

FAST SOLVERS FOR FLOW IN POROUS MEDIA BASED ON DISCONTINUOUS GALERKIN METHODS AND OPTIMAL REORDERING

JOSTEIN R. NATVIG¹, KNUT-ANDREAS LIE^{1,2}, AND BIRGITTE EIKEMO³

¹ SINTEF ICT, Dept. Applied Math, P.O. Box 124 Blindern, N-0314 Oslo, Norway,

² Centre of Mathematics for Applications (CMA), University of Oslo, Norway

³ University of Bergen, Dept. of Mathematics, Johs. Brunsgt. 12, N-5008 Bergen, Norway

ABSTRACT

We present a family of efficient solvers for hyperbolic transport equations modelling flow in porous media. The solvers are based on discontinuous Galerkin spatial discretisations and implicit temporal discretisation. By applying an optimal reordering algorithm, the corresponding discrete system of (non)linear equations can be solved in one grid-block at a time. This way, we avoid assembly of a full (non)linear system. Our approach allows us to handle large numbers of grid blocks with modest requirements on memory.

1. INTRODUCTION

In this paper we present efficient and accurate solution procedures for a class of linear and nonlinear boundary-value problems of the form

$$\begin{aligned} \alpha u + \nabla \cdot (\mathbf{v}F(u)) &= \beta, & \mathbf{x} \in \Omega, \\ u &= h(\mathbf{x}), & \mathbf{x} \in \partial\Omega^+. \end{aligned} \tag{1}$$

Here $F(u)$ is a flux function with positive characteristics, \mathbf{v} is a given (nearly) curl-free vector field, and $\partial\Omega^+$ denotes the inflow boundary on which $\mathbf{v} \cdot \mathbf{n} < 0$. Equations of this form arise either as simple models for single-phase flow, like e.g., the time-of-flight equation,

$$\mathbf{v} \cdot \nabla \tau = \phi, \tag{2}$$

or as the result of an implicit semi-discretisation of systems of hyperbolic conservation laws for multiphase and multicomponent flow of the form

$$\phi \partial_t u_i + \nabla \cdot (\mathbf{v}F_i(u)) = q_i, \quad i = 1, \dots, \ell - 1. \tag{3}$$

To discretise (1) we use a discontinuous Galerkin (dG) formulation. By this approach, we can easily achieve high-order accuracy with local, compact stencils where the only coupling is between elements sharing a common element face. This yields systems of (non)linear equations with predictable structure: Each (non)linear equation describes the interaction between the degrees-of-freedom of one element and its immediate neighbours sharing a common element face. If the normal velocity $\mathbf{n} \cdot \mathbf{v}$ is constant on each face, this structure can be greatly simplified by using an upwind approximation of the fluxes across element interfaces. In fact, in the linear case the upwind discretisation yields *reducible* systems of equations, for which we can find symmetric permutations that map the global

systems to block-triangular systems, where each block involves the degrees-of-freedom of one or a few elements. In the nonlinear case, the permutation of equations and unknowns yields a block-triangular Jacobian matrix. Finding the permutation (or reordering) is quite easy if we view the fluxes across element interfaces as edges in a directed graph and rephrase the permutation as a topological sort. From elementary graph theory it follows that the reordering can be found by using a depth-first traversal of the grid, in which each cell is visited only once.

A key point in our approach is to exploit this optimal reordering to develop very efficient (non)linear solvers. In the linear case [Natvig et al., 2006], we use a direct solver to factor the small diagonal blocks in the triangular system and thereby obtain a very efficient direct solver. In the nonlinear case (see [Natvig and Lie, 2006]), the nonlinear subsystems can be solved one-by-one according to the reordering, using for instance a Newton–Raphson method. In both cases the computational effort is reduced significantly from solving a large sparse (non)linear system for all degrees-of-freedom in the domain to solving a sequence of block problems involving a few (non)linear equations for each element. Moreover, in the nonlinear case, we may control the iterations separately for each subsystem and this will generally give better convergence than for the corresponding global nonlinear iteration. Finally, by using the optimal reordering one avoids assembling the global system.

The reordering idea is not new and has been described previously by Duff and Reid [1978]. Similarly, Dennis Jr. et al. [1994] explore the use of block-triangular structures to construct effective Newton-type nonlinear solvers. However, to the best of our knowledge, these ideas have not previously been used to compute transport in porous media, even though the idea is quite natural and can easily be motivated by the underlying physics: The triangular structure of the equations reflects the directional dependence of the continuous equation (1) that previously has been exploited in streamline methods, see [King and Datta-Gupta, 1998]. Due to the positive characteristics of F , the exact solution in each element K will only depend on the upstream points of all streamlines passing through K and be independent of the solution elsewhere in the domain. Using an upwind flux in our dG formulation preserves this one-sided domain-of-dependence, which is a prerequisite for the reordering approach.

2. DISCONTINUOUS GALERKIN DISCRETISATION

A discontinuous Galerkin method starts with a variational formulation. We thus partition the domain into non-overlapping elements $\{K\}$, multiply (1) with a function v from the space of arbitrary piecewise smooth functions V , and integrate by parts over K to get

$$\int_K (\alpha u - \beta)v \, dx - \int_K F(u)\mathbf{v} \cdot \nabla v \, dx + \int_{\partial K} v F(u)\mathbf{v} \cdot \mathbf{n} \, ds = 0, \quad \forall v \in V.$$

We seek a solution in a finite-dimensional subspace $V_h \subset V$ consisting of functions that are smooth inside each element, but may be discontinuous over the element boundaries. Due to the possible discontinuities, we must replace the flux $F(u_h)\mathbf{v} \cdot \mathbf{n}$ with a consistent and conservative numerical flux function $\hat{F}(a, b, \mathbf{v} \cdot \mathbf{n})$. This leads to the following discrete variational formulation: let

$$a_K(u_h, v_h) = \int_K (\alpha u_h - \beta)v_h \, dx - \int_K F(u_h)\mathbf{v} \cdot \nabla v_h \, dx + \int_{\partial K} \hat{F}(u_h, u_h^{\text{ext}}, \mathbf{v} \cdot \mathbf{n})v_h \, ds \quad (4)$$

and find u_h such that

$$a_K(u_h, v_h) = 0, \quad \forall K, \forall v_h \in V_h. \quad (5)$$

For the numerical flux \hat{F} we use an upwind approximation

$$\hat{F}(p, p^{\text{ext}}) = F(p) \max(\mathbf{v} \cdot \mathbf{n}, 0) + F(p^{\text{ext}}) \min(\mathbf{v} \cdot \mathbf{n}, 0), \quad (6)$$

for inner and outer values p and p^{ext} at element boundaries. Note that there are other consistent flux approximations are also consistent, but they may not preserve the directional dependency we rely on to compute a permutation of the unknowns. For instance, the well-known Lax-Friedrichs flux yields a consistent approximation of the inter-element fluxes, but creates a bidirectional dependence between all elements.

In the following we assume that our elements K are hexahedrals in a regular Cartesian grid and choose $V_h^{(n)} = \{v : v|_K \in \mathbb{Q}^{n-1}\}$, where $\mathbb{Q}^n = \text{span}\{x^p y^q z^r, 0 \leq p, q, r \leq n\}$. A simple basis for this space is the tensor-product of Legendre polynomials L_k . Thus $V_h^{(1)}$ is the space of elementwise constant functions giving a scheme that is formally first order; $V_h^{(2)}$ is the space of elementwise trilinear functions, giving a second-order scheme; etc. Henceforth, dG(n) will denote the discontinuous Galerkin scheme of formal order n having $m = n^d$ unknowns per element in d spatial dimensions.

Substituting the tensor-product Legendre basis functions into (5) and using an appropriate Gaussian quadrature rule to approximate the integrals, we end up with a system of nonlinear equations for the unknown degrees-of-freedom U

$$G_K(U) := a_K^h(u_h, L_k) = 0, \quad \forall K. \quad (7)$$

By writing $U|_K$ for the unknowns in element K and $U|_{\Omega \setminus K}$ for the unknowns outside K and separating G_K into the different terms of (4) and (6), we can write (7) as

$$M_K U_K - B_K + R_K(U|_K) + F_K^+(U|_K) + F^-(U|_{\Omega \setminus K}) = G_K^+(U|_K) + G^-(U|_{\Omega \setminus K}) = 0.$$

If we reorder the unknowns, all degrees-of-freedom on the upwind side of element K will be known, meaning that $G^-(U|_{\Omega \setminus K})$ is a known quantity. The only unknowns in (7) for each K are therefore the degrees-of-freedom in K .

3. SEQUENTIAL SOLUTION

As explained in the introduction, the key to obtaining a fast (non)linear solver is to find a reordering $p = (p_1, \dots, p_N)$ of the elements that renders the system of equations (7) into a block-triangular form

$$\begin{aligned} G_{p_1}^+(U_{p_1}) &= 0, \\ G_{p_2}^-(U_{p_1}) + G_{p_2}^+(U_{p_2}) &= 0, \\ \vdots & \\ G_{p_N}^-(U_{p_1}, \dots, U_{p_{N-1}}) + G_{p_N}^+(U_{p_N}) &= 0. \end{aligned} \quad (8)$$

We therefore consider the directed graph defined by assigning a vertex to each element K_i and a directed edge for each flux $(\mathbf{v} \cdot \mathbf{n})|_{\partial K_i \cap \partial K_j}$ between elements. Thus, an edge from vertex i to vertex j implies that the solution in K_j depends on the solution in K_i . The task of arranging vertices in a sequence according to their position in a directed graph is called a topological sort. To see how a suitable sequence can be constructed, note that $p_i < p_j$ for any vertex i that can be reached from vertex j by going backwards in the

graph. By traversing the graph backwards in a depth-first manner, adding vertex j to the sequence when the search backwards from j has been completed, we obtain a topologically sorted sequence. Since a depth-first search only visits each vertex once, the topological sort of a directed graph can be obtained in $\mathcal{O}(N)$ time for N vertices.

If the sequence cannot be found by a single depth-first traversal, the graph has at least one cycle of vertices that can reach any other vertex in the same cycle. Cycles correspond to groups of elements that are made mutually dependent by a nonzero circulation in the velocity field \mathbf{v} . The degrees-of-freedom in such a group of elements correspond to an irreducible diagonal block and must be computed simultaneously. Fortunately, cycles can be detected automatically by performing a *forward* depth-first search. By lumping all elements in a cycle into a single vertex, we obtain an acyclic graph where each vertex corresponds to one or a few elements that form an irreducible block of degrees-of-freedom.

For flow in porous media, the velocity field is typically computed by solving a pressure equation. For incompressible flow, the exact velocity field has zero circulation. A simple argument shows that the same is true for an approximate velocity field computed by a two-point pressure solver. A mixed finite-element solution, on the other hand, may give a velocity field with nonzero circulation. In compressible flow, we may also get nonzero circulation. In our experience, cycles that appear in velocity fields computed by the mixed finite-element method are small and sparse for incompressible and weakly compressible flows.

4. NUMERICAL EXAMPLES

In this section we present a few examples to demonstrate that our dG approach gives efficient and accurate solvers for single-phase and multiphase flow in porous media.

4.1. Steady-State Tracer Distribution. As our first example, we consider the stationary distribution of a set of tracers being continuously injected into a reservoir, modelled by the simple equation

$$\mathbf{v} \cdot \nabla c_\alpha = 0, \quad c_\alpha|_{\partial\Omega^+} \text{ given.}$$

The reservoir model is taken from [Christie and Blunt, 2001], has $220 \times 60 \times 85$ grid cells, and consists of a smooth shallow-marine Tarbert formation on-top of a fluvial Upper Ness formation. A vertical injection well is located in each of the four corner and a producer is located in the middle. By launching different tracers in each of the four injectors, we may determine the volumes swept by each injector and use this to delineate the reservoir into independent flow regions, as shown in Figure 1. The velocity field of this scenario was computed with a two-point flux approximation, with pressure-driven injection. The stationary tracer distributions were computed with a second-order discontinuous Galerkin scheme, and the boundaries between the swept volumes were computed as the 0.5 isosurface of the different tracer saturations.

4.2. Time-of-Flight. In our second example, we show time-of-flight computed in a quarter five-spot test case in the unit square. The flow is driven by point sources in the lower left and upper right corners. The permeability and porosity equal 10^{-6} in $(0.3, 0.7) \times (0.3, 0.7)$ and 1.0 elsewhere. The solutions computed on 100×100 elements with the dG-method are shown in Figure 2 together with a reference solution computed by

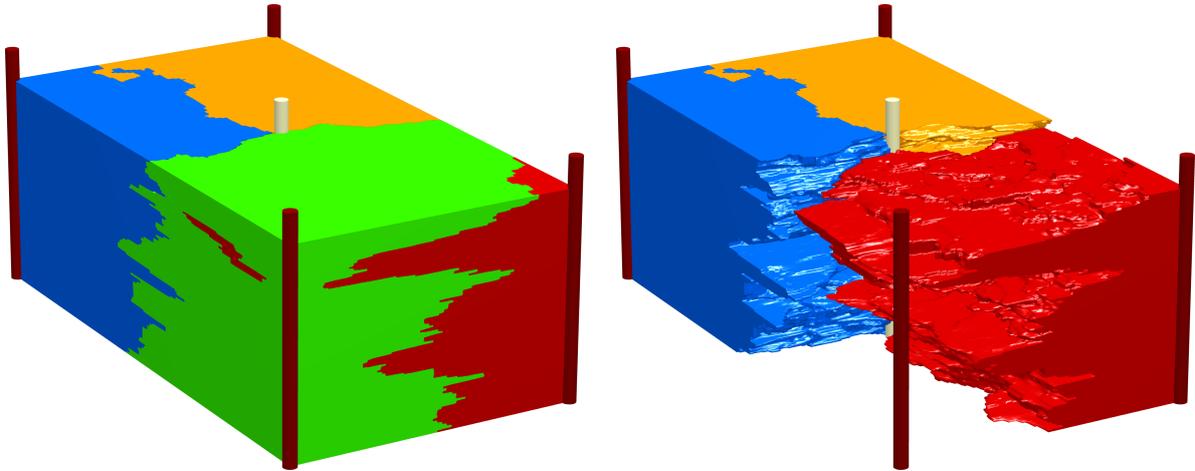


FIGURE 1. Stationary tracer distributions used to delineate the SPE 10 test case into independent flow regions. (Left) The four swept volumes in this scenario are shown in different colours. (Right) By removing one tracer, the intricate surfaces separating the swept volumes are revealed.

direct integration of time-of-flight along streamlines. To generate the plots the solutions have been sampled using 10×10 points in each element.

This seemingly innocent example turns out to be quite difficult to compute using any finite-difference or finite-volume method. The reason is that the time-of-flight solution has a large gradient downstream from the impermeable region. This rapid variation is impossible to capture accurately with polynomial elements, and will generate oscillations for elements with order higher than one.

Since our solution procedure computes the solution in one element at a time, we are able to implement a simple adaptive procedure. Clearly non-physical solutions can be detected by checking if the solution is increasing from inflow to outflow in each element K . If this is not the case (large) errors can propagate to the next element. To avoid this situation, we recompute the solution in K with a first-order method on a refined grid of $n \times n$ sub-elements. A computation based on this approach is shown in Figure 2.

4.3. Two-Phase Flow. In our next example we consider (3) for $\ell = 2$, modelling an oil-water system. The primary unknown is the water saturation s and the flux function is given by $f(s) = s^2/(s^2 + (1-s)^2)$. Applying a backward Euler temporal discretisation to (3), we get an equation of the form (1) for each time-step

$$\frac{\phi}{\Delta t} s^n + \nabla \cdot (\mathbf{v} f(s^n)) = \frac{\phi s^{n-1}}{\Delta t}.$$

The two-phase model will generally produce discontinuities at interfaces between injected water and resident oil. Near discontinuities, the dG scheme will tend to produce spurious oscillations, which can be suppressed by performing a post-processing with a nonlinear limiter function after each step, see [Natvig and Lie, 2006] for more details. Figure 3 shows a solution of a water injection scenario in Layer 37 of the model from [Christie and Blunt, 2001]. This layer is part of the fluvial Upper Ness formation characterised by high contrast and complex channel patterns. The computations are performed with

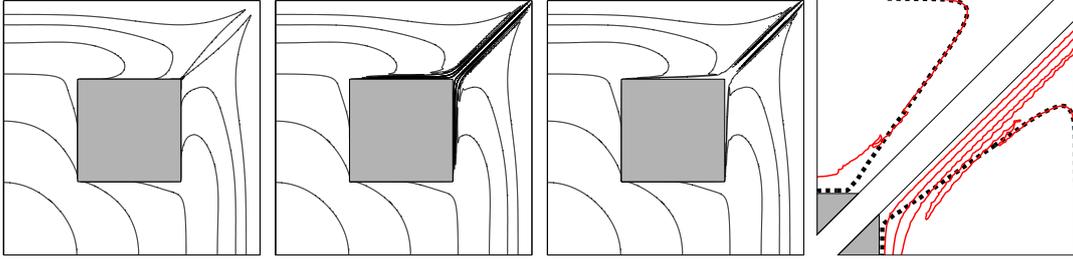


FIGURE 2. Time-of-flight in quarter five-spot with low-permeable region. Contours correspond to $[0.075 : 0.15 : 0.875]$ pore volumes injected. The solution is computed using (from left to right) streamline integration, $dG(3)$, and adaptive $dG(3)$. The figure on the right zooms on the 0.65 contour of $dG(3)$ (red) in the lower half and the adaptive $dG(3)$ (red) in the upper half, both with the streamline solution shown as a dashed line.

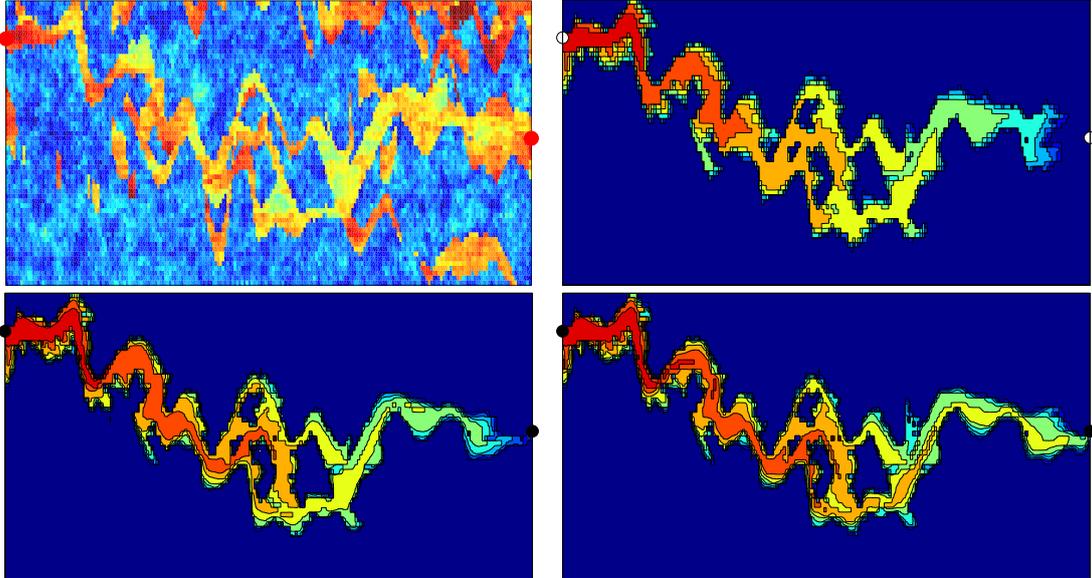


FIGURE 3. Two-phase flow in the fluvial Upper Ness formation of the SPE 10 test case. The figure shows (from top-left to bottom-right) the permeability field, the solutions at time $t = 0.2PVI$ computed using $dG(1)$, $dG(2)$, and $dG(3)$.

a fixed velocity field computed with a mixed finite-element method and saturations have been computed with $dG(1)$, $dG(2)$ and $dG(3)$ using very long time steps corresponding to a CFL-number of 2500. The resolution of thin fingers improve with increased order of accuracy as expected except at the front, where the nonlinear limiter function reduce the accuracy to second order.

4.4. Three-Phase WAG Injection. In our last example, we consider a water-alternating-gas scenario in a quarter five-spot. Here the primary unknowns are the water and

gas saturations, s_w and s_g . To define the flux functions, we introduce the phase mobilities

$$\begin{aligned}\lambda_w(s_w) &= s_w^2/\mu_w, & \lambda_g(s_g) &= (0.1s_g + 0.9s_g^2)/\mu_g, \\ \lambda_o(s_w, s_g) &= (1 - s_w - s_g)(1 - s_w)(1 - s_g)/\mu_o,\end{aligned}$$

where $\mu_w = 0.35$, $\mu_g = 0.012$ and $\mu_o = 0.8$. The two components of the flux function are $f_\alpha = \lambda_\alpha / (\sum_\alpha \lambda_\alpha)$ for $\alpha = w, g$. This system has only positive characteristics and is strictly hyperbolic except for the single point of 100% gas saturation, where the eigenvalues coincide [Juanes and Patzek, 2004]. Figure 4 shows the time evolution of a WAG injection cycle starting with the injection of 0.05 pore volumes of water, then 0.05 pore volumes of gas, etc. Before each injection step, the pressure and velocity fields are recomputed to account for the change in total mobility. The transport is computed with dG(2) using a CFL-number of 5000.

5. CONCLUDING REMARKS

In this paper we have demonstrated the capabilities of an implicit discontinuous Galerkin discretisation for linear and nonlinear transport in porous media. A sequential solution procedure based on reordering the equations yields a very fast method with the attractive feature that the runtime scales linearly with the number of elements. In the linear case, this offers a competitive alternative to streamline methods for delineating reservoirs. For nonlinear time-dependent problems this scheme yields a nonlinear solver that allows implicit time-stepping in large multiphase flow computations on desktop computers.

For very large problems, domain decomposition may be used to circumvent the memory limitations of a single computer. For time-dependent problems, the sequential solution procedure allows many time steps to be computed in parallel. Neither of these options have been tested yet.

REFERENCES

- Christie, M. A., and M. J. Blunt (2001), Tenth SPE comparative solution project: A comparison of upscaling techniques, *SPE Reservoir Eval. Eng.*, 4(4), 308–317, url: www.spe.org/csp.
- Dennis Jr., J. E., J. M. Martínez, and X. Zhang (1994), Triangular decomposition methods for solving reducible nonlinear systems of equations, *SIAM J. Opt.*, 4(2), 358–382.
- Duff, I. S., and J. K. Reid (1978), An implementation of Tarjans algorithm for block triangularization of a matrix, *ACM Trans. Math. Softw.*, 4(2), 137–147.
- Juanes, R., and T. W. Patzek (2004), Analytical solution to the Riemann problem of three-phase flow in porous media, *Transp. Porous Media*, 55(1), 47–70.
- King, M., and A. Datta-Gupta (1998), Streamline simulation: a current perspective, *In Situ*, 22, 91–140.
- Natvig, J. R., and K.-A. Lie (2006), Fast computation of multiphase flow in porous media using discontinuous Galerkin and optimal ordering, submitted.
- Natvig, J. R., K.-A. Lie, B. Eikemo, and I. Berre (2006), A discontinuous Galerkin method for computing single-phase flow in porous media, submitted.

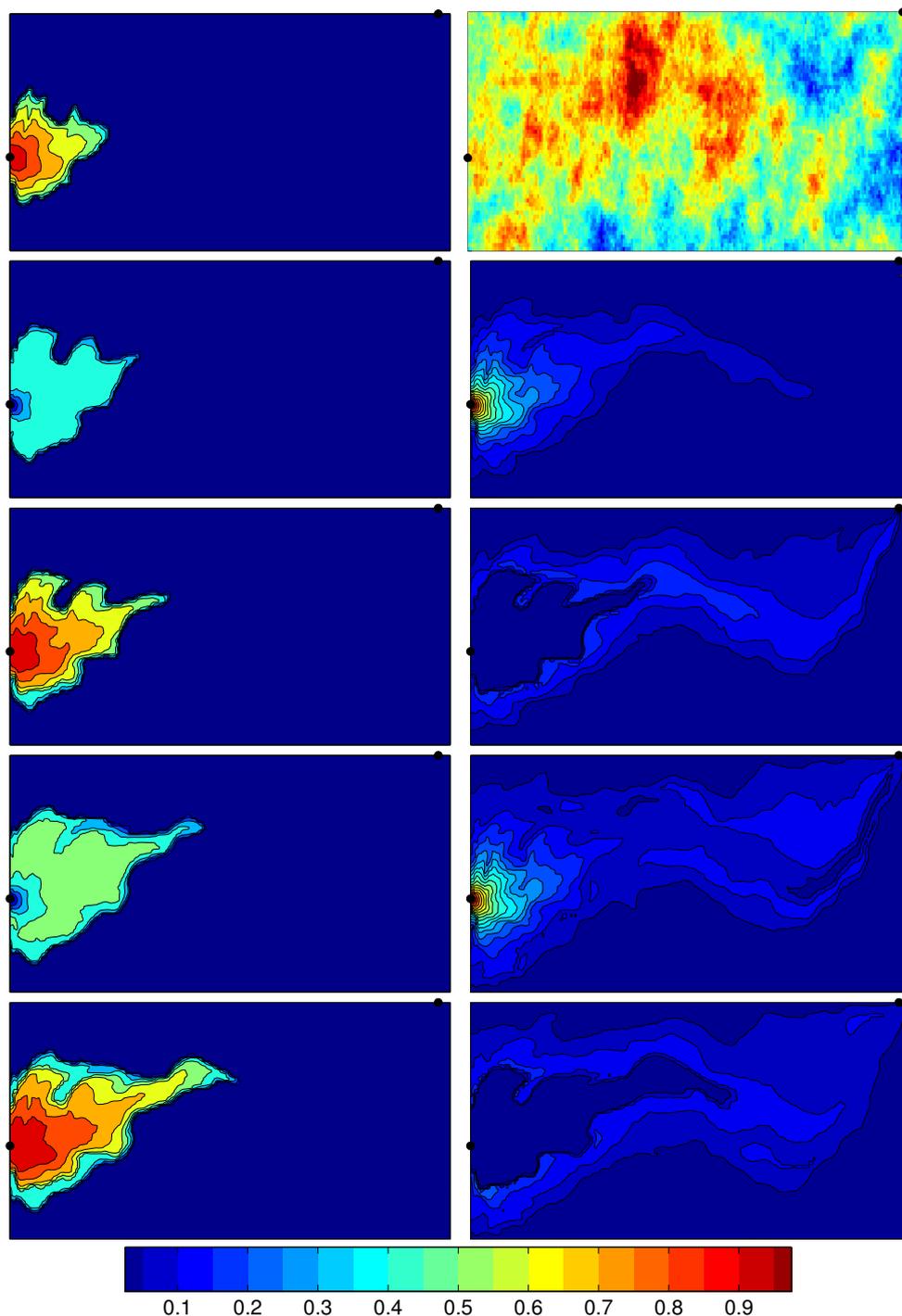


FIGURE 4. Three-phase flow in the smooth Tarbert formation in the 6th layer of the SPE 10 test case. The figures show (top, right) the permeability field with high permeability indicated by light shading, (left column) the water saturation and (right column) the gas saturation in five steps of a WAG cycle. The plotted contours are $[0 : 0.1 : 1]$ for the water saturation and $[0 : 0.05 : 1]$ for the gas saturation. The permeability field span values from $1e - 6$ to $1e - 12$.