight smoothing tolerance of $\epsilon_\sigma = 5 \times 10^{-8}$ to ensure that the numerical gradient method produces accurate enough gradients. First of all we can see that the fine-scale numerical gradient method and the iterative MS-gradient methods are of the same order of accuracy with respect to the perturbation δ , for all different cases considered. In all experiments, we can see the linear decreasing behaviour of the relative error values ϵ as the perturbation δ decreases. Also, note that the Adjoint and Direct methods provide the analytical gradient with the same level of accuracy. In the first experiment, a homogeneous test case with the permeability value of $1.0e^{-13}$ is used. The second experiment indicates the correctness of the method when it is applied to heterogeneous porous media problems.

4.2. Investigation of i-MSFV convergence behaviour and gradient quality

In this investigation, the error metric given by Eq. (54) is evaluated for a whole ensemble of heterogeneous permeability fields. The

Table 1
Validation experiments well setup.

Well	Fine scale position (I, J)	Well Type
INJE	(1, 1)	Injection
PROD	(21, 21)	Production
OBS1	(3, 3)	Observation
OBS2	(19, 19)	Observation

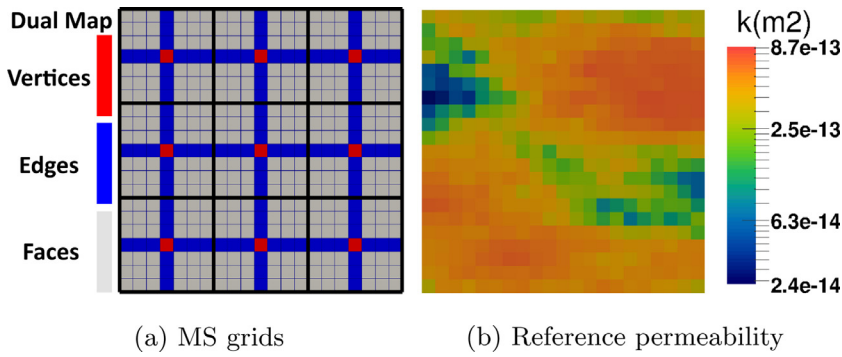


Fig. 1. Validation experiments setup. (a) Primal (bold black line), dual (identified by the blue cells) and fine grids (thin black lines). (b) Reference permeability field used in the twin experiment. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

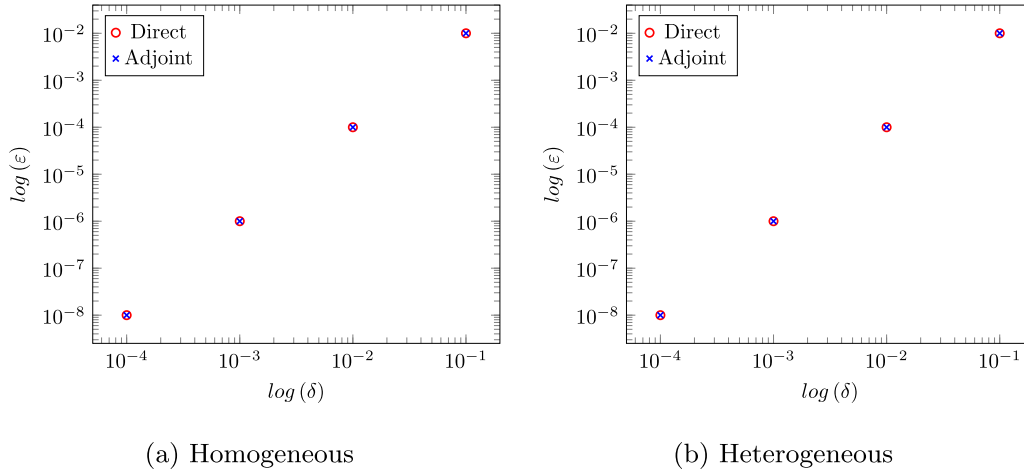


Fig. 2. Validation of the i-MSFV gradient computation method compared with a numerical gradient. (a) represents the homogeneous test case, while (b) represents the heterogeneous test case.

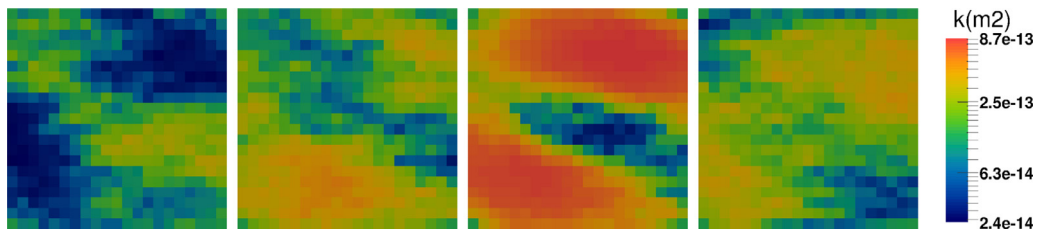


Fig. 3. Four different permeability realizations from the ensemble of 1000 members used in the 2D numerical experiments.

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

The fine-scale and coarse grids contain 21×21 and 7×7 cells, respectively. An inverted five-spot well pattern is employed, while four observation wells are placed close to the production wells. The well configuration is depicted in Table 2.

The synthetic observed pressures at the observation well locations are created via the classical twin-experiment strategy, where the permeability field is extracted from the 1000 geological realizations (the first one in Fig. 3).

The robustness of the method is illustrated in Fig. 4. It is possible to observe that the angle between fine-scale gradient and the i-MSFV gradient is smaller the tighter we make the outer-loop residual tolerance. Moreover, the variance also goes to almost zero as we set the residual tolerance to $1e^{-4}$. For this set of relatively small permeability contrast, perfect alignment with fine-scale gradient is reached if the tolerance is

Table 2

Well configuration for the homogeneous, two-dimensional case.

Well	Fine scale position (I, J)	Well type
INJE	(11, 11)	Injection
PROD1	(1, 1)	Production
PROD2	(21, 1)	Production
PROD3	(1, 21)	Production
PROD4	(21, 21)	Production
OBSWELL1	(3, 3)	Observation
OBSWELL2	(19, 3)	Observation
OBSWELL3	(3, 19)	Observation
OBSWELL4	(19, 19)	Observation

set to $1e^{-5}$. We highlight that the permeability contrast of this ensemble is not high. Next, we assess the robustness for the method for geological settings with higher permeability contrasts.

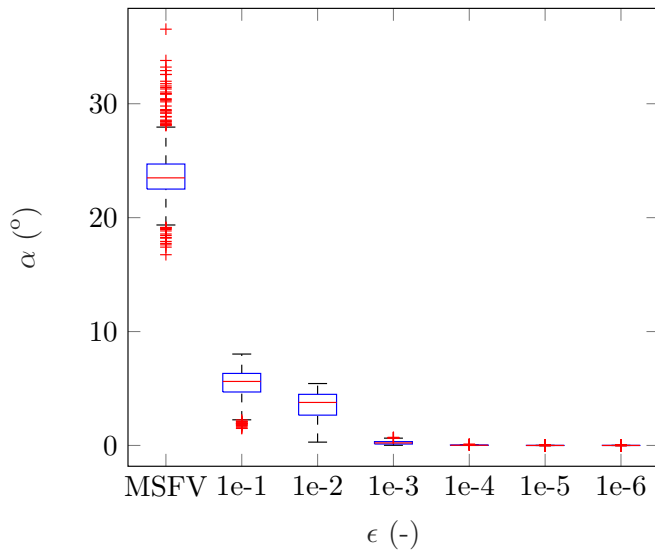


Fig. 4. Box-plot illustrating the angle α between fine-scale gradient and i-MSFV gradient computed for the 1000 member ensemble as a function of the outer-loop tolerance error ϵ . “No iteration” is equivalent to the MSFV gradient computation presented in de Moraes et al. (2017).

4.3. Robustness with respect to heterogeneity contrast and distribution

In order to further explore the point about the robustness of the method with respect to heterogeneity contrast and distribution, four sets of 20 equiprobable realizations of log-normally distributed permeability fields with a spherical variogram and dimensionless correlation lengths of $\Psi_1 = 0.5$ and $\Psi_2 = 0.02$ are generated using sequential Gaussian simulations (Remy et al., 2009). For each set, the variance and the mean of $\ln(k)$ are 2.0 and 3.0, respectively, where k is the grid block permeability. As depicted in Fig. 5, 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 (Fig. 5d). Compared with the previous set, the permeability contrast is much higher in this case.

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 3.

The observed data is generated from a twin-experiment associated with (the first) permeability realization of each set.

In this experiment, $\epsilon = 1.0e^{-6}$ and $\epsilon_\sigma = 1.0e^{-1}$. The box-plot shown in Fig. 6 summarizes the required total number of smoothing steps, for all outer i-MSFV steps.

The grid orientation effect (Aziz et al., 1993) impact on the performance of the i-MSFV method is clear in this example. The more the heterogeneity orientation is aligned with the flow orientation, the less

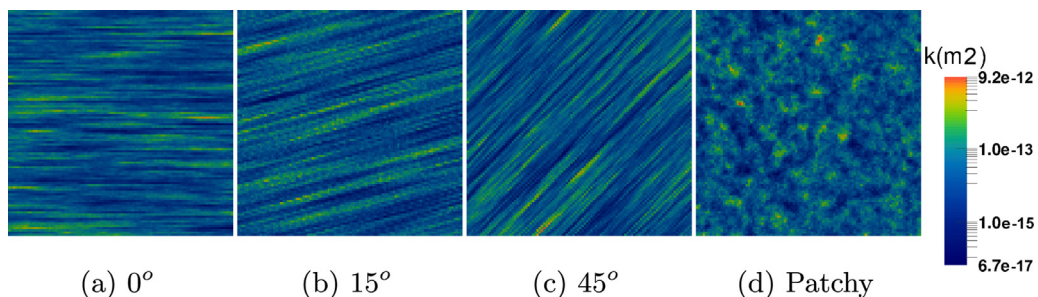


Fig. 5. Permeability distribution of four different realizations taken from the sets of 20 geostatistically equiprobable permeability fields with different correlation angles (a-c). Also, a patchy field (d) with a small correlation length is considered.

Table 3
Well configuration for Case 2.

Well	Fine scale position (I, J)	Well type
INJE	(1, 1)	Injection
PROD	(100, 100)	Production
OBSWELL1	(3, 3)	Observation
OBSWELL2	(98, 98)	Observation

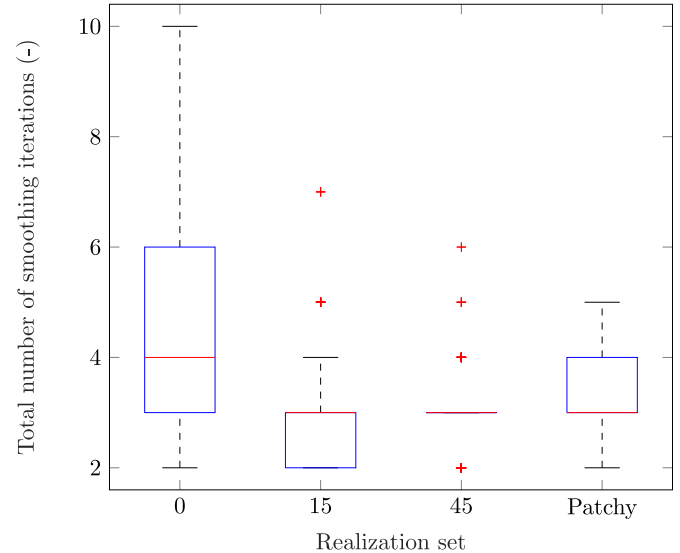


Fig. 6. Box-plot representing the total number of smoothing iterations required to compute the misfit OF gradient for the different permeability ensembles with correlation angles 0° , 15° and 45° and with a small correlation length (patchy).

is the number of required iterations. Also, in relation to Fig. 7, the more challenging the forward problem, the more challenging it is to compute i-MSFV gradient in accordance to the fine-scale gradient. Nevertheless, in all cases, almost all i-MSFV realization gradients are perfectly aligned with fine-scale gradient, demonstrating the robustness of the method.

4.4. SPE-10 comparative test case

Now, we investigate the performance of our method in the SPE-10 comparative case (Christie et al., 2001), regarded as challenging model for upscaling (Durufofsky, 2005) and multiscale simulation (Hajibeygi, 2011) techniques. Here, we consider the 2-D flow simulation of both top and bottom layer of the original 3D model, which permeability fields are illustrated in Fig. 8.

The fine grid dimensions is 60×220 , while we employ a 12×20 coarse grid in the MS simulation. A quarter five-spot well setting is

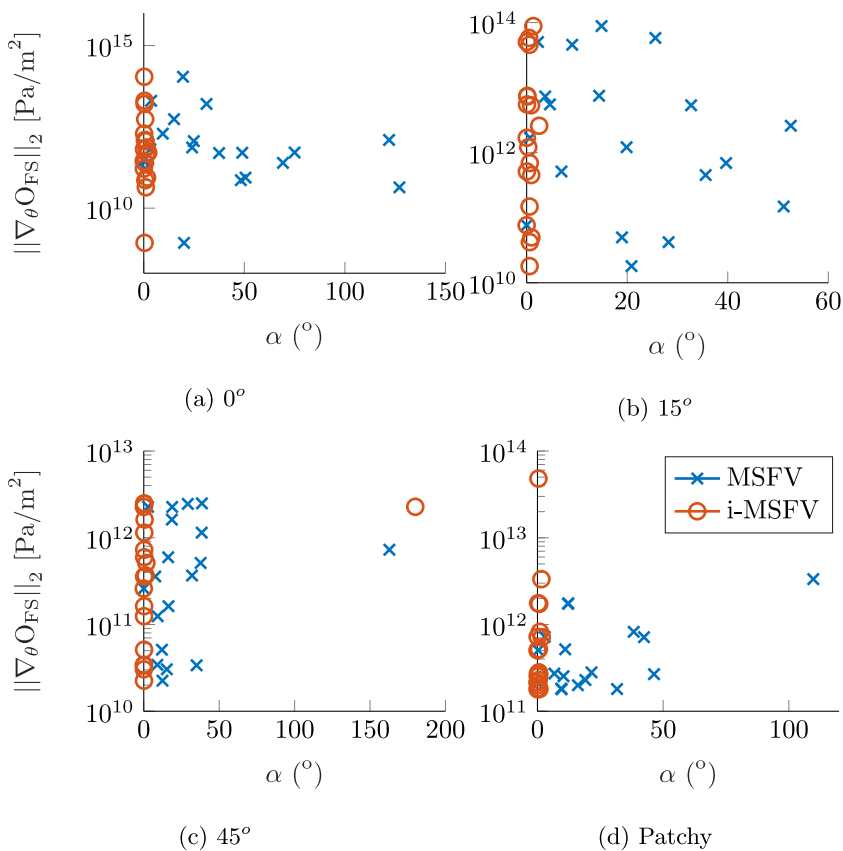


Fig. 7. Illustration of gradient quality improvement when an i-MSFV gradient computation strategy is employed in comparison to a MSFV computation strategy. The x-axis represent the angle α between fine-scale and the MSFV gradient (illustrated by blue crosses) and the i-MSFV gradient with error tolerance $\epsilon = 10^{-6}$ (illustrated by orange circles). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

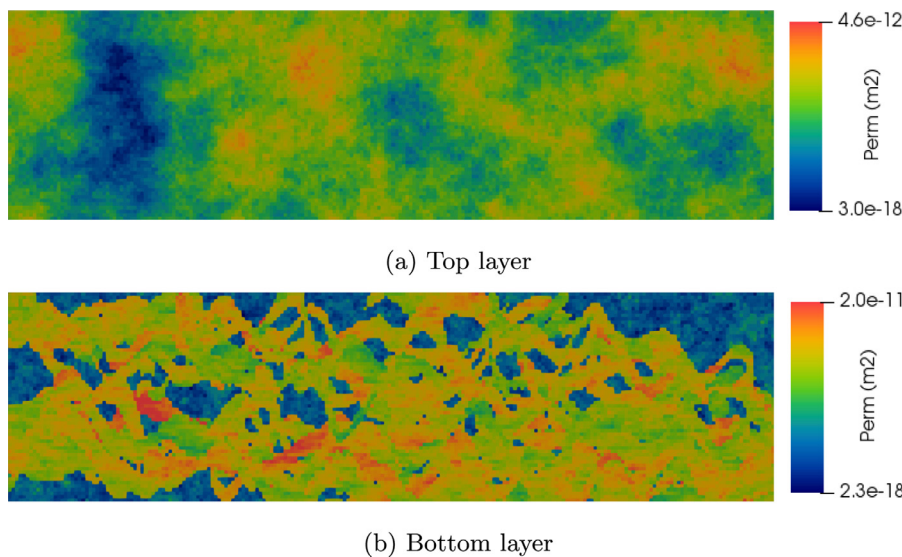


Fig. 8. SPE-10 comparative test case: top (a) and bottom (b) layer permeability fields.

considered. Four observation wells are deliberately positioned in low permeability regions, surrounded by high permeability regions. The well positions are described in Table 4.

We emphasize that it has been reported (see e.g. Hajibeygi, 2011) that the MSFV method provides non-monotone pressure solutions when solving the pressure for the bottom layer (Fig. 8b). Strategies to improve the MSFV solution, among them the i-MSFV (Hajibeygi and Jenny, 2011) here considered, are necessary to address the issue.

In this twin experiment, the reference permeability field used to compute the observations is considered homogeneous with a value of $1e-13$.

Table 4

Well configuration for the SPE-10 comparative test case.

Well	Fine scale position (I, J)	Well type
INJE	(1, 220)	Injection
PROD	(60, 1)	Production
OBSWELL1	(33, 5)	Observation
OBSWELL2	(28, 50)	Observation
OBSWELL3	(28, 83)	Observation
OBSWELL4	(43, 204)	Observation

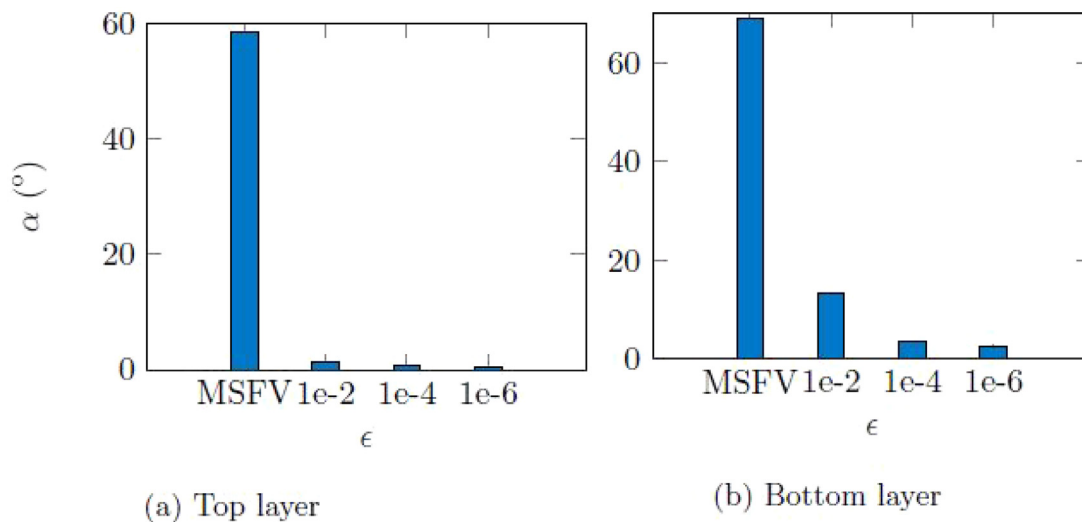


Fig. 9. i-MSFV gradient quality (angle α between fine-scale and i-MSFV gradients) as a function of residual error ϵ for the SPE-10 top layer (a) and bottom layer (b).

In this experiment, we fix $\epsilon_\sigma = 1.0e^{-2}$ and vary $\epsilon = 1.0e^{-2}$, $1.0e^{-4}$, $1.0e^{-6}$ to evaluate whether the same convergence behaviour of the i-MSFV gradient toward fine-scale gradient direction is observed.

Once again we observe that the method provides accurate gradients, even considering this challenging geological setting, for both the top and bottom layers of the model (see Fig. 9).

4.5. Discussion

Based on the results acquired from the different numerical cases of increasing complexity we demonstrated that our newly introduced method can provide accurate gradients, up to fine-scale accuracy, if small enough residual tolerances are employed in the i-MSFV forward simulation. Also, it is shown that only a few i-MSFV/smoothing iterations are necessary to have reasonably accurate gradients. However, we highlight that, from an optimization point of view, gradients that are fully in accordance with fine-scale gradient are not necessary. As long as the gradient roughly points to the correct up/downward direction, the optimization process is able to progress towards the maximum/minimum. This is particularly true in the early iterations, where there is a more steep region of the objective function. On the other hand, as the optimization process approaches the optimum, more accurate gradients are required for a precise stopping criteria evaluation. The convergent behaviour of the i-MSFV gradient, which is demonstrated to be directly related to the outer-residual tolerance, which is an estimate defined *a-priori*, should allow for an error control of the gradient computation, which ultimately would allow control of the optimization process performance.

The gradient computation methods here, by design, directly inherits the characteristics of the i-MSFV introduced in Hajibeygi et al. (2008). The solution of the backward (transposed) system of equations mirrors the solution strategy of the linear system of equations that arises from the forward problem. Even the system matrices from the forward and the backward system of equations share the same properties (e.g. conditioning number). Therefore, in principle, all studies in the literature related to the robustness and efficiency of the i-MSFV method with respect to grid size, level of heterogeneity, among others, should also be valid in the context of the gradient computation methods here proposed. Nevertheless, further numerical studies are important to investigate the robustness of the method itself with respect to such aspects.

5. Summary and research perspectives

We introduce iterative multiscale gradient computation algorithms based on the iterative Multiscale Finite Volume (i-MSFV) simulation method. By firstly re-casting the i-MSFV in an algebraic framework, we derive a flexible, multi-purpose derivative computation framework which accounts for the Direct and Adjoint methods. Their computational efficiency is discussed and it is shown that they share advantages equivalent to the i-MSFV method. The new methods are validated via numerical experiments against numerical differentiation are shown to address the challenges encountered by the MSFV gradient computation, specifically associated with high heterogeneity contrast. Fine-scale-accurate gradients are obtained for challenging geological models with high permeability contrasts (e.g. the SPE-10 comparative test case) if the i-MSFV model converges to an error tolerance of $1.0e^{-6}$. However, we highlight that such accuracy is not necessary and only a few i-MSFV/smoothing iterations are required for reasonably accurate gradients. Further control of the gradient quality via an *a-priori* error estimate along the optimization process should allow for a computationally efficient optimization method. More specifically, a less accurate gradient estimate usually suffices in the early optimization iterations, when the objective function convergence is more steep, while more accurate gradients are required in the flatter regions for a more precise stopping criterion. Hence, the determination of an *a-priori* error estimate should be a key development.

Acknowledgements

This work is part of Rafael Moraes' PhD project, sponsored by Petrobras' through its employee development program. The discussions held with the DARSim (Delft Advanced Reservoir Simulation) members shall be acknowledged.

References

- Anterion, F., Eymard, R., Karcher, B., 1989. Use of parameter gradients for reservoir history matching. SPE Symposium on Reservoir Simulation. Society of Petroleum Engineers. <https://doi.org/10.2118/18433-MS>.
- Aziz, K., et al., 1993. Reservoir simulation grids: opportunities and problems. J. Petrol. Tech. 45 (07), 658–663.
- Cardoso, M., Durlofsky, L., Sarma, P., 2009. Development and application of reduced-order modeling procedures for subsurface flow simulation. Int. J. Numer. Method. Eng. 77 (9), 1322–1350.
- Chavent, G., Dupuy, M., Lemmonier, P., 1975. History matching by use of optimal theory. Soc. Petrol. Eng. J. 15 (01), 74–86.

- Christie, M., Blunt, M., et al., 2001. Tenth spe comparative solution project: A comparison of upscaling techniques. SPE Reservoir Simulation Symposium. Society of Petroleum Engineers.
- van Doren, J.F., Markovinović, R., Jansen, J.-D., 2006. Reduced-order optimal control of water flooding using proper orthogonal decomposition. *Comp. Geosci.* 10 (1), 137–158. <https://doi.org/10.1007/s10596-005-9014-2>.
- Durlafsky, L.J., 2005. Upscaling and gridding of fine scale geological models for flow simulation. 8th International Forum on Reservoir Simulation Iles Borromees, Stresa, Italy, 2024.
- Fonseca, R.M., Kahrobaei, S.S., Van Gastel, L.J.T., Leeuwenburgh, O., Jansen, J.D., 2015. Quantification of the impact of ensemble size on the quality of an ensemble gradient using principles of hypothesis testing. SPE Reservoir Simulation Symposium. Society of Petroleum Engineers. <https://doi.org/10.2118/173236-MS>.
- Frank, P., 2017. Iterative multiscale adjoint gradient computation for incompressible flow optimization in porous media.
- Fu, J., Caers, J., Tchelepi, H.A., 2011. A multiscale method for subsurface inverse modeling: single-phase transient flow. *Adv. Water Resour.* 34 (8), 967–979.
- Fu, J., Tchelepi, H.A., Caers, J., 2010. A multiscale adjoint method to compute sensitivity coefficients for flow in heterogeneous porous media. *Adv. Water Resour.* 33 (6), 698–709.
- Hajibeygi, H., 2011. Iterative multiscale finite volume method for multiphase flow in porous media with complex physics Ph.D. thesis. ETH Zurich.
- Hajibeygi, H., Bonfigli, G., Hesse, M.A., Jenny, P., 2008. Iterative multiscale finite-volume method. *J. Comput. Phys.* 227 (19), 8604–8621.
- Hajibeygi, H., Jenny, P., 2011. Adaptive iterative multiscale finite volume method. *J. Comput. Phys.* 230 (3), 628–643.
- Hou, T.Y., Wu, X.-H., 1997. A multiscale finite element method for elliptic problems in composite materials and porous media. *J. Comput. Phys.* 134 (1), 169–189.
- Jansen, J.D., 2011. Adjoint-based optimization of multi-phase flow through porous media—a review. *Comp. Fluids* 46 (1), 40–51.
- Jansen, J.D., 2013. 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. [uuiid:6000459e-a0cb-40d1-843b-81650053e093](https://doi.org/10.1007/s10596-013-9093-3).
- Jansen, J. D., 2016. Gradient-based optimization of flow through porous media, v.3.
- Jenny, P., Lee, S.H., Tchelepi, H.A., 2003. Multi-scale finite-volume method for elliptic problems in subsurface flow simulation. *J. Comput. Phys.* 187 (1), 47–67.
- Kraaijevanger, J.F.B.M., Egberts, P.J.P., Valstar, J.R., Buurman, H.W., 2007. Optimal waterflood design using the adjoint method. SPE Reservoir Simulation Symposium. Society of Petroleum Engineers. <https://doi.org/10.2118/105764-MS>.
- Krogstad, S., Hauge, V.L., Gulbransen, A., 2011. Adjoint multiscale mixed finite elements. *SPE J.* 16 (01), 162–171.
- Li, R., Reynolds, A., Oliver, D., et al., 2003. History matching of three-phase flow production data. *SPEJ* 8 (04), 328–340.
- Lie, K.-A., Møyner, O., Natvig, J.R., Kozlova, A., Bratvedt, K., Watanabe, S., Li, Z., 2017. Successful application of multiscale methods in a real reservoir simulator environment. *Comp. Geosci.* 21 (5), 981–998. <https://doi.org/10.1007/s10596-017-9627-2>.
- Moraes, R., Rodrigues, J., Hajibeygi, H., Jansen, J., et al., 2017. Multiscale gradient computation for multiphase flow in porous media. In: SPE Reservoir Simulation Conference. Society of Petroleum Engineers.
- de Moraes, R. J., Rodrigues, J. R., Hajibeygi, H., Jansen, J. D., Gradient computation for sequentially coupled subsurface optimization models. In Preparation.
- de Moraes, R.J., Rodrigues, J.R., Hajibeygi, H., Jansen, J.D., 2017. Multiscale gradient computation for flow in heterogeneous porous media. *J. Comput. Phys.* 336, 644–663.
- Oliver, D.S., Reynolds, A.C., Liu, N., 2008. Inverse theory for petroleum reservoir characterization and history matching. Cambridge University Press.
- Remy, N., Boucher, A., Wu, J., 2009. Applied Geostatistics with SGeMS: A User's Guide. Cambridge University Press.
- Rodrigues, J.R.P., 2006. Calculating derivatives for automatic history matching. *Comput. Geosci.* 10 (1), 119–136.
- Saad, Y., 2003. Iterative methods for sparse linear systems. SIAM.
- Tene, M., Wang, Y., Hajibeygi, H., 2015. Adaptive algebraic multiscale solver for compressible flow in heterogeneous porous media. *J. Comput. Phys.* 300, 679–694.
- Wang, Y., Hajibeygi, H., Tchelepi, H.A., 2014. Algebraic multiscale solver for flow in heterogeneous porous media. *J. Comput. Phys.* 259, 284–303.
- Zhou, H., Tchelepi, H.A., 2008. Operator-based multiscale method for compressible flow. *SPE J.* 13 (02), 523–539.