Discontinuous Galerkin Methods for Advective Transport in Single-Continuum Models of Fractured Media ⋆

Birgitte Eikemo [a,*] Knut–Andreas Lie [b,a] Geir Terje Eigestad [a]
Helge K. Dahle [a]

[a] *University of Bergen, Deptartment of Mathematics, Johs. Brunsgt. 12, NO–5008 Bergen, Norway*

[b] *SINTEF ICT, Department of Applied Mathematics, P.O. Box 124 Blindern, NO–0314 Oslo, Norway*

Abstract

Accurate simulation of flow and transport processes in fractured rocks requires that flow in fractures and shear zones to be coupled with flow in the porous rock matrix. To this end, we will herein consider a single-continuum approach in which both fractures and the porous rock are represented as volumetric objects, i.e., as cells in an unstructured triangular grid with a permeability and a porosity value associated with each cell. Hence, from a numerical point of view, there is no distinction between flow in the fractures and the rock matrix. This enables modelling of realistic cases with very complex structures. To compute single-phase advective transport in such a model, we propose to use a family of higher-order discontinuous Galerkin methods. Single-phase transport equations are hyperbolic and have an inherent causality in the sense that information propagates along streamlines. This causality is preserved in our discontinuous Galerkin discretization. We can therefore use a simple topological sort of the graph of discrete fluxes to reorder the degrees-of-freedom such that the discretised linear system gets a lower block-triangular form, from which the solution can be computed very efficiently using a single-pass forward block substitution. The accuracy and utility of the resulting transport solver is illustrated through several numerical experiments.

Key words: transport in porous media, fractured media, time-of-flight, discontinuous Galerkin discretization, unstructured grids, linear solvers, directed asyclic graphs

1 Introduction

Fractured reservoirs are complex geological structures, for which fractures (cracks and joints created by rock stress) have higher permeability and porosity than the surrounding rock. Although the aperture of fractures is very small compared with the dimensions of the reservoir, the fracture network often forms the primary pathway for fluid flow and mass transfer and has a significant impact on the flow characteristics of the porous medium. The matrix blocks between the conducting fractures, on the other hand, can significantly increase the storage capacity of the rock. Accurate representation of fractures is a challenge in characterisation, modelling, and simulation of petroleum and groundwater reservoirs [1–3].

Fractured media have traditionally been modelled using either discrete or multicontinua models. In multicontinua models, one assumes that an representative elementary volume can be obtained both for the porous medium (the rock matrix) and for the fractured system. In a dual-porosity model, for instance, the rock is characterised as two overlapping continua, which are both treated as porous media, meaning that also the matrix blocks are assigned a value of porosity greater than zero. The interaction between the matrix and the fracture systems are described using exchange terms [4]. For a rock with large porous blocks between the conducting fractures, multicontinua models have been used to account for the release of fluid from storage in the matrix blocks into the fracture network. The primary advantage of multicontinua flow models is that they provide a mechanism to account for the delay in the hydraulic response of the rock caused by fluid that is resident in the less permeable matrix blocks.

The interaction of fracture and matrix porosities and permeabilities is very complex and often makes simple models highly inaccurate. Indeed, it is widely recognised that state-of-the-art simulation methods based upon multicontinua descriptions are not able to deliver sufficient resolution of the complex flow patterns that develop when a fractured reservoir is produced. Several approaches have therefore been taken to accurately describe fracture systems on a grid-cell scale, that is, based upon complex gridding schemes in which fractures are represented explicitly as lower-dimensional objects at the cell faces. With

⋆ We would like to thank Håkon Hægland for the streamline simulations and Rainer Helmig for useful feedback on fractured model issues. The research is funded in part by the Research Council of Norway through the GeoScale project (grant no. 158908/130).

* Corresponding author.
Email addresses: birgitte@math.uib.no (Birgitte Eikemo), Knut-Andreas.Lie@sintef.no (Knut–Andreas Lie), geirte@math.uib.no (Geir Terje Eigestad), helge.dahle@math.uib.no (Helge K. Dahle).

such discrete models, it is possible to model single- and multiphase flow and transport processes more accurately.

Herein we consider an even more ambitious modelling approach that has increased in popularity lately; in this approach fractures are represented explicitly as thin *volumetric* objects in a highly detailed geological model. In the following we consider single-phase flow in semi-realistic 2D models of fractured reservoirs and use unstructured, conforming triangular grids in which the fractures themselves are represented explicitly as cells with small width and high permeability (and porosity). This will lead to models with high contrasts in reservoir properties, and very complex hydraulic conductivities. For such a model, flow and transport in fractures and rock matrix are solved simultaneously using a single-continuum approach.

A simplified single-phase model is often sufficient to reveal the major displacement patterns in a subsurface model. Computing single-phase flow essentially amounts to solving an elliptic pressure equation. However, to further understand the flow mechanisms one can consider various derived quantities like streamlines, timelines, influence regions, reservoir partitioning, tracer profiles, well pairs, etc., that may be more visual and intuitive than pressure values and discrete fluxes. Such quantities are usually associated with, and computed by a streamline method [5]. The purpose of our paper is to develop an alternative approach based on conservative finite-volume methods. In particular, we will focus on the time-of-flight, which can be used to identify areas affected by contamination in groundwater flow or to determine drainage and flooded volumes in petroleum reservoirs if the effects of diffusion, dispersion, degradation, and sorption, are insignificant. The ideas presented herein are a continuation of the research in [6], in which we presented a family of discontinuous Galerkin schemes for simulating flow in idealised fractured media using rectangular grids.

To discretize the time-of-flight equation, we will use a higher-order discontinuous Galerkin (dG) method, which results in a linear system having a block structure in which each block corresponds to the degrees-of-freedom in a single element (i.e., cell). Blocks corresponding to neighbouring elements in the grid are coupled through the numerical flux function used to approximate the physical flux over element faces. By introducing an upwind flux approximation, the elements can be ordered to ensure that the linear system has a lower block-triangular form. Each block corresponds to the degrees-of-freedom in a single element or in a collection of elements having circular dependence due to rotation in the discrete flow field. Given the triangular form, the linear system can be solved very efficiently using a forward block substitution (i.e., a Gauss–Seidel iteration). This solution procedure also has very low memory requirements: once the elements have been reordered, the linear system can be assembled and solved in a local block-by-block fashion. The exact same

idea can be applied to the *nonlinear* boundary value problem arising from a implicit temporal semi-discretization of multiphase advective transport. This is discussed in detail by Natvig and Lie [7,8], who also present studies of the efficiency of the reordering method. Discontinuous Galerkin methods based on *explicit* temporal discretizations are discussed in [9].

In a certain sense, the reordering idea can be seen as a special case of the fast sweeping methodology developed for eikonal [10,11] and nonlinear Hamilton–Jacobi equations [12,13]. In essence, the fast sweeping method consists of upwind discretizations combined with nonlinear Gauss–Seidel iterations with alternating orderings to treat groups of characteristics in parallel. Altogether this gives an algorithm of complexity $\mathcal{O}(N)$ for N grid cells. There are two main differences between the fast sweeping method and our reordering method. First, the possibility of cyclic dependencies in the discrete flow field, a strict causality principle is not necessarily fulfilled for the discretized transport equation. Second, the transport equations considered herein (and in [7,8]) only have a single group of characteristics (i.e., streamlines) and thus only require a single ordering.

The rest of this paper is organised as follows: In Section 2 the equations used to model single-phase flow are described in detail. Next, Section 3 introduces the discontinuous Galerkin method used to discretize the fluid transport equations. Then, numerical results for single-phase transport in fractured 2D media are given in Section 4. We also verify the accuracy and convergence rates of our schemes using a simple unfractured case with known analytical solution. Finally, in Section 5 we summarise and give main conclusions.

2 Single-Phase Flow Models

Single-phase flow in an incompressible porous medium is typically modelled by a mass-balance equation in combination with Darcy's law. If we assume gravity to be negligible, the governing equations can be written

$$\nabla \cdot \mathbf{v} = f, \qquad \mathbf{v} = -\frac{1}{\mu}\mathbf{K}\nabla p, \qquad \mathbf{x} \in \Omega. \tag{1}$$

Given a specification of the fluid sources f, the rock permeability $\mathbf{K}$, the fluid viscosity μ, and proper conditions at the boundary $\partial\Omega$ of the physical domain Ω, the system (1) can be solved to compute the pressure p and the volumetric flow density $\mathbf{v}$. To simplify the presentation, we assume that there are no internal fluid sources or sinks and that the flow governed by (1) is driven entirely by conditions set on the inflow and outflow boundaries, denoted $\partial\Omega^-$ and $\partial\Omega^+$, respectively.

For many purposes, (1) does not give a sufficient description of the flow patterns and it is therefore customary to introduce additional transport equations to describe quantities like tracers, contaminants, etc. that are passively advected with the single-phase flow. For simplicity, we will henceforth assume that $\mathbf{v} = \mathbf{v}(\mathbf{x})$ is given and is steady, divergence free, and irrotational. (Later we will also assume that $\mathbf{v}$ is given implicitly in the form of fluxes that are constant on each element interface.) Given a fixed flow velocity, the concentration q of a passively advected quantity evolves according to the linear hyperbolic equation

$$\phi q_t + \mathbf{v} \cdot \nabla q = 0, \qquad q|_{\partial\Omega^-} = q^-(\mathbf{x}, t), \tag{2}$$

where ϕ is the porosity of the medium. The steady-state version of (2),

$$\mathbf{v} \cdot \nabla q = 0, \qquad q|_{\partial\Omega^-} = q^-(\mathbf{x}), \tag{3}$$

describes the stationary distribution of a tracer that is injected into a reservoir at the inflow boundary $\partial\Omega^-$. This equation can, for instance, be used to determine the spatial region influenced by an inflow boundary (or a fluid source), or by reversing the sign of $\mathbf{v}$, the region influencing an outflow boundary (or drained by a fluid sink). Within reservoir simulation, this could typically be used to compute the swept region of an injector or the drainage region of a producer (or combinations thereof).

Another quantity of interest is the time-of-flight, which is the time it takes a passive particle to travel from a point on the inflow boundary to the given point in the reservoir. For a steady velocity, the particle will travel along a streamline, that is, a curve that at any point is tangential to the velocity $\mathbf{v}$. The time-of-flight τ is therefore defined as

$$\tau(\mathbf{x}) = \int_{\psi} \frac{\phi(r)\, dr}{|\mathbf{v}(\mathbf{x}(r))|}, \tag{4}$$

where ψ denotes the streamline that connects $\mathbf{x}$ to an inflow boundary (or fluid source) and r is arclength along the streamline. Iso-contours of τ define natural timelines in a reservoir and therefore modern streamline methods use τ rather than r as a spatial coordinate. Equation (4) may alternatively be written in differential form as,

$$\mathbf{v} \cdot \nabla \tau = \phi, \quad \tau|_{\mathbf{x} \in \partial\Omega^-} = 0. \tag{5}$$

The transport equations (3) and (5) are special cases of the more general equation

$$\mathbf{v} \cdot \nabla q = H(q, \mathbf{x}), \qquad q|_{\Omega^-} = h(\mathbf{x}, t). \tag{6}$$

Similarly, (2) takes the form (6) if we introduce a semi-discretization in time. Accurate solution of (6) is important in areas such as oil recovery and groundwater hydrology to reveal the transport properties of $\mathbf{v}$. Solving (6) is rather easy for smooth velocities, but becomes harder when $\mathbf{v}$ has large spatial variations and exhibits fine-scale details that are important for the global flow pattern.

In the next section, we present an efficient strategy for solving (6) numerically on unstructured triangular grids.

3 Discontinuous Galerkin Schemes with Optimal Ordering

To develop higher-order discontinuous Galerkin methods, we start with a variational formulation of (6). We partition Ω into an unstructured grid consisting of non-overlapping triangular elements (cells) $\{T_k\}$ and seek solutions in a finite-dimensional space V_h consisting of piecewise smooth functions that may be discontinuous over element interfaces. Let $\mathbb{Q}^n = \text{span}\{x^r y^s : 0 \leq r+s \leq n\}$ be the space of polynomials of degree at most n, and let $V_h^{(n)} = \{\varphi : \varphi|_{T_k} \in \mathbb{Q}^n\}$. Thus, $V_h^{(0)}$ is the space of elementwise constant functions, which will give a scheme that is formally first-order accurate. Similarly, $V_h^{(1)}$ is the space of elementwise linear functions giving a formally second-order accurate scheme, and so forth. Henceforth, we use dG(n) to denote the discontinuous Galerkin approximation of polynomial order n. Inside each element T_k, the discrete solution q_h can be written

$$q_h(T_k) = \sum_{i=1}^{m_k} q_i^k L_i^k, \qquad \forall T_k. \tag{7}$$

where $\{L_i^k\}$ is some basis for $V_h^{(n)}$ on T_k, and m_k is the number of associated degrees-of-freedom. The unknown coefficients $\{q_i^k\}$ are collected in the vector Q for the whole domain and in (sub)vector Q_T for element T.

The approximate solution q_h is determined as the unique solution of the following weak formulation of (6)

$$a_T^h(q_h, \varphi_h) = b_T^h(q_h, \varphi_h) \quad \forall T, \quad \forall \varphi_h \in V_h^{(n)}, \tag{8}$$

where

$$\begin{aligned} a_T^h(q_h, \varphi_h) &= -\int_T (q_h \mathbf{v}) \cdot \nabla \varphi_h d\mathbf{x} + \int_{\partial T} \mathbf{v} \cdot \mathbf{n}\, q_h \varphi_h ds, \\ b_T^h(q_h, \varphi_h) &= \int_T H(q_h, \mathbf{x}) \varphi_h d\mathbf{x}. \end{aligned} \tag{9}$$

Since the solution is discontinuous over element interfaces, we will use an upwind flux to approximate the integrand of the second integral in $a_T^h(\cdot,\cdot)$,

$$\begin{aligned}\mathbf{v}\cdot\mathbf{n}\, q_h &\approx \hat{f}(q_h, q_h^{ext}, \mathbf{v}\cdot\mathbf{n}) \\ &= q_h \max(\mathbf{v}\cdot\mathbf{n}, 0) + q_h^{ext}\min(\mathbf{v}\cdot\mathbf{n}, 0).\end{aligned} \tag{10}$$

Here q_h and q_h^{ext} are the inner and outer approximations of the unknown quantity q (i.e., the concentration of the advected quantity in (2), the tracer concentration in (3), or the time-of-flight in (5)) at the element interfaces. The upwind approximation of the flux preserves the directional dependency of the underlying continuous equation (6). In other words, the solution in T will only be influenced by elements $\mathcal{U}(T)$ that are intermediate neighbours in the upwind direction, which we will later exploit to compute the solution in a blockwise fashion. Formally, $\mathcal{U}(T)$ consists of all elements E such that $(\mathbf{v}\cdot\mathbf{n}_T)|_{\partial E\cap\partial T} < 0$, where $\mathbf{n}_T$ is the outward-pointing normal to T.

For the special case of the time-of-flight equation (5), we introduce an additional clipping, $\max(q_h^{ext}, 0)$, to prevent negative values from propagating downstream. Time-of-flight is non-negative by definition, but negative values may arise when using high-order polynomials. Alternatively, undesired oscillations may be suppressed by introducing local *hp*-refinement, see [14] for more details.

To find a solution to (8), we choose trial functions $\varphi_h = L_i^k$ and evaluate the inner products (9) using appropriate quadrature rules. This gives a set of linear equations for the degrees-of-freedom in each element,

$$A_T Q = B_T, \qquad (A_T)_{ij} = a_T^h(L_i, L_j), \quad (B_T)_i = b_T^h(L_i, L_j).$$

For convenience, we split the coefficient matrix into the element stiffness matrix R_T and the coupling to other elements through the numerical flux integral $F_T(Q)$. Given the upwind approximation of the flux (10), we can split the flux integral in two part, where $F_T^+ Q_T$ denotes the flux out of element T and $F_T^- Q_{\mathcal{U}(T)}$ denotes the flux into element T. Hence, the following system of linear equations is obtained

$$\left(-R_T + F_T^+\right) Q_T + F_T^- Q_{\mathcal{U}(T)} = B_T, \qquad \forall T. \tag{11}$$

The coefficient matrix has a block-banded structure, where the size of each block is given by the number of degrees-of-freedom in each element or connected collection of elements, see [7] for a more detailed discussion.

A fast linear solver can now be constructed by observing that the solution in each element can be computed by inverting $(-R_T + F_T^+)$ once the solution is known in all upstream neighbours $\mathcal{U}(T)$ of T. We may therefore construct the solution locally, starting at inflow boundaries (or fluid sources) and proceeding

downstream, element by element. From a computational point of view, it is more convenient to look at this as an optimal ordering of unknowns that renders the system of equations (11) in lower block-triangular form. If such an ordering exists, it can be found in N_e operations for N_e elements. If the reordering does not exist, there must be circular dependence among some of the elements and these mutually dependent elements must be solved for simultaneously. Nevertheless, the reordering still applies, the only difference is that we locally get a block system associated with a set of interconnected elements. More details are found in [14,7].

We have previously shown that the dG-reordering method is both accurate and highly efficient for rectangular grids [14,6,15]. In [7], we demonstrated that the same technique can be applied to semi-discrete nonlinear transport equations of the form $\mathbf{v} \cdot \nabla \mathbf{F}(\mathbf{q}) = \mathbf{H}(\mathbf{q}, \mathbf{x}, t)$ that describe multiphase and multicomponent flow when gravity, capillarity, and dispersivity are neglected.

4 Numerical Examples

We have previously [6] used dG schemes to compute time-of-flight in fractured media represented as a rectangular grid, which restricted the orientation of fractures to be either horizontal or vertical. Here, we will consider more realistic fracture distributions modelled on triangular grids. Hybrid grids consisting of both rectangular and triangular cells are within reach because of the localised nature of the dG formulation, but are not considered herein. However, we show one example of adaptively refined grids. For each example, the forcing velocity field will either be given by an analytical expression or be computed by a standard conservative method for (1), in which case the velocity will be divergence free and nearly irrotational.

In all cases, except Case 2, we will use equations in non-dimensional form. This means that one time unit corresponds to the time it takes to inject one pore volume into the computational domain.

Case 1 (Convergence Study) *We start by assessing the accuracy and verifying convergence rates on triangular elements. To this end, we consider a rotating velocity field* $\mathbf{v} = (y, -x)$ *in the domain* $[1, 2] \times [1, 2]$*. We set* $\tau = 0$ *on the inflow boundaries (*$x = 1$ *and* $y = 2$*). Then the exact time-of-flight is given by:*

$$\tau(x, y) = -\arctan\left(\frac{y}{x}\right) + \arctan\left(\frac{\min(\sqrt{x^2 + y^2 - 1}, 2)}{\max(\sqrt{\max(x^2 + y^2 - 4, 0)}, 1)}\right). \quad (12)$$

Figures 3 and 4 present L_2*-errors and convergence rates for a grid-refinement*

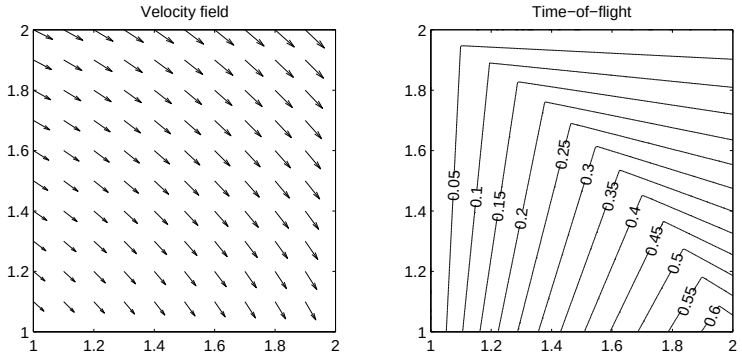

Fig. 1. Velocity field and the time-of-flight reference solution (12) for the convergence study.

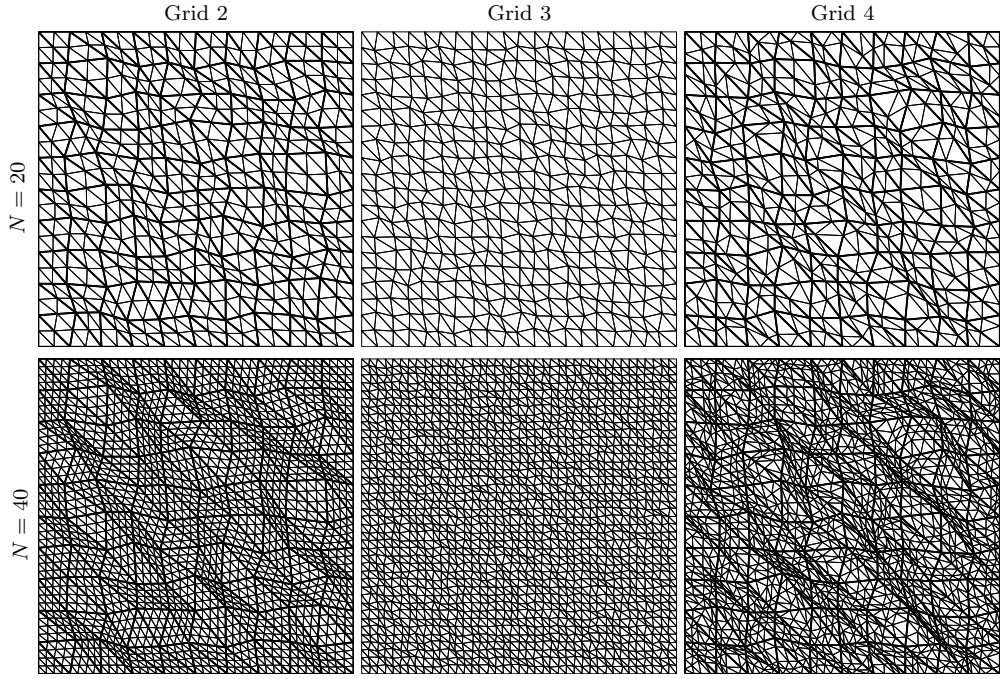

Fig. 2. Two refinement levels for Grids 2, 3, and 4.

study performed by increasing the order n in dG(n) on four grid types with increasing roughness (see Figure 2):

Grid 1 *triangulation of a uniform* $N \times N$ *Cartesian grid.*

Grid 2 *uniform refinement and triangulation of a* 10×10 *base grid in which each internal node has been given a random perturbation up to 20% in each spatial direction.*

Grid 3 *same as Grid 1, but with a perturbation up to 20% of* all *inner nodes on the* $2N \times N$ *grid.*

Grid 4 *same as Grid 2, but with a perturbation up to 20% of* all new *nodes on the* $2N \times N$ *grid for each refinement.*

Figure 3 shows L_2*-errors measured in a smooth part of the domain,* $[1, 1.3] \times$

$[1, 1.3]$, *while Figure 4 shows errors integrated over the whole domain. The figures indicate how different roughness*[1] *in the refined grids impacts the convergence rates. For the perturbed grids, convergence rates are computed by comparing to an h defined by averaging of the maximum edge of all elements. We also measured convergence versus the largest cell edge in the domain, but these results were somewhat inconclusive and are not included.*

Grids 1 and 2 are refined such that the elements approach half of parallelograms in the asymptotic limit, and hence we observe the expected order of accuracy in smooth regions. For the whole domain, however, we observe reduced convergence rates because of the kink in the solution along the circular arc $x^2 + y^2 = 5$ *(see Figure 1), as is to be expected. These results agree with the results in [14] for rectangular elements.*

Figure 5 shows histograms of the mesh sizes for Grid 3 and Grid 4, both with grid size $N = 80$*. Grid 3 has a normal distribution of the measured mesh sizes, which explains that the observed convergence rates are only marginally lower than the formal order of each method. The first refinement level of Grid 4 has a normal distribution, but as the grids are refined, the mesh distribution becomes more and more skewed, with a significant tail to the right, corresponding to elements with longer edges. Because of this kind of distribution (short triangle edges combined with longer triangle edges), the mesh quality diminishes as the grid is refined, which introduces an opposite effect to the pure reduction of edge lengths seen for the other grids. The observed convergence orders are therefore significantly lower than their theoretical value and also seem to decrease with the refinement level, indicating a stronger effect of the skewed mesh distribution for increasing numbers of grid cells. Note also that the decay increases with the order of the basis functions.*

These results should be compared with the theoretical results obtained in [16] and [17], where the convergence of the pressure equation (1) is studied for general permeability description and irregular geometry. When transforming the pressure equation on general quadrilateral grids with general permeability to computational space, the evaluation of a quantity which may be viewed as *the computational space permeability* depending on the Piola mapping, becomes important. Different evaluations of the computational space permeability may have a very different behaviour on rough grids, and convergence may be lost entirely for rough grids that do not handle this evaluation properly.

[1] Rough grids are defined in the literature (see e.g., [16]) as quadrilateral grids that do not approach parallelograms as the grids are refined. Here, the triangular grids are constructed by dividing each quadrilateral of a quadrilateral grid into two triangles which have one common edge. By this definition, Grids 3 and 4 are rough grids. Similar convergence studies have been performed in [17], where in general a decay in convergence rates may be seen.

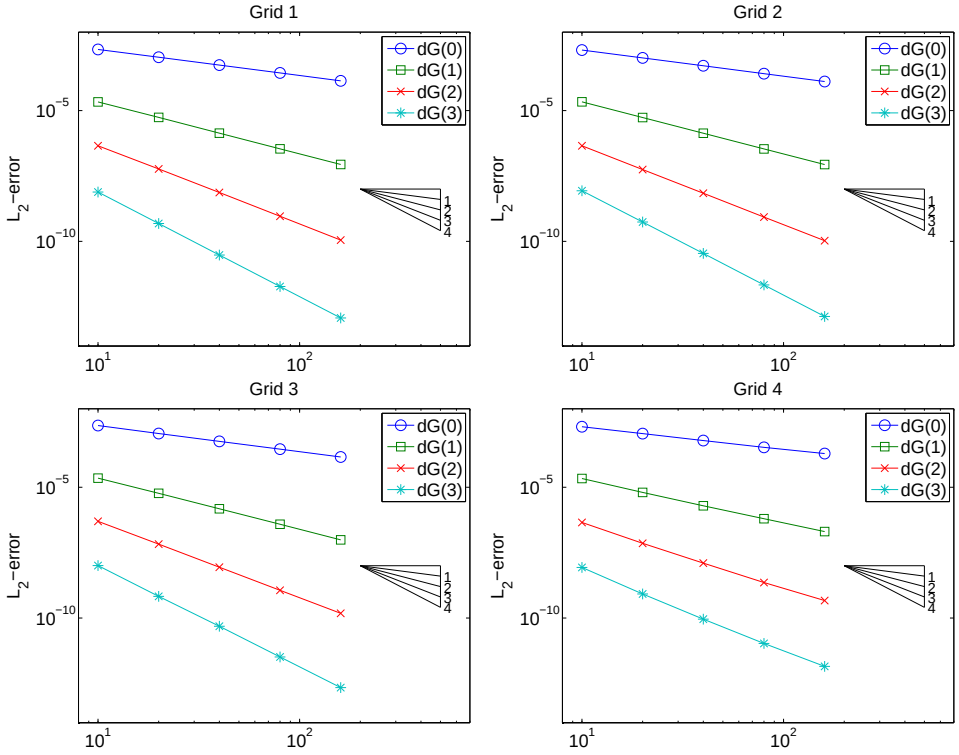

Fig. 3. L_2-errors and convergence rates over a smooth part of the domain, $[1, 1.3] \times [1, 1.3]$, for a grid refinement study with dG(n) on a series of $2N \times N$ grids for Grids 1, 2, 3, and 4.

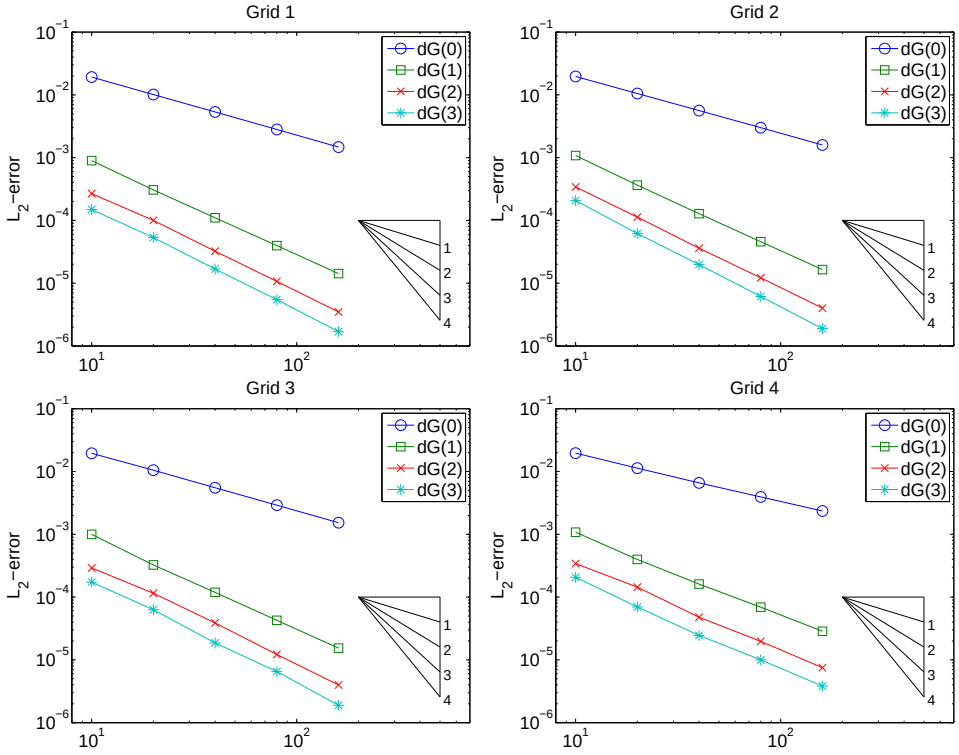

Fig. 4. Same as Figure 3, but with the L_2-errors and convergence rates measured over the whole domain, $[1, 2] \times [1, 2]$.

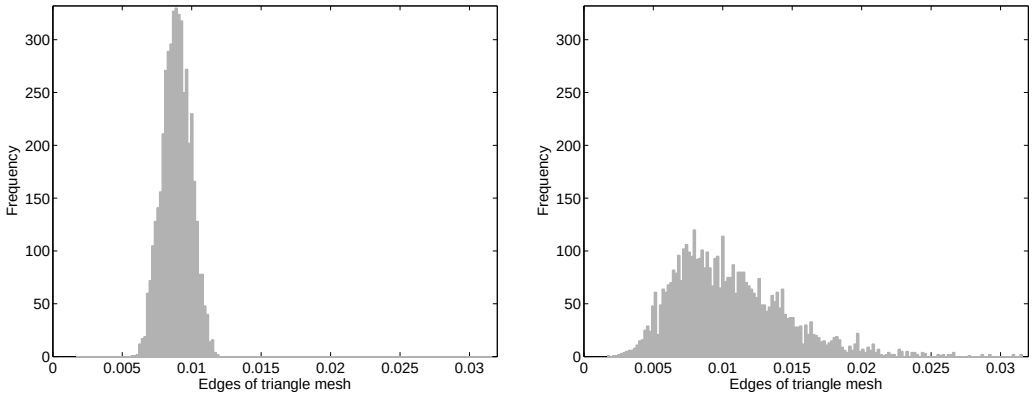

Fig. 5. Histogram of edges of triangle meshes for Grid 3 (left) and Grid 4 (right).

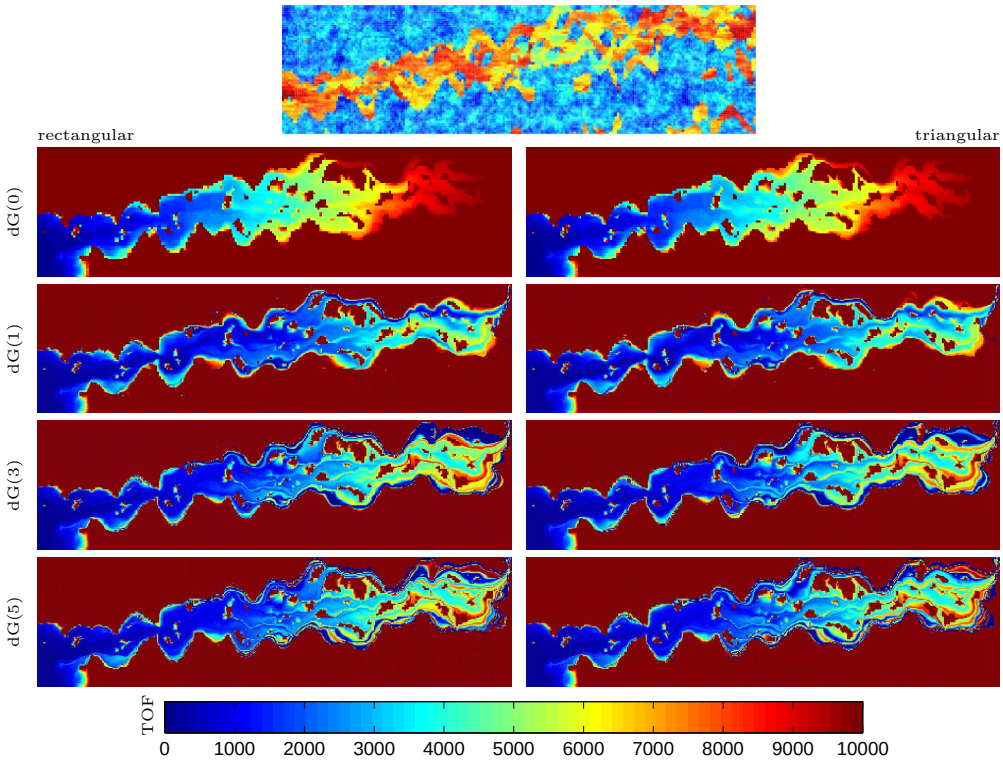

Fig. 6. Time-of-flights for Layer 77 from the SPE10 test case.

In the next example, we consider a case with strongly heterogeneous media properties.

Case 2 (A Fluvial Medium) *Consider a 2D quarter five-spot case with permeability and porosity data from Layer* 77 *of Model 2 from the* 10*th SPE Comparative Solution Project [18]. This permeability field is highly heterogeneous with permeability variation up to eleven orders of magnitude and strong contrasts between the low-permeability background and a set of intertwined high-permeability channels, see Figure 6. The rectangular grid consists of* 220×60 *cells, each of size* 10 *ft* $\times$20 *ft, while the triangular grid is created by dividing each rectangular element into two triangles. The right column in the*

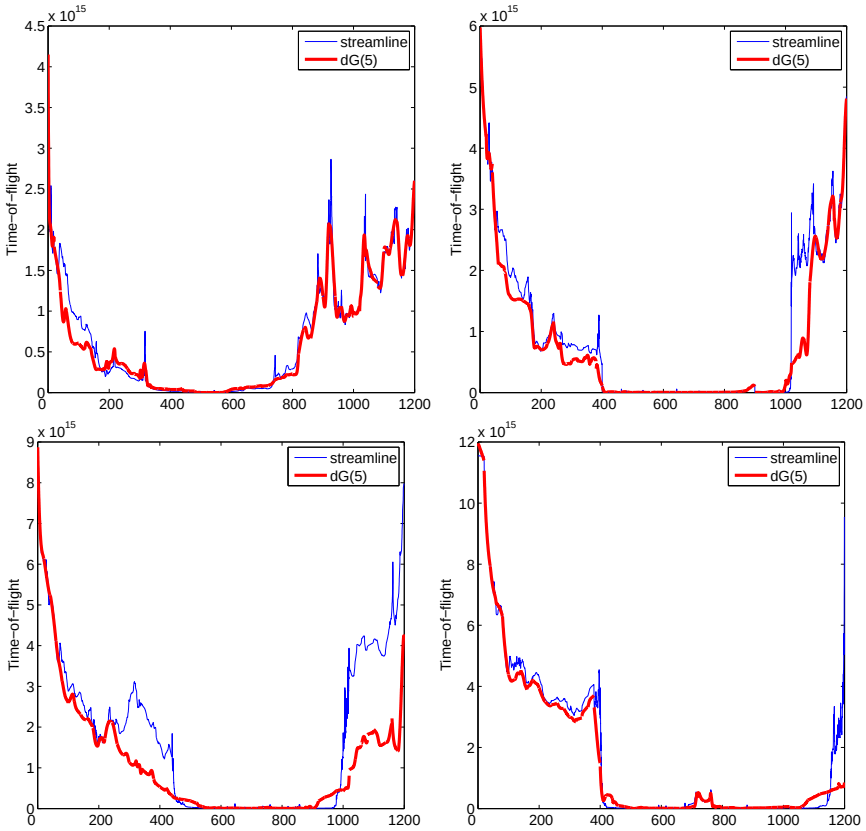

Fig. 7. The computed time-of-flights along vertical cross sections at $x = 550, 1100, 1650, 2200$ ft.

figure shows the computed time-of-flights on triangular elements for dG(n), $n = 0, 1, 3, 5$. *For comparison, the left column shows the corresponding solutions using the dG scheme on rectangular elements, see [14]. The plots were created by sampling the polynomial patches in* 10×10 *uniformly distributed points inside each rectangular element. In the visual norm, the accuracy is approximately the same on the triangular and on the rectangular grid.*

Figure 7 shows time-of-flights for four different vertical cross sections at $x = 550, 1100, 1650, 2200$ *ft. The thick lines show the dG(5) solution computed on the triangular grid and the thin lines show the solution obtained by back-tracing approximately* 8 000 *streamlines. The corresponding relative* L_1*-errors are present in Table 7. Altogether, we observe that strong heterogeneities in the permeability field influence the accuracy of the computed time-of-flights.*

Case 3 (Three Fractures) *In this example, we consider a case with three high-permeability fractures inside the unit square. We impose no-flow boundaries at bottom and top, inflow at the left boundary, and outflow at the right boundary. The aperture of the fractures is* 10^{-4} *length units and the permeability of the fracture* (K_f) *is assumed to be either* 10^3 *and* 10^5 *times higher than the permeability* (K_m) *of the homogeneous and isotropic background field.*

Table 1
The relative L_1-errors of the computed time-of-flights for different vertical cross sections. See Figure 7 for the computed time-of-flights.

x	dG(0)	dG(1)	dG(2)	dG(3)	dG(4)	dG(5)
550	2.07e-01	1.85e-01	1.74e-10	1.59e-01	1.40e-01	1.26e-01
1100	4.35e-01	3.30e-01	2.95e-01	2.58e-01	2.20e-01	2.02e-01
1650	6.15e-01	5.27e-01	5.01e-01	4.70e-01	4.32e-01	3.89e-01
2200	3.28e-01	1.83e-01	1.68e-01	1.44e-01	1.33e-01	1.25e-01

Table 2
Degrees-of-freedom for different order and grid resolution.

N	dG(0)	dG(1)	dG(2)	dG(3)	dG(4)	dG(5)
437	437	1311	2622	4370	6555	9177
5048	5048	15144	30288	50480	75720	106008
5301	5301	15903	31806	53010	79515	111321
23463	23463	70389	140778	234630	351945	492723

We compare time-of-flights computed using the four grids depicted in the top row of Figure 8. From the plots, we observe three qualitative tendencies: (i) for the same number of unknowns (see Table 2), the solution is better for the grid without *adaptivity; (ii) increased polynomial order is more important than increased grid resolution; and (iii) the improvements obtained by using finer grid resolutions decay with the polynomial order of the scheme. Finally, we observe that all dG solutions establish the qualitative structures of the flow pattern.*

Next, we consider the pointwise accuracy at the outflow boundary compared with a highly resolved solution computed by back-tracing approximately 16 000 *streamlines. Figure 9 shows the relative L_1-errors for the time-of-flight and the mass flux across the outflow boundary. Figure 10 shows the time-of-flight at the boundary for permeability ratio* $1 : 10^3$*. Similarly, Figure 11 shows the tracer production curve (average tracer concentration at the outflow boundary versus time) that results from injecting a tracer slug in the time interval* $t \in [0, 0.05]$*. As above, we observe that high polynomial order is more important than high grid resolution. In particular, Figure 11 shows that using dG(*3*) gives the same qualitative structures for all grid resolutions, whereas dG(*0*) fails to compute the correct tracer production on all grids. We also observe from Figure 9 that the error increases with increasing ratio between the matrix and fracture permeability. This observation agrees with the results in [14].*

When increasing the grid resolution in the example above, the grid inside the

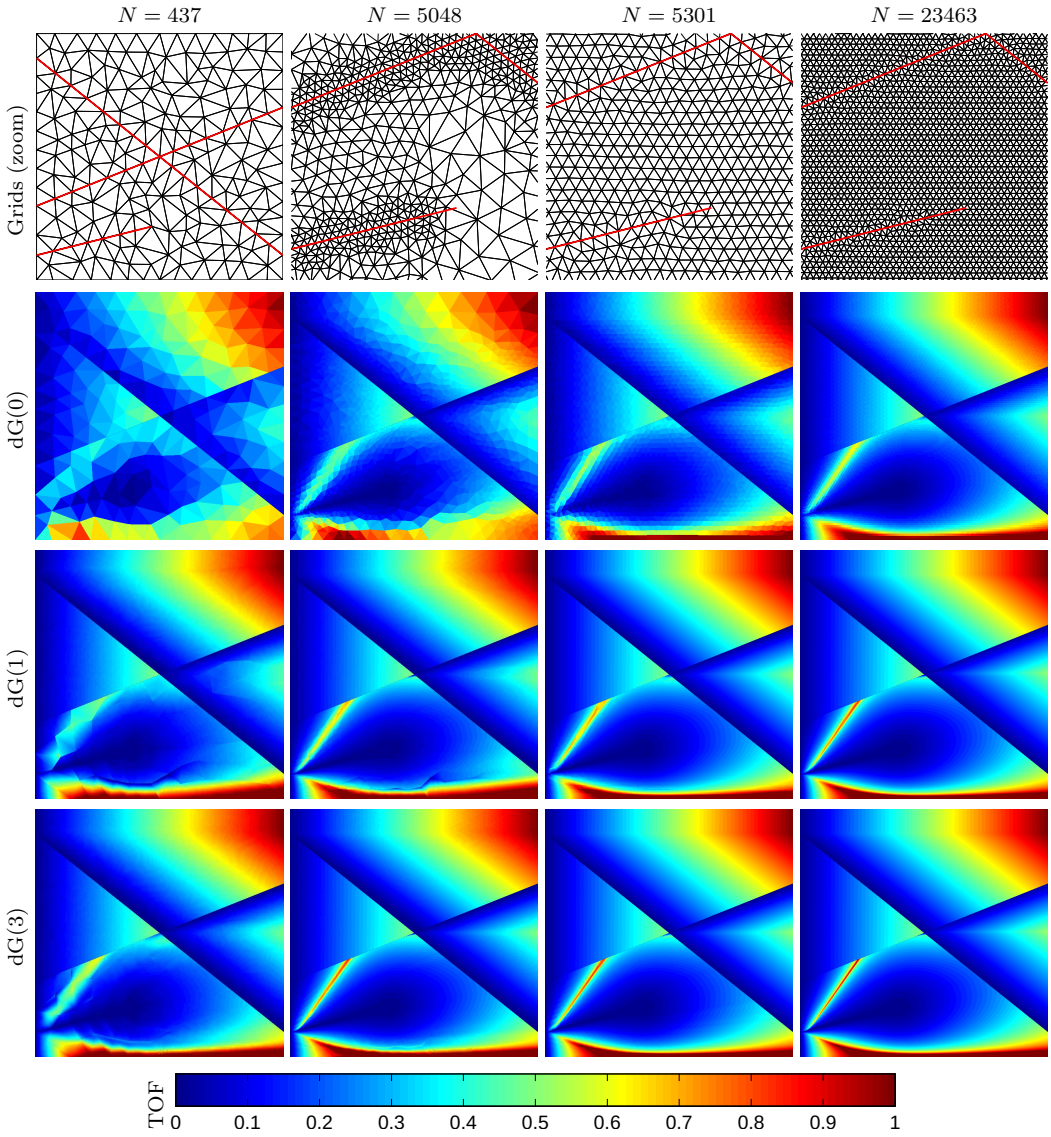

Fig. 8. Time-of-flight for Case 3 with ratio between matrix and fracture permeability equal $1 : 10^5$. In the first row, we only show a zoom of the area $[0.2, 0.6] \times [0.1, 0.5]$ for the three refined grids.

thin fractures only increased resolution in the longitudinal direction. For the simple Cartesian grids studied in [6], we observed that it was more important to increase the grid resolution in the *latitudinal* direction of the fractures to accurately resolve sharp transitions in time-of-flight arising when the flow changes from matrix to fracture and vice versa. In the next example, we therefore also consider refinement in the latitudinal direction of the fractures.

Case 4 (Latitudinal Refinement in Fractures) *Consider a unit square with flow from left to right and no-flow boundaries at bottom and top. The fracture network consists of five horizontal fractures and a skew vertical fracture extending from top to bottom. The aperture of the fractures is* 10^{-4} *unit*

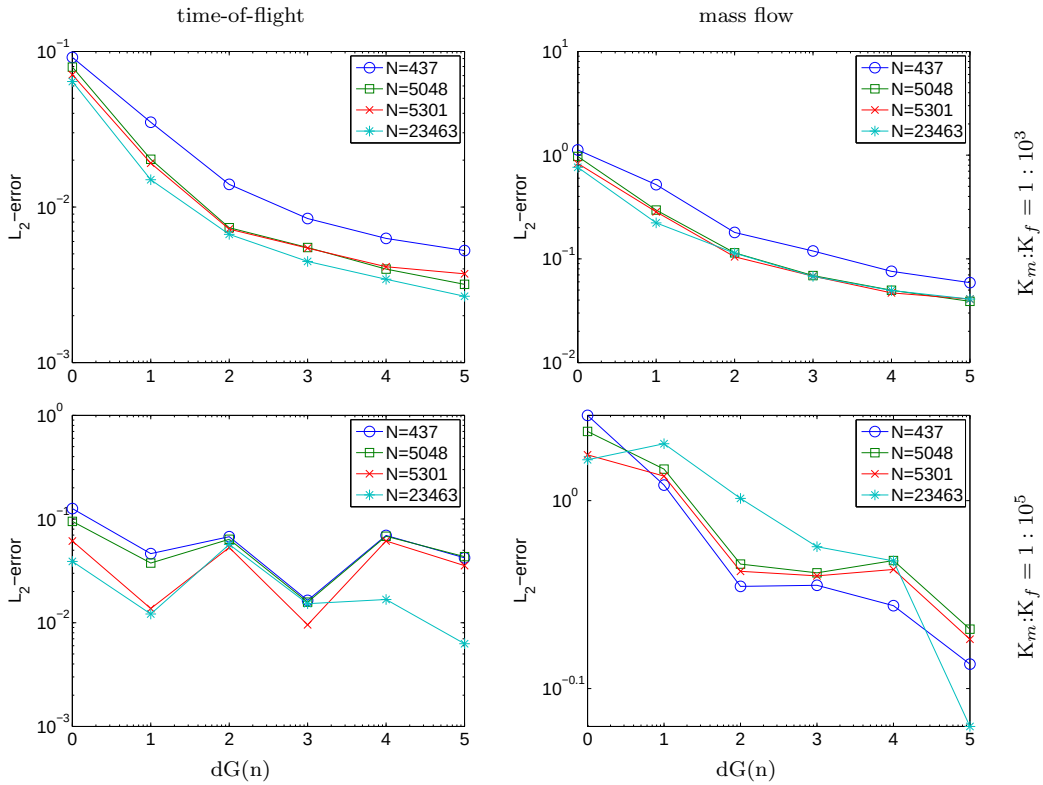

Fig. 9. Discrete relative L_1-errors in time-of-flight and mass flow at the outflow boundary for Case 3. The dG(n) solutions for $n = 0, \ldots, 5$ are compared with solutions computed by back-tracing approximately 16 000 streamlines.

lengths and the permeability ratio is $1 : 10^5$. *Figure 12 shows time-of-flight computed with dG(n) for* $n = 0, 1, 3$. *The upper row shows the time-of-flights computed on a grid in which each fracture is represented with one rectangular element divided into two triangle elements in the latitudinal direction. The lower row shows the solutions with eight rectangular elements (sixteen triangular elements) in the latitudinal direction.*

Table 3 reports the error in the time-of-flights and mass flow computed at the outflow boundary for the permeability ratios $1 : 10^3$ *and* $1 : 10^5$ *compared with a reference solution obtained by back-tracking streamlines from uniformly distributed points inside each element at the outflow boundary for a refined grid. With one exception, the errors decrease when refining the grid in the latitudinal direction inside the fractures. This agrees with results reported in [6], where we observed the importance of sufficient latitudinal grid resolution to correctly capture large spatial variations inside the fractures. Capturing these variations is necessary because the time-of-flight is an integrated quantity that is strongly affected globally by local discretization errors.*

Criteria to guide the choice between single and multicontinua (porosity) formulations in site-specific applications are not easily defined. A simple method is to consider by measuring the (outflow) concentration of some species present

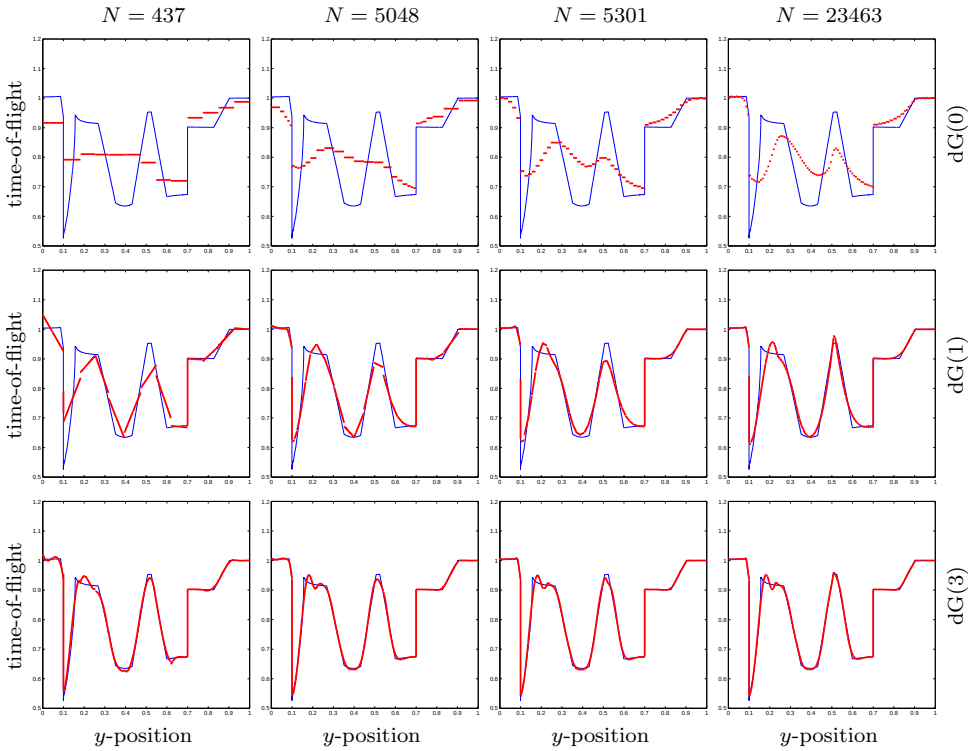

Fig. 10. Time-of-flight at the outflow boundary for Case 3 with ratio between matrix and fracture permeability equal $1 : 10^3$. The thick lines show the dG(n) solutions for $n = 0, 1, 3$ and the thin lines are solutions computed by tracing approximately 16 000 streamlines.

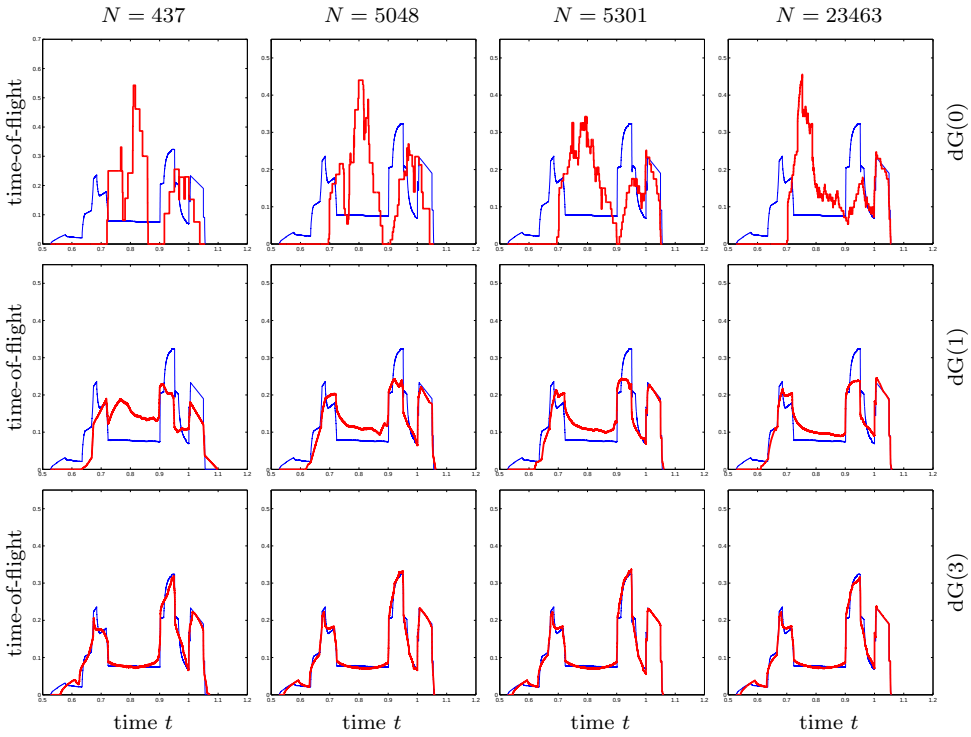

Fig. 11. Average tracer concentration over the outflow boundary as a function of time for the simulations shown in Figure 10.

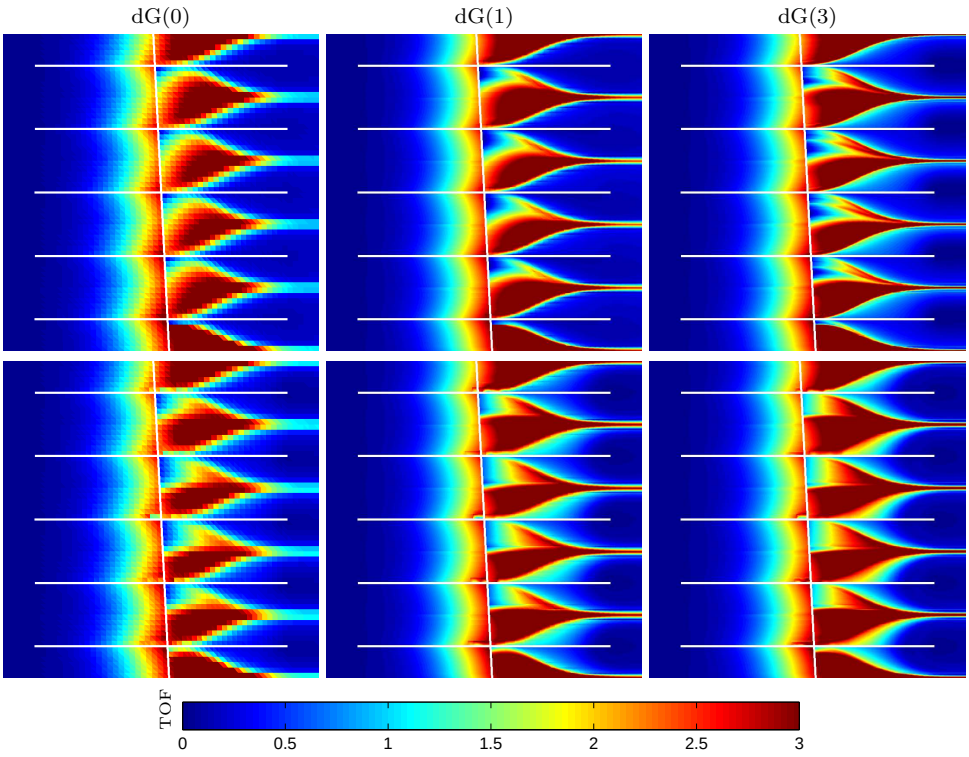

Fig. 12. Time-of-flight for Case 4 with ratio between matrix and fracture permeability equal $1 : 10^5$. The upper row shows the results using no latitudinal grid refinement, while the lower row shows the solutions for a refinement with eight rectangular (divided in sixteen triangular) elements in the latitudinal direction of the fractures. The distribution of fractures is depicted as thick lines.

Table 3
Discrete relative L_1-errors in time-of-flight (upper half) and mass flow (lower half) at the outflow boundary for Case 4 using M elements across the fractures. The solutions are compared to solutions computed by tracing approximately 4 000 streamlines.

K_m:K_f	M	dG(0)	dG(1)	dG(2)	dG(3)	dG(4)	dG(5)
$1 : 10^3$	1 (2)	1.58e-01	6.77e-02	3.83e-02	2.82e-02	2.07e-02	1.94e-02
	8 (16)	1.53e-01	5.74e-02	3.29e-02	2.30e-02	1.70e-02	1.62e-02
$1 : 10^5$	1 (2)	7.80e-01	5.86e-01	3.97e-01	3.59e-01	3.22e-01	2.93e-01
	8 (16)	7.49e-01	4.66e-01	2.11e-01	1.51e-01	1.32e-01	1.07e-01
$1 : 10^3$	1 (2)	1.11e-00	7.59e-01	6.45e-01	5.53e-01	4.34e-01	4.31e-01
	8 (16)	1.20e-00	5.91e-01	3.85e-01	3.00e-01	2.34e-01	2.42e-01
$1 : 10^5$	1 (2)	1.47e-00	1.29e-00	6.21e-01	6.37e-01	3.92e-01	3.77e-01
	8 (16)	8.36e-01	2.64e-01	2.54e-01	2.66e-01	1.82e-01	1.92e-01

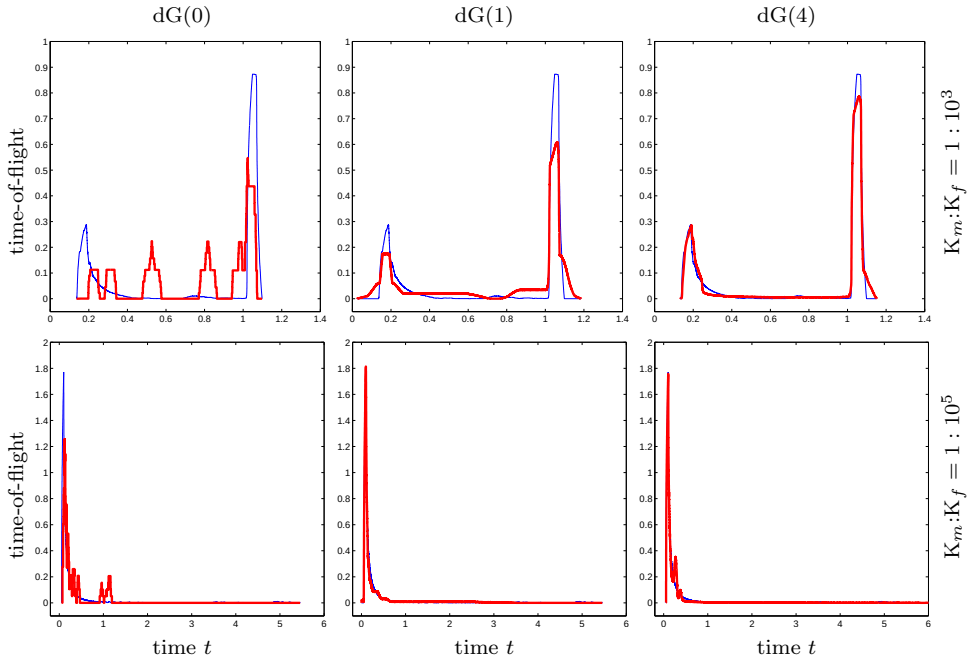

Fig. 13. Computed mass flow over the outflow boundary. The thick lines give the dG(n) solutions and the thin lines give solutions computed by back-tracing approximately 4 000 streamlines.

in a reservoir model during some predefined time interval. Here we consider breakthrough curves resulting from the injection of a tracer slug/pulse. If the curve has two peaks, there are two distinct transport mechanisms corresponding to flow in fractures and matrix. On the other hand, if the curve has a single peak, the medium can be modelled using a discrete model. In the next example we demonstrate that our dG scheme can provide a fast and easy method for evaluating tracer-breakthrough curves for flow in fractured porous media.

Case 5 (Discrete Versus Multicontinua Model) *Consider the same test example as in Case 4, now with grid refinement in the latitudinal direction of the fractures. We measure the concentration over the outflow boundaries resulting from a pulse injection of tracer in the time interval,* $t \in [0, 0.05]$. *Figure 13 shows the mass flow over the outflow boundary computed using dG(n) for* $n = 0, 1, 4$ *compared with a highly resolved streamline simulation on a refined grid. For permeability ratio* $1 : 10^3$, *shown in the upper row, we obtain multiple peaks, where the first peak represents the tracer going through the fractures and the next peak represent tracer flowing through the rock matrix, which has lower permeability. The results for permeability ratio* $1 : 10^5$ *only show a single peak, that breaks through very early, meaning that the tracer goes straight through the fractures and that this is the predominant transport mechanism. Thus, for the first case it is necessary to use a multicontinua model, while for the second case a discrete model may be appropriate.*

The previous example demonstrates that the time-of-flight formalism can be

used to find breakthrough curves for highly resolved small-scale models in which fractures are represented explicitly as volumetric object. This may be used as a guide when choosing an appropriate conceptual model to be used on a larger scale. Hence, our method may serve as a technical guide for the choice of single and multicontinua formulation in fractured rocks.

In the next example, we demonstrate how our dG methods can be used to delineate the reservoir by determining swept and drainage volumes and well connectivities. To this end, we will solve the steady tracer-concentration equation (3) rather than the time-of-flight equation. The stationary tracer equation describes the steady concentration arising if we continuously inject tracer at a certain part of the inflow boundary. Hence, if the tracer concentration is positive at a point, the point is influenced by the part of the inflow boundary from which we inject tracer. To partition a reservoir, we define the swept/drained volumes as the volumes having a concentration larger than 0.5. Notice in particular that due to the efficient sequential solution procedure, computing each drainage volume is a single-sweep computation that can be performed with high accuracy and modest demands on storage and computing power.

Case 6 (Stationary Tracer Distribution) *We consider the stationary tracer distribution for a fractured reservoir, in which the permeability ratio between the matrix and the fractures is* $1 : 10^5$ *and the aperture of the fractures is* 10^{-4} *length units. One injection well is located in each corner and two production wells are located inside the domain. Figure 14 shows the tracer distribution for each injector computed using basis functions of increasing order. The four sweep areas are shown in different grey-tones, with boundaries between the sweep areas defined as the* 0.5 *contour of the different tracer concentrations. The figure illustrates that low-order approximations generally provide sufficient accuracy to delineate the reservoir. This was also observed in [14].*

Figure 15 shows the stationary tracer distribution for a reservoir with more complex geometry. The distribution of the fractures is depicted in the figures, and the permeability ratio between matrix and the fractures is $1 : 10^3$*. One producer is located in the lower left corner and three injectors are located in the three other corners. Each row in Figure 15 shows the sweep areas for the three different injectors computed using dG(0) in the first column and dG(2) in the second column.*

5 Final Remarks

We have previously shown that the combination of a discontinuous Galerkin spatial discretization and an optimal ordering of cells is a robust, accurate, and efficient numerical approach for the solution of incompressible flow of

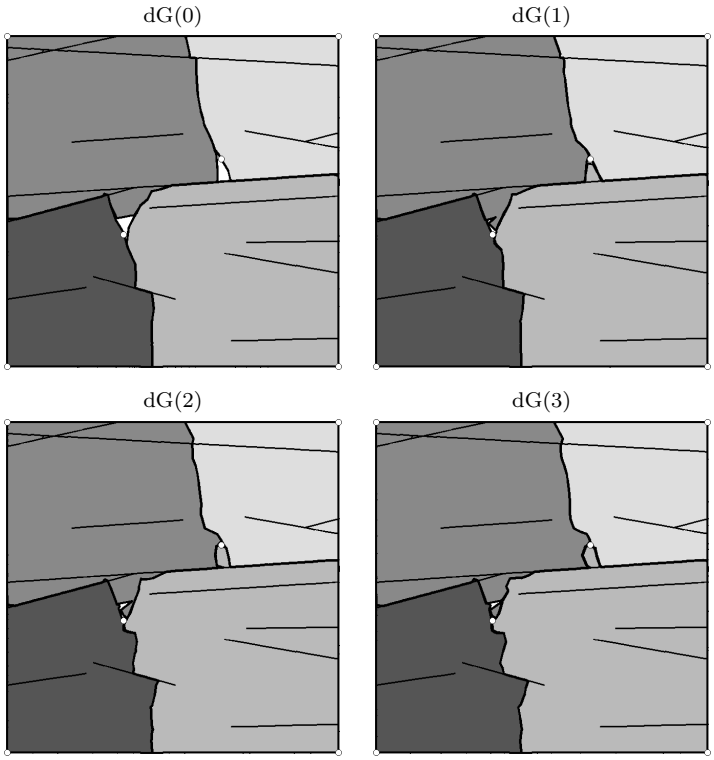

Fig. 14. Stationary tracer distribution with one injector placed in each corner and two producers placed inside the domain.

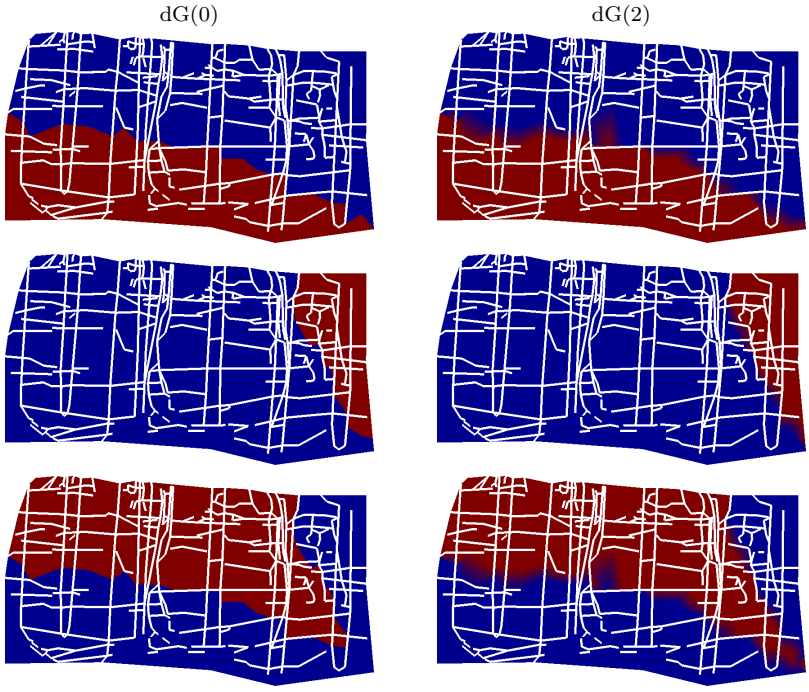

Fig. 15. Tracer distribution for three injectors placed in three of the corners and one producers placed in the lower left corner.

fluids in porous media, see [14,7]. For multiphase flow [7,8] and single-phase flow in media with mild heterogeneity, our experience indicates that a low-order dG method (the standard upwind method, dG(0), or the second-order dG(1)) is sufficient to accurately capture the fluid transport. For single-phase flow in strongly heterogeneous media, one may need to increase the order to accurately capture integrated quantities like time-of-flight and steady tracer concentration.

For fractured media, explicit modelling of complex fracture networks will give rise to very complex structures, and using unstructured triangular (tetrahedral) grids, at least locally, may be necessary to accurately model realistic cases. In this paper, we have made the first steps toward extending our dG methodology to unstructured grids by presenting results for triangular elements in 2D, from which the extension to tetrahedral elements in 3D is straightforward.

Two features with our methodology may prove very useful when attacking complex 3D models. First, by using a discontinuous Galerkin discretization in combination with an upwind flux, we localise the degrees-of-freedom (and their assembly) and simplify the coupling of different element types. Second, the optimal ordering of the unknowns enables us to compute the solutions in an element-by-element fashion that is quite general and applies to *any* grid for which the inter-element dependence can be described by a graph.

For triangular grids, the dG method is convergent for smooth solutions, but loses accuracy near discontinuities. Case 1 in Section 4 shows how the roughness of randomly perturbed grids impacts the accuracy, leading to reduced convergence rates. Considering polynomial degree versus grid resolution, our examples indicate that increasing the order of the basis functions is more important than increasing the grid resolution (provided the flux is resolved with sufficient accuracy). Our experience is that a dG discretisation of sufficiently high order is a relatively robust alternative to streamlines and performs well for a wide range of realistic cases. However, high permeability contrasts reduce the accuracy of the solution. This may be countermanded by introducing a *hp*-refinement [14], in which we reduce the order of the basis functions and refine the grid in areas with high media contrasts. Finally, to accurately compute time-of-flight in fractured porous media, it is important to have a sufficient grid resolution in the latitudinal direction of the fractures. This is necessary since the time-of-flight is an integrated quantity that is very sensitive to small-scale variations in media properties, and contains large spatial variation, in particular within and close to fractures.

We have also demonstrated how the framework can be used to compute accurate approximations to the stationary tracer distribution in a reservoir. Two test cases indicate that low-order approximations have sufficient accuracy to

produce reasonable delineations of a reservoir volume.

Altogether, we have demonstrated that the dG schemes in most cases can accurately compute time-of-flight and stationary tracer distribution. These quantities are of practical importance for applications in petroleum reservoir simulation and groundwater modelling. For petroleum reservoir simulation, time-of-flight gives the timelines in the reservoir, whereas computing the tracer distribution can determine the spatial regions swept or drained by well. Within groundwater applications, time-of-flight may be an important tool to visualise the spreading of contaminants and to help understanding the different transport processes.

References

[1] P. Dietrich, R. Helmig, M. Sauter, H. Hötzl, J. Köngeter, G. Teutsch, Flow and Transport in Fractured Porous Media, Springer, 2005.

[2] V. Reichenberger, R. Helmig, H. Jakobs, P. Bastian, J. Niessner, Complex gas-water processes in discrete fracture-matrix systems: Upscaling, mass-conservative discretization and efficient multilevel solution, Institut für Wasserbau, Universität Stuttgart (2004).

[3] N. R. Council (Ed.), Rock fractures and fluid flow, National Academy Press, 1996.

[4] L. Neunhäuserer, A. Hemminger, R. Helmig, Influence of fracture-matrix-interaction on flow and transport processes and the resulting effective parameters in fractured porous systems, in: Hydraulic Engineering for sustainable Water Resources Management at the Turn of the Millennium, XXVIII IAHR Congress, Graz, Austria, 1999.

[5] A. Datta-Gupta, M. J. King, Streamline Simulation: Theory and Practice, Vol. 11 of SPE Textbook Series, Society of Petroleum Engineers, 2007.

[6] B. Eikemo, I. Berre, H. K. Dahle, K.-A. Lie, J. R. Natvig, A discontinouos Galerkin method for computing time-of-flight in discrete-fracture models, in: P. J. Binning, P. K. Engesgaard, H. K. Dahle, G. F. Pinder, W. G. Gray (Eds.), Proceedings of the XVI International Conference on Computational Methods in Water Resources, Copenhagen, Denmark, 2006, `http://proceedings.cmwr-xvi.org/`.

[7] J. R. Natvig, K.-A. Lie, Fast computation of multiphase flow in porous media by implicit discontinuous Galerkin schemes with optimal ordering of elements, J. Comput. Phys. 227 (24) (2008) 10108–10124, dOI: 10.1016/j.jcp.2008.08.024.

[8] J. R. Natvig, K.-A. Lie, On efficient implicit upwind schemes, in: ECMOR XI, 11th European Conference on the Mathematics of Oil Recovery, EAGE, 2008, `http://www.earthdoc.org/detail.php?paperid=A03&edition=62`.

[9] H. Hoteit, A. Firoozabadi, Multicomponent fluid flow by discontinuous Galerkin and mixed methods in unfractured and fractured media, Water Resour. Res. 41 (W11412), doi:10.1029/2005WR004339.

[10] J. Qian, Y.-T. Zhang, H.-K. Zhao, Fast sweeping methods for eikonal equations on triangular meshes, SIAM J. Numer. Anal. 45 (1) (2007) 83–107 (electronic).

[11] F. Li, C.-W. Shu, Y.-T. Zhang, H. Zhao, A second order discontinuous Galerkin fast sweeping method for Eikonal equations, J. Comput. Phys. 227 (17) (2008) 8191–8208, dOI: 10.1016/j.jcp.2008.05.018.

[12] Y.-T. Zhang, H.-K. Zhao, J. Qian, High order fast sweeping methods for static Hamilton-Jacobi equations, J. Sci. Comput. 29 (1) (2006) 25–56.

[13] C.-Y. Kao, S. Osher, J. Qian, Legendre-transform-based fast sweeping methods for static Hamilton–Jacobi equations on triangulated meshes, J. Comput. Phys. 227 (24) (2008) 10209–10225, doi:10.1016/j.jcp.2008.08.016.

[14] J. R. Natvig, K.-A. Lie, B. Eikemo, I. Berre, An efficient discontinuous Galerkin method for advective transport in porous media, Adv. Water Resour. 30 (12) (2007) 2424–2438.

[15] J. R. Natvig, K.-A. Lie, B. Eikemo, Fast solvers for flow in porous media based on discontinuous Galerkin methods and optimal reordering, in: P. J. Binning, P. K. Engesgaard, H. K. Dahle, G. F. Pinder, W. G. Gray (Eds.), Proceedings of the XVI International Conference on Computational Methods in Water Resources, Copenhagen, Denmark, 2006, `http://proceedings.cmwr-xvi.org/`.

[16] R. Klausen, R. Winther, Robust convergence of multi point flux approximation on rough grids, Numer. Math. 104 (2006) 317–337, doi:10.1007/s00211-006-0023-4.

[17] I. Aavatsmark, G. Eigestad, R. Klausen, M. Wheeler, I. Yotov, Convergence of a symmetric MPFA method on quadrilateral grids, Comput. Geosci. 11 (2007) 333–345, doi:10.1007/s10596-007-9056-8.

[18] M. A. Christie, M. J. Blunt, Tenth SPE comparative solution project: A comparison of upscaling techniques, SPE Reservoir Eval. Eng. 4 (4) (2001) 308–317, `http://www.spe.org/csp`.