Comput Geosci (2009) 13:317–329
DOI 10.1007/s10596-008-9120-z

ORIGINAL PAPER

Identification of diffusion parameters in a nonlinear convection–diffusion equation using the augmented Lagrangian method

T. K. Nilssen · K. H. Karlsen · T. Mannseth · X.-C. Tai

Received: 10 November 2008 / Accepted: 10 November 2008 / Published online: 3 December 2008
© Springer Science + Business Media B.V. 2008

Abstract Numerical identification of diffusion parameters in a nonlinear convection–diffusion equation is studied. This partial differential equation arises as the saturation equation in the fractional flow formulation of the two-phase porous media flow equations. The forward problem is discretized with the finite difference method, and the identification problem is formulated as a constrained minimization problem. We utilize the augmented Lagrangian method and transform the minimization problem into a coupled system of nonlinear algebraic equations, which is solved efficiently with the nonlinear conjugate gradient method. Numerical experiments are presented and discussed.

Keywords Parameter estimationa · Inverse problems · Nonlinear convection–diffusion equation · Augmented Lagrangian methods · Porous media flow · Permeability

This work was partially supported by the Research Council of Norway (NFR), under grant 128224/431.

T. K. Nilssen (✉)
Department of Economics and Business Administration, University of Agder, Servicebox 422, 4604 Kristiansand, Norway
e-mail: trygve@rizklab.no

K. H. Karlsen
Centre of Mathematics for Applications, University of Oslo, P.O. Box 1053, Blindern, 0316 Oslo, Norway
e-mail: kennethk@math.uio.no

T. Mannseth
Centre for Integrated Petroleum Research, P. O. Box 7800, 5020 Bergen, Norway
e-mail: Trond.Mannseth@cipr.uib.no

T. Mannseth · X.-C. Tai
Department of Mathematics, University of Bergen, Johannes Brunsgt. 12, 5008 Bergen, Norway

X.-C. Tai
e-mail: Xue-Cheng.Tai@mi.uib.no

1 Introduction

In this paper, we investigate the estimation of diffusion coefficient $q(x)$ in the following two-dimensional equation

$$u_t + \frac{\partial}{\partial x} f(u) + \frac{\partial}{\partial y} g(u) - \nabla \cdot (q(x)N(u)\nabla u) = s(x,t)$$

$$\text{in } \Omega \times (0, T), \tag{1.1}$$

with the initial-boundary conditions

$$u(x, 0) = u_0(x) \text{ in } \Omega, \quad \text{and} \quad u(x,t) = 0 \text{ on } \partial\Omega \times (0, T).$$

The nonlinear functions f and g are S-shaped flux functions of Buckley–Leverett type, in the x and y directions respectively. The nonlinearity in the diffusion term, $N(u)$, is a positive function. For simplicity, we assume that Ω is the unit square. $s(x,t)$ is a given source term, which is piecewise smooth. We will use the following notation for the flux term

$$\nabla \cdot (f, g) = \frac{\partial}{\partial x} f(u) + \frac{\partial}{\partial y} g(u).$$

Equation 1.1 is related to two-phase porous media flow. The immiscible displacement of oil by water in a porous medium without gravity effects can be described by the following set of partial differential equations (see, e.g., [7]):

$$\nabla \cdot \mathbf{V} = f_1(x, t) \tag{1.2}$$

$$\mathbf{V} = -q(x)\lambda(x, u)(\nabla \overline{p} - \rho(u)\nabla h) \tag{1.3}$$

$$\phi(x)u_t + \nabla \cdot (f(u)\mathbf{V} + f_g(u)q(x)\nabla h) - \nabla \cdot (q(x)N(u)\nabla u) = f_2(x, t), \tag{1.4}$$

where f_1 and f_2 denote injection/production wells, $q(x)$ is the permeability, $\phi(x)$ is the porosity, $\mathbf{V}$ is the total Darcy velocity, u is the saturation of the wetting phase, $\lambda(x, u)$ denotes the total mobility of the phases, $\overline{p}$ is the global pressure, h is the height, $\rho(u)$ is the density of the wetting phase, and $N(u)$ is a given nonlinear diffusion function. Equation 1.4 is the fractional flow formulation of the mass balance equation for water, and $f(u)$ is a nonlinear fractional flow function, which is typically S-shaped. Further, we have

$$f_g(u) = (\rho_w - \rho_o) f(u)\lambda_o,$$

where ρ_w and ρ_o are the densities of the wetting and nonwetting phases, respectively, and λ_o is the phase mobility of the nonwetting phase.

Equation 1.1 is similar to Eq. 1.4 except for the convection term and the time derivative term. The time derivative terms are equal if we assume that $\phi(x) = 1$. The difference in the convection terms is that Eq. 1.1 does has no permeability dependence and no varying coefficient.

For realistic simulations of two-phase porous media flow, the values of $N(u)$ may attain values close to zero, making Eq. 1.4 a degenerate parabolic equation. However, in a parameter estimation setting, this is very difficult. The problem is that there is no information about $q(x)$ available when $q(x)N(u)$ attains small values. This problem is avoided herein by simply assuming that $N(u)$ is bounded away from zero.

A large amount of literature (e.g., [2, 3, 10, 11, 14]) is devoted to the augmented Lagrangian method for identification of $q(x)$ within the linear elliptic equation

$$-\nabla \cdot (q(x)\nabla u) = f(x). \tag{1.5}$$

Less work has been done on parameter estimation in the linear parabolic equation

$$u_t - \nabla \cdot (q(x)\nabla u) = f(x, t). \tag{1.6}$$

Recovery of $q(x)$ in Eq. 1.6 using the augmented Lagrangian method is investigated in [9, 13, 17], and other methods are studied in [5, 12, 15].

In [16], the augmented Lagrangian method for recovery of $q(x)$ within the nonlinear parabolic equation,

$$u_t - \nabla \cdot (q(x)N(\nabla u, u)\nabla u) = f(x, t), \tag{1.7}$$

is studied.

Equations 1.5–1.6 can be used to described one-phase flow processes in porous media where $q(x)$ is the permeability. Equation 1.7 is not used in two-phase flow simulations, but it has interesting mathematical properties. The aim of this paper is to take the augmented Lagrangian methodology one step further towards the problem of estimating permeability in real porous media simulations. This is done by studying parameter estimation in Eq. 1.1, which contains interesting features of the model Eqs. 1.2–1.4. For practical problems related to the recovery of permeability in such models, we refer the interested reader to, e.g., [4, 8, 18, 19].

Our approach for estimating $q(x)$ in Eq. 1.1 is based on observations $u_d(x^i, t)$, $i = 1, \ldots, n_p$; $t \in (0, T)$ of $u(x, t)$. These observations may contain noise. The inverse problem is formulated in a similar manner to [16, 17].

$$\min_{r(q,u)=0} \int_0^T \sum_{i=1}^{n_p} \frac{1}{2}(u(x^i, t) - u_d(x^i, t))^2 dt + \beta R(q), \tag{1.8}$$

with $r(q, u) = 0$ as the equation constraint (i.e., Eq. 1.1 fulfilled) and $q \in W$, where

$$W = \{q \in L^1(\Omega) | \quad 0 < q_{\min} \le q \le q_{\max} < \infty\}.$$

$r(q, u)$ is defined as the left-hand side minus the right-hand side of the equation, i.e.,

$$r(q, u) = u_t + \nabla \cdot (f, g) - \nabla \cdot (qN(u)\nabla u) - s.$$

$r(q, u)$ is referred to as the residual. The second term of the objective function consists of $R(q)$, which is a regularization functional, and β, which is a regularization parameter. These will be specified in the numerical examples. In the rest of this paper, we assume that $R(q)$ is quadratic in q.

The minimizer for Eq. 1.8 can be found by the augmented Lagrangian method. The augmented Lagrangian functional is defined by

$$L_c(q, u, \lambda) = \int_0^T \sum_{i=1}^{n_p} \frac{1}{2}(u(x^i, t) - u_d(x^i, t))^2 dt + \beta R(q) + \int_0^T (r, \lambda) dt + \frac{c}{2} \int_0^T \|r\|^2 dt,$$

where $(\cdot, \cdot)$ and $\|\cdot\|$ denote the L^2 inner product and norm, respectively. A saddle point for L_c together with the equation constraint fulfilled is a local minima for Eq. 1.8.

In [17] and [16], the models (Eq. 1.6 and Eq. 1.7, respectively) are discretized with the finite element method. For equations like Eq. 1.1, where there is a convection term included, the finite difference method is a highly suitable discretization technique, which we will use in this paper. The convection term could be discretized with central difference approximation, which is second-order accurate. However, a drawback of the central difference approach is that the numerical scheme needs very small time steps when the problem is convection-dominated. A remedy out of this is to use an upwind scheme for the convection term. Herein, we use a difference scheme based on the Engquist–Osher numerical flux [6] for the convection part and centered differencing for the diffusion part. This means that the differencing of the convective flux is biased in the direction of "incoming waves," which makes it possible to resolve solutions with steep gradients without too small time steps.

Another aspect is that, when solving Eq. 1.8 with an optimization technique, we need the derivative of the objective function with respect to the discretized solution. Consequently, it is important that we use a difference scheme whose numerical flux function is a differentiable function of the discretized solution, which is the case with the schemes used in this paper.

Traditionally, parameter estimation problems in complicated models, like the equations for multiphase flow, are formulated as output least squares problems, which are solved with methods like quasi-Newton or Gauss–Newton. With such an approach, the objective functional is nonquadratic in q. Here, L_c is quadratic in q for fixed u. In [17], L_c is also quadratic in u for fixed q, but that is not true in this paper because of the nonlinearities in the functions $f(u)$, $g(u)$, and $N(u)$.

Note that, in our minimization formulation, we do not use the interpolated version of u_d. In the formulation, we only calculate the distance between u and u_d at the observation points. Note also that we minimize over both q and u. These things make the augmented Lagrangian methodology a flexible formulation of the inverse problem, which has proven to give good results for finding global minima.

The remaining part of this paper is organized as follows: First, we present the numerical scheme that is used to discretize the forward problem Eq. 1.1. Then, we formulate the inverse problem in a discrete setting and explain how this can be solved with the augmented Lagrangian method. Next, we show how the nonlinear conjugate gradient method can be used to solve the subminimization problems of the augmented Lagrangian formulation. Finally, we present some numerical results with the proposed method.

2 Discretization and the inverse problem

In [17] and [16], Eqs. 1.6 and 1.7 are discretized with a finite element method. In this work, we use a finite difference method to discretize Eq. 1.1. This is a more suitable discretization method for equations including convection terms. The numerical schemes we present are implicit in the diffusion term and explicit in the convection term. We will, in the numerical experiments, use uniform grid in both space and time.

2.1 Finite difference discretization

Equation 1.1 is discretized with finite difference methods and we write the discretized equation as

$$u_{t_h} + \nabla_h \cdot (f, g) - \nabla_h \cdot (q(x) N(u) \nabla_h u) = s_h, \tag{2.1}$$

where the subscript h denotes the discretization parameter.

We assume that the discrete functions are defined on an $n_1 \times n_2$−grid in space, i.e., in $\Omega = (0, 1) \times (0, 1)$. The following notation will be used: $x_i = i\Delta x$, $y_j = j\Delta y$, $t^n = n\Delta t$, $u^n_{i,j} = u(x_i, y_j, t^n)$, and $s^n_{i,j} = s(x_i, y_j, t^n)$. Here, $\Delta x = \frac{1}{n_1}$, $\Delta y = \frac{1}{n_2}$, $\Delta t = \frac{T}{M}$, $i = 1, \dots n_1$, $j = 1, \dots n_2$, and $n = 1, \dots M$. The full vector of the n-th time level of a variable, for example, u, will be denoted by $u^n \in \mathbf{R}^{n_1 \times n_2}$. The discrete derivatives in the x direction are denoted

$$D^x_- u^n_{i,j} = \frac{u^n_{i,j} - u^n_{i-1,j}}{\Delta x} \quad \text{and} \quad D^x_+ u^n_{i,j} = \frac{u^n_{i+1,j} - u^n_{i,j}}{\Delta x}$$

for the backward and forward difference approximations, respectively. We use a corresponding notation for the discrete derivatives in the y and t directions.

In the following, we give a specific description of how the terms in Eq. 2.1 are discretized. The discretization of the time derivative term is

$$u_{t_h} = D^t_- u^n_{i,j}.$$

The discretized permeability, q, is defined by $q_{i+\frac{1}{2},j+\frac{1}{2}} = q(x_{i+\frac{1}{2}}, y_{j+\frac{1}{2}})$. The mean values are defined as

$$\overline{q}^x_{i,j+\frac{1}{2}} = \frac{1}{2}\left(q_{i+\frac{1}{2},j+\frac{1}{2}} + q_{i-\frac{1}{2},j+\frac{1}{2}}\right) \quad \text{and}$$

$$\overline{q}^y_{i+\frac{1}{2},j} = \frac{1}{2}\left(q_{i+\frac{1}{2},j+\frac{1}{2}} + q_{i+\frac{1}{2},j-\frac{1}{2}}\right).$$

For the nonlinear function $N(u)$, we denote $N^n_{i,j} = N\left(u^n_{i,j}\right)$, and the mean values as defined as

$$\left(\overline{N}^x\right)^n_{i+\frac{1}{2},j} = \frac{1}{2}\left(N^n_{i+1,j} + N^n_{i,j}\right) \quad \text{and}$$

$$\left(\overline{N}^y\right)^n_{i,j+\frac{1}{2}} = \frac{1}{2}\left(N^n_{i,j+1} + N^n_{i,j}\right).$$

The nonlinear diffusion term is discretized by

$$\nabla_h \cdot (q_{i,j} N^n_{i,j} \nabla_h u^n_{i,j}) = D^x_-\big(\overline{q}^y_{i+\frac{1}{2},j} \big(\overline{N}^x\big)^n_{i+\frac{1}{2},j} D^x_+ u^n_{i,j}\big) + D^y_-\big(\overline{q}^x_{i,j+\frac{1}{2}} \big(\overline{N}^y\big)^n_{i,j+\frac{1}{2}} D^y_+ u^n_{i,j}\big).$$

For the convection term, we use the Engquist–Osher upwind scheme (see [6])

$$\nabla^u_h \cdot \big(f(u^{n-1}_{i,j}), g(u^{n-1}_{i,j})\big) = D^x_- f^{EO}\big(u^{n-1}_{i,j}, u^{n-1}_{i+1,j}\big) + D^y_- g^{EO}\big(u^{n-1}_{i,j}, u^{n-1}_{i,j+1}\big), \quad (2.2)$$

where the Engquist–Osher numerical flux functions $f^{EO}(u_{i,j}, u_{i+1,j})$ and $g^{EO}\big(u_{i,j}, u_{i,j+1}\big)$ are defined by

$$f^{EO}(u_{i,j}, u_{i+1,j}) = \frac{1}{2}\big(f(u_{i,j}) + f(u_{i+1,j})\big) - \frac{1}{2}\int_{u_{i,j}}^{u_{i+1,j}} |f'(\xi)| d\xi,$$

$$g^{EO}\big(u_{i,j}, u_{i,j+1}\big) = \frac{1}{2}\big(g(u_{i,j}) + g(u_{i,j+1})\big) - \frac{1}{2}\int_{u_{i,j}}^{u_{i,j+1}} |g'(\xi)| d\xi.$$

In Appendix A, we calculate explicit formulas for f^{EO} and g^{EO}, for examples of f and g.

The upwind scheme Eq. 2.2 is a first-order scheme, and it can solve problems that are convection-dominated. In the next subsection, we comment on the regularity with respect to the numerical solution u of the numerical flux functions.

We can now write Eq. 2.1 as

$$D^t_- u^n_{i,j} + \nabla^u_h \cdot \big(f(u^{n-1}_{i,j}), g(u^{n-1}_{i,j})\big) - \nabla_h \cdot \big(q_{i,j} N^n_{i,j} \nabla_h u^n_{i,j}\big) = s^n_{i,j}, \quad n = 1, \ldots, M, \quad (2.3)$$

where $u^0_{i,j} = u_0(x_i, y_j)$ and $u^n_{i,j} = 0$ for $(x_i, y_j) \in \partial\Omega$. Notice that, in Eq. 2.3, the convection term is defined explicitly in time, but the diffusion term is discretized implicitly.

2.2 Regularity of the numerical flux function

In this subsection, we discuss the regularity of numerical flux function. In particular, we wish to explicitly point out that $\nabla^u_h \cdot \big(f(u^{n-1}_{i,j}), g(u^{n-1}_{i,j})\big)$ is a differentiable function of u^{n-1}.

For simplicity of notation, we consider only the one-dimensional case, i.e., we study the regularity of $\nabla^u_h \cdot f(u_i)$ with respect to u. The analysis of $\nabla^u_h \cdot g(u_i)$ is identical.

We have

$$\nabla^u_h \cdot f(u_i) = D^x_- f^{EO}(u_i, u_{i+1}) = \frac{1}{\Delta x}\left[f^{EO}(u_i, u_{i+1}) - f^{EO}(u_{i-1}, u_i)\right],$$

and we see that the regularity of $\nabla^u_h \cdot f(u_i)$ with respect to u depends on the regularity of f^{EO}. We further calculate the first- and second-order derivatives of f^{EO} with respect to u_i and u_{i+1}

$$f^{EO}(u_i, u_{i+1}) = \frac{1}{2}(f(u_i) + f(u_{i+1})) - \frac{1}{2}\int_{u_i}^{u_{i+1}} |f'(\xi)| d\xi,$$

$$\frac{\partial}{\partial u_i} f^{EO}(u_i, u_{i+1}) = \frac{1}{2} f'(u_i) + \frac{1}{2}|f'(u_i)|,$$

$$\frac{\partial^2}{\partial u_i^2} f^{EO}(u_i, u_{i+1}) = \frac{1}{2} f''(u_i) + \frac{1}{2}\mathrm{sgn}(|f'(u_i)|) f'(u_i),$$

$$\frac{\partial}{\partial u_{i+1}} f^{EO}(u_i, u_{i+1}) = \frac{1}{2} f'(u_{i+1}) - \frac{1}{2}|f'(u_{i+1})|,$$

$$\frac{\partial^2}{\partial u_{i+1}^2} f^{EO}(u_i, u_{i+1}) = \frac{1}{2} f''(u_{i+1}) - \frac{1}{2}\mathrm{sgn}(|f'(u_{i+1})|) f''(u_i),$$

where $\mathrm{sgn}(\cdot)$ denotes the sign function. By these calculations, we can then conclude that $\nabla^u_h \cdot f(u_i)$ and its first-order derivatives with respect to u_i and u_{i+1} are continuous, but the second-order derivatives of $\nabla^u_h \cdot f(u_i)$ with respect to u_i and u_{i+1} are discontinuous.

For the nonlinear conjugate gradient method (see Section 3.1) to be convergent, it is crucial that the gradient of the objective function is continuous (cf. [1]). In our setting, with the augmented Lagrangian method to do parameter estimation in convection–diffusion equations, the numerical flux function needs to be continuously differentiable with respect to the numerical solution. In Section 3.2 and Appendix B, we give explicit descriptions of the gradients of the objective functions for our minimization problems.

2.3 Fix point iteration

To integrate the forward problem Eq. 2.3 in time, we have to solve a nonlinear system for each time level. The systems are nonlinear because of the implicit treatment of the nonlinear diffusion term. We use a fix-point iteration to solve these nonlinear systems.

If we write the nonlinear system on the form $A(z)z = b$, where $A(z)$ is a nonsingular matrix depending nonlinearly on the vector z, then we iterate

$$A\left(z^{k-1}\right)z^k = b,$$

until convergence, with z^0 given. Here, z^0 is typically the solution from the previous time level. If this method converges, it is often an efficient way to solve a nonlinear problem. This method is only used to solve the forward problem Eq. 2.3.

2.4 Coarse grid for **q**

In earlier works (cf. [17] and [16]), the diffusion coefficient, q, has been defined in a coarse grid. This is straightforward when working with finite element functions. In this work, we will also describe the parameter, q, with a possible reduced number of degrees of freedom compared to the finest mesh. We define

$$q \in \mathbf{R}^{m_1 \times m_2},$$

and we use a constant prolongation when we need $q \in \mathbf{R}^{n_1 \times n_2}$. In the numerical experiments, we assume that the fine mesh is a refinement of the coarse mesh. Constant prolongation means that an entry of q on the fine mesh is obtained by taking the value from the closest point of the coarse mesh. In practice, the dimension of this space typically depends on the information available from the measurements.

2.5 The residual

We define the residual as the left-hand side minus the right-hand side of the discretized equation

$$r_{i,j}^n = D_-^t u_{i,j}^n + \nabla_h^u \cdot \left(f\left(u_{i,j}^{n-1}\right), g\left(u_{i,j}^{n-1}\right)\right)$$
$$-\nabla_h \cdot \left(q_{i,j} N_{i,j}^n \nabla_h u_{i,j}^n\right) - s_{i,j}^n,$$

where the prolongated $q \in \mathbf{R}^{n_1 \times n_2}$ is used.

For the minimization algorithms presented in next section, we need the derivatives of the residual with respect to u and q. Those calculations are presented in Appendix B.

2.6 Discretized minimization

We formulate a finite dimensional problem corresponding to Eq. 1.8 as follows:

$$\min_{r(q,u)=0} \sum_n \Delta t E(u^n) + \beta R(q), \tag{2.4}$$

subject to $q \in \mathbf{R}^{m_1 \times m_2}$ satisfying $q_{\min} \le q \le q_{\max}$ and $u \in \mathbf{R}^{n_1 \times n_2 \times M+1}$ satisfying $u_{i,j}^0 = u_0(x_i, y_j)$ and $u_{i,j}^n = 0$ for $(x_i, y_j) \in \partial\Omega$. Here,

$$E(u^n) = \sum_{(i,j)\in I_{\text{obs}}} \frac{1}{2}\left(u_{i,j}^n - (u_d)_{i,j}^n\right)^2, \tag{2.5}$$

and the regularization term, $\beta R(q)$, will be defined in the numerical experiments. Even though u is defined over $M+1$ time levels, the following minimization algorithm will only vary u in M time levels because the initial time level is fixed.

Above, I_{obs} denotes the set of all indices where the observation points are located (cf. x^i, $i = 1, \ldots, n_p$ in Eq. 1.8). In our numerical experiments, we have that the observation points are on the grid points. However, if we have observation points that are not located at grid points, we should do a spatial linear interpolation of u^n to execute $E(u^n)$. In this way, it will be similar to the finite element formulation of [16] and [17], where the discretized functions are continuously defined.

2.7 Minimization algorithm

We solve the discretized minimization problem Eq. 2.4 by the augmented Lagrangian method. The discretized augmented Lagrangian functional $L_c : \mathbf{R}^{m_1 \times m_2} \times \mathbf{R}^{n_1 \times n_2 \times M+1} \mathbf{R}^{n_1 \times n_2 \times M} \to \mathbf{R}$ is defined

$$L_c(q,u,\lambda) = \sum_{n=1}^{M} \Delta t E(u^n) + \beta R(q)$$
$$+ \sum_{n=1}^{M} \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} \Delta_{xyt} \left(\lambda_{i,j}^n r_{i,j}^n + \frac{c}{2}\left(r_{i,j}^n\right)^2\right), \tag{2.6}$$

where $\Delta_{xyt} = \Delta x \Delta y \Delta t$. Here, λ is a Lagrange multiplier. Notice that $\lambda \in \mathbf{R}^{n_1 \times n_2 \times M}$, while $u \in \mathbf{R}^{n_1 \times n_2 \times M+1}$. The reason for this is, as pointed out above, that the initial level of u is fixed, and therefore, λ is not defined for time level zero. $c > 0$ is a penalization constant, which is determined experimentally. In the discrete setting, it is known that L_c has a saddle point and that this point is a minimizer for Eq. 2.4, see [3, 11, 14].

We will use the following augmented Lagrangian method to find saddle points for this functional.

Here, $u_k^n \in \mathbf{R}^{n_1 \times n_2}$ denotes u at time level n and iteration k, and $u_k \in \mathbf{R}^{n_1 \times n_2 \times M+1}$ denotes the full vector for iteration k of Algorithm 2.1. The initial guess in the augmented Lagrangian method $u_0 \in \mathbf{R}^{n_1 \times n_2 \times M+1}$ should not be confused with the continuous function $u_0(x)$ describing the initial condition of Eq. 1.1.

Algorithm 2.1 (The augmented Lagrangian method)

Step 1 Choose initial values for $\lambda_0 \in \mathbf{R}^{n_1 \times n_2 \times M}$, $u_0 \in \mathbf{R}^{n_1 \times n_2 \times M+1}$ and set $k = 1$.

Step 2 Find $q_k \in \mathbf{R}^{m_1 \times m_2}$, $q_{\min} \le q_k \le q_{\max}$ satisfying

$$L_c(q_k, u_{k-1}, \lambda_{k-1}) = \min_{\substack{q \in \mathbf{R}^{m_1 \times m_2}, \\ q_{\min} \le q_k \le q_{\max}}} L_c(q, u_{k-1}, \lambda_{k-1}). \tag{2.7}$$

Step 3 Set $u_k^0 = u^0$ and find $\{u_k^n\}_{n=1}^M$ satisfying

$$L_c(q_k, u_k, \lambda_{k-1}) = \min_{u \in \mathbf{R}^{n_1 \times n_2 \times M+1}} L_c(q_k, u, \lambda_{k-1}). \tag{2.8}$$

Step 4 Update the Lagrange multiplier by

$$\lambda_k = \lambda_{k-1} + c r(q_k, u_k).$$

If not converged: Set $k = k + 1$ and go to step 2.

3 Implementation with the conjugate gradient method

In this section, we consider the problem of how to efficiently solve the two subminimization problems Eqs. 2.7 and 2.8 defined in Algorithm 2.1.

3.1 The nonlinear conjugate gradient method

In this subsection, we review the nonlinear conjugate gradient method to solve

$$\min_{z \in \mathbf{R}^d} F(z), \tag{3.1}$$

where F is a smooth function, and we have its gradients available. Nonlinear conjugate gradient methods are known to be good at solving large-scale problems and take the following form:

$k = 1,\ z_0$ given, $g_1 = -\nabla F(z_0)$,

while $\|\nabla F(z_k)\| > \epsilon$,

$$z_k = z_{k-1} + \alpha_k g_k,$$

$$g_{k+1} = -\nabla F(z_k) + \beta_k g_k,$$

$$k = k + 1.$$

end

Here, α_k is a step size, which is determined by a one-dimensional line search

$$\alpha_k = \arg\min_{\alpha} F(z_k + \alpha g_k). \tag{3.2}$$

The scalar β_k can be chosen as either (see [1])

$$\beta_k^1 = \frac{\|\nabla F(z_k)\|^2}{\|\nabla F(z_{k-1})\|^2} \quad \text{or}$$

$$\beta_k^2 = \frac{(\nabla F(z_k), \nabla F(z_k) - \nabla F(z_{k-1}))}{\|\nabla F(z_{k-1})\|^2}.$$

Here, $\|\cdot\|$ and $(\cdot,\cdot)$ denotes the norm and inner product of l^2. The latter choice, β_k^2, is most stable with respect to nonoptimal line search. If F is a quadratic function, $\nabla F(z_k)$ and $\nabla F(z_{k-1})$ are orthogonal, and thus, $\beta_k^1 = \beta_k^2$.

For quadratic functions F, the exact solution of the line search is

$$\alpha_k = \frac{\|\nabla F(z_{k-1})\|^2}{g_k^T H_k g_k}, \tag{3.3}$$

where H_k is the Hessian of F in the point z_k, $H_k = \nabla^2 F(z_k)$. In this paper, we approximate the line search by Eq. 3.3 in the nonquadratic cases.

When using the nonlinear conjugate gradient method, we need to calculate $\nabla F(z_k)$ and $g_k^T \nabla^2 F(z_k) g_k$ for given z_k and g_k. We do not need to form the Hessian. In the next subsection, we study how the nonlinear conjugate gradient method can be efficiently used to minimize the augmented Lagrangian functional. For fixed (u, λ), the functional $L_c(q, u, \lambda)$ is quadratic with respect to q, but for fixed (q, λ), the functional $L_c(q, u, \lambda)$ is nonquadratic with respect to u. In order to use the conjugate gradient method, we need to calculate the Gateaux derivatives of L_c. The next subsection contains these calculations.

3.2 Gateaux derivatives of the augmented Lagrangian functional

In order to use the conjugate gradient method to solve Algorithm 2.1, we calculate the derivatives of the augmented Lagrangian functional. The Gateaux derivatives of the augmented Lagrangian functional in a given direction will be denoted $L_c' \cdot p = \frac{\partial L_c}{\partial q} \cdot p$ and $L_c' \cdot v = \frac{\partial L_c}{\partial u} \cdot v$. Here, the direction p will be a vector of the same dimension as q, and correspondingly, v has the same dimension as u. Note that, when writing $L_c' \cdot p$, the p indicates that we take the derivative with respect to q in the direction p. Similarly, the v in $L_c' \cdot v$ indicates the derivative with respect to u in the direction v. For second-order derivatives, we use the notation $L_c'' \cdot (p, p) = \frac{\partial^2 L_c}{\partial q^2} \cdot (p, p)$ and $L_c'' \cdot (v, v) = \frac{\partial^2 L_c}{\partial u^2} \cdot (v, v)$.

First, we calculate the derivative of L_c with respect to q in the direction $p \in \mathbf{R}^{m_1 \times m_2}$

$$L_c' \cdot p = \beta R'(q) \cdot p + \sum_{n,i,j} \Delta_{xyt}((r_{i,j}^n)' \cdot p)(\lambda_{i,j}^n + c r_{i,j}^n). \tag{3.4}$$

In the implementations, the full vector $\frac{\partial L_c}{\partial q}$ is needed. Let $\{e_{a,b}^1\}$ be the unit basis vectors of $\mathbf{R}^{m_1 \times m_2}$. We can calculate $\frac{\partial L_c}{\partial q}$ by letting the direction p in Eq. 3.4 run through all basis functions $e_{a,b}^1$, i.e., $(\frac{\partial L_c}{\partial q})_{a,b} = L_c' \cdot e_{a,b}^1$, for $a = 1, \ldots, m_1$, $b = 1, \ldots, m_2$.

The Gateaux derivative of L_c with respect to u in the direction $v \in \mathbf{R}^{n_1 \times n_2 \times M}$ (v is not defined for time level zero) is

$$\begin{aligned} L_c' \cdot v &= \sum_n \Delta t E'(u^n) \cdot v^n \\ &\quad + \sum_{n,i,j} \Delta_{xyt}\big((r_{i,j}^n)' \cdot v\big)\big(\lambda_{i,j}^n + c r_{i,j}^n\big) \\ &= \sum_{(i,j)\in I_{\text{obs}},n} \Delta t\big(u_{i,j}^n - (u_d)_{i,j}^n\big) v_{i,j}^n \\ &\quad + \sum_{n,i,j} \Delta_{xyt}\big((r_{i,j}^n)' \cdot v\big)\big(\lambda_{i,j}^n + c r_{i,j}^n\big). \end{aligned}$$

Let $\{e_{a,b}^2\}$ be the unit basis vectors of $\mathbf{R}^{n_1 \times n_2}$. We can calculate entry a, b at time level n by

$$\begin{aligned} \left(\frac{\partial L_c}{\partial u}\right)_{a,b}^n &= \sum_{(i,j)\in I_{\text{obs}}} \Delta t\big(u_{i,j}^n - (u_d)_{i,j}^n\big) e_{a,b}^2 \\ &\quad + \sum_{i,j} \Delta_{xyt}\big((r_{i,j}^n)' \cdot e_{a,b}^2\big)\big(\lambda_{i,j}^n + c r_{i,j}^n\big). \end{aligned}$$

The second-order derivatives are

$$L_c'' \cdot (p, p) = \beta R''(q) \cdot (p, p) + c \sum_{n,i,j} \Delta_{xyt}\big((r_{i,j}^n)' \cdot p\big)^2,$$

because r is linear in q, and

$$\begin{aligned} L_c'' \cdot (v, v) &= \sum_{n;(i,j)\in I_{\text{obs}}} \Delta t\big(v_{i,j}^n\big)^2 + c \sum_{n,i,j} \Delta_{xyt}\big((r_{i,j}^n)' \cdot v\big)^2 \\ &\quad + \sum_{n,i,j} \Delta_{xyt}\big((r_{i,j}^n)'' \cdot (v, v)\big)\big(\lambda_{i,j}^n + c r_{i,j}^n\big). \end{aligned}$$

See Appendix B for calculations of the derivatives of r.

4 Numerical experiments

We present numerical experiments for the proposed method. The test problem is Eq. 1.1 with $\Omega = (0, 1) \times (0, 1)$, $T = 0.05$, $u_0(x) = \sin(\pi x)\sin(\pi y)$, $s(x, t) = 0$ and the true parameter to be estimated, $q(x)$, is piecewise constant

$$q(x) = \begin{cases} q_1^c, & x \in [0, 0.5] \times [0, 0.5] \\ q_2^c, & x \in [0.5, 1] \times [0, 0.5] \\ q_3^c, & x \in [0, 0.5] \times [0.5, 1] \\ q_4^c, & x \in [0.5, 1] \times [0.5, 1]. \end{cases} \tag{4.1}$$

where $q_i^c = i, i = 1, \ldots 4$, in the first two examples.

The flux function in the x direction is an S-shaped Buckley–Leverett flux with gravitational effects

$$f(u) = \frac{u^2(1 - 5(1 - u^2))}{u^2 + (1 - u)^2}, \tag{4.2}$$

and the flux function in the y direction is an S-shaped Buckley–Leverett flux

$$g(u) = \frac{u^2}{u^2 + (1 - u)^2}. \tag{4.3}$$

For the nonlinear diffusion function, we use

$$N(u) = 1 + u(u - 1).$$

In all the numerical examples, the problem is discretized with a uniform grid with $h = \frac{1}{n_1 - 1} = \frac{1}{n_2 - 1}$ as grid cell size for u and $H = \frac{1}{m_1 - 1} = \frac{1}{m_2 - 1}$ as the grid cell size for q. The grid for u is a refinement of the grid for q. The number of time steps is $M = \frac{T}{\Delta t}$.

In Example 4.2, we test the parameter estimation algorithm with noise in the observations. It is added multiplicatively, i.e.,

$$\begin{aligned} &(u_d)_{i,j}^n = (1 + \sigma \cdot \text{rand}(i, j, n)) u_{i,j}^n, \\ &(i, j) \in I_{\text{obs}}, \quad n = 1, \ldots, M. \end{aligned} \tag{4.4}$$

Here, $\text{rand}(i, j, n)$ is a vector of normally distributed numbers with zero mean and standard deviation 1. We refer to $\sigma \in \mathbf{R}$ as the noise level.

The stopping criteria for the nonlinear conjugate gradient method from Section 3.1 is

$$\frac{\|\nabla F(z_k)\|}{\|\nabla F(z_0)\|} \le \epsilon,$$

where F is the augmented Lagrangian functional, ∇ denotes the derivative in either the $u-$ or $q-$direction, and $\|\cdot\|$ is the l^2-norm defined by

$$\|v\| = \sqrt{\frac{\sum_{i,j} v_{i,j}^2}{\sum_{i,j} 1}}.$$

In the examples, we use $\epsilon = 10^{-6}$.

The regularization functional is $R(q) = \|\nabla q\|^2$. The regularization parameter, β, and the augmented Lagrangian parameter, c, are determined by trial and error, i.e., we test several β and c values and check which performs best. Advanced methods for determining β and c values is not discussed in this paper. When there is no noise in the observations, the regularization parameter, β, can be set to zero.

In the examples, we illustrate the convergence of q_k, u_k, and $r_k = r(q_k, u_k)$ for the augmented Lagrangian method. In all examples, we plot the l^2−norm of the error. More specifically, for increasing k value, we plot

$$\|q_k - q\|,$$

to illustrate the convergence of q_k, where q is the true diffusion parameter,

$$\|u_k - u_d\| = \Delta t \sum_{n=1}^{M} E(u_k^n),$$

to illustrate the convergence of u_k, where u_d is the observations and $E(u_k^n)$ is defined in Eq. 2.5, and

$$\|r_k\| = \Delta t \sum_{n=1}^{M} \|r_k^n\|,$$

to illustrate the convergence of r_k. The augmented Lagrangian algorithm is stopped by inspection of these plots.

The initial values for q and u in the conjugate gradient method are the solutions from the previous iteration of Algorithm 2.1. In the first iteration of Algorithm 2.1, we use the spatial linear interpolant of $u_d^n(x^i)$ for u_0, and q_0 equal to a constant. The constant is chosen as the average of the true permeability, i.e., $q_0 = \frac{1}{|\Omega|}\int_\Omega q\ dx$. The Lagrange multiplier is initially $\lambda_0 = 0$. In the experiments, the constraints on W, $q_{\min} \leq q \leq q_{\max}$, were never active.

Example 4.1 This example is specified by $h = \frac{1}{8}$, $H = \frac{1}{4}$, and $M = \frac{T}{\Delta t} = 20$. The observations are placed in every other grid point of V_h in the x and y directions. Since $H = 2h$, this means that we have one observation point for each grid point in V_H, i.e., $n_p = 16$. The c value was set to $7 \cdot 10^{-5}$. Algorithm 2.1 is tested, and in Fig. 1, we show the convergence of $\|q_k - q\|$, $\|u_k - u_d\|$, and $\|r_k\|$.

Example 4.2 In this example, we test our algorithm when there is multiplicative noise added to the observations (cf. Eq. 4.4). The example is specified by $h = \frac{1}{24}$, $H = \frac{1}{12}$ (i.e., $n_p = 144$), $M = 10$, and $c = 3 \cdot 10^{-5}$. As above, we have observation points in every other grid point of V_h in the x and y directions, and since $H = 2h$, we have one observation point in each of the 144 grid points of V_H.

For noise level $\sigma = 10^{-3}$, Fig. 2 show the convergence of q, u, and r. We have tested both L^2-norm and H^1-seminorm regularizations, i.e., $R(q) = \int_\Omega q^2 dx$ and $R(q) = \int_\Omega \nabla q \cdot \nabla q\, dx$, respectively. The H^1-seminorm regularization performs better, and the optimal choice of the regularization parameter is $\beta = 6 \cdot 10^{-11}$.

In Fig. 3, we show the result after 12 iterations of Algorithm 2.1. The estimated parameter q_{15} is shown

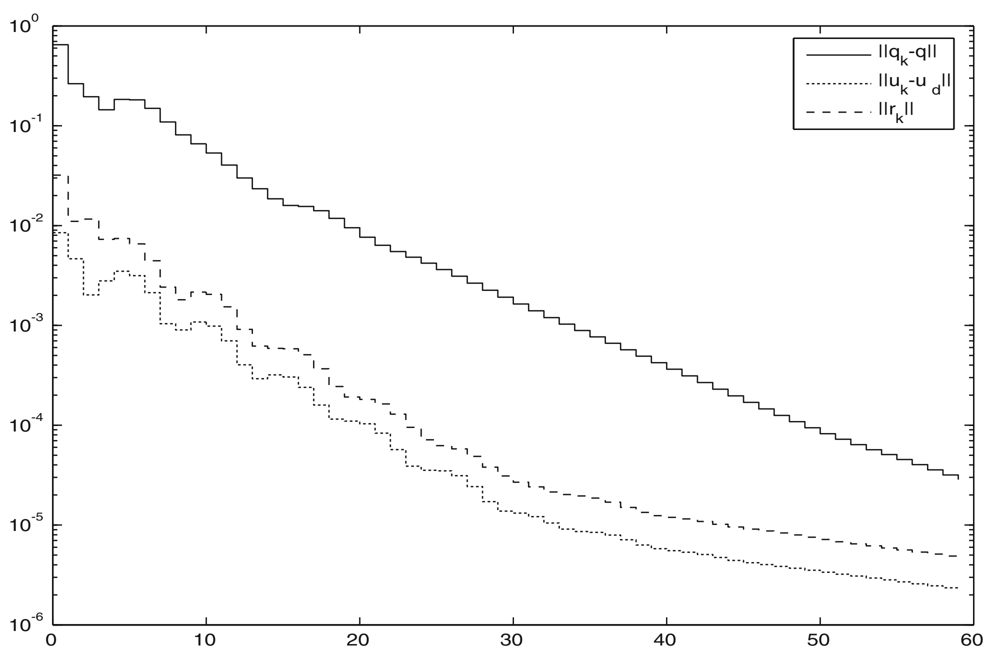


Fig. 1 $\|q_k - q\|$, $\|u_k - u_d\|$, and $\|r_k\|$ vs k. Logarithmic scale on the *vertical axis*

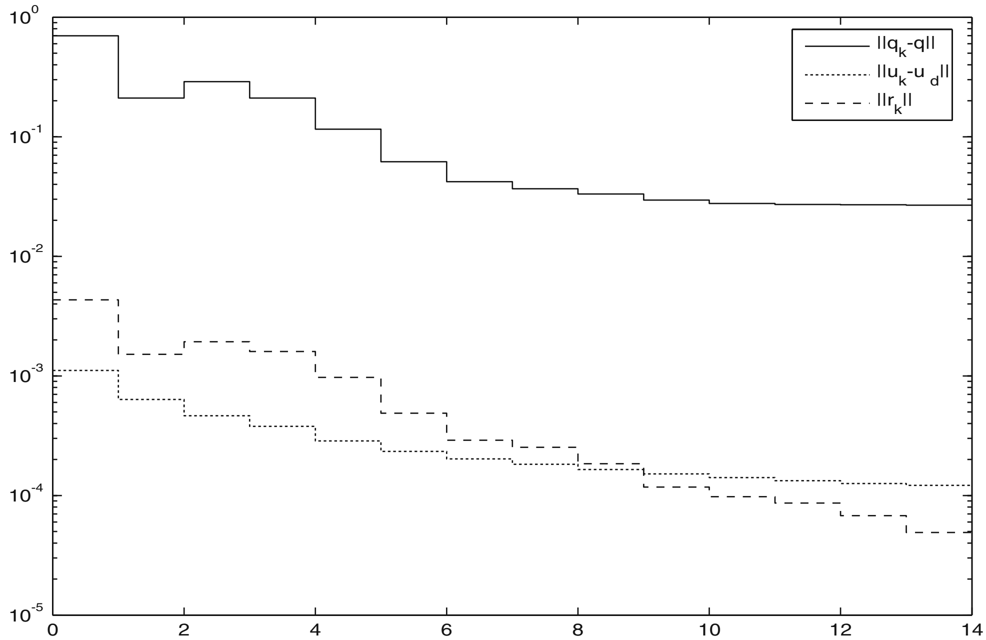


Fig. 2 $\|q_k - q\|$, $\|u_k - u_d\|$, and $\|r_k\|$ vs k with noise level $\sigma = 10^{-3}$. Logarithmic scale on the *vertical axis*

together with the true parameter q and the error $q_{15} - q$. The relative l^2 error after the last iteration is $\frac{\|q-q_{15}\|}{\|q\|} \approx 0.014$.

Example 4.3 Here, we test a more convection-dominated example. This example is described as Example 4.1, but we use $q_i^c = i \cdot 10^{-1}$. The convergence of q_k to q is shown in Fig. 4. The final time is chosen shorter, $T = 0.02$. This is to make sure that we do not get transport of information across the boundaries due to the convection term. In this example, the c value was set to $1.6 \cdot 10^{-3}$.

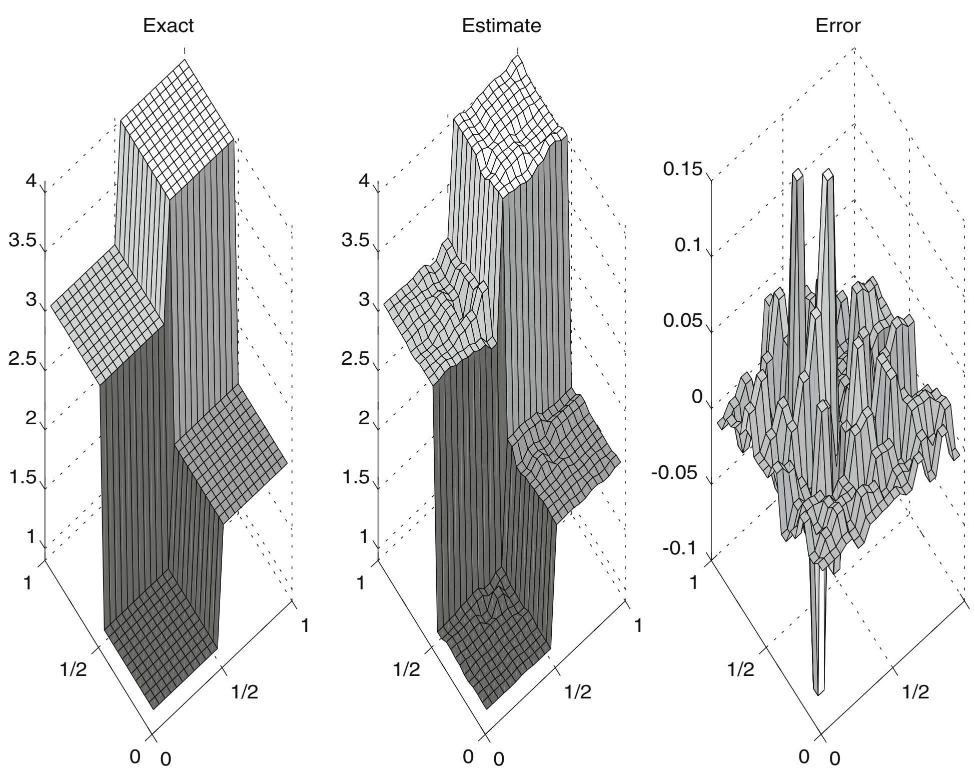
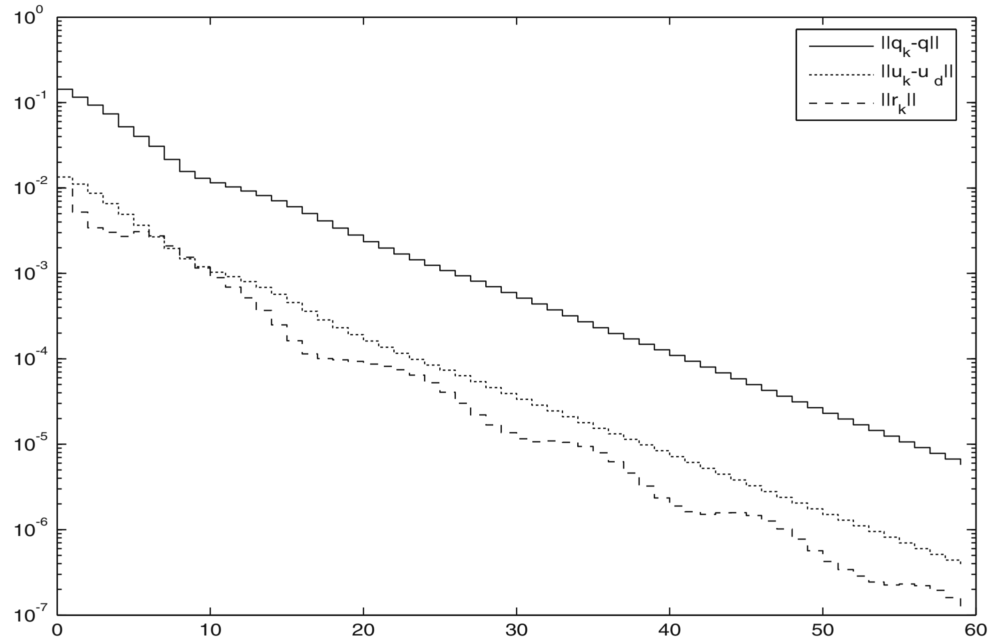


Fig. 3 The exact, the estimated and the error in permeability $q(x)$ with noise level $\sigma = 10^{-3}$

Fig. 4 $\|q_k - q\|$, $\|u_k - u_d\|$, and $\|r_k\|$ vs k. Logarithmic scale on the *vertical axis*



5 Concluding remarks

We have studied the augmented Lagrangian method for recovering a diffusion parameter in a convection–diffusion equation. The forward problem Eq. 1.1 contains some of the mathematical and numerical challenges that are present in models for multiphase porous media flow. By this work, we have taken a step further towards developing the augmented Lagrangian method to solve permeability estimation problems within such models.

We have observed good performance for the augmented Lagrangian method in our examples. The convergence plots of q_k to q are approximately the same as with the corresponding algorithms in [16]. The difference is less than 10%. However, the presented algorithm performs worse in examples where the convection term dominates the diffusion term. This is a field for further research.

The numerical experiments would have been even more physically relevant if the nonlinear diffusion function were $N(u) = u(u-1) + \epsilon$ for a small ϵ. As pointed out in Section 1, this is difficult in a parameter estimation setting. From numerical experiments, it seems that parameter estimation is hard when $\epsilon \leq 0.3$.

Appendix A: The numerical flux function

In this appendix, we give an explicit formula for the numerical flux function f^{EO} and g^{EO} and their discrete derivatives, used in the upwind scheme for the convection term. For notational simplicity we only use one subscript index in the following calculation.

A.1 Buckley–Leverett flux function

The flux function in the y direction is an S-shaped Buckley–Leverett flux function, cf. Eq. 4.3. Noticing that $g'(u) \geq 0$, for $u \in (0, 1)$, we see that

$$g^{EO}(u_i, u_{i+1}) = g(u_i),$$

which gives

$$D_-^x(g^{EO}(u_i, u_{i+1})) = \frac{1}{\Delta x}[g(u_i) - g(u_{i-1})].$$

A.2 Buckley–Leverett flux function with gravitational effects

The flux function in the x direction is an S-shaped Buckley–Leverett flux function with gravitational effects, cf. Eq. 4.2. In order to calculate $\int_{u_i}^{u_{i+1}} |f'(\xi)|d\xi$, we study the sign of f' in (0,1). f' has only one zero in (0, 1), and this is given by

$$x_{\text{zero}} \approx 0.37.$$

Thus, we see that

$$f'(\xi) \begin{cases} < 0, & \xi \in (0, x_{\text{zero}}) \\ = 0, & \xi = x_{\text{zero}} \\ > 0, & \xi \in (x_{\text{zero}}, 1). \end{cases}$$

Additionally, we can calculate the integral

$$\int_{u_i}^{u_{i+1}} |f'(\xi)|d\xi = \begin{cases} f(u_{i+1}) - f(u_i), & u_i,\ u_{i+1} \geq x_{\text{zero}} \\ f(u_i) - f(u_{i+1}), & u_i,\ u_{i+1} < x_{\text{zero}} \\ f(u_i) + f(u_{i+1}) - 2f(x_{\text{zero}}), & u_i < x_{\text{zero}},\ u_{i+1} \geq x_{\text{zero}} \\ 2f(x_{\text{zero}}) - f(u_i) - f(u_{i+1}), & u_i \geq x_{\text{zero}},\ u_{i+1} < x_{\text{zero}}, \end{cases}$$

and by the definition of f^{EO}, we then have

$$f^{EO}(u_i, u_{i+1}) = \begin{cases} f(u_i), & u_i,\ u_{i+1} \geq x_{\text{zero}} \\ f(u_{i+1}), & u_i,\ u_{i+1} < x_{\text{zero}} \\ f(x_{\text{zero}}), & u_i < x_{\text{zero}},\ u_{i+1} \geq x_{\text{zero}} \\ f(u_i) + f(u_{i+1}) - f(x_{\text{zero}}), & u_i \geq x_{\text{zero}},\ u_{i+1} < x_{\text{zero}}. \end{cases}$$

By this, we have an explicit formula for the numerical flux function in the x direction.

Appendix B: The derivatives of the residual

In the implementation of the nonlinear conjugate gradient method, we need the derivatives of the residual. In this section, we calculate the first- and second-order derivatives with respect to u and q. By this, we complete the calculations of Section 3.2.

B.1 Derivatives with respect to q

We first calculate the derivative with respect to q. The residual is linear in q, and thus, the derivative of the residual with respect to q in the direction p is

$$\frac{\partial r^n_{i,j}}{\partial q} \cdot p = -\nabla_h \cdot \left(p_{i,j} N^n_{i,j} \nabla_h u^n_{i,j}\right).$$

The second-order derivative is zero.

B.2 Derivatives with respect to u

Then, we calculate $\frac{\partial r^n_{i,j}}{\partial u} \cdot v$. For the discrete time derivative term, we have that the derivative with respect to u in the direction v is

$$\frac{\partial D^t_{-} u^n_{i,j}}{\partial u} \cdot v = D^t_{-} v^n_{i,j}.$$

For the convection term, the derivative can be similarly calculated. First, we calculate in 1D

$$\begin{aligned} \frac{\partial f^{EO}(u_i, u_{i+1})}{\partial u_i} &= \frac{1}{2} f_c'(u_i) + \frac{1}{2} |f_c'(u_i)| \\ &= f_c'(u_i) H(f_c'(u_i)) \quad \text{and} \\ \frac{\partial f^{EO}(u_i, u_{i+1})}{\partial u_{i+1}} &= \frac{1}{2} f_c'(u_{i+1}) - \frac{1}{2} |f_c'(u_{i+1})| \\ &= -f_c'(u_{i+1}) H(-f_c'(u_{i+1})), \end{aligned}$$

where $H(\cdot)$ is the Heaviside function. This gives

$$\begin{aligned} \frac{\partial f^{EO}(u_i, u_{i+1})}{\partial u} \cdot v = {} & f_c'(u_i) H(f_c'(u_i)) \cdot v_i \\ & - f_c'(u_{i+1}) H(-f_c'(u_{i+1}) \cdot v_{i+1}), \end{aligned}$$

and then in 2D,

$$\begin{aligned} & \frac{\partial \left[\nabla^u_h \cdot \left(f\left(u^{n-1}_{i,j}\right), g\left(u^{n-1}_{i,j}\right)\right)\right]}{\partial u} \cdot v \\ & = \frac{1}{\Delta x} \left(\left|f'\left(u^{n-1}_{i,j}\right)\right| \cdot v^{n-1}_{i,j} - f'\left(u^{n-1}_{i+1,j}\right) H\left[-f'\left(u^{n-1}_{i+1,j}\right)\right] \right. \\ & \qquad \left. \cdot v^{n-1}_{i+1,j} + f'\left(u^{n-1}_{i-1,j}\right) H\left[f'\left(u^{n-1}_{i-1,j}\right)\right] \cdot v^{n-1}_{i-1,j}\right) \\ & \quad + \frac{1}{\Delta y} \left(\left|g'\left(u^{n-1}_{i,j}\right)\right| \cdot v^{n-1}_{i,j} - g'\left(u^{n-1}_{i,j+1}\right) H\left[-g'\left(u^{n-1}_{i,j+1}\right)\right] \right. \\ & \qquad \left. \cdot v^{n-1}_{i,j+1} + g'\left(u^{n-1}_{i,j-1}\right) H\left[g'\left(u^{n-1}_{i,j-1}\right)\right] \cdot v^{n-1}_{i,j-1}\right). \end{aligned}$$

For the diffusion term, we have

$$\frac{\partial\left[\nabla_h\cdot\left(q_{i,j}N^n_{i,j}\nabla_h u^n_{i,j}\right)\right]}{\partial u}\cdot v$$
$$= D^x_-\left(\overline{q}^y_{i+\frac{1}{2},j}\left(\overline{N}^x\right)^n_{i+\frac{1}{2},j}D^x_+v^n_{i,j}\right)$$
$$+ D^x_-\left(\overline{q}^y_{i+\frac{1}{2},j}\left(\overline{N'\cdot v}^x\right)^n_{i+\frac{1}{2},j}D^x_+u^n_{i,j}\right)$$
$$+ D^y_-\left(\overline{q}^x_{i,j+\frac{1}{2}},\left(\overline{N}^y\right)^n_{i,j+\frac{1}{2}}D^y_+v^n_{i,j}\right)$$
$$+ D^y_-\left(\overline{q}^x_{i,j+\frac{1}{2}}\left(\overline{N\cdot v}^y\right)^n_{i,j+\frac{1}{2}}D^y_+u^n_{i,j}\right)$$
$$= D^x_-\left(\overline{q}^y_{i+\frac{1}{2},j}\left[\left(\overline{N}^x\right)^n_{i+\frac{1}{2},j}D^x_+v^n_{i,j}\right.\right.$$
$$\left.\left.+\left(\overline{N'\cdot v}^x\right)^n_{i+\frac{1}{2},j}D^x_+u^n_{i,j}\right]\right)$$
$$+ D^y_-\left(\overline{q}^x_{i,j+\frac{1}{2}}\left[\left(\overline{N}^y\right)^n_{i,j+\frac{1}{2}}D^y_+v^n_{i,j}\right.\right.$$
$$\left.\left.+\left(\overline{N\cdot v}^y\right)^n_{i,j+\frac{1}{2}}D^y_+u^n_{i,j}\right]\right).$$

From these calculations, we have an explicit formula for $(r^n_{i,j})'\cdot v$,

$$\frac{\partial r^n_{i,j}}{\partial u}\cdot v = \frac{\partial D^t_- u^n_{i,j}}{\partial u}\cdot v + \frac{\partial\left[\nabla_h\cdot\left(f(u^{n-1}_{i,j}),g(u^{n-1}_{i,j})\right)\right]}{\partial u}\cdot v$$
$$-\frac{\partial\left[\nabla_h\cdot(q_{i,j}N^n_{i,j}\nabla_h u^n_{i,j})\right]}{\partial u}\cdot v.$$

We also need the double derivative of the residual

$$\left(r^n_{i,j}\right)''\cdot(v,v) = \left[\nabla_h\cdot\left(f(u^{n-1}_{i,j}),g(u^{n-1}_{i,j})\right)\right]''\cdot(v,v)$$
$$+\left[\nabla_h\cdot\left(q_{i,j}N^n_{i,j}\nabla_h u^n_{i,j}\right)\right]''\cdot(v,v).$$

For the convection term, we have

$$\left[\nabla^u_h\cdot\left(f(u^{n-1}_{i,j}),g(u^{n-1}_{i,j})\right)\right]''\cdot(v,v)$$
$$=\frac{1}{\Delta x}\left[f''(u^{n-1}_{i,j})\,\mathrm{sgn}\left(f'(u^{n-1}_{i,j})\right)\left(v^{n-1}_{i,j}\right)^2\right.$$
$$-f''(u^{n-1}_{i+1,j})H\left(-f'(u^{n-1}_{i+1,j})\right)\left(v^{n-1}_{i+1,j}\right)^2$$
$$\left.+f''(u^{n-1}_{i-1,j})H\left(f'(u^{n-1}_{i-1,j})\right)\left(v^{n-1}_{i-1,j}\right)^2\right]$$
$$+\frac{1}{\Delta y}\left[g''(u^{n-1}_{i,j})\,\mathrm{sgn}\left(g'(u^{n-1}_{i,j})\right)\left(v^{n-1}_{i,j}\right)^2\right.$$
$$-g''(u^{n-1}_{i,j+1})H\left(-g'(u^{n-1}_{i,j+1})\right)\left(v^{n-1}_{i,j+1}\right)^2$$
$$\left.+g''(u^{n-1}_{i,j-1})H\left(g'(u^{n-1}_{i,j-1})\right)\left(v^{n-1}_{i,j-1}\right)^2\right].$$

For the diffusion term, we have

$$\left[\nabla_h\cdot\left(q_{i,j}N^n_{i,j}\nabla_h u^n_{i,j}\right)\right]''\cdot(v,v)$$
$$= D^x_-\left(\overline{q}^y_{i+\frac{1}{2},j}\left(\overline{N'\cdot v}^x\right)^n_{i+\frac{1}{2},j}D^x_+v^n_{i,j}\right)$$
$$+ D^x_-\left(\overline{q}^y_{i+\frac{1}{2},j}\left(\overline{N'\cdot v}^x\right)^n_{i+\frac{1}{2},j}D^x_+v^n_{i,j}\right)$$
$$+ D^x_-\left(\overline{q}^y_{i+\frac{1}{2},j}\left(\overline{N''\cdot(v,v)}^x\right)^n_{i+\frac{1}{2},j}D^x_+u^n_{i,j}\right)$$
$$+ D^y_-\left(\overline{q}^x_{i,j+\frac{1}{2}},\left(\overline{N'\cdot v}^y\right)^n_{i,j+\frac{1}{2}}D^y_+v^n_{i,j}\right)$$
$$+ D^y_-\left(\overline{q}^x_{i,j+\frac{1}{2}},\left(\overline{N'\cdot v}^y\right)^n_{i,j+\frac{1}{2}}D^y_+v^n_{i,j}\right)$$
$$+ D^y_-\left(\overline{q}^x_{i,j+\frac{1}{2}}\left(\overline{N''\cdot(v,v)}^y\right)^n_{i,j+\frac{1}{2}}D^y_+u^n_{i,j}\right)$$
$$= D^x_-\left(\overline{q}^y_{i+\frac{1}{2},j}\left[2\left(\overline{N'\cdot v}^x\right)^n_{i+\frac{1}{2},j}D^x_+v^n_{i,j}\right.\right.$$
$$\left.\left.+\left(\overline{N''\cdot(v,v)}^x\right)^n_{i+\frac{1}{2},j}D^x_+u^n_{i,j}\right]\right)$$
$$+ D^y_-\left(\overline{q}^x_{i,j+\frac{1}{2}},\left[2\left(\overline{N'\cdot v}^y\right)^n_{i,j+\frac{1}{2}}D^y_+v^n_{i,j}\right.\right.$$
$$\left.\left.+\left(\overline{N''\cdot(v,v)}^y\right)^n_{i,j+\frac{1}{2}}D^y_+u^n_{i,j}\right]\right),$$

where the mean value notation from Section 2 is used. In the examples, we use $N(u) = 1 + u(u-1)$, which gives $N'\cdot v = (2u-1)v$ and $N''\cdot(v,v) = 2v^2$. This gives

$$\left(\overline{N'\cdot v}^x\right)^n_{i+\frac{1}{2},j} = \frac{1}{2}\left(\left(2u^n_{i+1,j}-1\right)v^n_{i+1,j}\right.$$
$$\left.+\left(2u^n_{i,j}-1\right)v^n_{i,j}\right),$$
$$\left(\overline{N''\cdot(v,v)}^x\right)^n_{i+\frac{1}{2},j} = \left(v^n_{i+1,j}\right)^2+\left(v^n_{i,j}\right)^2,$$

and corresponding for the y direction.

References

1. Bertsekas, D.P.: Nonlinear programming. Athena Scientific, Thousand Oaks (1990)
2. Chan, T.F., Tai, X.–C.: Identification of discontinuous coefficient from elliptic problems using total variation regularization. Technical Report CAM–97–35, University of California at Los Angeles, Department of Mathematics (1997)
3. Chen, Z., Zou, J.: An augmented Lagrangian method for identifying discontinuous parameters in elliptic systems. SIAM J. Control Optim. **37**, 892–910 (1999)
4. Cuypers, M., Dubrule, O., Lamy, P., Bissell, R.: Optimal choice of inversion parameters for history matching with the pilot point method. In: Proc. 6th European Conference on the Mathematics of Oil Recovery, Peebles, UK (1998)
5. Engl, H.W., Zou, J.: A new approach to convergence rate analysis of Tikhonov regularization for parameter identification in heat conduction. Inverse Probl. **16**, 1907–1923 (2000)

6. Engquist, B., Osher, S.: One–sided difference approximations for nonlinear conservation laws. Math. Comput. **36**(154), 321–351 (1981)
7. Espedal, M.S., Karlsen, K.H.: Numerical solution of reservoir flow models based on large time step operator splitting algorithm. In: Filtration in Porous Media and Industrial Applications, Lecture Notes in Math., vol. 1734, pp. 9–77. Springer, Berlin (2000)
8. Grimstad, A.A., Mannseth, T., Nordtvedt, J.E., Nævdal, G.: Reservoir characterization through scale splitting. In: Proc. 7th European Conference on the Mathematics of Oil Recovery, Baveno, Italy (2000)
9. Gou, B., Zou, J.: An augmented Lagrangian method for parameter identification in parabolic systems. J. Math. Anal. Appl. **263**, 49–68 (2001)
10. Heimsund, B.O., Chan, T.F., Nilssen, T.K., Tai, X.–C.: Level set methods for a parameter identification problem. In: Barbu, V., Lasiecka, I., Tiba, D., Varsan, C. (eds.) Analysis and optimization of differential systems, pp. 189–200. Kluwer Academic, Boston (2003)
11. Ito, K., Kunisch, K.: The augmented Lagrangian method for parameter estimation in elliptic systems. SIAM J. Control Optim. **28**, 113–136 (1990)
12. Keung, Y.L., Zou, J.: Numerical identifications of parameters in parabolic equations. Inverse Probl. **14**(1), 83–100 (1998)
13. Kunisch, K., Peichl, G.: Estimation of a temporally and spatially varying diffusion coefficient in a parabolic system by an augmented Lagrangian technique. Numer. Math. **59**, 473–509 (1991)
14. Kunisch, K., Tai, X.-C.: Sequential and parallel splitting methods for bilinear control problems in Hilbert spaces. SIAM J. Numer. Anal. **34**, 91–118 (1997)
15. Kärkkäinen, T.: Error estimates for distributed parameter identification in parabolic problems with output least squares and Crank–Nicolson method. Appl. Math. **42**, 259–277 (1997)
16. Nilssen, T.K., Mannseth, T., Tai, X.-C.: Permeability estimation with the augmented Lagrangian method for a nonlinear diffusion equation. Comput. Geosci. **7**, 27–47 (2003)
17. Nilssen, T.K., Tai, X.-C.: Parameter estimation with the augmented Lagrangian method for a parabolic equation. J. Optim. Theory Appl. **124**(2), 435–453 (2005)
18. Reynolds, A.C., He, N., Chu, L., Oliver, D.: Reparameterization techniques for generating reservoir descriptions conditioned to variograms and well–test pressure data, SPE 30588. In: Proc. SPE Annual Technical Conference and Exhibition, Dallas, USA (1995)
19. Shah, P.C., Gavalas, G.R., Seinfeld, J.H.: Error analysis in history matching, the optimum level of parameterization. Soc. Pet. Eng. J. **6**, 219–228 (1978)