A DISCONTINUOUS GALERKIN METHOD FOR TRANSPORT IN FRACTURED MEDIA USING UNSTRUCTURED TRIANGULAR GRIDS

B. EIKEMO, K.-A. LIE, G. T. EIGESTAD, AND H. K. DAHLE

Abstract. The possibility to couple discrete (fractures, shear zones) and continuous (rock matrix) model elements is a prerequisite for simulating flow and transport processes in fractured rocks. The method described in this paper uses unstructured triangular grids to explicitly represent the fractures and matrix rock as a single continuum in which one can compute the transport using a higher-order discontinuous Galerkin method. By modelling the complex fracture networks explicitly, very complex structures can be modelled and using unstructured triangular grids may be necessary to accurately model realistic cases. Herein we consider single-phase equations for advective transport, which have an inherent causality in the sense that information propagates along streamlines. Our discontinuous Galerkin discretization preserves this causality. 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.

1. Introduction

Accurate representation of fractured reservoirs represents a challenge for the characterisation, modelling, and simulation of petroleum and groundwater reservoirs, see [4, 12, 3]. Fractured reservoirs are complex geological structures, where fractures (cracks and joints created by rock stress) have higher permeability and porosity than the surrounding rock (matrix). 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.

Models for fractured media have traditionally been of two general types: discrete or multicontinua (porosity) models. In a discrete model, the fractures are considered as discrete structures integrated in the surrounding rock matrix. With such a model we have the possibility to model single- and multiphase flow and transport processes accurately. Using multicontinua models, we have to make assumption that an representative elementary volume cannot only be obtained for porous medium–the rock matrix–but also for the fractured system.

Date: Draft August 20, 2008.

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.

It is principally possible to use different flow and transport models for the different continua. Exchange terms describing the interaction between the matrix system and the fracture system are very important using multicontinua models, see [11]. For a rock mass 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 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-fault 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. Herein we consider an even more ambitions modelling approach that has increased in popularity lately; in this approach fractures are represented explicitly as thin volumetric cells 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, where the fractures themselves are represented explicitly as cells with small width and high permeability (and porosity). This will lead to models with highly contrasting reservoir properties and very complex hydraulic conductivities. To accurately model the flow and transport in regions characterised by high contrast in permeability between the fractures and the matrix, we will also briefly investigate the use of local adaptive refinement.

A simple single-phase model is often sufficient to reveal the major displacement patterns in a fractured medium (e.g., if represented as a single continuum with fractures as volumetric objects). 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 timelines, influence regions, reservoir partitioning, tracer profiles, well pairs, etc., that may be more visual and intuitive than pressure values and discrete fluxes. One particular quantity of interest is the time-of-flight, which can be used to identify areas affected by contaminations in groundwater flow or to determine drainage and flooded volumes in petroleum reservoirs.

Most of such derived quantities are often associated with, and computed by, streamlines methods. However, since they all can be described by (steady-state) transport equations, one could equally well use a grid-based method: the purpose of our paper is to develop a finite-volume method for solving time-of-flight type equations to characterise flow patterns and to compute fluid transport for highly detailed models with explicit fracture modelling. 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 where each block corresponds to the degrees-of-freedom in a single cell (which we sometimes will refer to as an element). Blocks corresponding to neighbouring cells in the grid are coupled through the numerical flux function used to approximate the physical flux over cell interfaces. By introducing an upwind flux approximation, the elements can be ordered to ensure that the linear system has a lower block-triangular form, where each block corresponds to the degrees-of-freedom in a single cell or in a collection of cells having circular dependence due to rotation in the velocity field. Given the triangular form, the linear system can be decomposed to a set of small problems, one for each block and solved using a forward block substitution. This solution procedure is very efficient and 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. For more details on the efficiency of the reordering method, we refer the reader to [8, 7], in which the same ideas are applied to multiphase flow. The ideas presented herein are a continuation of the research in [5], where we presented a family of discontinuous Galerkin schemes for simulating flow in idealised fractured media using rectangular grids. In this representation, the orientation of the fractures are restricted to being horizontal or vertical.

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

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

This system can be solved to compute the pressure p and the volumetric flow velocity $\mathbf{v}$ if 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. Alternatively, the system can be written as a second-order elliptic equation for the pressure. 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. This paper focuses on such transport equations. For simplicity, we will henceforth assume that $\mathbf{v} = \mathbf{v}(\mathbf{x})$ is given and is 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 $\tau = \tau(\mathbf{x})$, which is defined as the time needed for a passive particle to travel from a point on the inflow boundary to a given point $\mathbf{x}$. Iso-contours of τ define natural timelines in a reservoir. To define τ, we introduce streamlines, which are a family of curves that at any point are tangential to the velocity vector $\mathbf{v}$ of the flow. For a steady velocity (as considered herein), streamlines coincide with the path traced out by a passive particle moving with the flow field. The time-of-flight τ is 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 denotes the arclength along the streamline. Note that modern streamline methods use the time-of-flight τ rather than the arclength r as spatial coordinates. 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) comes on the form (6) if we introduce an appropriate 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 following we present an efficient strategy for solving transport equations on the form (6) on unstructured triangular grids where we combine higher-order discontinuous Galerkin (dG) spatial discretizations with an upwind numerical flux function that creates a one-sided dependency between the elements in the grid and ensures that we can find a reordering of the elements such that the resulting system becomes lower block-triangular and can be solved block-by-block. We have previously studied the dG-reordering method for rectangular grids [10, 5, 9], for which it proved to be both accurate and highly efficient. In [8], 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.

3. Discontinuous Galerkin Schemes with Optimal Ordering

To develop higher-order discontinuous Galerkin methods, we start with a variational formulation of (6). We then partition the solution domain 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 $\mathbb{Q}^n = \operatorname{span}\{x^p y^q : 0 \le p+q \le n\}$ 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 $a_T^h(\cdot,\cdot)$,

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

Here q_h and q_h^{ext} are the inner and outer approximations 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 on T will only be influenced by elements $\mathcal{U}(T)$ that are intermediate neighbours in the upwind direction, which we 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. Notice that compared with a standard upwind approximation, we have introduced an additional clipping, $\max(q_h^{ext}, 0)$, to prevent negative values from propagating downstream. Negative values are unphysical, but may arise when using high-order polynomials.

To find a solution to (8), we choose trial functions $\varphi_h = L_i^k$ and evaluate (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 parts. Let $F_T^+ Q_T$ denote the flux out of element T and $F_T^- Q_{\mathcal{U}(T)}$ denote the flux into element T. Hence, the following system of linear equations is obtained

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

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 [8] 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 of T. We may therefore construct the solution locally, starting at inflow boundaries (or fluid sources) and proceeding element by element downstream. 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 N_e is the number of elements, such an ordering can be found in N_e operations if it exists.

If the reordering of elements does no 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 instead of a single element. More details are found in [10, 8].

4. Numerical Examples

In [5], we presented a dG scheme for computing time-of-flight in fractured porous media represented on rectangular grids. The use of rectangular grid restricts the orientation of fractures to be either horizontal or vertical. In this section we will consider more realistic fracture distributions modelled on triangular grids present results from selected numerical experiments using higher-order dG schemes and optimal ordering of triangular elements. For each example, the forcing velocity field will either be given by an analytical expression or be computed by a standard conservative method for solving the first-order system of flow equations (1) or its second-order counterpart, in which case the velocity will be divergence free and nearly irrotational. We will mainly apply the dG schemes to compute time-of-flight in semi-realistic examples of fractured media. Because of the localised nature of the dG formulation, using hybrid grids consisting of both rectangular and triangular cells are within reach, but is not considered herein. However, we show one example of adaptively refined grids.

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

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

Tables 1 and 2 present L_2-errors and convergence rates for a grid-refinement study performed by increasing the order n in dG(n) on four grid types with increasing roughness (see Figure 1):

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

Grid 2: uniform refinement and triangulation of a 10×10 base grid where 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.

In Table 1 the L_2-errors are measured in a smooth part of the domain, $[1, 1.3] \times [1, 1.3]$, while in Table 2 the error is integrated over the whole domain. The tables indicate how different roughness[1] in the refined grids impacts the L_2-errors and the convergence rates. For the perturbed grids the rates of convergence are computed by comparing to two different mesh sizes: the average maximum mesh size, which is the average of the maximum cell edges of each element, and by the maximum mesh size, which is the largest cell edge in the domain.

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 get reduced convergence rates because of the kink in the solution along the circular arc $x^2 + y^2 = 5$. This agrees with the results in [10] for rectangular elements. Note that the kink may impact the regularity of the analytical solution, such that the decays of convergence rates are expected.

[1] Rough grids are defined in the literature (see e.g., [6]) 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 [1], where in general a decay in convergence rates may be seen.

TABLE 1. 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 to 4. Convergence rates on Grids 3 and 4 are computed in terms of the average maximum and maximum mesh size, respectively.

	dG(0)		dG(1)		dG(2)		dG(3)	
N	Error	Rate	Error	Rate	Error	Rate	Error	Rate
10	2.17e-03	—	2.14e-05	—	4.52e-07	—	7.67e-09	—
20	1.09e-03	1.00	5.47e-06	1.97	5.88e-08	2.94	4.81e-10	4.00
40	5.47e-04	1.00	1.37e-06	1.99	7.38e-09	2.99	3.01e-11	4.00
80	2.74e-04	1.00	3.45e-07	1.99	9.12e-10	3.02	1.88e-12	4.00
160	1.37e-04	1.00	8.70e-08	1.99	1.12e-10	3.02	1.18e-13	4.00
10	2.05e-03	—	2.16e-05	—	4.50e-07	—	8.62e-09	—
20	1.03e-03	1.00	5.43e-06	1.99	5.55e-08	3.02	5.51e-10	3.97
40	5.15e-04	1.00	1.36e-06	2.00	6.91e-09	3.01	3.48e-11	3.99
80	2.58e-04	1.00	3.42e-07	1.99	8.50e-10	3.02	2.16e-12	4.01
160	1.29e-04	1.00	8.62e-08	1.99	1.06e-10	3.00	1.35e-13	4.01
10	2.28e-03	—/—	2.22e-05	— /—	4.99e-07	—/—	1.01e-08	—/—
20	1.13e-03	1.01/1.12	5.86e-06	1.92/2.18	6.68e-08	2.90/3.22	6.76e-10	3.90/4.52
40	5.67e-04	0.99/1.04	1.50e-06	1.96/2.05	8.74e-09	2.94/3.03	4.86e-11	3.80/4.03
80	2.85e-04	0.99/1.03	3.84e-07	1.97/2.03	1.15e-09	2.93/3.12	3.25e-12	3.90/3.92
160	1.43e-04	1.00/1.03	9.76e-08	1.98/2.04	1.52e-10	2.92/2.97	2.20e-13	3.88/4.01
10	2.05e-03	—/—	2.16e-05	—/—	4.50e-07	—/—	8.62e-09	—/—
20	1.12e-03	0.90/0.91	6.36e-06	1.81/1.83	7.19e-08	2.71/2.74	8.32e-10	3.46/3.49
40	6.09e-04	0.90/0.93	1.96e-06	1.75/1.80	1.26e-08	2.59/2.67	9.09e-11	3.30/3.40
80	3.35e-04	0.88/0.90	6.25e-07	1.69/1.73	2.28e-09	2.53/2.59	1.07e-11	3.16/3.24
160	1.94e-04	0.80/0.82	2.02e-07	1.67/1.71	4.66e-10	2.35/2.39	1.42e-12	2.99/3.05

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

	dG(0)		dG(1)		dG(2)		dG(3)	
N	Error	Rate	Error	Rate	Error	Rate	Error	Rate
10	1.92e-02	—	8.95e-04	—	2.67e-04	—	1.49e-04	—
20	1.01e-02	0.92	3.06e-04	1.55	1.00e-04	1.42	5.34e-05	1.48
40	5.34e-03	0.92	1.10e-04	1.47	3.23e-05	1.63	1.68e-05	1.67
80	2.81e-03	0.92	3.97e-05	1.48	1.07e-05	1.60	5.46e-06	1.62
160	1.47e-03	0.93	1.42e-05	1.48	3.48e-06	1.61	1.69e-06	1.69
10	1.96e-02	—	1.08e-03	—	3.41e-04	—	2.07e-04	—
20	1.05e-02	0.90	3.65e-04	1.57	1.13e-04	1.60	6.20e-05	1.74
40	5.62e-03	0.90	1.28e-04	1.51	3.61e-05	1.64	1.98e-05	1.64
80	3.00e-03	0.91	4.58e-05	1.48	1.22e-05	1.57	6.15e-06	1.69
160	1.59e-03	0.92	1.65e-05	1.47	4.01e-06	1.60	1.89e-06	1.70
10	1.95e-02	—/—	9.97e-04	—/—	2.91e-04	—/—	1.73e-04	—/—
20	1.05e-02	0.89/0.97	3.25e-04	1.62/1.62	1.15e-04	1.35/1.45	6.29e-05	1.46/1.56
40	5.51e-03	0.93/0.93	1.19e-04	1.45/1.50	3.88e-05	1.56/1.61	1.86e-05	1.76/1.82
80	2.91e-03	0.92/0.95	4.27e-05	1.48/1.50	1.22e-05	1.67/1.71	6.50e-06	1.51/1.52
160	1.53e-03	0.93/0.95	1.54e-05	1.47/1.50	3.99e-06	1.61/1.65	1.89e-06	1.78/1.82
10	1.96e-02	—/—	1.08e-03	—/—	3.41e-04	—/—	2.07e-04	—/—
20	1.13e-02	0.82/0.83	3.98e-04	1.48/1.50	1.44e-04	1.28/1.30	6.97e-05	1.61/1.63
40	6.58e-03	0.80/0.82	1.61e-04	1.34/1.37	4.77e-05	1.64/1.67	2.45e-05	1.55/1.58
80	3.93e-03	0.77/0.78	6.91e-05	1.26/1.29	1.97e-05	1.31/1.34	1.00e-05	1.32/1.36
160	2.35e-03	0.76/0.78	2.85e-05	1.31/1.34	7.51e-06	1.43/1.46	3.81e-06	1.44/1.47

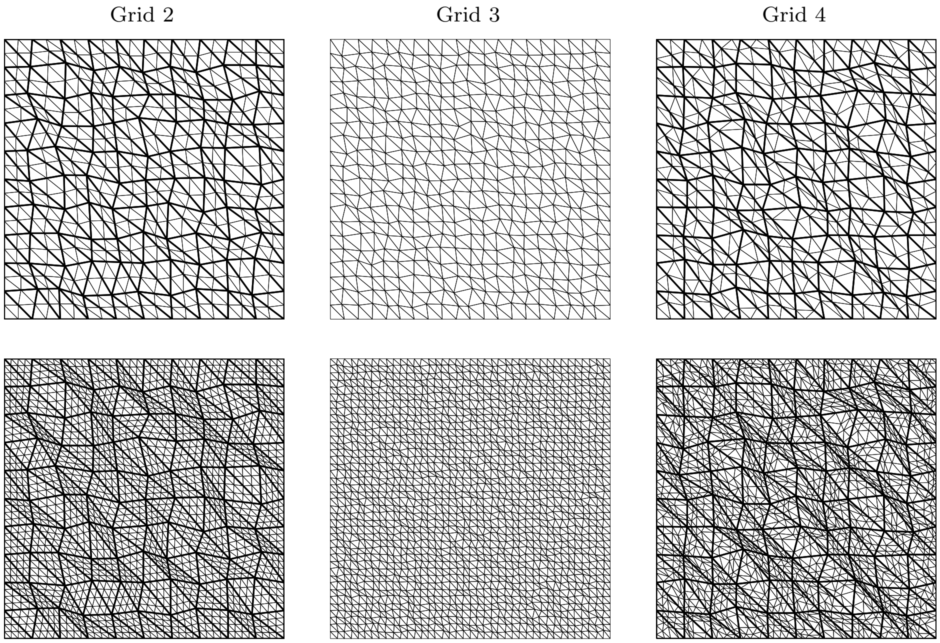

FIGURE 1. Two refinement levels for Grids 2, 3 and 4.

For the two rough grids, Grids 3 and 4, the errors increase on each refinement level. Grid 3 seems to yield marginally lower convergence rates than the formal order of the method when comparing to the average maximum mesh size. Note, however, that when using the maximum mesh size to compute rates, they fluctuate above and below the formal convergence rates, which may indicate that the average maximum mesh size is a somewhat non-conclusive mesh size to compare convergence against. This may be explained by the nature of random grid perturbations and the impact this has on the numerically calculated solutions on each refinement level. A long cell edge may impact the shapes of the neighbouring triangles and may yield larger errors at nodes associated with these.

Grid 4 experiences loss of convergence orders; this is observed when comparing to both the average maximum mesh size and the maximum mesh size. This may be explained by the diminishing grid quality such perturbations lead to. In Figure 2, the histograms for the mesh sizes of the triangular grids have been plotted for both Grids 3 and 4 for the grid size $N = 80$. Grid 3 has a normal distribution of the measured mesh size h for each element, whereas the mesh distribution in Grid 4 is skewed with a long tail in the interval corresponding to triangular grid cells with longer edges. Note that by the definition of the grid perturbation, the first refinement level of Grid 4 is a normal distribution. As the grids are refined, the mesh distribution becomes more and more skewed, with a significant tail to the right. 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

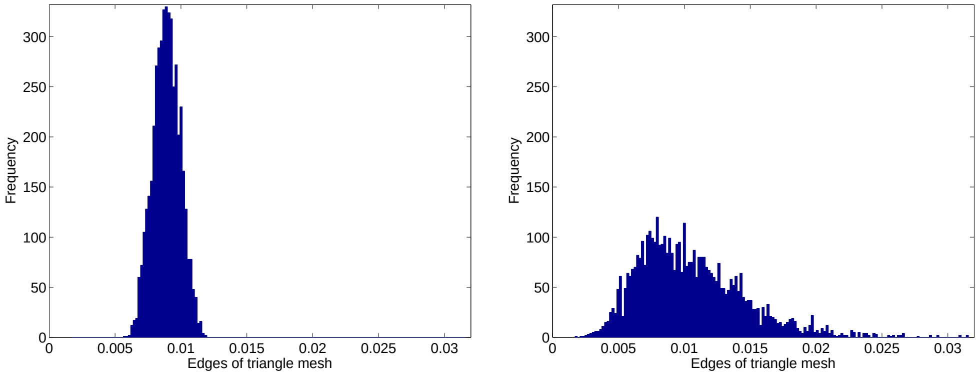

FIGURE 2. Histogram of edges of triangle meshes for Grid 3 (left) and Grid 4 (right).

reduction of triangle edge sizes the other grids experience. The observed order of dG(n) is significantly lower than $n + 1$ for the Grid 4 refinement and also seems to decrease as the grids are refined, 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 [6] and [1], where the convergence of the pressure equation is studied for general permeability description and irregular geometry. When a transformation to a computational space is performed for the pressure equation on general quadrilateral grids with general permeability, 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.

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 in the 10th SPE Comparative Solution Project [2]. This layer contains sharp contrasts in permeability (and porosity) between the low-permeable background and a set of intertwined high-permeable channels. The strongly heterogeneous structure in this permeability field is shown in the upper plot in Figure 3; the permeability variation is up to eleven orders of magnitude. The right column in the figure shows the computed time-of-flights on triangular elements for dG(n), $n = 0, 1, 3$, and 5. For comparison, the left column shows the corresponding solutions using the dG scheme on rectangular elements, see [10]. The grid size is 220×60 for the rectangular grid, while the triangular grid is created by dividing each rectangular element into two triangles. The plots were created by sampling the polynomial patches in 10×10 uniformly distributed points inside each rectangular elements. In the

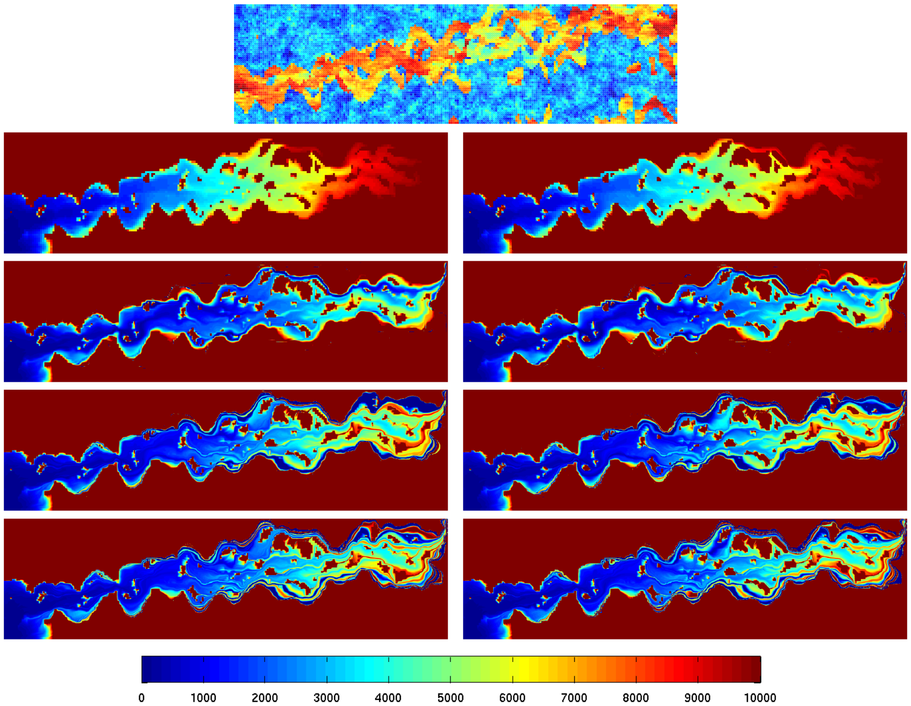

Figure 3. Time-of-flights for Layer 77 from the SPE10 test case computed using dG(n), $n = 0, 1, 3$, and 5 on the original rectangular grid (left column) and on a triangular grid (right column) created by splitting each rectangular cell in two.

Table 3. The relative L_1-errors of the computed time-of-flights for different vertical cross sections. See Figure 4 for the computed time-of-flights.

x	dG(0)	dG(1)	dG(2)	dG(3)	dG(4)	dG(5)
55	2.0719e-01	1.8535e-01	1.7421e-10	1.5946e-01	1.3955e-01	1.2579e-01
110	4.3507e-01	3.3024e-01	2.9480e-01	2.5810e-01	2.2044e-01	2.0217e-01
165	6.1523e-01	5.2704e-01	5.0089e-01	4.7045e-01	4.3190e-01	3.8929e-01
220	3.2763e-01	1.8338e-01	1.6834e-01	1.4444e-01	1.3316e-01	1.2459e-01

visual norm, the accuracy is approximately the same on the triangular and on the rectangular grid.

For this case we investigate the computed time-of-flights for four different vertical cross sections at $x = 55, 110, 165, 220$. The red graphs in Figure 4 show the dG(5) solution for the different cross sections and the blue graphs show the solution obtained by back-tracing approximately 8 000 streamlines. The corresponding relative L_1-errors for time-of-flights computed in different cross sections are present in Table 4. Altogether, we observe that strong heterogeneities in the permeability field influence the accuracy of the computed time-of-flights.

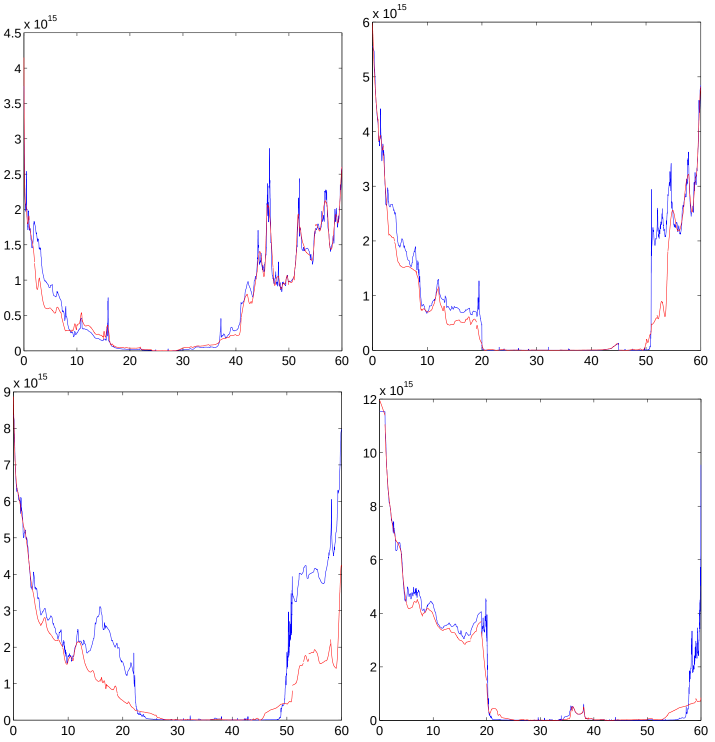

FIGURE 4. The computed time-of-flights along vertical cross sections at $x = 55, 110, 165, 220$. The red graphs give the dG(5) solution and the blue graphs give the solutions computed using approximately 8 000 streamlines.

Case 3 (Discrete Fracture Model). In this example, we consider a case with three high-permeable 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. Two cases are considered with the fractures having a permeability of 10^3 and 10^5, respectively, relative to the homogeneous and isotropic background field. The aperture of the fracture is 10^{-4} length units.

We compare the computed time-of-flights on a triangular grid with 5 048 elements that are adapted around and along the fractures with results on a coarser grid with 437 elements, a finer grid with 23 463 elements, and a grid with 5 301 elements but without adaptivity. The four grids are depicted in the top row of Figure 5. The permeability ratio between matrix and the fractures is $1 : 10^5$. From the plots, we observe three qualitative tendencies: (i) for the same number of unknowns (see Table 4), 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 decays with the polynomial order of the scheme. Finally, we observe that all the dG solutions establish the qualitative structures of the flow pattern.

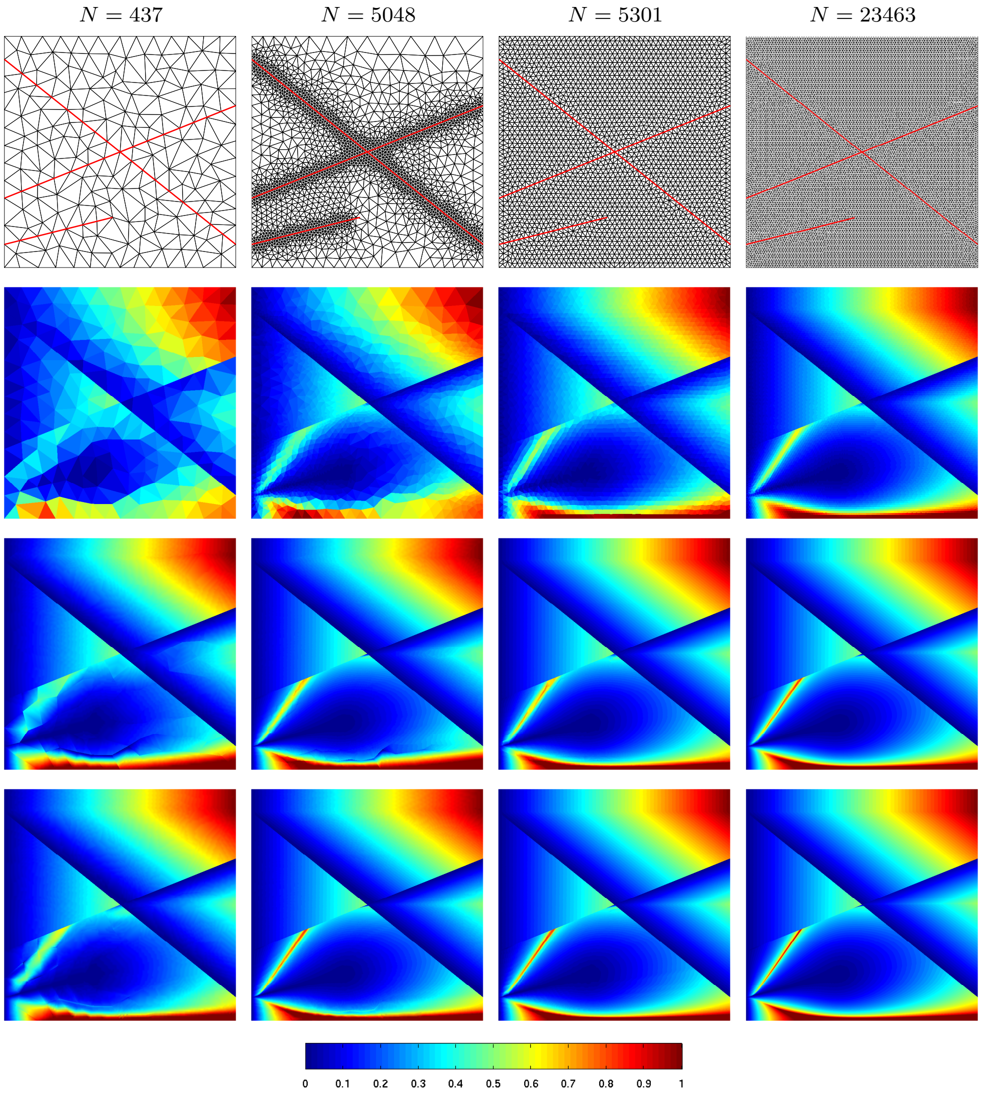

FIGURE 5. Case 3 with ratio between matrix and fracture permeability equal $1 : 10^5$. The rows present the computed time-of-flights using dG(0), dG(1), and dG(3).

TABLE 4. 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

Next, we consider the pointwise accuracy at the outflow boundary compared with a highly resolved solution computed by back-tracing approximately 16 000 streamlines. Table 5 shows the discrete relative L_1-errors for the time-of-flight

TABLE 5. Discrete relative L_1-errors in time-of-flight (upper half) and mass flow (lower half) at the outflow boundary for Case 3 with ratio K_m:K_f between the matrix and fracture permeability. The solutions are compared with solutions computed by tracing approximately 16 000 streamlines.

K_m:K_f	N	dG(0)	dG(1)	dG(2)	dG(3)	dG(4)	dG(5)
$1 : 10^3$	437	9.1575e-02	3.5080e-02	1.3979e-02	8.4314e-03	6.2851e-03	5.2484e-03
	5048	7.9320e-02	2.0271e-02	7.3588e-03	5.4863e-03	3.9867e-03	3.1843e-03
	5301	7.0833e-02	1.9132e-02	7.2066e-03	5.4397e-03	4.1306e-03	3.7196e-03
	23463	6.4046e-02	1.4980e-02	6.6697e-03	4.4689e-03	3.4315e-03	2.6675e-03
$1 : 10^5$	437	1.2607e-01	4.6378e-02	6.7521e-02	1.6426e-02	6.9459e-02	4.2250e-02
	5048	9.4916e-02	3.7634e-02	6.3852e-02	1.5811e-02	6.8247e-02	4.3420e-02
	5301	6.1381e-02	1.3697e-02	5.3138e-02	9.5423e-03	6.1411e-02	3.5569e-02
	23463	3.9014e-02	1.2143e-02	5.6907e-02	1.5259e-02	1.6766e-02	6.2730e-03
$1 : 10^3$	437	1.1240e-00	5.1969e-01	1.7987e-01	1.1910e-01	7.5825e-02	5.8945e-02
	5048	9.7077e-01	2.9547e-01	1.1434e-01	6.8977e-02	4.9809e-02	3.8912e-02
	5301	8.3155e-01	2.8433e-01	1.0473e-01	6.8073e-02	4.7003e-02	4.0898e-02
	23463	7.6335e-01	2.2220e-01	1.1358e-01	6.7778e-02	4.9492e-02	4.0935e-02
$1 : 10^5$	437	1.1102e-00	1.0192e-00	9.0023e-01	9.0149e-01	8.7923e-01	8.1843e-01
	5048	1.0885e-00	1.0394e-00	9.2523e-01	9.1535e-01	9.2921e-01	8.5429e-01
	5301	1.0577e-00	1.0305e-00	9.1703e-01	9.1194e-01	9.1910e-01	8.4378e-01
	23463	1.0514e-00	1.0723e-00	1.0027e-00	9.4507e-01	9.2882e-01	7.5827e-01

and the mass flux across the outflow boundary. Figure 6 shows the time-of-flight at the boundary for permeability ratio $1 : 10^3$. Similarly, Figure 7 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 7 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 Table 5 that the error increases with increasing ratio between the matrix and fracture permeability. This observation agrees with the results in [10].

When increasing the grid resolution in the example above, the grid inside the thin fractures only increased resolution in the longitudinal direction. For the simple Cartesian grids studied in [5], 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 lengths and the permeability ratio is $1 : 10^5$. Figure 8 shows time-of-flight computed with dG(n) for $n = 0$, 1, and 3. The upper row shows the time-of-flights computed on a grid where each fracture is represented with one rectangular element divided into two triangle elements in the latitudinal direction. The lower row shows the

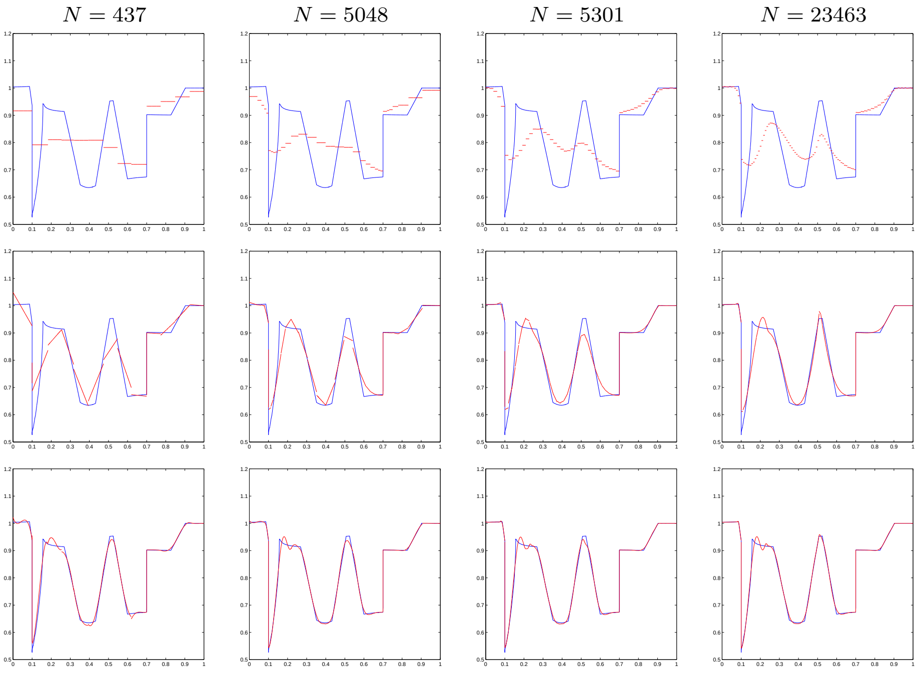

Figure 6. Time-of-flight for Case 3 with ratio between matrix and fracture permeability equal $1:10^3$. The red graphs show the dG(n) solutions for $n = 0$, 1, and 3 (from top to bottom) and the blue graphs are solutions computed by tracing approximately 16 000 streamlines.

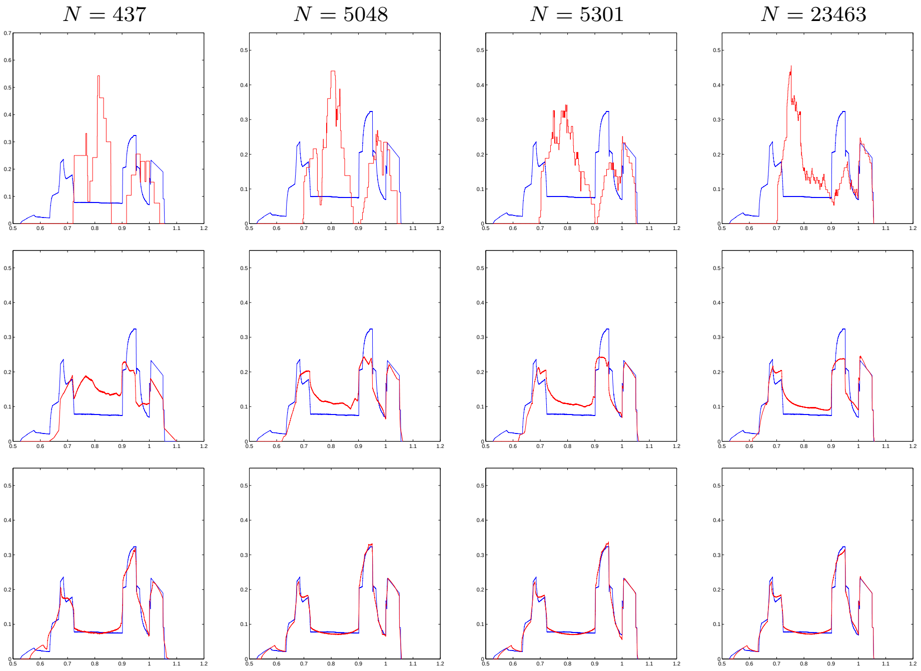

Figure 7. Average tracer concentration over the outflow boundary as a function of time for the simulations shown in Figure 6.

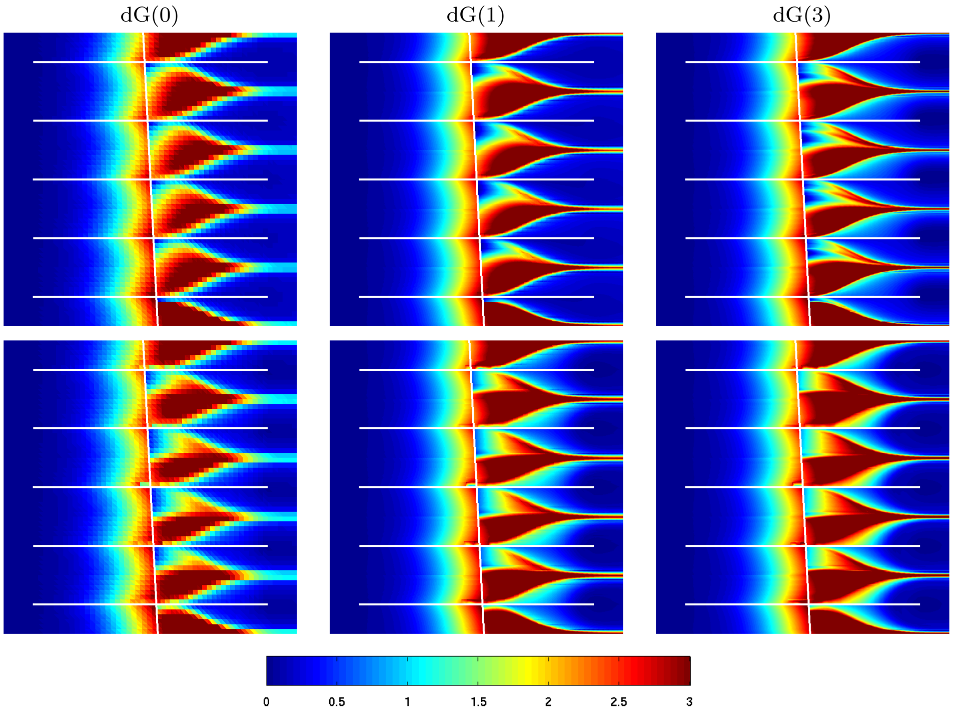

Figure 8. Time-of-flight for Case 4 with ratio between matrix and fracture permeability equal $1 : 10^5$. The upper row shows the results using no latitudinal grid refinement, while the lower row shows the solutions for a refinement with eight rectangular (divided in sixteen triangular) elements in the latitudinal direction of the fractures. The distribution of fractures is depicted as white lines.

solutions with eight rectangular elements (sixteen triangular elements) in the latitudinal direction.

Table 6 reports the time-of-flights and mass flow computed at the outflow boundary for the permeability ratios $1 : 10^3$ and $1 : 10^5$. We compare the computed time-of-flights 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. These results agree with the results in [5], where we observed the importance of sufficient latitudinal grid resolution to correctly capture large spatial variations inside the fractures. Capturing these variations is necessary since 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 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

Table 6. 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)	1.5772e-01	6.7707e-02	3.8263e-02	2.8205e-02	2.0743e-02	1.9426e-02
	8 (16)	1.5260e-01	5.7438e-02	3.2888e-02	2.3043e-02	1.7009e-02	1.6180e-02
$1 : 10^5$	1 (2)	7.7989e-01	5.8586e-01	3.9670e-01	3.5901e-01	3.2230e-01	2.9336e-01
	8 (16)	7.4916e-01	4.6600e-01	2.1120e-01	1.5121e-01	1.3237e-01	1.0724e-01
$1 : 10^3$	1 (2)	1.1149e-00	7.5882e-01	6.4524e-01	5.5346e-01	4.3380e-01	4.3060e-01
	8 (16)	1.1992e-00	5.9138e-01	3.8487e-01	2.9969e-01	2.3405e-01	2.4164e-01
$1 : 10^5$	1 (2)	1.4736e-00	1.2881e-00	6.2057e-01	6.3658e-01	3.9217e-01	3.7723e-01
	8 (16)	8.3593e-01	2.6399e-01	2.5384e-01	2.6574e-01	1.8156e-01	1.9228e-01

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 model 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. The tracer is a pulse injection for a short time; in our case for $t \in [0, 0.05]$. Figure 9 shows the mass flow over the outflow boundary computed using dG(n) for $n = 0$, 1, and 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 lower-permeable rock matrix. The results for permeability ratio $1 : 10^5$ only has 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 where 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 on

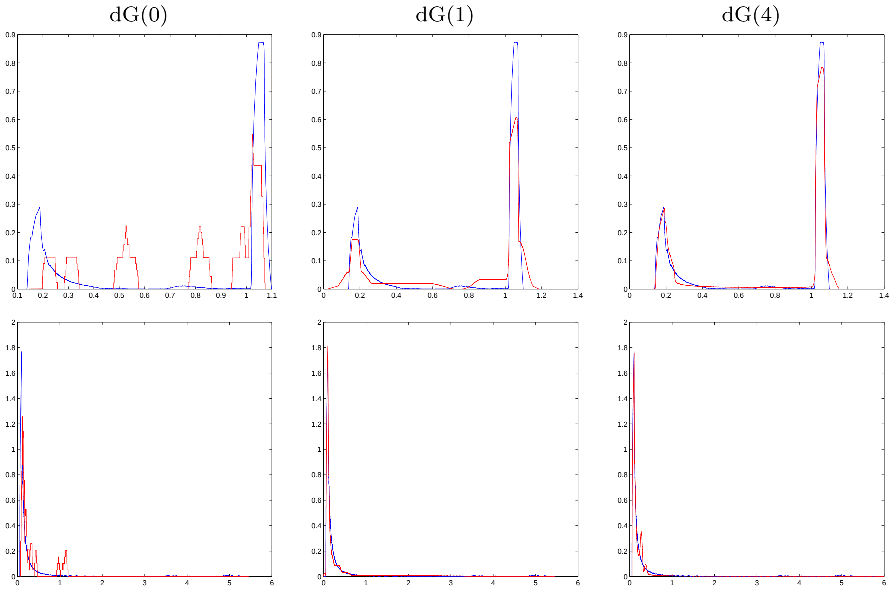
dG(0) dG(1) dG(4)

FIGURE 9. Computed mass flow over the outflow boundary with permeability ratio between matrix and the fractures equal $1 : 10^3$ in the upper row and $1 : 10^5$ in the lower row. The red graphs give the dG(n) solutions and the blue graphs give solutions computed by back-tracing approximately 4 000 streamlines.

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 computations that can be performed with high order accuracy and modest demands on storage and computing power.

Case 6 (Approximation of Stationary Tracer Distribution). We consider the stationary tracer distribution for a fractured reservoir shown in Figure 10. The permeability ratio between the the matrix and the fractures is $1 : 10^5$ and the the aperture of the fractures is 10^{-4} length units. Four injection wells are located in each corner and two production wells are located inside the domain. Figure 10 shows the tracer distribution for each injector computed using basis functions of increasing order. The different swept areas are shown in different colours/shading, and the boundaries between the swept areas correspond to the 0.5 contour of the different tracer concentrations. The figure illustrates that low-order approximations do, in general, provide sufficient accuracy to delineate the reservoir. This was also observed in [10].

Figure 11 shows the stationary tracer distribution in a more challenging fractured reservoir. 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 11 shows the swept areas for the

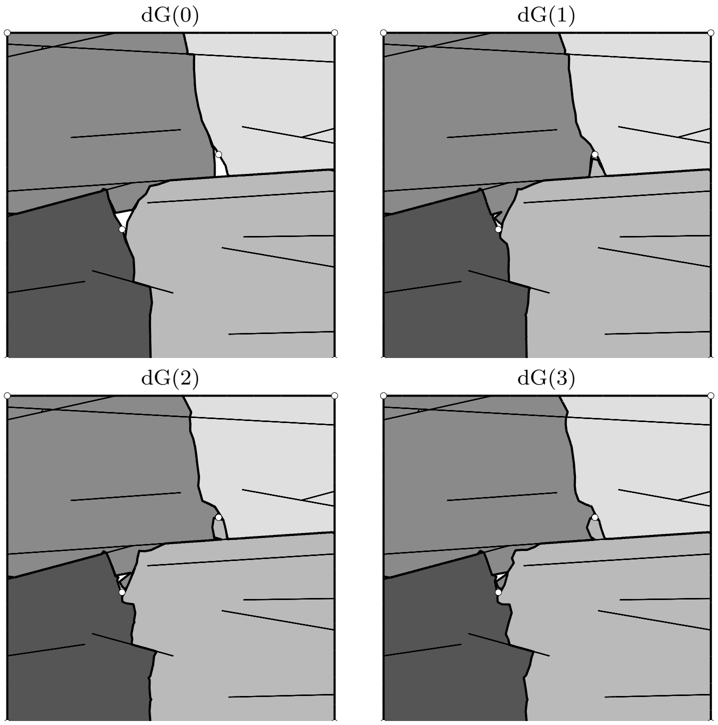

FIGURE 10. Stationary tracer distribution for four injectors placed one in each corner and two producers are placed inside the domain.

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 a optimal ordering of cells is a robust, accurate, and efficient numerical approach for the solution of incompressible flow of fluids in porous media, see [10, 8]. For multiphase flow [8, 7] 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, all our results have so far been presented for Cartesian grids. However, 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.

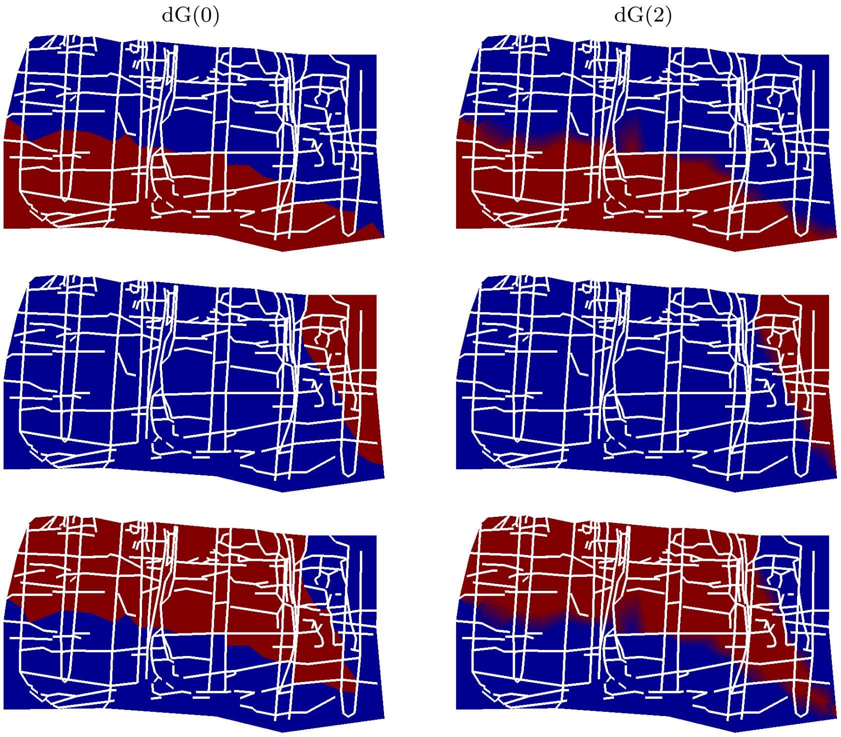

FIGURE 11. Tracer distribution for three injectors placed in three of the corners and one producers placed in the lower left corner.

There are two features with our methodology that may prove very useful when attacking complex 3D models. First of all, 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. Secondly, we use an optimal ordering of the unknowns that allows us to compute the solutions in an element-by-element fashion. This method is quite general and applies to *any* grid where 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 impact the accuracy, leading to reduced convergence rates for rough grids. Considering polynomial degree versus grid resolution, some of the other 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 relative robust alternative to streamlines that performs well in a wide range of realistic cases. However, high permeability contrasts reduce the accuracy of the solution. This may be countermanded by introducing a sort of a slope limiter as used in [10], where 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 having 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, the time-of-flight gives the timelines in the reservoir, whereas computing the tracer distribution can determine the spatial regions swept or drained by a fluid from a source or a sink. Within groundwater applications, the evaluation of the time-of-flight may be a important tool to visualise the spreading of contaminants and to help understanding the different transport processes.

Acknowledgements

We would like to thank Håkon Hægland for the streamline simulations, and Rainer Helmig for useful feedback on fractured model issues. The research is founded in part by the Research Council of Norway through the GeoScale project with grant 158908/130.

References

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

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

[3] N. R. Council, editor. *Rock fractures and fluid flow*. National Academy Press, 1996.

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

[5] B. Eikemo, I. Berre, H. K. Dahle, K.-A. Lie, and 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, and W. G. Gray, editors, *Proceedings of the XVI International Conference on Computational Methods in Water Resources*, Copenhagen, Denmark, June 2006. http://proceedings.cmwr-xvi.org/.

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

[7] J. Natvig and K.-A. Lie. On efficient implicit upwind schemes. In *Proceedings of ECMOR XI*, Bergen, Norway, 8-11 September 2008. http://folk.uio.no/kalie/papers/ecmorxi-ro.pdf.

[8] J. Natvig and K.-A. Lie. Fast computation of multiphase flow in porous media by implicit discontinuous Galerkin schemes with optimal ordering of elements. *J. Comput. Phys.*, to appear. http://folk.uio.no/kalie/papers/dg-multiphase.pdf.

[9] J. Natvig, K.-A. Lie, and 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, and W. G. Gray, editors, *Proceedings of the XVI International*

Conference on Computational Methods in Water Resources, Copenhagen, Denmark, June 2006. `http://proceedings.cmwr-xvi.org/`.

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

[11] L. Neunhäuserer, A. Hemminger, and 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*, Graz, Austria, 22-27 August 1999. XXVIII IAHR Congress.

[12] V. Reichenberger, R. Helmig, H. Jakobs, P. Bastian, and 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.

(Birgitte Eikemo) University of Bergen, Dept. of Mathematics,
Johs. Brunsgt. 12, NO-5008 Bergen, Norway
E-mail address: `birgitte@math.uib.no`

(Knut–Andreas Lie) SINTEF ICT, Applied Mathematics,
P.O. Box 124, Blindern, NO–0314 Oslo, Norway
E-mail address: `Knut-Andreas.Lie@sintef.no`

(Knut–Andreas Lie) University of Bergen, Dept. of Mathematics,
Johs. Brunsgt. 12, NO-5008 Bergen, Norway

(Geir Terje Eigestad) University of Bergen, Dept. of Mathematics,
Johs. Brunsgt. 12, NO-5008 Bergen, Norway
E-mail address: `geirte@math.uib.no`

(Helge K. Dahle) University of Bergen, Dept. of Mathematics,
Johs. Brunsgt. 12, NO-5008 Bergen, Norway
E-mail address: `helge.dahle@math.uib.no`