Level-set techniques for facies identification in reservoir modeling

This article has been downloaded from IOPscience. Please scroll down to see the full text article.

2011 Inverse Problems 27 035008

(http://iopscience.iop.org/0266-5611/27/3/035008)

View the table of contents for this issue, or go to the journal homepage for more

Download details:
IP Address: 18.84.6.234
The article was downloaded on 11/02/2011 at 16:01

Please note that terms and conditions apply.
Level-set techniques for facies identification in reservoir modeling

Marco A Iglesias and Dennis McLaughlin

Department of Civil and Environmental Engineering, Massachusetts Institute of Technology, 15 Vassar Street, Cambridge, MA 02139, USA

E-mail: marcoi@mit.edu

Received 26 June 2010, in final form 21 December 2010
Published 11 February 2011
Online at stacks.iop.org/IP/27/035008

Abstract

In this paper we investigate the application of level-set techniques for facies identification in reservoir models. The identification of facies is a geometrical inverse ill-posed problem that we formulate in terms of shape optimization. The goal is to find a region (a geologic facies) that minimizes the misfit between predicted and measured data from an oil–water reservoir. In order to address the shape optimization problem, we present a novel application of the level-set iterative framework developed by Burger in (2002 Interfaces Free Bound. 5 301–29; 2004 Inverse Problems 20 259–82) for inverse obstacle problems. The optimization is constrained by (the reservoir model) a nonlinear large-scale system of PDEs that describes the reservoir dynamics. We reformulate this reservoir model in a weak (integral) form whose shape derivative can be formally computed from standard results of shape calculus. At each iteration of the scheme, the current estimate of the shape derivative is utilized to define a velocity in the level-set equation. The proper selection of this velocity ensures that the new shape decreases the cost functional. We present results of facies identification where the velocity is computed with the gradient-based (GB) approach of Burger (2002) and the Levenberg–Marquardt (LM) technique of Burger (2004). While an adjoint formulation allows the straightforward application of the GB approach, the LM technique requires the computation of the large-scale Karush–Kuhn–Tucker system that arises at each iteration of the scheme. We efficiently solve this system by means of the representer method. We present some synthetic experiments to show and compare the capabilities and limitations of the proposed implementations of level-set techniques for the identification of geologic facies.

(Some figures in this article are in colour only in the electronic version)
1. Introduction

The success of oil recovery depends, among other factors, on the ability to provide reliable simulations of the reservoir performance. The reservoir dynamics are usually modeled by a distributed parameter dynamic system whose parameters represent the petrophysical properties of the reservoir. A proper characterization of those reservoir properties is fundamental for an accurate reservoir simulation. However, the petrophysical properties are highly heterogeneous and only a limited number of measurements are available at certain locations in the reservoir. Therefore, in order to obtain accurate predictions of the reservoir performance, it is essential to possess reliable estimates of the petrophysical properties in locations where measurements are not available.

When historic production data are available at reservoir wells, global estimates of the petrophysical properties can be obtained by means of data inversion. In other words, given production data, we would like to identify the parameters whose corresponding output fits these production data. Let us denote by \( K : D \rightarrow \mathbb{R} \) the petrophysical property of interest defined on a reservoir whose domain is denoted by \( D \). Let \( f(K) \) denote a vector of predicted data generated by an abstract reservoir model. In general \( f : K_{ad} \rightarrow \mathbb{R}^M \) is a nonlinear operator defined on an admissible parameter space of functions \( K_{ad} \). Then, given data from the reservoir \( d \in \mathbb{R}^M \), the parameter identification problem is usually formulated as

\[
J(K) = \|d - f(K)\|^2 \rightarrow \min_{K \in K_{ad}}.
\]  

(1)

It is well known \([19, 20]\) that, even if the solution to problem (1) exists, it may not be continuous with respect to \( d \). Therefore, the computation of \( K \) in (1) is usually unstable. This type of inverse ill-posed problem is typically addressed by means of regularization. Either by traditional Tikhonov regularization or by iterative regularization, the aim is to define \( a \) \textit{priori} a parameter space whose norm will be used to stabilize (1). Most of the existing techniques for parameter identification in reservoir modeling \([17–20]\) choose \( K_{ad} \) as a space of smooth functions. In \([17, 18, 20]\), for example, the Sobolev space \( H^3(D) \) was utilized for a problem like (1) where \( f \) was the predicted data generated by an oil–water reservoir model. Although the smoothness of \( K_{ad} \) seems to be necessary for regularization purposes, it does not account for a broad class of petrophysical properties that we usually observe in reservoirs. In particular, defining \( K_{ad} \) a space of smooth functions and using some regularization technique to solve (1) eliminates the possibility of identifying discontinuous reservoir properties. In this case, the standard techniques can still be applied but the result will be an overly smooth estimate that may compromise future simulations.

In this paper, the petrophysical property of interest is the absolute permeability of the reservoir. In particular, we consider discontinuous permeabilities of the form

\[
K \equiv K_e + (K_i - K_e)\chi_\Omega,
\]  

(2)

where \( K_i \) and \( K_e \) are constant values of the permeability in the facies denoted by \( \Omega \) and \( \Omega - D \), respectively. In expression (2) \( \chi_\Omega \) is the characteristic function of \( \Omega \) defined by

\[
\chi_\Omega = \begin{cases} 
1 & \text{if } x \in \Omega, \\
0 & \text{if } x \in D - \Omega.
\end{cases}
\]  

(3)

Let us assume that \( K_i \) and \( K_e \) are known; then, \( K \) in (2) depends only on \( \Omega \), i.e. \( K = K(\Omega) \). If we define

\[
F(\Omega) \equiv f(K(\Omega)),
\]  

(4)

problem (1) can be reformulated as

\[
J(\Omega) = \|d - F(\Omega)\|^2 \rightarrow \min_{\Omega \in \Omega_{ad}}.
\]  

(5)
where $S_{ad}$ is an admissible class of shapes. Therefore, the parameter identification problem can be formulated as a shape reconstruction problem where the unknowns are geologic facies.

The level-set technique, first developed by Osher and Sethian [24], was proposed by Santosa in [28] for the computational solutions to inverse obstacle problems like (2). This type of geometrical inverse problem arises in many disciplines. Although there are several techniques to compute solutions to shape inverse problems, the level-set method has become the standard technique because of its computational efficiency and its ability to describe topological changes. In the level-set technique, the shape $\Omega$ is implicitly parameterized as follows:

$$\Omega(\tau) = \{ x \in D | \phi(x, \tau) < 0 \}, \quad \text{(6)}$$

where $\phi$ is the so-called level-set function and $\tau$ is an artificial time variable that parameterizes the evolution of $\Omega$. With this implicit parameterization, a geometrical deformation of $\Omega$ with normal velocity of magnitude $V$ defined on the boundary $\partial \Omega(\tau)$ can be obtained by solving the following Hamilton–Jacobi equation [24]:

$$\frac{\partial \phi}{\partial \tau} + \hat{V} |\nabla \phi| = 0. \quad \text{(7)}$$

In the previous expression, $\hat{V} : D \rightarrow \mathbb{R}$ is an extension of $V$ on $D$. Santosa proposed a gradient-based iterative scheme where, at each iteration level, an expression for $\hat{V}$ is derived so that a new shape $\Omega(\tau + \Delta \tau) = \{ x \in D | \phi(x, \tau + \Delta \tau) < 0 \}$, obtained by solving (7) on $[\tau, \tau + \Delta \tau]$, satisfies $J(\Omega(\tau + \Delta \tau)) \leq J(\Omega(\tau))$. Therefore, an evolution of $\Omega(\tau)$ is generated where the limit $\Omega(\tau \rightarrow \infty)$ is expected to minimize (5). Although Santosa applied his ideas to deconvolution and diffraction screen reconstruction problems, his methodology has been widely used for inverse problems in several disciplines. Interesting reviews of level-set techniques with application to inverse scattering can be found in [11, 12]. For a general survey on level-set techniques we refer the reader to the survey presented in [8]. It is of particular relevance for this paper to mention the work of Burger [6] who developed a functional analytic framework that extends the evolution (gradient-based) approach of Santosa [28]. That framework was used again by Burger in [7] to construct a Levenberg–Marquardt (LM) level-set technique. Both approaches were applied and compared in [7] for the identification of cavities and inverse source problems. Those applications showed that the LM produces similar results in fewer iterations than the gradient-based (GB) technique. However, the overall advantage of the LM method depends on the ability to efficiently solve the linear system that arises from the Karush–Kuhn–Tucker (KKT) condition at each iteration of the algorithm.

Level-set techniques have also been recently applied to the identification of petrophysical properties in reservoirs [4, 5, 13, 21]. Dorn et al [13] follow the ‘evolution approach’ of Santosa [28] where a gradient-based algorithm is proposed for the reconstruction of shapes. The shape derivatives needed for the algorithm of [13] are formally computed via a ‘level-set derivative’ and the resulting Dirac deltas are approximated with a narrowband function. In addition, Dorn et al uses a projection of the level-set update in the Sobolev space $H^1(D)$ to maintain some regularity in the level-set function during their gradient-based scheme. A different approach is presented in [4, 5, 21] where the inverse problem is formulated as an optimization problem in terms of a level-set function. Following the ‘optimization approach’ of Santosa [28], the work in [4, 5, 21] utilizes a Newton-type iterative scheme where the level-set function is updated at each iteration. In this approach, regularization is accomplished by a reparameterization of the level-set function. It is worth mentioning that in the previous approaches [4, 5, 13, 21], the optimal shape is obtained from a level-set function which is in turn updated with the optimization scheme. Therefore, the level-set function is not obtained by solving (7) for some $V$. The approaches previously described provided encouraging results.
that show the potential of level-set techniques for facies identification and provide a good foundation for further research.

In this paper we present the application of the level-set framework of [6, 7] for facies identification from oil–water reservoir production data. The problem is formulated as a shape reconstruction problem as in (5). In contrast to the approach of [4, 5, 13, 21] we pose the problem in terms of the unknown shape $\Omega_1$. Then, the level-set technique is used to evolve the shape, via (7), within the optimization schemes that we present in the following sections. In order to address the problem in a geometrical fashion (i.e. in terms of $\Omega$), we use an integral formulation of the reservoir model. This is also required for the computation of the shape derivatives that we need for the application of the analytic framework of [7]. More precisely, the shape derivatives of the forward model and measurement functional are utilized to set a variational problem for the velocity $V$. Once this velocity is computed, an extension is obtained to solve (7) and find a shape that decreases (5). For our application we study both GB and LM approaches. An adjoint formulation is required to efficiently incorporate the constraint into the optimization problem.

Due to the complexity and the large scale of the forward operator (the reservoir model) and its resulting KKT system, the LM method represents a big computational challenge and a potential disadvantage over a gradient-based formulation. Fortunately, since only a small number of observations are typically available for the present application, the aforementioned KKT system in the LM approach can be solved reasonably efficiently with the representer method. This technique has been successfully used in [17, 18] for parameter identification in reservoir modeling with iterative Newton-type regularization techniques.

The rest of the paper is structured as follows. In section 2 we introduce the reservoir model and the measurement functional. In section 3 we define the facies identification problem. Then, we present an iterative level-set technique for solving the inverse problem. The shape derivatives and the adjoint operators are also presented in section 3. Technical aspects of the level-set techniques are discussed in section 4 and the numerical implementation is presented in section 5. In section 6 we present and discuss numerical results obtained with our level-set implementation. Conclusions and future research are discussed in section 7.

2. The forward model

2.1. An oil–water reservoir model

In this section we present the mathematical model that describes an oil–water reservoir. For $n \in \{2, 3\}$ let $D \subset \mathbb{R}^n$ with smooth boundary that we identify with the physical domain of a reservoir. Let $T > 0$ and let $[0, T]$ be the time interval of interest for the simulation of the reservoir dynamics. The porosity of the reservoir is denoted by $\varphi$. Define an index $\alpha$ that distinguishes the water phase ($\alpha = w$) from the oil phase ($\alpha = o$), and denote by $c_w$ and $c_o$ the $\alpha$-phase fluid and rock compressibility, respectively. Let $s(x, t)$ and $p(x, t)$ be the water saturation and the pressure of the reservoir respectively. We denote by $\lambda_w(s)$ and $\lambda(s)$ the water and total mobility defined by

$$
\lambda_w(s) = \frac{k_{wr}(s)}{\mu_w}, \quad \lambda(s) = \frac{k_{wr}(s)}{\mu_w} + \frac{k_{or}(s)}{\mu_o}
$$

(8)

where $k_{or}$ denotes the relative permeability of the $\alpha$-phase. The relative permeability curves are prescribed (see section 6) nonlinear functions of the water saturation $s$. In expression (8) $\mu_w$ denotes the $\alpha$-phase viscosity. Furthermore, we assume that only water is injected at $N_I$ injection wells located at $\{x_I^l\}_{l=1}^{N_I}$. Water and oil are produced at $N_P$ production wells located at $\{x_P^l\}_{l=1}^{N_P}$. The water flux and total flux are denoted by $u_w(x, t)$ and $u(x, t)$, respectively. We
assume that both injection and production wells are operated under prescribed bottom hole pressure denoted by \( \{ P_{bh,i}^{l}(t) \}_{i=1}^{N_I} \) and \( \{ P_{bh,p}^{l}(t) \}_{l=1}^{N_P} \), respectively. In order to simplify the subsequent analysis, we define

\[
\theta^l(s) = \begin{cases} 
\lambda_w(s) & \text{if } l \in [1, \ldots, N_I], \\
\lambda(s) & \text{if } l \in [N_I + 1, \ldots, N],
\end{cases}
\]  

(9)

where \( N = N_P + N_I \) is the total number of wells. Assuming that both fluids are slightly compressible and that gravity and capillary forces are neglected, it can be shown [17] that the oil–water reservoir can be described by the solution of the following system:

\[
\varphi(c_r + c_w s + c_o (1-s)) \frac{\partial p}{\partial t} + \nabla \cdot \mathbf{u} - \sum_{l=1}^{N} \omega^l \theta^l(s) K \left[ P_{bh}^{l} - p \right] \delta(x - x^l) = 0,
\]

(10)

\[
\varphi(c_r + c_w s) \frac{\partial p}{\partial t} + \frac{\partial s}{\partial t} + \nabla \cdot \mathbf{u}_w - \sum_{l=1}^{N} \omega^l \theta^l(s) K \left[ P_{bh}^{l} - p \right] \delta(x - x^l) = 0,
\]

(11)

where

\[
\mathbf{u} = -K \lambda(s) \nabla p,
\]

(12)

\[
\mathbf{u}_w = \frac{\lambda_w}{\lambda} \mathbf{u},
\]

(13)

and \( K \) defined in (2). The system (10)–(13) is solved on \( (0, T] \) with initial conditions

\[
p(x, 0) = p_0,
\]

(14)

\[
s(x, 0) = s_0.
\]

(15)

Finally, for simplicity homogeneous Neumann boundary conditions are considered

\[
\mathbf{u} \cdot \mathbf{n} = 0,
\]

(16)

\[
\mathbf{u}_w \cdot \mathbf{n} = 0.
\]

(17)

In (10)–(11), \( \omega^l \) is a geometrical factor for the well model [9] corresponding to the \( l \)th well. Analogously, \( \delta(x - x^l) \) is an approximation to the Dirac delta centered at the \( l \)th well.

Remark 2.1. For the derivation of the reservoir model described by (10)–(17) [17], Peaceman’s well model [25] was utilized as well as the assumption that the bottom hole pressure is given at wells. However, we emphasize that the subsequent analysis can be carried out for PDE reservoir models that account for different or even more general production scenarios.

For this paper we assume that the following set of parameters is given:

\[
\mathcal{Q} \equiv \left\{ c_{w}, c_{r}, k_{rw}, \left\{ P_{bh}^{l} \right\}_{l=1}^{N}, \omega^l, p_0, x_0, \mu_o, D, T, \varphi, K_e, K_i \right\}.
\]

(18)

Given \( \mathcal{Q} \) and \( \Omega \), a solution \( (p, s, \mathbf{u}, \mathbf{u}_w) \) to (10)–(17) is called the state of the reservoir. Expressions (12)–(13) can be substituted in (10)–(11) to have a formulation only in terms of \( (p, s) \). However, level-set methods for shape inversion requires the computation of sensitivities with respect to \( \Omega \) and those cannot be computed directly from (10) to (17). Therefore, we proposed the use of a mixed (weak) formulation similar to the one presented in [29], which is also consistent with our numerical discretization of (10)–(17). We start by defining the following spaces [15, 29]:
\( \mathcal{X} = H^1([0, T]; L^2(\Omega)) \cap L^\infty(\Omega_T), \)
\[
\mathcal{V}_i = L^2(\Omega), \quad i \in \{1, 2, 5, 6\},
\]
\[
\mathcal{W}_1 = \mathcal{W}_2 = \{ v \in \mathcal{X} | v(\cdot, 0) = 0 \text{ a.e. in } \Omega \},
\]
\[
\mathcal{U} = \{ u \in [L^2(\Omega_T)]^p | \nabla \cdot u(\cdot, t) \in L^2(\Omega) \text{ and } u(\cdot, t) \cdot n = 0 \text{ a.e. in } (0, T) \},
\]
\[
\mathcal{U}_2 = \{ u \in [L^2(\Omega)]^p | \nabla \cdot u \in L^2(\Omega) \},
\]
\[
\mathcal{V}_3 = \mathcal{V}_4 = \mathcal{W}_4 = \mathcal{U},
\]
\[
\mathcal{H} = \mathcal{X} \times \mathcal{U}^2,
\]
\[
\mathcal{H}_0 = \{ h = (h_1, h_2, h_3, h_4) \in \mathcal{H} | h_1(\cdot, 0) = h_2(\cdot, 0) = 0 \text{ a.e. in } \Omega \}.
\]

Following the approaches of [27, 29], we propose the following weak formulation.

**Definition 2.1 (Forward model).** For every \( \Omega \subset S_{ad} \), \( h = (h_1, h_2, h_3, h_4) \in \mathcal{H} \) and \( v_i \in \mathcal{V}_i \), \( i = 1, \ldots, 8 \), we define the following functionals:

\[
G_1(\Omega, h, v_1) = \int_D \left[ \varphi(c_2) \frac{\partial h_1}{\partial t} + \nabla \cdot h_3 - \sum_{i=1}^N K_c \phi(h_2) \left[ P_{bh}^i - h_1 \right] \delta(x - x^i) \right] v_1 \, dx
\]
\[
- \int_{\Omega} \sum_{i=1}^N [K_i - K_c] \phi(h_2) \left[ P_{bh}^i - h_1 \right] \delta(x - x^i) v_1 \, dx,
\]
\[
G_2(\Omega, h, v_2) = \int_D \left[ \varphi(c_2 + c_w) h_2 \frac{\partial h_1}{\partial t} + \varphi h_2 \frac{\partial h_2}{\partial t} + \nabla \cdot h_4 \right] \left[ K_c \phi(h_2) \left[ P_{bh}^i - h_1 \right] \delta(x - x^i) \right] v_2 \, dx
\]
\[
- \int_{\Omega} \sum_{i=1}^N [K_i - K_c] \phi(h_2) \left[ P_{bh}^i - h_1 \right] \delta(x - x^i) v_2 \, dx,
\]
\[
G_3(\Omega, h, v_3) = \int_D \left[ \frac{1}{\lambda(h_2) K_c} h_3 \cdot v_3 - h_1 \nabla \cdot v_3 \right] \, dx - \int_{\Omega} [K_i - K_c] h_3 \cdot v_3 \, dx,
\]
\[
G_4(\Omega, h, v_4) = \int_D \left[ h_4 - \frac{\lambda_w(h_2)}{\lambda(h_2)} h_3 \right] \cdot v_4 \, dx,
\]
\[
G_5(\Omega, h, v_5) = \int_D [h_1(\cdot, 0) - p_0] v_5 \, dx,
\]
\[
G_6(\Omega, h, v_6) = \int_D [h_2(\cdot, 0) - s_0] v_6 \, dx.
\]

Given \( \Omega \in S_{ad} \), a solution to the forward model is \( h \in \mathcal{H} \) such that

\[
G_i(\Omega, h, v_i) = 0 \quad i \in \{1, \ldots, 6\}
\]

for all \( v_i \in \mathcal{V}_i \) a.e. in \([0, T] \).
To obtain (27)–(28) and (30)–(32) we first substitute \( [h_1, h_2, h_3, h_4] = [p, s, u, u_w] \) in (10)–(17). Then, we multiply expressions (10)–(11) and (13)–(15) by a test function and integrate the result over \( D \). To obtain the linear functional (29) we divide (12) by \( \lambda(s)K \) with \( K \) from (2). The resulting expression is projected on a test function \( v_3 \) and integrated by parts. Then, we use the boundary conditions (16)–(17) which are incorporated into the weak formulation through definition (22). Finally, (3) is used to arrive at (29). The integral formulation presented above allows us to formally apply standard results from the shape calculus to compute the shape derivatives required for the functional-analytic approach of [6]. This mixed formulation for which \( u_w \) and \( u \) are unknowns is suitable for a mixed finite-element discretization of (10)–(17) [29]. We remark that, to our best knowledge, the existence and uniqueness of the previous forward model is still an open problem. Therefore, we need to assume the following.

**Assumption 2.1.** *For every \( \Omega \in \mc{S}_{ad} \), there exists a unique \( h \in \mc{H} \) solution to the forward model.*

In addition to the forward model previously defined, our subsequent analysis also requires the linearization (sensitivities) of (33) with respect to the state \( h \) and the shape \( \Omega \). We now introduce some concepts and notation related to the aforementioned linearization. Let us take \( i (i \in \{1, \ldots, 6\}) \) arbitrary and fixed and consider the corresponding \( G_i : \mc{S}_{ad} \times \mc{H} \times \mc{V}_i \to \mathbb{R} \) defined above. The linearization of \( G_i \) with respect to \( h \in \mc{H} \) can be carried out by a standard ‘formal’ Frechet differentiation [22]. In order to introduce the notation let us take \( j (j \in \{1, \ldots, 4\}) \) and \( (\Omega, h, v_i) \in \mc{S}_{ad} \times \mc{H} \times \mc{V}_i \) arbitrary but fixed. Then, for every \( \Sigma_j \in \mc{H}_j \) we denote by \( D_h G_i (\Omega, h, v_i) \Sigma_j \) the ‘partial’ Frechet derivative with respect to \( h_j (\in \mc{H}_j) \) of the operator \( G_i \) (evaluated at \( (\Omega, h, v_i) \)) applied to \( \Sigma_j \). We formally define the derivative of \( G_i \) with respect to \( h \) applied to \( \Sigma = [\Sigma_1, \ldots, \Sigma_4] \in \mc{H}_j^4 \) by

\[
D_h G_i (\Omega, h, v_i) \Sigma = \sum_{j=1}^{4} D_h G_i (\Omega, h, v_i) \Sigma_j.
\]

The linearization of \( G_i \) with respect to \( \Omega \) involves the concept of the shape derivative. Let us assume that \( \Omega \) admits a parameterization \( \Omega = \Omega (\tau) \) that describes the geometrical deformation of \( \Omega (\tau) \) obtained by applying a velocity field in the normal direction at \( \partial \Omega \). Let us denote the magnitude of that velocity field by \( V \) and let \( (h, v_i) \in \mc{H} \times \mc{V} \) arbitrary. The ‘partial’ shape derivative of \( G_i \) at \( (\Omega (\tau), h, v_i) \) applied to \( V \) is formally defined by

\[
D_\Omega G_i (\Omega (\tau), h, v_i) V \equiv \lim_{\Delta \tau \to 0} \frac{1}{\Delta \tau} [G_i (\Omega (\tau + \Delta \tau), h, v_i) - G_i (\Omega (\tau), h, v_i)]
\]

A rigorous definition of the shape derivative based on the speed method can be found in [10]. For our analysis we assume that the shape derivative is a linear functional of \( V \). This assumption, under certain regularity conditions, can be made rigorous by results from the shape analysis [10].

### 2.2. Measurement functionals

In this section we introduce the measurement functionals that we used to relate the solution to the forward model to each observation collected at the reservoir. We assume that we can measure the flow rate at the locations of the production and injection wells. Once the well configuration is given with a number of (production plus injection) \( N \) wells, the number of spatial measurements \( N \) is fixed. Then, for every well \( l \in \{1, \ldots, N\} \), we specify a set of \( M^l \) measurement times denoted by \( \{t^l_m\}_{m=1}^{M^l} \) when measurements are collected. Note that,
for a given well configuration, $M \equiv \sum_{l=1}^{N} M'$ defines the total number of measurements. The number of measurement times $M'$ collected at the $l$th well depends on the measurement procedure. When downhole permanent sensors are available, measurement can be collected almost on real time. However, for the subsequent implementation the upper bound for $M'$ is given by the number of time steps in the numerical discretization.

To derive an expression of the flow rate in terms of the solution to the forward model $(s, p)$ we use Peaceman’s well model \[25\]. This model, also used to derive equations (10)–(11) \[17\] postulates that the total flow rate is given by the expression

$$q^l(t) = \int_{\Omega} \omega^l K(x) \lambda(s(x, t)) \left[ P_{bh}^l(t) - p(x, t) \right] \delta(x - x^l) \, dx$$

(36)

for $l \in \{1, \ldots, N\}$. We now use model (36) to construct a measurement functional that defines 'predicted data' which in turn can be compared with real measured data from wells. From (36) with $K$ defined as in (2) and recalling that $h \equiv (h_1, h_2, h_3, h_4)$, we define the measurement functional

$$\mathcal{M}^l_m(\Omega, h) = \int_{0}^{T} \int_{\Omega} \omega^l \lambda(h_2) K_e \left[ P_{bh}^l - h_1 \right] \delta(t - t_m^l) \delta(x - x^l) \, dx \, dt$$

$$+ \int_{\Omega} \int_{0}^{T} \omega^l \lambda(h_2) K_e \left[ P_{bh}^l - h_1 \right] \delta(t - t_m^l) \delta(x - x^l) \, dx \, dt$$

(37)

for every $l \in \{1, \ldots, N\}$ and $m \in \{1, \ldots, M'\}$. The Frechet derivative with respect to $h$ and applied to $\Sigma$ is defined analogous to (34) and denoted by $D_h \mathcal{M}^l_m(\Omega, h) \Sigma$. Similarly, the shape derivative of $\mathcal{M}^l_m(\Omega, h)$ is defined analogous to (35) and denoted by $D_{\mathcal{M}^l_m(\Omega, h)} V$.

Additionally, for each $l \in \{1, \ldots, N\}$ we define the following vectors of dimension $M'$:

$$\mathcal{M}^l(\Omega, h) \equiv \left[ \mathcal{M}^1(\Omega, h), \ldots, \mathcal{M}^{M'}(\Omega, h) \right]^T,$$

(38)

$$D_h \mathcal{M}^l(\Omega, h) \Sigma \equiv \left[ D_h \mathcal{M}^1(\Omega, h) \Sigma, \ldots, D_h \mathcal{M}^{M'}(\Omega, h) \Sigma \right]^T,$$

(39)

$$D_{\mathcal{M}^l_m(\Omega, h)} V \equiv \left[ D_{\mathcal{M}^l_m(\Omega, h)} V, \ldots, D_{\mathcal{M}^l_m(\Omega, h)} V \right]^T.$$  

(40)

Furthermore, we define

$$\mathcal{M}(\Omega, h) \equiv \left[ \mathcal{M}^1(\Omega, h), \ldots, \mathcal{M}^N(\Omega, h) \right],$$

$$D_h \mathcal{M}(\Omega, h) \Sigma \equiv \left[ D_h \mathcal{M}^1(\Omega, h) \Sigma, \ldots, D_h \mathcal{M}^N(\Omega, h) \Sigma \right],$$

(41)

$$D_{\mathcal{M}^l_m(\Omega, h)} V \equiv \left[ D_{\mathcal{M}^l_m(\Omega, h)} V, \ldots, D_{\mathcal{M}^l_m(\Omega, h)} V \right].$$

(42)

of dimension $M \equiv \sum_{l=1}^{N} M'$.

Given $\Omega$, the solution $h$ to the forward model is obtained by solving (33). Then, from (38) to (41) we may compute the 'predicted data' $\mathcal{M}(\Omega, h)$. As we indicated earlier, 'predicted data' will be used to compare with 'measured data' (usually called 'production data') collected at the reservoir. For this comparison, a norm in the data space needs to be defined. For every $l \in \{1, \ldots, N\}$ let $C_l \in \mathbb{R}^{M' \times M'}$ be a positive definite matrix. Then, for any $v \in \mathbb{R}^M$ with a structure like (38)–(41), we define

$$\|v\| = \sum_{l=1}^{N} [v^l]^T C_l^{-1} [v^l].$$

(43)
The particular definition of $C_l$ will be given in section 6. On one hand, the weighting matrix $C_l$ will ensure that the norm (43) has the correct units of production data. On the other hand, it allows us to weight production data according to some prior information of the measurement error of production data. Furthermore, let $C \in \mathbb{R}^{M \times M}$ be the following block diagonal matrix:

$$C \equiv \begin{bmatrix} C_1 & \cdots & \cdots & C_N \\ \vdots & & & \vdots \\ & & & \vdots \\ & & & C_N \end{bmatrix}. \quad (44)$$

3. The inverse problem

Let $\Omega'$ denote the 'true' facies in the reservoir. In other words, $\Omega'$ is the geological facies such that the permeability of the reservoir takes value $K_i$ in $\Omega'$. Let $h^\star$ be the solution to the forward model (33) with $\Omega = \Omega'$ and define $d \equiv M(\Omega', h^\star)$. If we assume that the forward model (33), the permeability characterization (2) and the measurement functionals (37) provide an exact representation of the reservoir dynamics, then $d$ should coincide with 'measured data' collected at the reservoir wells. Under those assumptions, the inverse problem can be formulated in the following statement: given reservoir data $d$, find $\Omega_1^\star \in S_{ad}$ such that $d = M(h, \Omega_1^\star)$ where $h \in H$ is the solution to the weak forward model in the sense of definition 2.1.

In a real scenario, the measurements collected at the reservoir wells are typically contaminated with errors. Therefore, we cannot expect to exactly reproduce the vector of measurements $d$. Instead, the noisy measurements form a vector $d^\eta$ that we suppose satisfies

$$\|d - d^\eta\| \leq \eta \quad (45)$$

with the norm defined in (43). In the general case where only $d^\eta$ is available, the solution to the aforementioned inverse problem should be understood in the following least-squares sense.

**Definition 3.1** (The inverse problem). Given a set of production data $d^\eta$, solve

$$J(\Omega) = \|d^\eta - M(h, \Omega)\|^2 \rightarrow \min_{\Omega \in S_{ad}} \quad (46)$$

where $h = (h_1, h_3, h_3, h_4)$ is a solution to the forward model in the sense of definition 2.1.

3.1. Level-set techniques

We now assume that $\Omega = \Omega(\tau)$ is parameterized in terms of a level-set function as in (6). This parameterization allows us to define an iterative scheme where a geometrical deformation is obtained at each iteration level by solving (7). The general idea of constructing level-set-based solutions to (46) [6, 7, 28], is to select a velocity $V$ so that after solving (7), the new shape (implicitly parameterized by $\phi$) produces a decrease of $J$ in (46). In this paper we report the implementation of two approaches for the computation of $V$: the gradient-based (GB) approach of [6] and the Levenberg–Marquardt (LM) approach presented in [7]. Let us denote by $H^{1/2}(\partial \Omega)$ the standard fractional Sobolev space [1]. We now present the level-set iterative technique with the two possible computations for the velocity.

**Algorithm 1** [Level-set technique [6, 7]]. Set $\tau > 1$ and select $\phi^0$ such that $\{\phi^0\} = \Omega(0)$ where $\Omega(0)$ is the initial facies. For each $n = 1, \ldots$
(1) Given $\Omega = \Omega(\tau)$ find $h$ by solving the forward model in definition 2.1.

(2) If

$$\|d - \mathcal{M}(\Omega, h)\| \leq \tau \eta$$

stop. Output: $\Omega(\tau)$.

(3) Compute $V$ by solving one of the following variational problems: (GB) Find $V$ such that

$$\langle V, W \rangle_{H^{1/2}(\partial \Omega)} = -D_{\Omega} J(\Omega) W$$

for all $W \in H^{1/2}(\partial \Omega)$ where $J$ is the cost functional defined in (46). (LM) Find $V$ that minimizes

$$J_{LM}(V) = \|d - \mathcal{M}(\Omega, h) - D_h \mathcal{M}(\Omega, h) \Sigma - D_{\Omega} \mathcal{M}(\Omega, h) V\|^2 + \alpha \|V\|^2_{H^{1/2}(\partial \Omega)}$$

subject to

$$D_h G_i(\Omega, h, v_i) \Sigma + D_{\Omega} G_i(\Omega, h, v_i) V = 0.$$  

(4) Find $\hat{V}$, an extension of $V$ on $D$.

(5) Solve the Hamilton–Jacobi equation (7) over the interval $(\tau, \tau + \Delta \tau)$ for some $\Delta \tau > 0$.

(6) Set $\tau = \tau + \Delta \tau$ and $\Omega(\tau) = \{ \phi(\cdot, \tau + \Delta \tau) < 0 \}$.

As we indicated earlier, the shape derivatives can be considered linear functionals of a velocity field $V$ defined on $\partial \Omega$. In the subsequent analysis we present the explicit formulas for every shape derivative that appears in algorithm 1. From those formulas (expressions (60)–(63) and (68)) and some formal arguments we assume (assumption 3.1) that the shape derivatives for this application are bounded linear functionals on the Sobolev space $H^{1/2}(\partial \Omega)$. This assumption is essential for the well-posedness of step (3) in algorithm 1 since for problems (GB) and (LM) the velocity is computed by solving a variational problem in $H^{1/2}(\partial \Omega)$. Although we cannot prove assumption 3.1 rigorously, the space $H^{1/2}(\partial \Omega)$ is chosen for this application from the numerical evidence that $H^{1/2}(\partial \Omega)$ provides the minimum regularity for $V$ so that the solution of (7) can be numerically computed. Due to the choice of $H^{1/2}(\partial \Omega)$, the (GB) problem of algorithm 1 corresponds to the ‘Stefan-like flow’ of [6]. A similar choice for the velocity space was also needed in [2] for the identification of cavities.

Note that if there exists $V \in H^{1/2}(\partial \Omega)$ that satisfies the variational problem (48), then for $W = V$ in (48) we have

$$\frac{d}{d\tau} J(\Omega(\tau)) = D_{\Omega} J(\Omega) V = -\|V\|^2_{H^{1/2}(\partial \Omega)} \leq 0.$$  

Therefore, a geometrical perturbation of $\Omega(\tau)$ with normal velocity of magnitude $V$ will produce a new shape $\Omega(\tau + \Delta \tau)$ such that $J(\Omega(\tau + \Delta \tau)) < J(\Omega(\tau))$. This new shape, is obtained from steps (4)–(6) of algorithm 1.

As with the gradient-based approach, the main goal of the LM technique is to obtain a decrease of the objective functional $J$ (46). It is clear that the level-set LM technique is an extension of the LM approach for inverse ill-posed problems [17]. In other words, in (49) we minimize the squared norm of the linearized data misfit plus a stabilization term weighted by a Tikhonov parameter $\alpha$. In addition, the constraint (50) is the linearized model. It is not straightforward to see that $V$ computed from (49) will result in a decrease of $J$, so we refer the reader to the proof in [7].
3.2. Sensitivity

For the implementation of the LM algorithm presented above, the computations of $D_{h_i} \mathcal{M}(\Omega, \mathbf{h}) \Sigma$, $D_{\mathcal{M}}(\Omega, \mathbf{h}) V$, $D_{h_i} G_i(\Omega, \mathbf{h}, v_i) \Sigma$ and $D_{\mathcal{M}} G_i(\Omega, \mathbf{h}, v_i) V$ are required. Let us examine $G_3$ in (29). From formal arguments it follows that

$$D_{h_i} G_3(\Omega, \mathbf{h}, v_3) \Sigma_1 = - \int_D \Sigma_1 \nabla \cdot v_3 \, dx,$$

$$D_{h_i} G_3(\Omega, \mathbf{h}, v_3) \Sigma_2 = - \int_D \frac{\lambda'(h_2)}{\lambda^2(h_2) K_e} \Sigma_2 \mathbf{h}_3 \cdot v_3 \, dx + \int_\Omega \frac{[K_i - K_e] \lambda'(h_2) \Sigma_2}{\lambda(h_2) K_e} \mathbf{h}_3 \cdot v_3 \, dx,$$

$$D_{h_i} G_3(\Omega, \mathbf{h}, v_3) \Sigma_3 = \int_D \frac{1}{\lambda(h_2) K_e} \Sigma_3 \cdot v_3 \, dx - \int_\Omega \frac{[K_i - K_e]}{\lambda(h_2) K_e} \Sigma_3 \cdot v_3 \, dx,$$

$$D_{h_i} G_3(\Omega, \mathbf{h}, v_3) \Sigma_4 = 0. \quad (52)$$

Furthermore, we observe that the $\Omega$-dependent term in $G_3$ has the prototypical integral form required for the formal application of theorem 4.2 in [10] which yields

$$D_{\mathcal{M}} G_3(\Omega, \mathbf{h}, v_3) V = \int_{\partial \Omega} V(\sigma) \frac{[K_i - K_e]}{\lambda(h_2) K_e} \mathbf{h}_3 \cdot v_3 \, d\sigma. \quad (53)$$

A similar argument applies for the computation of the derivatives of (27)–(32) as well as the derivatives of $\mathcal{M}(\Omega, \mathbf{h})$ in (37)–(41). From the previous analysis and the chain rule, we obtain that the derivatives with respect to $\mathbf{h}$, $D_{h_i} G_i(\Omega, \mathbf{h}, v_i) : \mathcal{H} \times H^{1/2}(\partial \Omega) \to \mathbb{R}$ at $(\Omega, \mathbf{h}, v_i)$ are given by the following relations:

$$D_{h_i} G_1(\Omega, \mathbf{h}, v_1) \Sigma = \int_D \left[ \phi'(h_2) \frac{\partial h_1}{\partial t} \Sigma_2 + \nabla \cdot \Sigma_3 + \phi(h_2) \frac{\partial \Sigma_1}{\partial t} \right] \mathbf{h},$$

$$- \sum_{i=1}^N \left[ \frac{\lambda}{\lambda} \Sigma_2 \phi'_{\ell} - \phi \lambda(h_2) \Sigma_1 K_e \right] \delta(x - x') \, v_1 \, dx,$$

$$- \int_\Omega \sum_{i=1}^N \left[ \frac{K_i - K_e}{K_e} \right] \left[ \frac{\lambda}{\lambda} \Sigma_2 \phi'_{\ell} - \phi \lambda(h_2) \Sigma_1 K_e \right] \delta(x - x') \, v_1 \, dx, \quad (54)$$

$$D_{h_i} G_2(\Omega, \mathbf{h}, v_2) \Sigma = \int_D \left[ \phi(c_x + c_w) \Sigma_2 \frac{\partial p}{\partial t} + \phi \lambda(h_2) \Sigma_1 K_e \right] \mathbf{h},$$

$$- \sum_{i=1}^N \left[ \frac{D\theta}{\lambda} \Sigma_2 \phi'_{\ell} - K_e \theta \Sigma_1 K_e \right] \mathbf{h},$$

$$- \int_\Omega \sum_{i=1}^N \left[ \frac{K_i - K_e}{K_e} \right] \left[ \frac{D\theta}{\lambda} \Sigma_2 \phi'_{\ell} - K_e \theta \Sigma_1 K_e \right] \mathbf{h}, \quad (55)$$

$$D_{h_i} G_3(\Omega, \mathbf{h}, v_3) \Sigma = \int_D \left[ \frac{1}{\lambda K_e} \Sigma_3 - \frac{\lambda}{\lambda} \Sigma_2 \mathbf{h}_3 \right] \mathbf{h}_3 \cdot v_3 - \Sigma_1 \nabla \cdot v_3 \, dx,$$

$$- \int_\Omega \frac{[K_i - K_e]}{\lambda K_e} \Sigma_3 - \frac{\lambda}{\lambda} \Sigma_2 \mathbf{h}_3 \cdot v_3 \, dx, \quad (56)$$

$$D_{h_i} G_4(\Omega, \mathbf{h}, v_4) \Sigma = \int_D \Sigma_4 \cdot v_4 - \int_D \left[ \frac{\lambda_s}{\lambda} \right] \Sigma_2 \mathbf{h}_3 \cdot v_4 \, dx - \int_D \frac{\lambda_s}{\lambda} \Sigma_3 \cdot v_4 \, dx, \quad (57)$$
\[
\text{D}_h G_5(\Omega, h, v_5) \Sigma = \int_D \Sigma_1(\cdot, 0) v_5 \, dx,
\]
(58)
\[
\text{D}_h G_6(\Omega, h, v_6) \Sigma = \int_D \Sigma_2(\cdot, 0) v_6 \, dx.
\]
(59)

The shape derivatives of \( G_i \) are given by
\[
\text{D}_h G_1(\partial/\Omega_1, h, v_1) V \equiv - \int_{\partial/\Omega_1 \cap T_0} \Sigma \left[ K_i - K_e \right] q_1^i \delta(\sigma - x^i) V(\sigma) \, d\sigma,
\]
(60)
\[
\text{D}_h G_2(\partial/\Omega_1, h, v_2) V = - \int_{\partial/\Omega_1 \cap T_0} \Sigma \left[ K_i - K_e \right] \frac{\theta^j}{\lambda} q_2^j \delta(\sigma - x^j) V(\sigma) v_2 \, d\sigma,
\]
(61)
\[
\text{D}_h G_3(\partial/\Omega_1, h, v_3) V = \int_{\partial/\Omega_1 \cap T_0} V(\sigma) \left[ K_i - K_e \right] \frac{\lambda}{\lambda K_i K_e} q_3 \delta(\sigma - x^3) \, d\sigma,
\]
(62)
\[
\text{D}_h G_i(\partial/\Omega_1, h, v_4) V \equiv 0, \quad i \in \{4, 5, 6\}.
\]
(63)

In expressions (54)–(63) we have used the following definitions:
\[
\lambda \equiv \lambda(h_2), \quad \theta^j \equiv \theta^j(h_2), \quad q_1^j \equiv K_e \lambda \left[ p_{hh}^j - p \right].
\]
(64)

With similar arguments we can show that
\[
\text{D}_h \mathcal{M}_m(\Omega, h) \Sigma_1 = - \int_D \int_0^T \omega^i \lambda K_i \Sigma_1 \delta(t - t_m^i) \delta(x - x^i) \, dx \, dt
\]
\[
- \int_{\partial/\Omega_1 \cap T_0} \omega^j \lambda \left[ K_i - K_e \right] \Sigma_1 \delta(t - t_m^j) \delta(x - x^j) \, d\sigma \, dt,
\]
(65)
\[
\text{D}_h \mathcal{M}_m(\Omega, h) \Sigma_2 = \int_D \int_0^T \frac{\lambda'}{\lambda} \Sigma_2 q_2^j \delta(t - t_m^j) \delta(x - x^j) \, dx \, dt
\]
\[
+ \int_{\partial/\Omega_1 \cap T_0} \frac{\left[ K_i - K_e \right]}{K_e} \frac{\lambda'}{\lambda} \Sigma_2 q_2^j \delta(t - t_m^j) \delta(x - x^j) \, d\sigma \, dt,
\]
(66)
\[
\text{D}_h \mathcal{M}_m(\Omega, h) \Sigma_i \equiv 0, \quad i \in \{3, 4\}.
\]
(67)

Additionally,
\[
\text{D}_h \mathcal{M}_m(\Omega, h) V = \int_{\partial/\Omega_1 \cap T_0} \frac{\left[ K_i - K_e \right]}{K_e} q_1^j \delta(t - t_m^j) \delta(x - x^j) V(\sigma) \, d\sigma.
\]
(68)

From our formal application of [10] (theorem 4.2) we find that the shape derivatives (60)–(63) are linear functionals of \( V \). Note that if \( q_1^j \delta(\sigma - x^j) v_1 \in L^2(\partial \Omega) \) is bounded, then by extension arguments (60) is a linear functional bounded on \( L^2(\partial \Omega) \). However, as we indicated earlier in this section, our experiments show that additional regularity must be imposed in \( V \). This should not be surprising if we realize that (60) consists of terms such as \( \delta(\sigma - x^j) \) which is the trace of an approximation to a Dirac delta. Therefore, \( q_1^j \delta(\sigma - x^j) v_1 \in L^2(\partial \Omega) \) may be an unrealistic assumption that cannot be used in this analysis. From the previous reasons and our numerical experiments, we assume that (60)–(63) are linear functionals bounded on the smaller space \( H^{1/2}(\partial \Omega) \).
Assumption 3.1. For every $\Omega \in S_{ad}$ and $i \in \{1, \ldots, 6\}$, the shape derivative $D_{\Omega}G_i(\Omega, h, v_i)V$ is a bounded linear functional on $H^{1/2}(\partial \Omega)$. Furthermore, let $(\Omega, h) \in S_{ad} \times H$ be arbitrary but fixed. Then for every $V \in H^{1/2}(\partial \Omega)$ we assume that
\[ D_{h}G_i(\Omega, h, v_i)\Sigma + D_{\Omega}G_i(\Omega, h, v_i)V = 0 \] (69)
admits a unique solution $\Sigma \in H$. Moreover, there exists $C > 0$ such that $\|\Sigma\|_H \leq C\|V\|_{H^{1/2}(\Omega)}$.

Remark 3.1. Consider now the mappings $h : \{\Omega : \Omega \subset D\} \rightarrow H$ where $h = h(\Omega) \in H$ is the solution to the forward model. From assumption 3.1 it follows that, for every $\Omega \in S_{ad}$, the shape derivative of $h(\Omega)$ exists. This derivative is denoted by $D_{\Omega}h(\Omega)V$. Additionally, the shape derivative is the solution to (69).

Remark 3.2. Define $F : S_{ad} \rightarrow \mathbb{R}^M$ by
\[ F(\Omega) \equiv \mathcal{M}(\Omega, h(\Omega)), \] (70)
where $h(\Omega)$ is defined in remark 3.1 and let us denote by $D_{\Omega}h(\Omega)V \equiv \Sigma$ the shape derivative. Then, for every $V \in H^{1/2}(\Gamma)$, we can formally apply the chain rule to find
\[ D_{\Omega}F(\Omega)V = D_{h}\mathcal{M}(\Omega, h(\Omega))\Sigma + D_{\Omega}\mathcal{M}(\Omega, h(\Omega))V \] (71)
where $\Sigma$ is the solution to (69).

From (46) and remark 3.2, it follows formally that
\[ D_{\Omega}J(\Omega)V = -[d^\theta - \mathcal{M}(\Omega, h(\Omega))]C^{-1}[D_{h}\mathcal{M}(\Omega, h(\Omega))\Sigma + D_{\Omega}\mathcal{M}(\Omega, h(\Omega))V] \] (72)
where $\Sigma \in H$ is the shape derivative of $h = h(\Omega)$, i.e. $\Sigma \equiv D_{\Omega}h(\Omega)V$ that satisfies
\[ D_{h}G_i(\Omega, h(\Omega), v_i)\Sigma + D_{\Omega}G_i(\Omega, h(\Omega), v_i)V = 0, i \in \{1, \ldots, 4\} \] (73)
for all $v_i \in V$. From the previous arguments it can be observed that an adjoint formulation is now desirable to reduced the effort of calculating (73) for every $V \in H^{1/2}(\partial \Omega)$.

3.3. The adjoint operators

For both the gradient-based and the LM techniques, expression (73) is a constraint for the solution of the variational problem that generates the velocity for the level-set equation. In order to eliminate the aforementioned constraint, an adjoint formulation will be derived. In this section we define the adjoint operators associated with (73). In order to simplify the subsequent analysis, for every $(\Omega, h)$ fixed, we first define the following linear operators:

\[ B_1(X, Y) \equiv \sum_{i=1}^{N} a_i[\lambda X + \theta^i Y]\delta(x - x^i), \] (74)

\[ B_2(X, Y) \equiv \sum_{i=1}^{N} [\lambda^i X + D\theta^i Y]^{1/\lambda} q_i^i\delta(x - x^i), \] (75)

\[ B_3(X, Y) \equiv \sum_{i=1}^{N} q_i^i[\lambda X + \theta^i Y]\delta(x - x^i). \] (76)
We now define $L^* G_i : \mathcal{H} \times W_i \rightarrow \mathbb{R}$ for $i = 1, \ldots, 4$ by the following relations:

$$L^* G_1(A, w_1) \equiv \int_0^T \int_D \left[ -\varphi \frac{\partial (c(h_2)A_1)}{\partial t} - \varphi (c_w + c_r) \frac{\partial (sA_2)}{\partial t} - \nabla \cdot A_3 + K_e B_1(A_1, A_2) \right] w_1$$

$$+ \int_D [c(h_2)A_1 + (c_r + c_w)h_2 A_2] \varphi w_1 \big|_{t = T}$$

$$+ \int_0^T \int_{\Omega} [K_i - K_e] B_1(A_1, A_2) w_1 \, dx \, dt,$$

(77)

$$L^* G_2(A, w_2) \equiv \int_D \left[ \varphi c'(h_2) \frac{\partial h_1}{\partial t} A_1 + \varphi (c_r + c_w) \frac{\partial h_2}{\partial t} A_2 - \varphi \frac{\partial A_2}{\partial t} - B_2(A_1, A_2) \right.$$

$$- \frac{\lambda'}{\lambda^2 K_e} h_3 \cdot A_3 - \left[ \frac{\lambda_w}{\lambda} \right] \left[ h_1 \cdot A_4 \right] w_2 + \int_D \varphi \left[ w_2 A_2 \right]_{t = T}$$

$$- \int_\Omega \left[ \frac{K_j - K_e}{K_e} B_2(A_1, A_2) - \frac{K_j - K_e}{K_i K_e} \frac{\lambda'}{\lambda^2} h_3 \cdot A_3 \right] w_2,$$

(78)

$$L^* G_3(A, w_3) \equiv \int_0^T \int_D \left[ \nabla \cdot w_3 A_1 + \frac{1}{\lambda K_e} w_3 \cdot A_3 - \frac{\lambda_w}{\lambda} w_3 \cdot A_4 \right]$$

$$- \int_0^T \int_\Omega \frac{K_j - K_e}{\lambda K_i K_e} w_3 \cdot A_3,$$

(79)

$$L^* G_4(A, w_4) \equiv \int_0^T \int_D A_4 \cdot w_4 + \int_0^T \int_D A_2 \nabla \cdot w_4.$$

(80)

We now state the following.

**Lemma 3.1.** Let $\Omega \in S_{ad}$ and $h \in \mathcal{H}$. Let $D_h G_i$ be defined as in (54)–(59). Operators (77)–(80) satisfy

$$\sum_{j=1}^4 L^* G_j(A, w_j) = \sum_{i=1}^4 \int_0^T D_h G_i(\Omega, h, A_i) w$$

(81)

for all $w = [w_1, \ldots, w_4] \in \prod_{j=1}^4 W_j$ and all $A = [A_1, \ldots, A_4] \in \mathcal{H}$.

**Proof.** See appendix A.

4. Application of the level-set techniques

In this section we discuss the computation of the velocity $V$ in step (3) of algorithm 1.

4.1. The gradient-based level-set algorithm

In order to compute (48), we first consider the following adjoint problem: find $A \in \mathcal{H}$ such that

$$L^* G_j(A, w_j) = \left[ d^g - \mathcal{M}(\Omega, h) \right] C^{-1} D_hj, \mathcal{M}(\Omega, h) w_j, \quad j \in \{1, \ldots, 4\}$$

(82)

for all $w_j \in W_j$. From (65) to (68) and (74) to (80) it is not difficult to see that only $\Omega$ and $h$ are needed for the construction of problem (82). Fortunately, $(\Omega, h)$ are given at the
beginning of each iteration level (see step (1) of algorithm 1). Existence and uniqueness of (82) will be assumed for the subsequent analysis. Then we add (82) for \( j \in \{1, \ldots, 4\} \) to find that
\[
\sum_{j=1}^{4} L^* G_j(A, w_j) = [d^0 - \mathcal{M}(\Omega, h)] C^{-1} D_h \mathcal{M}(\Omega, h) w,
\]
(83)
where \( w = [w_1, \ldots, w_4] \). On the other hand, we use (81) with \( A \) the solution to (82) and the result is combined with (83) to find that
\[
[d^0 - \mathcal{M}(\Omega, h)] C^{-1} D_h \mathcal{M}(\Omega, h) w = \sum_{i=1}^{4} \int_{0}^{T} D_h G_i(\Omega, h, A_i) w
\]
(84)
for all \( w \in \prod_{i=1}^{4} \mathcal{V}_i \). Let \( w = \Sigma \) be the solution to (73). Then, from (73) and (84) it follows that (72) becomes
\[
D_Q J(\Omega) V = -[d^0 - \mathcal{M}(\Omega, h)] C^{-1} [D_h \mathcal{M}(\Omega, h) \Sigma + D_{Q2} \mathcal{M}(\Omega, h) V]
\]
\[
= \sum_{i=1}^{4} \int_{0}^{T} D_h G_i(\Omega, h, A_i) V - [d^0 - \mathcal{M}(\Omega, h)] C^{-1} D_{Q2} \mathcal{M}(\Omega, h) V.
\]
(85)
Once \( A \) is computed with (82), we use (85) in (48) to compute \( V \) by solving
\[
\langle V, W \rangle_{H^{1/2}(\partial \Omega)} = -\sum_{i=1}^{4} \int_{0}^{T} D_h G_i(\Omega, h, A_i) W + [d^0 - \mathcal{M}(\Omega, h)] C^{-1} D_{Q2} \mathcal{M}(\Omega, h) W
\]
(86)
for every \( W \in H^{1/2}(\partial \Omega) \). We summarize algorithm 1 with the GB approach.

1. Given \( \Omega = \Omega(\tau) \) find \( h \) by solving the forward model.
2. If \( \|d^0 - \mathcal{M}(\Omega, h)\| \leq \tau h \), stop. Output: \( \Omega(\tau) \).
3. [3.1] Compute \( A \) by solving the adjoint problem (82).
4. [3.1] Compute \( V \) from (86).
5. Find \( \bar{V} \), an extension of \( V \) on \( D \).
6. Solve the Hamilton–Jacobi equation (7) over the interval \( (\tau, \tau + \Delta \tau) \) for some \( \Delta \tau > 0 \).
7. Set \( \tau = \tau + \Delta \tau \) and \( \Omega(\tau) = [\phi, \tau + \Delta \tau) < 0] \).

4.2. The Levenberg–Marquardt scheme

The computation of the velocity with the LM approach requires the solution to (49)–(50). This can be found from the first order optimality condition (KKT conditions) stated in the following proposition.

**Proposition 4.1 (KKT conditions).** The optimality condition for (49)–(50) is \((\Sigma, V, A) \in \mathcal{H} \times H^{1/2}(\partial \Omega) \times \mathcal{H}\) such that
\[
D_h G_i(\Omega, h, v_i) \Sigma + D_{Q2} G_i(\Omega, h, v_i) V = 0, i = 1, \ldots, 4
\]
(87)
\[
\alpha[V, W]_{H^{1/2}(\partial \Omega)} = -\sum_{i=1}^{4} \int_{0}^{T} D_h G_i(\Omega, h, A_i) W + Q[\Sigma, V] \cdot D_{Q2} \mathcal{M}(\Omega, h) W,
\]
(88)
\[
L^* G_i(A, w_i) = Q[\Sigma, V] \cdot D_h \mathcal{M}(\Omega, h) w_i, i = 1, \ldots, 4
\]
(89)
for all \( v_i \in \mathcal{V}_i \) \( (i = 1, \ldots, 4) \), \( W \in H^{1/2}(\partial \Omega) \) and \( w_i \in \mathcal{V}_i \) where
\[
Q[\Sigma, V] \equiv [d - \mathcal{M}(\Omega, h) - D_h \mathcal{M}(\Omega, h) \Sigma - D_{Q2} \mathcal{M}(\Omega, h) V] C^{-1}
\]
(90)
for \( l \in \{1, \ldots, N\} \) and \( m \in \{1, \ldots, M\} \).
Proof. We adjoin the constraint (50) in (49) by defining the following Lagrangian functional on $\Pi_{j=m}^4 W_j \times H^1(\Omega) \times H$: 

$$
\hat{J}(\Sigma, V, A) \equiv \left\| d - \mathcal{M}(\Omega, h) - D_h \mathcal{M}(\Omega, h) \Sigma - D_{\mathcal{Q}} \mathcal{M}(\Sigma, h) \right\|^2 + \alpha \| V \|^2_{H^{1/2}(\partial \Omega)} 
+ \int_0^T \sum_{i=1}^4 [D_h G_i(\phi, h, A_i(\cdot, t)) \Sigma + D_{\mathcal{Q}} G_i(\phi, h, A_i(\cdot, t)) V] \, dt. 
$$

(91)

From (58) and (59) note that (50) for $i \in \{5,6\}$ are the weak forms of the homogeneous initial conditions for $\Sigma_1$ and $\Sigma_2$. Then, for simplicity those constraints are included in the space $W_2$ (21) that we use to define the Lagrangian (91). It is straightforward to see that 

$$
D_\Sigma \hat{J}(\Sigma, V, A)w = -[d - \mathcal{M}(\Omega, h) - D_h \mathcal{M}(\Omega, h) \Sigma - D_{\mathcal{Q}} \mathcal{M}(\Sigma, h) V] \mathbf{C}^{-1} D_h \mathcal{M}(\Omega, h)w 
+ \int_0^T \sum_{i=1}^4 D_h G_i(\phi, h, A_i(\cdot, t))w \, dt
$$

(92)

for all $w \in \Pi_{j=m}^4 W_j$. Then, from (90) and (81)

$$
D_\Sigma \hat{J}(h, V, A)w = -Q(\Sigma, V) \cdot D_h \mathcal{M}(\Omega, h)w + \sum_{j=1}^4 L^*G_j(A, w_j).
$$

(93)

Similarly we have

$$
D_y \hat{J}(\Sigma, V, A)W = -Q(\Sigma, V) \cdot D_{\mathcal{Q}} \mathcal{M}(\Sigma, h)W + \alpha [V, W]\big|_{H^{1/2}(\partial \Omega)} 
+ \int_0^T \sum_{i=1}^4 [D_{\mathcal{Q}} G_i(\phi, h, A_i(\cdot, t)) W] \, dt
$$

(94)

for every $W \in H^{1/2}(\partial \Omega)$. Trivially

$$
D_\Lambda \hat{J}(\Sigma, V, A)v = \int_0^T \sum_{i=1}^4 [D_h G_i(\phi, h, v_i(\cdot, t)) \Sigma + D_{\mathcal{Q}} G_i(\phi, h, v_i(\cdot, t)) V] \, dt
$$

(95)

for all $v \in H$. The first order optimality condition for (91) can be stated as

$$
D_\Sigma \hat{J}(\Sigma, V, A)w + D_y \hat{J}(\Sigma, V, A)W + D_\Lambda \hat{J}(\Sigma, V, A)v = 0
$$

(96)

for all $w \in \mathcal{W}$, $W \in H^{1/2}(\partial \Omega)$ and $v \in H$. From (93) to (95), expression (96) implies formally (87)–(89).

In contrast to the GB, in the LM approach the adjoint equation (89) and the velocity equation (88) are coupled. This coupling in the KKT system is typical in the application of Newton-type methods for PDE-constrained optimization problems. An approach for solving the KKT system (87)–(89) is to discretize the equations and apply some preconditioning strategy to solve the resulting large-scale linear system. Although this technique is efficient for some applications [7], it becomes prohibitively expensive for the problem under consideration. Therefore, for inverse reservoir modeling, Newton-type methods had been traditionally avoided until the recent applications of the representer method [17, 18]. Due to the efficiency of the representer method to solve KKT systems that arises in LM approaches for inverse problems in reservoir models, we propose its application for the solution of (87)–(89). In this technique we seek a solution of (87)–(89) that has the form

$$
\begin{bmatrix}
\Sigma \\
V \\
A
\end{bmatrix} = \sum_{l=1}^N \sum_{m=1}^{M^l} q_{l,m}^1 \begin{bmatrix}
\frac{1}{a} R_h^{l,m}_1 \\
\frac{1}{a} R_{y,v}^{l,m}_1 \\
B_A^{l,m}
\end{bmatrix}
$$

(97)
for some $\beta_l^m \in \mathbb{R}$, $R_A^{l,m} \in \mathcal{H}$, $R_V^{l,m} \in H^{1/2}(\partial \Omega)$ and $R_h^{l,m} \in \mathcal{H}$ where $l \in \{1, \ldots, N\}$ and $m \in \{1, \ldots, M_l\}$. In appendix B we present the algorithm that allows us to compute $R_A^{l,m}$, $R_V^{l,m}$ and $R_h^{l,m}$ so that (97) satisfies the KKT conditions (87)–(89). Note that the main term of our interest in (97) is the velocity $V$ which is needed for step (4) of the LM algorithm.

For a rigorous analysis of the representer method we refer the reader to [18]. One of the assumptions for the application of the representer method is that the number of measurements must be finite. This assumption after discretization implies that the number of measurements must be small compared to the dimension of $V$. Fortunately, this assumption is valid for this application where only a small number of measurements are typically available.

In the previous sections we have discussed both the GB and the LM approach to compute the velocity that we need to solve the level-set equation (7). As we indicated earlier, the solution of (7) on $[\tau, \tau + \Delta \tau]$ yields, by construction, a shape that decreases the functional in (5) for $\Delta \tau$ small. To our best knowledge, a convergence theory for algorithm 1 is an open problem. Nevertheless, our numerical experiments show that convergence of the level-set algorithm is achieved according to the discrepancy principle (47).

5. Implementation

In this section we discuss the numerical implementation of the level-set technique. We denote by $H$ the Heaviside function defined by

$$H(\tau) = \begin{cases} 0 & \text{if } \tau \leq 0, \\ 1 & \text{if } \tau > 0. \end{cases}$$

(98)

For every level-set function $\phi$, we define

$$K(\phi) = K_e + [K_i - K_e]H(-\phi),$$

(99)

$$\xi(\phi) = K_i - [K_i - K_e]H(-\phi).$$

(100)

Let $H_\epsilon$ be a continuous approximation of $H$ and let $K^\epsilon$ and $\xi^\epsilon$ be the corresponding expressions (99)–(100) after replacing $H$ with $H_\epsilon$. Furthermore, assume $\Omega$ is a rectangular domain in $\mathbb{R}^2$. Let $\mathcal{T}$ be a partition of $\Omega$ into rectangles. Consider the lowest order Raviart–Thomas (RT) spaces on $\mathcal{P}$ [9], i.e. for $P \in \mathcal{P}$

$$\mathcal{V}_h(P) = \{ w \in \mathbb{R} \},$$

(101)

$$\mathcal{U}_h(P) = \{(a_1 + a_2 x, a_3 + a_4 y) a_i \in \mathbb{R}, i \in \{1, \ldots, 4\}\},$$

(102)

$$\tilde{\mathcal{U}}_h(P) = \{ u \in \mathcal{U}_h(P)| u \cdot n|_{\partial D} = 0 \}.$$  

(103)

5.1. Implementation of the forward model

We first discuss some aspects of the implementation of the forward model (33) that we compute at every iteration level of algorithm 1. Note that some terms in (27)–(29) involve integrals on $\Omega$. This in turn is parameterized by $\phi$, the current estimate of the level-set function. Therefore, we use (98) to replace integrals on $\Omega$ by integrals on $D$ weighted by $H(-\phi)$. By using this parameterization, expression (27), for example, becomes

$$G_1 = \int_D \left[ \omega c(h_2) \frac{\partial h_1}{\partial t} + \nabla \cdot h_3 - \sum_{l=1}^N K(\phi) \omega l(h_2) \left[ P_{bh}^l - h_1 \right] \delta(x - x^l) \right] v_1 \, dx,$$

(104)
where we have used the definition of $K(\phi)$ from (99). Analogous expressions can be obtained for (28)–(32). We now replace $H(\phi)$ by $H_\epsilon(\phi)$ (equivalently $(\phi, \xi)$ by $(\phi, \xi)$ and consider the following mixed finite element approximation for the resulting expressions: find, a.e. in $(0, T)$, $(h_1(\cdot, t), h_2(\cdot, t), h_3(\cdot, t), h_4(\cdot, t)) \in V_h^2 \times \tilde{U}_h^2$ such that

\begin{align}
&\int_D \left[ \psi c(h_2) \frac{\partial h_1}{\partial t} + \nabla \cdot h_3 - \sum_{i=1}^N K^e(\phi) \frac{\partial \theta^i(h_2)}{\partial t} \left[ P_{bh} - h_1 \right] \delta(x - x^i) \right] v_1 \mathrm{d}x = 0, \\
&\int_D \left[ \psi(c_r + c_w) h_2 \frac{\partial h_1}{\partial t} + \psi \frac{\partial h_2}{\partial t} + \nabla \cdot h_4 \\
&\quad - \sum_{i=1}^N K^e(\phi) \frac{\partial \theta^i(h_2)}{\partial t} \left[ P_{bh} - h_1 \right] \delta(x - x^i) \right] v_2 \mathrm{d}x = 0, \\
&\int_D \left[ \frac{\xi^e(\phi)}{\lambda(h_2) K_i} h_1 \cdot v_3 - h_1 \nabla \cdot v_3 \right] \mathrm{d}x, \\
&\int_D \left[ h_4 - \frac{\lambda_w(h_2)}{\lambda(h_2)} h_3 \right] \cdot v_4 \mathrm{d}x = 0, \\
&\int_D [h_1(\cdot, 0) - p_0] v_5 \mathrm{d}x = 0, \\
&\int_D [h_2(\cdot, 0) - s_0] v_6 \mathrm{d}x = 0
\end{align}

for every $(v_1, v_2, v_3, v_4) \in V_h^2 \times \tilde{U}_h^2$. From the definition of the RT spaces, at each element $P$, the degrees of freedom of the $j$-coordinate of $h_i$ ($i \in \{3, 4\}$) are its values at the midpoint of the edges of $P$ in the $j$-direction. On the other hand, $h_1$ and $h_2$ are piecewise constant functions evaluated at the center of each element. Then, by choosing the appropriate quadrature rules [27], from equations (107) and (108) we can find explicit relations for the nodal values of $h_3$ and $h_4$ in terms of the nodal values of $h_2$ and $h_1$. In order to evaluate the nonlinear function $\lambda_w(h_2)/\lambda(h_2)$ of (108) at an interface of two elements, an upwind scheme is utilized. Variables $h_3$ and $h_4$ can be eliminated from (105) and (106) by using the aforementioned relation of their nodal values in terms of $h_1$ and $h_2$. Finally, a suitable quadrature rule applied to (105) and (106) yields a semi-discrete cell-center finite difference approximation for $h_1$ and $h_2$. This time-dependent system is discretized with an implicit backward Euler scheme. The resulting nonlinear system is solved with the sequential solution technique [9].

5.2. Implementation of the adjoint equations

From equations (82) and (89) we observe that the adjoint equations for the GB and the LM techniques differ only on their right-hand side. Therefore, the description of the numerical implementation of the adjoint equations for both techniques are particular cases of the following problem. Given $v \in \mathbb{R}^M$, find $A \in \mathcal{H}$ such that

\begin{equation}
L^* G_j(A, w_j) = v \cdot D_{h_j} \mathcal{M}(\Omega, h) w_j, j \in \{1, \ldots, 4\}
\end{equation}

for all $w_j \in V_j$. As before, we use the current estimate of $\phi$ to replace integrals over $\Omega$ in (77)–(80) by integrals over $D$ weighted by $H(\phi)$. In addition, we formally translate the
weak form of the time-dependent formulation into the following problem. For a.e. in \((0, T]\), find \((A_1(t), A_2(t), A_3(t), A_4(t)) \in \mathcal{H}\) such that

\[
\begin{aligned}
\int_D \left[ -\varphi \frac{\partial (c(h_2)A_1)}{\partial t} - \varphi (c_w + c_r) \frac{\partial (h_2 A_2)}{\partial t} - \nabla \cdot A_3 + K(\phi) B_1(A_1, A_2) \right] w_1 \\
= - \sum_{m=1}^{M} v_m \int_D \frac{\omega^2}{\lambda} K(h_2) \delta(t - t^l_m) \delta(x - x^l_m) w_1,
\end{aligned}
\]  

(112)

\[
\begin{aligned}
\int_D \left[ \varphi' (h_2) \frac{\partial h_1}{\partial t} A_1 + \varphi (c_r + c_w) \frac{\partial h_1}{\partial t} A_2 - \varphi \frac{\partial A_2}{\partial t} - K(\phi) \frac{1}{K_e} B_2(A_1, A_2) \\
- \frac{\lambda^2}{\lambda^2 K_e K_i} \xi(\phi) h_3 \cdot A_3 - \left[ \frac{\lambda_w}{\lambda} \right] h_3 \cdot A_4 \right] w_2 \\
= \sum_{m=1}^{M} v_m \int_D \frac{\lambda^2}{\lambda K_e} K(\phi) q^l_m \delta(t - t^l_m) \delta(x - x^l) w_2,
\end{aligned}
\]  

(113)

\[
\begin{aligned}
\int_D A_1 \cdot w_1 + \int_0^T \int_D A_2 \nabla \cdot w_2 = 0
\end{aligned}
\]  

(115)

for all \((w_1, w_2, w_3, w_4) \in V_1^2 \times U_2^2\), where \(V_1\) and \(U_2\) are defined in (20) and (23), respectively. Problems (112)–(115) are furnished with the following final time conditions:

\[
\begin{aligned}
\varphi[(c(h_2)A_1 + (c_r + c_w) h_2 A_2)]_{t=T} = 0, \\
\varphi[A_2]_{t=T} = 0.
\end{aligned}
\]  

(116)

(117)

Since \(c(h_2) > 0, h_2 > 0, \varphi > 0\), (116)–(117) implies the final time conditions for the adjoint variables \(A_2(t), T = 0 = A_1(t, T)\). We now replace \(H(\varphi)\) by \(H(\varphi)\) in (112)–(115), use (99)–(100) and discretize on the RT spaces to obtain

\[
\begin{aligned}
\int_D \left[ -\varphi \frac{\partial (c(h_2)A_1)}{\partial t} - \varphi (c_w + c_r) \frac{\partial (h_2 A_2)}{\partial t} - \nabla \cdot A_3 + K'(\phi) B_1(A_1, A_2) \right] w_1 \\
= - \sum_{m=1}^{M} v_m \int_D \frac{\omega^2}{\lambda} K(h_2) \delta(t - t^l_m) \delta(x - x^l_m) w_1,
\end{aligned}
\]  

(118)

\[
\begin{aligned}
\int_D \left[ \varphi' (h_2) \frac{\partial h_1}{\partial t} A_1 + \varphi (c_r + c_w) \frac{\partial h_1}{\partial t} A_2 - \varphi \frac{\partial A_2}{\partial t} - K'(\phi) \frac{1}{K_e} B_2(A_1, A_2) \\
- \frac{\lambda^2}{\lambda^2 K_e K_i} \xi'(\phi) h_3 \cdot A_3 - \left[ \frac{\lambda_w}{\lambda} \right] h_3 \cdot A_4 \right] w_2 \\
= \sum_{m=1}^{M} v_m \int_D \frac{\lambda^2}{\lambda K_e} K'(\phi) q^l_m \delta(t - t^l_m) \delta(x - x^l) w_2,
\end{aligned}
\]  

(119)

\[
\begin{aligned}
\int_D \left[ \nabla \cdot w_3 A_1 + \frac{1}{\lambda K_e K_i} \xi'(\phi) w_3 \cdot A_3 - \frac{\lambda_w}{\lambda} w_3 \cdot A_4 \right] = 0,
\end{aligned}
\]  

(120)
for \( w_1, w_2 \in \mathcal{V}_h \) and \( w_3, w_4 \in \tilde{\mathcal{U}}_h \). In addition, we consider the same quadrature rules that we indicated earlier \([27]\) to solve \( A_3 \) and \( A_4 \) in terms of \( A_2 \) and \( A_1 \) which in turn reduces \((118)–(120)\) to a cell-center finite-difference scheme. Then, an implicit backward Euler scheme in time is utilized and the resulting linear system for \( A_2 \) and \( A_1 \) is solved simultaneously.

5.3. Implementation of the velocity equation

Analogous to the adjoint equations, it is easy to see that the variational problem for the velocity in both the GB and the LM approaches (see equations \((86)\) and \((88)\)) are particular cases of the following problem. Given \( \tilde{\alpha} > 0 \) and \( v \in \mathbb{R}^M \), find \( V \in H^{1/2}(\partial \Omega) \)

\[
\tilde{\alpha} \langle V, W \rangle_{H^{1/2}(\partial \Omega)} = - \sum_{i=1}^{4} \int_0^T D_\Omega G_i(\Omega, h, A_i)W + v \cdot D_\Omega \mathcal{M}(\Omega, h)W \quad (121)
\]

for all \( W \in H^{1/2}(\partial \Omega) \). From \((60)\) to \((63)\), \((68)\) and \((76)\), we rewrite the previous expression as

\[
\langle V, W \rangle_{H^{1/2}(\partial \Omega)} = \int_0^T \int_{\partial \Omega} f(\sigma)W(\sigma) \, d\sigma \quad (122)
\]

where

\[
f(\sigma) \equiv - \left[ \frac{K_i - K_e}{\lambda K_i \tilde{\alpha}} \left[ \frac{1}{K_i} h_3 \cdot A_3 + B_3(A_1, A_2) - \lambda q M \sum_{m=1}^{M} v_m \delta(t - t_m) \delta(\sigma - x^i) \right] \right]. \quad (123)
\]

In order to solve \((122)\) we follow the strategy suggested in \([7]\) for the definition of the inner product in \( H^{1/2}(\partial \Omega) \). More precisely, extensions \( \tilde{W}, \tilde{V} \in H^1(D) \) (on \( D \)) of \( W \) and \( V \) are constructed and used in the following definition:

\[
\langle V, W \rangle_{H^{1/2}(\partial \Omega)} = \langle \tilde{V}, \tilde{W} \rangle_{H^1(D)} = \int_D \tilde{V} \tilde{W} \, dx + \frac{\kappa}{l^2} \int_D \nabla \tilde{V} \cdot \nabla \tilde{W} \, dx. \quad (124)
\]

In the second equality of the previous expressions, the term corresponding to the \( L^2 \) inner product of the gradient is weighted by a positive number \( \kappa / l^2 \) where \( l \) is some characteristic length and \( \kappa \) is a dimensionless positive number. We include the weighting factor \( 1/l^2 \) for dimension purposes. In addition, the objective of \( \kappa \) is to control the regularity in the velocity \( V \). Note that \( \kappa \to 0 \) will yield a velocity in \( V \in L^2(\partial \Omega) \) that does not provide satisfactory numerical results. Let \( \tilde{V} \in H^1(D) \) be the weak solution to the following problem:

\[
- \frac{\kappa}{l^2} \nabla^2 \tilde{V} + \tilde{V} = 0 \quad \text{in} \quad D - \partial \Omega, \quad (125)
\]

\[
\nabla \tilde{V} \cdot n = 0 \quad \text{in} \quad \partial D, \quad (126)
\]

\[
\frac{\kappa}{l^2} [\nabla \tilde{V} \cdot n] = f(\sigma) \quad \text{in} \quad \partial \Omega, \quad (127)
\]

where \([\cdot]\) denotes the jump on \( \partial \Omega \) and \( \kappa > 0 \) arbitrary. For every \( W \in H^{1/2}(\partial \Omega) \), let \( \tilde{W} \in H^1(D) \) be an extension of \( W \). Multiply \((125)\) by \( \tilde{W} \), integrate by parts and use conditions \((126)\) and \((127)\) to find that \( \tilde{V} \), the weak solution of \((125)–(127)\), satisfies

\[
\int_D \tilde{V} \tilde{W} \, dx + \frac{\kappa}{l^2} \int_D \nabla \tilde{V} \cdot \nabla \tilde{W} \, dx = \int_0^T \int_{\partial \Omega} f(\sigma)W(\sigma) \, d\sigma. \quad (128)
\]

Therefore, from \((124), (122)\) and the trace theorem it follows that \( V = \tilde{V}|_{\partial \Omega} \) solves the variational problem \((121)\).
Remark 5.1. Note that the previous procedure to find $V$ (i.e. solving (128)) automatically provides the extension velocity $\hat{V} \in H^1(D)$ for step (4) of algorithm 1.

For the numerical implementation of (128), we replace the boundary integral in (128) by a volume integral over $D$ weighted by $H'_\epsilon(-\phi)|\nabla \phi|$. Since $H'_\epsilon$ is an approximation to the Dirac delta centered at the boundary, an application of the coarea formula [16] allows us to formally approximate the right-hand side of (128) by the right-hand side of

$$
\int_D \hat{V} \hat{W} \, dx + \frac{k}{\rho} \int_D \nabla \hat{V} \cdot \nabla \hat{W} \, dx = \int_0^T \int_D f(\xi)H'_\epsilon(-\phi)|\nabla \phi|\hat{W} \, dx.
$$

(129)

With the numerical implementation of the adjoint and the velocity equations we finalize the discussion of the numerical implementation of the GB. For the LM technique we observe that we still need to solve (87). For this system we use the same arguments and discretizations on the RT spaces that we utilize in this section for the implementation of forward and adjoint models.

5.4. The level-set equation

Once the velocity extension $\hat{V}$ has been obtained (see remark 5.1) we numerically solve (7) with the level-set MATLAB toolbox developed by Mitchell [23]. We use a fifth-order weighted essentially non-oscillatory upwind approximation for the spatial derivative in (7) [24]. The time scheme is an explicit third order total variation diminishing Runge–Kutta technique. With this discretization scheme, the time step for the solution to the level-set equation (7) is determined by the CFL condition.

We recall that, by construction of $\hat{V}$, the solution of the level-set equation (7) on $[\tau, \tau + \Delta \tau]$ generates a shape that decreases (5) for $\Delta \tau$ small. To our best knowledge the proper selection of $\Delta \tau$ is still an open problem. Nevertheless, when using (7) in the context of geometrical inverse problems the standard approach has been to take $\Delta \tau$ according to the CFL condition, i.e.

$$
\Delta \tau \leq \frac{ch}{\sup_{x \in D} |\hat{V}(x)|}
$$

(130)

for some $c < 1$. In the previous expression $h$ denotes the mesh size. A decrease of the cost functional is usually obtained in our experiments with the LM technique by using $c = 0.9$ in (130). However, for the GB approach, the same criteria sometimes yields high oscillations in $J$. As stated in [7] the aforementioned issue can be resolved by successively bisecting $\Delta \tau$ until either a decrease of $J$ is achieved or the original $\Delta \tau$ is reduced by some prescribed factor. For our experiments, only one bisection is carried out when $J$ does not decrease with the given $\Delta \tau$. We found that additional bisections result in extremely slow convergence or even stagnation of the GB technique.

In order to initialize algorithm 1, the initial shape $\Omega(0)$ is implicitly parameterized with a signed distance function [24]. This function is the initial condition for solving (7) with the velocity $\hat{V}$ computed in the first iteration of the scheme. It is well known that flattening and/or steepening of the solution to (7) may occur. A very steep or flat level set may prevent us from computing the boundary of $\partial \Omega$ which is needed in the computation of (128). In order to avoid the aforementioned behavior, after some fixed number of iterations, we reinitialize the level-set function to a signed distance function of the current estimate of $\Omega$.21
6. Numerical results

6.1. Experiment I

In figure 1 (left) we present a 2D field generated stochastically by sequential indicator simulation with the geostatistical software SGEMS [26]. This type of field is typically used for the simulation of reservoir properties. For this experiment, the field in figure 1 (left) represents the absolute permeability given by expression (2) with \( K_i = 1500 \) md, \( K_e = 150 \) md and \( \Omega \) defined by the red region of figure 1 (left). The well configuration for this experiment, also shown in figure 1, consists of nine production wells (black dots) and four injection wells (white squares). This is a standard well configuration used for example in [13, 17]. Furthermore, the relative permeability curves in (8) are defined by

\[
kr_w(s) = a_w \left[ \frac{s - s_{iw}}{1 - s_{iw} - s_{or}} \right]^{\theta_w},
\]

\[
kr_o(s) = a_o \left[ \frac{1 - s - s_{or}}{1 - s_{iw} - s_{or}} \right]^{\theta_o},
\]

where the parameters \( a_w, \theta_w \) as well as other pertinent reservoir parameters are displayed in table 1. The absolute permeability field with corresponding facies of figure 1 (left) is interpolated on a \( 180 \times 180 \) grid and the forward model is solved on \([0, T]\) divided in 80 time steps. Then, a vector of exact measurements \( \mathbf{d} \) is computed from (37) where nine collection times \( \{t_{lm}^m\} \) are defined for each well \( l \in \{1, \ldots, 13\} \). Therefore, for this experiment there are \( M = 9 \times 13 = 117 \) total number of measurements. Synthetic production data \( \mathbf{d}_\eta \) are generated from \( \mathbf{d} \) by adding 1% of Gaussian noise. More precisely for \( m \in \{1, \ldots, 9\}, l \in \{1, \ldots, 13\} \) we define

\[
d_{m,l}^\eta = d_{m,l} + 0.01d_{m,l}\epsilon_{m,l}
\]

where \( \epsilon_{m,l} \) is a Guassian random variable of zero mean and unit variance. These synthetic data are used for the identification of \( \Omega \) by means of algorithm 1. In order to avoid inverse crimes, the reconstruction is conducted on a \( 60 \times 60 \) grid and the interval of time is divided into 40 time steps. For simplicity, we choose \( \mathbf{C} \) in (44) a diagonal matrix with entries defined by \( [\mathbf{C}]_{m,m} = \min(0.01d_{m,l}^\eta)^2, 150 \) bbl/day\(^2\)\). The selection of matrix \( \mathbf{C} \) should reflect prior knowledge of the measurement error typically available from measurement devices. In this
Table 1. Reservoir description.

<table>
<thead>
<tr>
<th>Experiment</th>
<th>I and II</th>
<th>III</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dimension (m$^2$)</td>
<td>$900 \times 900 \times 1$</td>
<td>$900 \times 900 \times 1$</td>
</tr>
<tr>
<td>Grid blocks (synthetic data)</td>
<td>$180 \times 180 \times 1$</td>
<td>$180 \times 180 \times 1$</td>
</tr>
<tr>
<td>Grid blocks (inversion)</td>
<td>$60 \times 60 \times 1$</td>
<td>$60 \times 60 \times 1$</td>
</tr>
<tr>
<td>Time step size (days) (synthetic data)</td>
<td>25</td>
<td>25</td>
</tr>
<tr>
<td>Time step size (days) (inversion)</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>Time interval $[0, T]$ (days)</td>
<td>$[0, 2000]$</td>
<td>$[0, 3000]$</td>
</tr>
<tr>
<td>Initial pressure$^a$, $P_0$ (Pa)</td>
<td>$2.0 \times 10^7$</td>
<td>$1.5 \times 10^7$</td>
</tr>
<tr>
<td>Initial saturation$^a$, $S_0$</td>
<td>0.2</td>
<td>0.2</td>
</tr>
<tr>
<td>BHP at injection wells$^b$, $P_{bh,I}(t)$ (Pa)</td>
<td>$2.45 \times 10^7$</td>
<td>$1.7 \times 10^7$</td>
</tr>
<tr>
<td>BHP at production wells$^b$, $P_{bh,P}(t)$ (Pa)</td>
<td>$2.25 \times 10^7$</td>
<td>$1.6 \times 10^7$</td>
</tr>
<tr>
<td>Boundary conditions$^c$, $B(x, t)$</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$c_w$ (Pa$^{-1}$)</td>
<td>$4.5 \times 10^{-10}$</td>
<td>$4.5 \times 10^{-10}$</td>
</tr>
<tr>
<td>$c_0$ (Pa$^{-1}$)</td>
<td>$1.34 \times 10^{-9}$</td>
<td>$1.34 \times 10^{-9}$</td>
</tr>
<tr>
<td>$c_r$ (Pa$^{-1}$)</td>
<td>$6 \times 10^{-10}$</td>
<td>$6 \times 10^{-10}$</td>
</tr>
<tr>
<td>$\mu_w$ (Pa s)</td>
<td>$0.5 \times 10^{-3}$</td>
<td>$0.5 \times 10^{-3}$</td>
</tr>
<tr>
<td>$\mu_o$ (Pa s)</td>
<td>$3 \times 10^{-3}$</td>
<td>$3 \times 10^{-3}$</td>
</tr>
<tr>
<td>$K_i$ (millidarcies, md)</td>
<td>1500</td>
<td>2500</td>
</tr>
<tr>
<td>$K_r$ (millidarcies, md)</td>
<td>150</td>
<td>100</td>
</tr>
<tr>
<td>$s_{uw}$</td>
<td>0.2</td>
<td>0.2</td>
</tr>
<tr>
<td>$s_{or}$</td>
<td>0.2</td>
<td>0.2</td>
</tr>
<tr>
<td>$a_w$</td>
<td>0.4</td>
<td>0.4</td>
</tr>
<tr>
<td>$a_o$</td>
<td>0.9</td>
<td>0.9</td>
</tr>
<tr>
<td>$\theta_w$</td>
<td>2.5</td>
<td>2.5</td>
</tr>
<tr>
<td>$\theta_o$</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>$l$ (m)</td>
<td>900</td>
<td>900</td>
</tr>
</tbody>
</table>

$^a$ Constant on $\Omega_1$.
$^b$ Constant on $[0, T]$.
$^c$ Constant on $\Omega_T$.

In some cases, this variance may take small values resulting in scaling issues detrimental to the convergence rate of the present algorithms. For this reason a lower bound is specified in the previous definition of $[C^l]_{m,m}$. For our numerical experiments, the value selected above provides a realistic and convenient lower bound for the variance of the flow rate measurement error.

For the approximation of the Heaviside function (see section 5) we define

$$H_{\epsilon} = \begin{cases} 
0 & \text{if } \tau < -\epsilon, \\
1 + \frac{\tau}{\epsilon} & \text{if } \tau \in [-\epsilon, 0], \\
1 & \text{if } \tau > 0,
\end{cases}$$

with $\epsilon = 1.5h$ and $h$ the mesh size. The red region of figure 1 (middle) defines the shape that we utilize to initialize our scheme. Note that the (facies) permeability values at the wells are the same for both the initial figure 1 (middle) and the true region figure 1 (left). Assuming that the permeability and therefore the lithofacies are known at the reservoir well is usually a
valid assumption that has been used in [13] for the generation of initial guesses like the one we used for this experiment.

In figure 2 we show the final facies configuration obtained with the GB approach for different values of $\kappa$ (see equation (124)). The reconstructions for $\kappa = 1, \kappa = 10^{-1}, \kappa = 10^{-2}$, and the true permeability are presented (from left to right) in the top row of figure 2. The bottom row of figure 2 (left) shows the log10 of the data misfit $\|d - M(h, \Omega)\|$. In figure 2 (bottom-right) we also present the L1-error of the reconstructed region given by

$$
\|\chi/\Omega^* - \chi/\Omega\|_{L_1(D)} \equiv \int_D |\chi/\Omega^* - \chi/\Omega| \, dx.
$$

From figure 2 (bottom-right) we observe that for all values of $\kappa$, the L1-error decreases to approximately the same value ($\|\chi/\Omega^* - \chi/\Omega\|_{L_1(D)} \approx 0.2$). Note that the cost functional figure 2 (bottom-left) also decreases for every $\kappa$. Moreover, slower convergence is obtained for smaller values of $\kappa$. In addition, convergence is achieved in the sense of the discrepancy principle (47) for values $\tau = 3.75$ ($\kappa = 1$), $\tau = 3.0$ ($\kappa = 10^{-1}$), $\tau = 2.6$ ($\kappa = 10^{-2}$) and $\tau = 2.5$ ($\kappa = 10^{-3}$). We recall that $\tau \approx 2$ has been observed to produce reasonable results in iterative regularization algorithms where the discrepancy principle (47) is used to terminate the scheme [14]. Therefore, for this experiment we observe that convergence for smaller $\kappa$’s results in smaller values $\tau$ that apparently approach the aforementioned reported value $\tau \approx 2$.

From (125) to (127) it is easy to see that the regularity in $V$ increases with $\kappa$. On the other hand, from expression (7) and $\partial \Omega = \{x \in D | \phi(x) = 0\}$, it follows that the regularity of $V$ determines the regularity of the boundary of $\Omega$. Indeed, we observe from figure 2 that $\partial \Omega$ for the reconstructed (red) region seems to be overly smooth for large values of $\kappa (\kappa \geq 1)$. On the other hand, for small values of $\kappa$ (e.g., $\kappa = 10^{-3}$) the boundary becomes rougher. However, if a value $\kappa < 10^{-3}$ is chosen in (124), the resulting velocity field (129) produces a
non-smooth level-set function from which the boundary of \( \Omega \) cannot be computed. From this first set of experiments we see that \( \kappa = 10^{-2} \) produces a reasonable reconstruction after 180 iterations. For this case, in the first row of figure 5 we show the configuration of the inverted region at iterations 8, 20, 40, 70 and 160. Furthermore, the computational cost of the GB algorithm for the reconstruction of \( \Omega \) for \( \kappa = 10^{-2} \) was approximately of 500 forward model (FM) runs.

In the second part of experiment I, we repeat the same exercise as before except that we use the LM approach. In addition to \( \kappa \) in expression (124), the regularity of \( V \) in the LM approach also depends on the Tikhonov parameter \( \alpha \) in (88). In figure 3 we show the results of the first part of this experiment. In this figure (top row) we observe the reconstruction for \( \kappa = 10^{-1} \) fixed and different values of \( \alpha \) (from left to right): \( \alpha = 10^{-4} \), \( \alpha = 10^{-5} \), \( \alpha = 10^{-6} \) and \( \alpha = 10^{-7} \). In the bottom row of figure 3 we also show the corresponding data misfit (left) and the L1-error (right) as defined in (135). Note that convergence is slower for smaller values of the Tikhonov parameter \( \alpha \) (\( \alpha \leq 10^{-6} \)). In this case, however, smaller L1-error is accomplished and a less regular boundary of \( \Omega \) is obtained. In contrast, overly smooth boundary of the reconstructed field is obtained for large values of \( \alpha \) (\( \alpha = 10^{-3} \)). In figure 4 we present the results for \( \alpha = 10^{-6} \) fixed and (from left to right) \( \kappa = 1 \), \( \kappa = 10^{-1} \), \( \kappa = 10^{-2} \) and \( \kappa = 10^{-3} \). In this case the combination (\( \alpha = 10^{-6} \), \( \kappa = 10^{-3} \)) seems to provide the minimum regularity on \( V \) for which the reconstruction can be numerically obtained. While large values of \( \kappa \) provide overly smooth reconstruction (like with the GB approach), smaller \( \kappa \)'s are detrimental to the boundary regularity as well as the rate of convergence. For this experiment, the combination (\( \alpha = 10^{-6} \), \( \kappa = 10^{-2} \)) provides reasonable outcomes in terms of both (135) and computational effort. More precisely, convergence is achieved after 50 iterations with a computational cost of 600 FM model runs. Therefore, we observe that the identification of facies in this particular experiment (\( \alpha = 10^{-6} \), \( \kappa = 10^{-2} \) with LM) is computationally more
expensive than conducting the identification with the analogous ($\kappa = 10^{-2}$) GB approach (whose cost corresponds to 500 FM runs). Note, however, that both approaches provide about the same accuracy of the reconstructed shape (in the norm (135)). Although for this experiment our LM implementation was less efficient than the GB technique, in the following section we will observe that this result is reversed. More precisely, our LM implementation outperforms the GB technique when we initialize both algorithms with a shape that differs more substantially from the true one.

6.2. Experiment II

In experiment II we repeat the experiments from experiment I except that the initial shape is given by the high permeability region of figure 1 (right). Unlike the initial guess for the previous experiments, the permeability values at some wells of figure 1 (right) do not coincide with those in figure 1 (left). Although the assumption of knowing the facies at the reservoir wells is valid, we consider relevant to test the methodology under more demanding situations. In figure 6 we present the results for experiment II with the GB algorithm. Several values of $\kappa$ in (124) were tested. Note that convergence is slower than that achieved in experiment I (with the GB approach). Moreover, for values $\kappa \geq 1 \times 10^{-1}$, the algorithm stagnates at the suboptimal field shown in figure 6 (top-left). For small values of $\kappa$ ($\kappa \approx 2.5 \times 10^{-3}$) we can further decrease the limit value of the data misfit figure 6 (bottom-left). Note that this decrease does not improve the accuracy of the reconstruction in terms of the L1-error figure 6 (bottom-right). However, the small value $\kappa \approx 2.5 \times 10^{-3}$ produces a splitting of the red region in the lower-left corner of the reservoir. It seems that, for that small $\kappa$, the velocity (see expression (124)) develops singularities apparently needed to move and split the high permeability region while decreasing the cost.
Figure 5. Facies configuration. First row: experiment I GB ($\kappa = 10^{-2}$) at iterations (from left to right) 8, 20, 40, 70 and 160. Second row: experiment I LM ($\kappa = 10^{-2}$, $\alpha = 10^{-6}$) at iterations (from left to right) 5, 15, 25, 35 and 45. Third row: experiment II GB ($\kappa = 2.5 \times 10^{-3}$) at iterations (from left to right): 30, 80, 250, 500 and 800. Fourth row: experiment II LM ($\kappa = 5 \times 10^{-3}$, $\alpha = 10^{-6}$) at iterations (from left to right) 15, 30, 80, 100 and 115.

In the third row of figure 5 we show the evolution of the reconstructed facies (for $\kappa = 2.5 \times 10^{-3}$) at iterations 30, 80, 250, 500 and 800. In this figure, the extremely slow convergence of the GB approach can be observed. Note that from the 250th to the 500th iteration, a very small change in the identified region was obtained. Finally, we remark that convergence for $\kappa = 2.5 \times 10^{-3}$ with the GB approach was achieved after 1100 iterations with a computational cost of approximately 3100 FM runs.

In figure 7 we present the results obtained with the LM for the reconstruction of figure 1 (left) with the initial guess of figure 1 (right). In figure 7 we show the identifications for fixed $\alpha = 10^{-6}$ and (from left to right) $\kappa = 1, \kappa = 10^{-1}, \kappa = 10^{-2}$ and $\kappa = 5 \times 10^{-3}$. Analogous to the GB approach, the velocity obtained with large values of $\kappa$ (in this case $\kappa = 1$) does not split the facies at the lower-left corner of the reservoir. Even though the $L1$-error is still reasonable compared to other estimates, the facies is not reconstructed at the lower-left corner where a production well is located. This is responsible for the stagnation at a high value of the cost functional. For the smallest value $\kappa = 5 \times 10^{-3}$, convergence was achieved after 120 iterations. The fourth row of figure 5 shows some intermediate iterations. It is important to mention that the computational cost of the LM approach for the case $\kappa = 5 \times 10^{-3}$ and $\alpha = 10^{-6}$ was
Figure 6. Experiment II GB. Top row: identified facies for (from left to right) $\kappa = 1 \times 10^{-1}$, $\kappa = 1 \times 10^{-2}$, $\kappa = 2.5 \times 10^{-3}$. Top-right: true facies. Bottom: data misfit (left) and L1-error (right).

Figure 7. Experiment II LM ($\alpha = 10^{-6}$). Top row: identified facies for (from left to right) $\kappa = 1$, $\kappa = 10^{-1}$, $\kappa = 10^{-2}$, $\kappa = 5 \times 10^{-3}$ and true facies (right). Bottom row: Data misfit (left) and L1-error (right).
1500 FM runs. In this case, clearly the efficiency of the LM approach outperformed the GB technique (3100 FM).

6.3. Experiment III

From experiments I and II we learn that it is possible to obtain estimates of facies that capture the main features of the true facies distribution. However, we recall that the inverse problem under consideration is severely ill-posed. In particular, due to non-uniqueness, there may be several reservoir configurations for which the level-set technique may result in substantially different facies estimates than the one we used for the generation of synthetic data. Nevertheless, since our main focus is the evaluation of the level-set techniques for facies identification, we utilize reservoir configurations that favor uniqueness of the reconstruction. This is the case of experiment III where we consider a true shape defined by the red region of figure 8 (top-left). In this figure we also show the well configuration that consists of five production wells (white squares) and ten production wells (black dots). The first goal of this experiment is to simulate the properties of fluvial reservoirs where channelized structures are typically observed. The second goal is to test the level technique for splitting shapes in order to decrease the cost functional. Synthetic data are generated with the same procedure of the previous experiments. Inverse crimes are avoided as before. The initial shape is the red disk presented in the second column of figure 8 (top row). After conducting the reconstruction with both the GB and LM approaches, reasonable reconstructions were obtained for $\kappa = 5 \times 10^{-2}$ (GB) and $\alpha = 1 \times 10^{-5}$ and $\kappa = 5 \times 10^{-2}$ (LM). These reconstructions are shown in the third and fourth column of figure 8, respectively. Moreover, convergence was achieved according to the discrepancy principle with $\tau = 1.5$ for both schemes and their performance can be observed in figure 8 (bottom row). Although the reconstructions obtained with both approaches provide the same
accuracy (see figure 8, bottom-right), the LM approach was computationally more efficient. More precisely, convergence of the LM scheme was achieved after 90 iterations equivalent to 1100 FM runs. In contrast, the GB approach took about 600 iterations to converge with an associated computational cost of 1700 FM runs. Some of the intermediate iterations can be observed in figure 9.

In this experiment we also analyze the convergence of the LM technique for the reconstruction of figure 8 (top-left) from synthetic data with noise levels of 35%, 25%, 10%, 5% and 1%. These results are shown in figure 10. The objective function for each error level converges to a value that satisfies the equality in (47) for \( \tau = 0.98 \) (for 35% and 25%), \( \tau = 0.88 \) (for 10%), \( \tau = 0.92 \) (for 5%) and \( \tau = 1.5 \) (for 1%). In contrast to experiment I (where \( \tau > 2 \)), this result indicates that the stopping criterion (47) seems to provide stable solutions for \( \tau \approx 1 \). These are still reasonable values consistent with the standard assumption (\( \tau > 1 \)) of iterative regularization techniques [14]. Note that the splitting of the initial shape cannot be performed when the data are contaminated with large errors (\( \geq 25\% \)).

From the results of this experiment we observe that both techniques provide an excellent reconstruction of facies from accurate production data. However, as we indicated earlier, the well configuration of this experiment has been chosen to reduce the lack of uniqueness in the identification of facies. Indeed, note from expressions (129) and (123) that, at each iteration, the main contribution to the velocity comes from the term \( h_3 \cdot A_3 \) at points close to \( \partial \Omega_1 \). The adjoint flux \( A_3 \) depends on the misfit between predictions and observations. Then, \( A_3 \) takes maximum values only at points close to the reservoir wells where measurements are collected. Similarly, the term \( h_3 \) corresponds to the current estimate of the reservoir flux. Although this flux also takes large values at the well locations, the well configuration is such that, even if the current estimate of \( \partial \Omega \) is not close to the measurement wells, the \( x \)-component of \( h_3 \) is sufficiently large to generate a velocity that moves \( \partial \Omega \) in a direction that decreases the cost functional. It is worth mentioning that, if for example the true facies were channels parallel to those of figure 8 (top-left), both level-set approaches stagnate at suboptimal shapes. Furthermore, note that adding more measurement locations also increases the number of points in \( D \) where the adjoint flux \( A_3 \) may contribute to the velocity. Therefore, having more measurement locations may be beneficial to the identification of facies with level sets. In
a secondary oil recovery process, drilling more wells to increase the number of (production data) measurement locations may not be practical. However, incorporating other types of measurements related to the state of the reservoir may be possible by means of seismic and/or geodetic data.

7. Conclusions

We have addressed the problem of facies identification by applying gradient-based and Newton-type level-set techniques. These techniques are developed according to the functional analytic framework of [6, 7] where the shape derivatives of the forward model determine the velocity in the level-set equation. By using a weak forward model consistent with our numerical discretization, we formally compute the shape derivatives from the standard shape analysis.

From our experiments we observe that the iterative level-set framework, applied to the identification of facies, inherits convergence and regularizing properties of standard iterative regularization techniques [14]. In particular, we observe that the squared data misfit decreases until it reaches and oscillates around a value that satisfies the discrepancy principles for typical values of $\tau$. In addition, as in the standard case, convergence of the LM technique is achieved in fewer iterations than the GB approach.

The accuracy and efficiency of both the LM and the GB implementations depend on the proper selection of the parameters $\kappa$ (for GB and LM) and $\alpha$ (for LM) that control the regularity of the velocity computed at each iteration of the scheme. Our numerical experiments validate our theoretical choice $V \in H^{1/2}(\partial\Omega)$ in the level-set framework. Indeed, for values $\kappa$ in (124) less than some lower bound, the velocity computed with the level-set equation does not possess the minimum regularity to obtain (via the level-set equation) a new shape whose boundary $\partial\Omega$ can be computed numerically. On the other hand, large values of $\kappa$ and $\alpha$ (for LM) are detrimental to performance in terms of number of iterations and accuracy. To our
best knowledge, the optimal choice of these parameters is still an interesting open problem that may significantly improve the computational efficiency of the proposed implementations.

When the algorithm is initialized with a shape that has the ‘true’ facies at wells, both techniques exhibit fast convergence. Moreover, the identified facies distribution has the same accuracy for both the GB and the LM technique. In this case, however, the GB technique is more efficient than our representer-based LM approach in terms of computational cost. While substantial computational savings of the LM can be obtained from a parallel implementation of the represent algorithm 2 (appendix B), our serial representer-based implementation of the LM approach was computationally more efficient than the GB technique for those experiments where the initial shape does not have the true facies at the reservoir wells.

The results of this paper show that the level-set technique provides an efficient framework to compute reasonable estimates of geologic facies from production data. It is important, however, to remark that alleviating the ill-posedness of the identification problem (e.g. by adding some other source of data) is essential to the development of a robust level-set framework that can potentially become a tool in petroleum applications. Finally, we emphasize that our framework can be extended for the solution of more complex problems. In particular, analogous approaches to those presented in [4, 13, 21] can be followed to include the identification of permeability values at each facies, as well as the identification of multiple facies.

Acknowledgment

Support for this work was provided by ENI S.p.A. as part of the project ‘Integrated Multiscale Flow Physics, Characterization, and Control’.

Appendix A. Proof of lemma 3.1.

From definitions (54) to (59) we have,

\[
D_h G_1(\Omega, h, v_1) \Sigma_1 = \int_D \left[ \psi(h_2) \frac{\partial \Sigma_1}{\partial t} + \sum_{l=1}^{N} \alpha_l \lambda(h_2) \Sigma_1 K_l \delta(x - x^l) \right] v_1 \\
+ \int_{\Omega} \sum_{l=1}^{N} \left[ K_j - K_e \right] \alpha_l \lambda \Sigma_1 K_e \delta(x - x^l) v_1 \, dx \, dt
\]  
(A.1)

\[
D_h G_2(\Omega, h, v_2) \Sigma_1 = \int_D \left[ \psi(c_r + c_w) h_2 \frac{\partial \Sigma_1}{\partial t} + \sum_{l=1}^{N} K_e \omega^l \theta^l \Sigma_1 \delta(x - x^l) \right] v_2 \\
+ \int_{\Omega} \sum_{l=1}^{N} \left[ K_j - K_e \right] K_e \omega^l \theta^l \Sigma_1 v_2 \delta(x - x^l)
\]  
(A.2)

\[
D_h G_3(\Omega, h, v_3) \Sigma_1 \equiv -\int_D \Sigma_1 \cdot \nabla v_3
\]  
(A.3)

\[
D_h G_5(\Omega, h, v_5) \Sigma_1 \equiv \int_D \Sigma_1(\cdot, 0) v_5
\]  
(A.4)

\[
D_h G_i(\Omega, h, v_i) \Sigma_1 \equiv 0, \text{ } i \in \{4, 6\}
\]  
(A.5)
Therefore,
\[
\sum_{i=1}^{6} \int_{0}^{T} D_{i} G_{i}(\Omega, \mathbf{h}, v_{i}) \mathbf{\Sigma}_{1} = \int_{0}^{T} \int_{D} \left[ \varphi c(h_{2}) \frac{\partial \mathbf{\Sigma}_{1}}{\partial t} v_{1} + \varphi(c_{r} + c_{w}) h_{2} \frac{\partial \mathbf{\Sigma}_{1}}{\partial t} v_{2} + K_{r} B_{1}(v_{1}, v_{2}) \mathbf{\Sigma}_{1} \right] + \int_{0}^{T} \int_{\Omega} K_{r} B_{1}(v_{1}, v_{2}) \mathbf{\Sigma}_{1} \, dx \, dt
\]
(A.6)

where we have used (74)–(76). Let us now assume that \( v_{1}, v_{2} \in \mathcal{X} \) (see expression (19)). Then, since we are also assuming \( \Sigma_{1}, h_{2} \in \mathcal{X} \), the product rule [16] (theorem 4) implies that \( c(h_{2}) v_{1}, h_{2} v_{2} \in \mathcal{X} \). Therefore, from the Sobolev embedding \( H^{1}([0, T]; L^{2}(D)) \hookrightarrow C([0, T]; L^{2}(D)) \), the first and the second terms in the integral above can be integrated by parts with respect to time to obtain
\[
\sum_{i=1}^{6} \int_{0}^{T} D_{i} G_{i}(\Omega, \mathbf{h}, v_{i}) \mathbf{\Sigma}_{1} = \int_{0}^{T} \int_{D} \left[ -\varphi \frac{\partial c(h_{2}) v_{1}}{\partial t} \mathbf{\Sigma}_{1} - \varphi(c_{r} + c_{w}) \frac{\partial (h_{2} v_{2})}{\partial t} \mathbf{\Sigma}_{1} v_{2} + K_{r} B_{1}(v_{1}, v_{2}) \mathbf{\Sigma}_{1} \right] + \int_{0}^{T} \int_{\Omega} K_{r} B_{1}(v_{1}, v_{2}) \mathbf{\Sigma}_{1} \, dx \, dt
\]
(A.7)

where in the last equality we have used (77). Therefore, for \( \Sigma_{1} \in \mathcal{W}_{1}, \Sigma_{1}(\cdot, 0) = 0 \) a.e. in \( \Omega \) and therefore
\[
\sum_{i=1}^{6} \int_{0}^{T} D_{i} G_{i}(\Omega, \mathbf{h}, v_{i}) \mathbf{\Sigma}_{1} = L^{*} G_{1}(\mathbf{v}, \Sigma_{1}).
\]
(A.8)

We recall that this expression is valid for \( v_{1}, v_{2} \in \mathcal{X} \) and \( \Sigma_{1} \in \mathcal{W}_{1} \). With similar arguments we can show that
\[
\sum_{i=1}^{6} \int_{0}^{T} D_{i} G_{i}(\Omega, \mathbf{h}, v_{i}) \mathbf{\Sigma}_{j} = L^{*} G_{1}(\mathbf{v}, \Sigma_{j}) \quad j \in \{1, 2, 3, 4\}
\]
(A.9)

for all \( \mathbf{v} \in \mathcal{H} \) and \( \Sigma_{j} \in \mathcal{W}_{j} \). Expression (A.9) implies trivially (81).

\[\square\]

Appendix B. The representer algorithm for the solution of the KKT system

We now present the algorithm for the computation of \( \beta_{m}^{i}, R_{h}^{i,m}, R_{V}^{i,m} \) and \( R_{A}^{i,m} \) so that \( (\Sigma, V, A) \) defined by (97) satisfies the KKT system (87)–(89).

Algorithm 2 [Representer’s algorithm]. Given \( \Omega(\tau) \) and \( h(\Omega) \) the solution to (33)
(1) For every \( l \in \{1, \ldots, N\} \) and \( m \in \{1, \ldots, M^l\} \), [1.1] Compute \( \mathbf{R}^l,m_{\lambda} \in \mathcal{H} \) by solving
\[
L^*G_i(\mathbf{R}^l,m_{\lambda}, \mathbf{w}_i) = D_hM^l_{m}(\Omega, \mathbf{h})\mathbf{w}_i
\]
for all \( \mathbf{w}_i \in \mathcal{W}_i \), [1.2] Find \( R^l,m_{\nu} \) from
\[
\langle R^l,m_{\nu}, \mathcal{W} \rangle_{H^1/2(\partial \Omega)} = -\frac{4}{\alpha} \sum_{i=1}^{4} \int_0^T D_\Omega G_i(\Omega, \mathbf{h}, R^{l,m}_{A,i})\mathbf{V} + D_\Omega M^l_{m}(\Omega, \mathbf{h})\mathbf{W}
\]
for all \( \mathbf{W} \in H^{1/2}(\partial \Omega) \), [1.3] Find \( R^l,m_{h} \) by solving
\[
D_hG_i(\Omega, \mathbf{h}, \mathbf{v}_i) R^l,m_{h} + D_\Omega G_i(\Omega, \mathbf{h}, \mathbf{v}_i) R^l,m_{\nu} = 0
\]
for all \( \mathbf{v}_i \in \mathcal{V}_i \), [1.4] For \( n \in \{1, \ldots, N\} \) and \( k \in \{1, \ldots, M^n\} \), compute the entries
\[
[R^{l,n}_{j,k}] = D_h M^n_{j}(\Omega, \mathbf{h})R^{l,n}_{h} + D_\Omega M^n_{j}(\Omega, \mathbf{h})R^{l,n}_{\nu}
\]
of the matrix \( R^{l,n} \in \mathbb{R}^{M_l \times M_n} \).

(2) Define
\[
\mathbf{R} = \begin{bmatrix} \mathbf{R}^1 \\ \vdots \\ \mathbf{R}^N \end{bmatrix}, \quad \mathbf{R}' = \begin{bmatrix} \mathbf{R}^1 & \cdots & \mathbf{R}^N \end{bmatrix},
\]
and solve
\[
[\mathbf{R} + \alpha \mathbf{C}]^{-1} \beta = \mathbf{d} - \mathbf{M}(\Omega, \mathbf{h})
\]
where \( \mathbf{C} \) is defined in (44).

(3) Compute \( \mathbf{A} \) from (B.12).

(4) Compute \( \mathbf{V} \) from (B.11).

Let \((\Sigma, \mathcal{V}, \mathbf{A})\) be defined by (97) with \( \beta \) computed from (B.6) and for \( l \in \{1, \ldots, N\} \) and \( m \in \{1, \ldots, M^l\} \), \( R^l,m_{\lambda} \), \( R^l,m_{\nu} \) and \( R^l,m_{h} \) are computed as in step (1) of algorithm 2. We now show that (97) solves the KKT system. From (B.6), (B.5) and (44), it follows that
\[
C_l \beta^l \equiv \mathbf{d}' - \mathbf{M}'^l(\Omega, \mathbf{h}) - \frac{1}{\alpha} \mathbf{R}' \beta
\]
which in turn, from (B.4), implies
\[
[C_l \beta^l]_j = \mathbf{d}'_j - \mathbf{M}'^l(\Omega, \mathbf{h}) - \frac{1}{\alpha} \sum_{n=1}^{N} \sum_{k=1}^{M^n} \beta^l_n [D_h M^n_{j}(\Omega, \mathbf{h})R^{n,k}_{h} + D_\Omega M^n_{j}(\Omega, \mathbf{h})R^{n,k}_{\nu}].
\]

From the linearity of the derivatives of the measurement functionals, the previous expression becomes
\[
\beta \equiv [\mathbf{d} - \mathbf{M}(\Omega, \mathbf{h}) - D_h \mathbf{M}(\Omega, \mathbf{h})\Sigma - D_\Omega \mathbf{M}(\Omega, \mathbf{h})\mathbf{V}]\mathbf{C}^{-1}
\]
with \( \Sigma \) and \( \mathbf{V} \) defined in (97). On the other hand, we now multiply equations (B.1)–(B.3) by \( \beta^l_m \in \mathbb{R} \). Then, we sum the resulting expressions for \( m \in \{1, \ldots, M^l\} \) and then we sum over \( l \in \{1, \ldots, N\} \). Finally we use the linearity and definition (97) to find
\[
D_h G_i(\Omega, \mathbf{h}, \mathbf{v}_i) \Sigma + D_\Omega G_i(\Omega, \mathbf{h}, \mathbf{v}_i) \mathbf{V} = 0, \ i = 1, \ldots, 4
\]
\[ \alpha(V, W)_{H^{1/2}(\partial \Omega)} = -\frac{4}{\sum_{i=1}^{4}} \int_{0}^{T} D \Omega G_i(\Omega, h, A_i)V + \beta \cdot D \Omega \mathcal{M}(\Omega, h) W, \]  
(B.11)

\[ L^* G_i(A, w_i) = \beta \cdot D h_i \mathcal{M}(\Omega, h) w_i, i = 1, \ldots, 4, \]  
(B.12)

for all \( v_i \in V_i \) \( (i = 1, \ldots, 4) \), \( W \in H^{1/2}(\partial \Omega) \) and \( w_i \in W_i \). We now compare (B.9)–(B.12) with (87)–(89) to see that (97) solves the KKT system. Note that we do not need to store \( [R_l^m, R_l^m, R_l^m] \) in step (1). Moreover, the computation of each representer \( [R_l^m, R_l^m, R_l^m] \) in step (1) is independent of each other. Therefore, this step of the representer algorithm is embarrassingly parallelizable [3].

References


[29] Yotov I 1996 Mixed finite element methods for flow in porous media PhD Thesis TR96-09, Department of Computational and Applied Mathematics, Rice University, Houston, TX