Multigrid for Higher Order Discontinuous Galerkin Finite Elements Applied to Groundwater Flow

Peter Bastian, Volker Reichenberger
Interdisziplinäres Zentrum für Wissenschaftliches Rechnen,
Universität Heidelberg
Im Neuenheimer Feld 368, 69120 Heidelberg,
email: Peter.Bastian@iwr.uni-heidelberg.de
Volker.Reichenberger@iwr.uni-heidelberg.de

Abstract

In this paper we present a multigrid method for higher order discontinuous Galerkin (DG) finite elements applied to the groundwater flow equation. It uses an incomplete LU decomposition on an element-wise block structure as smoother and Galerkin coarse grid correction with high order approximation on the coarse grid. It is shown experimentally that the method converges independent of the mesh size parameter and polynomial degree. The case of discontinuous permeability requires appropriate ordering of the unknowns. Furthermore, we compare the approximation quality of DG with vertex centered finite volumes and mixed finite elements.

1 Discontinuous Galerkin Finite Element Method

The discontinuous Galerkin method for diffusion problems has been formulated in [6].

1.1 Notation

In this paper we wish to solve the groundwater flow problem

$$\nabla \cdot (\varrho \mathbf{j}) = q \qquad \text{in } \Omega, \tag{1a}$$

$$\mathbf{j} = -\frac{\boldsymbol{K}}{\mu}(\boldsymbol{\nabla} p - \varrho \mathbf{g}), \tag{1b}$$

$$p = g \qquad \text{on } \Gamma_D \subseteq \partial\Omega, \tag{1c}$$

$$\varrho \mathbf{j} \cdot \mathbf{n} = h \qquad \text{on } \Gamma_F = \partial\Omega \setminus \Gamma_D, \tag{1d}$$

where ϱ, μ are density and viscosity of the fluid, $\boldsymbol{K}$ is the absolute permeability, $\mathbf{g}$ is a vector pointing in direction of gravity with magnitude equal to

the gravitational acceleration and p is the unknow pressure. Ω is a bounded Lipschitz domain in $\mathbb{R}^d$, $d = 2, 3$ and $\mathbf{n}$ denotes the outer unit normal.

Let $\mathcal{T}_h$ be a partitioning of the domain Ω into elements. The elements of the partitioning are denoted by

$$\mathcal{T}_h = \{\Omega_1, \ldots, \Omega_e, \ldots, \Omega_f, \ldots, \Omega_{n_h}\}$$

The partitioning may be very general, it is only required that

$$\bar{\Omega} = \bigcup_{e=1}^{n_h} \bar{\Omega}_e \quad \text{and} \quad \Omega_e \cap \Omega_f = \emptyset. \tag{2}$$

The set of internal edges (faces in 3d) Γ_{int} of the partitioning is

$$\Gamma_{\text{int}} = \bigcup_{\Omega_e, \Omega_f} (\partial\Omega_e \cap \partial\Omega_f). \tag{3}$$

On each internal edge/face $\Gamma_{ef} = \partial\Omega_e \cap \partial\Omega_f$ with $e > f$ we have the normal $\mathbf{n}_{ef}$ pointing from Ω_e to Ω_f. Γ_D is the set of element edges/faces on the Dirichlet boundary and Γ_F the element edges/faces on the boundary where fluxes are prescribed.

The discontinuous Galerkin (DG) method approximates the weak solution of (1) in finite-dimensional subspaces of the broken Sobolev space

$$H^m(\mathcal{T}_h) = \{v \in L_2(\Omega) | \;\; v|_{\Omega_e} \in H^m(\Omega_e) \; \forall \Omega_e \in \mathcal{T}_h\} \tag{4}$$

with $m \geq 1$, i. e. functions may be discontinuous at element boundaries.

In the discrete scheme the spaces $H^m(\Omega_e)$ will be replaced by polynomials $P_{r_e}(\Omega_e)$ of degree r_e on element Ω_e. These polynomials are generated from the polynomials on the reference element $\hat{\Omega}$ via

$$P_{r_e}(\Omega_e) = \left\{ \varphi \mid \varphi = \hat{\varphi} \circ T_{\Omega_e}^{-1}, \; \hat{\varphi} \in \hat{P} = P_{r_e}(\hat{\Omega}) \right\}, \tag{5}$$

where $T_{\Omega_e} : \hat{\Omega} \to \Omega_e$ is the mapping from the reference element to the transformed element.

The degree r_e may vary from element to element but in our application it will be held fixed, i. e. $r_e = r$ for any e. Our discrete solution space therefore is

$$V_{h,r}(\mathcal{T}_h) = \prod_{e=1}^{n_h} P_r(\Omega_e). \tag{6}$$

Finally, we need the jump of a function $v \in H^m(\mathcal{T}_h)$ at a point $\mathbf{x}$ on the edge/face Γ_{ef}

$$[v]_{ef}(\mathbf{x}) = v|_{(\partial\Omega_e \cap \Gamma_{ef})}(\mathbf{x}) - v|_{(\partial\Omega_f \cap \Gamma_{ef})}(\mathbf{x}), \quad e > f, \tag{7}$$

and the average

$$\langle v\rangle_{ef}(\mathbf{x}) = \frac{1}{2}\left(v|_{(\partial\Omega_e\cap\Gamma_{ef})}(\mathbf{x}) + v|_{(\partial\Omega_f\cap\Gamma_{ef})}(\mathbf{x})\right), \quad e > f. \tag{8}$$

For two functions $v, w \in H^m(\mathcal{T}_h)$ we have the formula

$$[vw]_{ef} = [v]_{ef}\langle w\rangle_{ef} + \langle v\rangle_{ef}[w]_{ef}. \tag{9}$$

1.2 Weak Formulation

Taking the exact solution of (1), testing with a function $v \in H^m(\mathcal{T}_h)$, $m > \frac{3}{2}$, and integrating by parts we get

$$\begin{aligned}
\int_\Omega \nabla\cdot(\varrho\mathbf{j})v\ d\mathbf{x} &= \sum_{\Omega_e\in\mathcal{T}_h}\int_{\Omega_e} \nabla\cdot(\varrho\mathbf{j})v d\mathbf{x} \\
&= \sum_{\Omega_e\in\mathcal{T}_h}\left[-\int_{\Omega_e}\varrho\mathbf{j}\cdot\boldsymbol{\nabla}v\ d\mathbf{x} + \int_{\partial\Omega_e} v\varrho\mathbf{j}\cdot\mathbf{n}\ ds\right] \\
&= \sum_{\Omega_e\in\mathcal{T}_h} -\int_{\Omega_e}\varrho\mathbf{j}\cdot\boldsymbol{\nabla}v\ d\mathbf{x} + \sum_{\Gamma_{ef}\subseteq\Gamma_{\text{int}}}\int_{\Gamma_{ef}}[v\varrho\mathbf{j}\cdot\mathbf{n}_{ef}]\ ds \\
&\quad + \sum_{\Gamma_e\subseteq\Gamma_D}\int_{\Gamma_e} v\varrho\mathbf{j}\cdot\mathbf{n}\ ds + \sum_{\Gamma_e\subseteq\Gamma_F}\int_{\Gamma_e} vh\ ds\ .
\end{aligned} \tag{10}$$

For the jump term occuring in (10) we use (9) to obtain

$$[v\varrho\mathbf{j}\cdot\mathbf{n}_{ef}] = [v]\,\langle\varrho\mathbf{j}\cdot\mathbf{n}_{ef}\rangle + \langle v\rangle\,[\varrho\mathbf{j}\cdot\mathbf{n}_{ef}]\ .$$

The second term in this equation is the jump in the flux over inter-element boundaries which is zero (assuming that $\mathbf{j}$ is the solution of (1)), i. e. we get

$$[v\varrho\mathbf{j}\cdot\mathbf{n}_{ef}] = [v]\,\langle\varrho\mathbf{j}\cdot\mathbf{n}_{ef}\rangle\ . \tag{11}$$

Inserting Darcy's law (1b) into (10), using (11) and collecting all terms involving gravity we arrive at

$$\begin{aligned}
\int_\Omega \nabla\cdot(\varrho\mathbf{j})v\ d\mathbf{x} &= \sum_{\Omega_e\in\mathcal{T}_h}\int_{\Omega_e}\varrho\frac{\boldsymbol{K}}{\mu}\boldsymbol{\nabla}p\cdot\boldsymbol{\nabla}v\ d\mathbf{x} \\
&\quad - \sum_{\Gamma_{ef}\subseteq\Gamma_{\text{int}}}\int_{\Gamma_{ef}}[v]\,\langle\varrho\frac{\boldsymbol{K}}{\mu}\boldsymbol{\nabla}p\cdot\mathbf{n}_{ef}\rangle\ ds - \sum_{\Gamma_e\subseteq\Gamma_D}\int_{\Gamma_e} v\varrho\frac{\boldsymbol{K}}{\mu}\boldsymbol{\nabla}p\cdot\mathbf{n}\ ds \\
&\quad + \sum_{\Gamma_e\subseteq\Gamma_F}\int_{\Gamma_e} vh\ ds - \sum_{\Omega_e\in\mathcal{T}_h}\int_{\Omega_e}\varrho^2\frac{\boldsymbol{K}}{\mu}\mathbf{g}\cdot\boldsymbol{\nabla}v\ d\mathbf{x} \\
&\quad + \sum_{\Gamma_{ef}\subseteq\Gamma_{\text{int}}}\int_{\Gamma_{ef}}[v]\,\langle\varrho^2\frac{\boldsymbol{K}}{\mu}\mathbf{g}\cdot\mathbf{n}_{ef}\rangle\ ds + \sum_{\Gamma_e\subseteq\Gamma_D}\int_{\Gamma_e} v\varrho^2\frac{\boldsymbol{K}}{\mu}\mathbf{g}\cdot\mathbf{n}\ ds\ .
\end{aligned} \tag{12}$$

If we want to use (12) for the definition of a weak solution some form of continuity of p over inter-element boundaries is needed. This is done by requiring

$$\sum_{\Gamma_{ef} \subseteq \Gamma_{\text{int}}} \int_{\Gamma_{ef}} \langle \varrho \frac{\boldsymbol{K}}{\mu} \boldsymbol{\nabla} v \cdot \mathbf{n}_{ef} \rangle [p]_{ef} \; ds = 0, \tag{13}$$

i. e. $\langle \varrho \frac{\boldsymbol{K}}{\mu} \boldsymbol{\nabla} v \cdot \mathbf{n}_{ef} \rangle$ is used as a test function in this weak continuity. Dirichlet boundary conditions are also imposed in a weak form by requiring

$$\sum_{\Gamma_e \subseteq \Gamma_D} \int_{\Gamma_e} \left(\varrho \frac{\boldsymbol{K}}{\mu} \boldsymbol{\nabla} v \cdot \mathbf{n} \right) (p - g) \; ds = 0. \tag{14}$$

These two weak continuity requirements are *not* enforced strictly but rather are added as a *penalty* to the weak form (12).
Now we are able to state the complete bilinear form

$$\begin{aligned} a_h(p, v) = & \sum_{\Omega_e \in \mathcal{T}_h} \int_{\Omega_e} \varrho \frac{\boldsymbol{K}}{\mu} \boldsymbol{\nabla} p \cdot \boldsymbol{\nabla} v \; d\mathbf{x} \\ & + \sum_{\Gamma_{ef} \subseteq \Gamma_{\text{int}}} \int_{\Gamma_{ef}} \langle \varrho \frac{\boldsymbol{K}}{\mu} \boldsymbol{\nabla} v \cdot \mathbf{n}_{ef} \rangle [p] - [v] \, \langle \varrho \frac{\boldsymbol{K}}{\mu} \boldsymbol{\nabla} p \cdot \mathbf{n}_{ef} \rangle \; ds \\ & + \sum_{\Gamma_e \subseteq \Gamma_D} \int_{\Gamma_e} \left(\varrho \frac{\boldsymbol{K}}{\mu} \boldsymbol{\nabla} v \cdot \mathbf{n}_{ef} \right) p - v \left(\varrho \frac{\boldsymbol{K}}{\mu} \boldsymbol{\nabla} p \cdot \mathbf{n} \right) \; ds \end{aligned} \tag{15}$$

and right hand side

$$\begin{aligned} l_h(p, v) = & \sum_{\Omega_e \in \mathcal{T}_h} \int_{\Omega_e} \left(\varrho^2 \frac{\boldsymbol{K}}{\mu} \mathbf{g} \right) \cdot \boldsymbol{\nabla} v - qv \; d\mathbf{x} \\ & - \sum_{\Gamma_{ef} \subseteq \Gamma_{\text{int}}} \int_{\Gamma_{ef}} [v] \, \langle \varrho^2 \frac{\boldsymbol{K}}{\mu} \mathbf{g} \cdot \mathbf{n}_{ef} \rangle \; ds - \sum_{\Gamma_e \subseteq \Gamma_F} \int_{\Gamma_e} vh \; ds \\ & + \sum_{\Gamma_e \subseteq \Gamma_D} \int_{\Gamma_e} \left(\varrho \frac{\boldsymbol{K}}{\mu} \boldsymbol{\nabla} v \cdot \mathbf{n} \right) g - v \varrho^2 \frac{\boldsymbol{K}}{\mu} \mathbf{g} \cdot \mathbf{n} \; ds \; . \end{aligned} \tag{16}$$

The discrete problem to be solved then reads: Find $p_h \in V_{h,r}(\mathcal{T}_h)$ such that

$$a_h(p_h, v_h) = l_h(v_h) \quad \forall v_h \in V_{h,r}(\mathcal{T}_h). \tag{17}$$

1.3 Basic Properties

The bilinear form a_h is non-symmetric and positive semi-definite, i. e. we have $a_h(v_h, v_h) \geq 0$. The DG method exhibits optimal order convergence in the H^1-norm, i. e. $O(h^r)$ for degree r polynomials, if the solution is

sufficiently regular. Convergence in L_2 is $O(h^{r+1})$ for r odd and (suboptimal) $O(h^r)$ for r even. These results hold for polynomial degree $r \geq 2$. For $r < 2$ the method is unstable. Moreover, the DG approximation is locally mass conservative over elements making it very attractive for flow and transport calculations. For proofs of these results and the necessary assumptions we refer to [6], [7].

Another nice feature of the method is that the polynomial space

$$P_r = \{\mathbf{x}^{\boldsymbol{\alpha}} |\ |\boldsymbol{\alpha}| \leq r\} \tag{18}$$

can be used for every element type (here we use the usual multiindex notation). As an example we consider the 2d case with $r = 2$. Then the polynomials generated by the basis $\{1, x, y, x^2, xy, y^2\}$ are used for triangular as well as quadrilateral elements. There are six degrees of freedom per element in this case. Degrees of freedom in one element are coupled with each other and with all degrees of freedom in neighboring elements.

In the implementation we do not use the monomials as a basis for P_r. In order to avoid conditioning problems we use basis polynomials $\hat{\varphi}_i$ that are L_2-orthonormal on the corresponding reference element $\hat{\Omega}$:

$$(\hat{\varphi}_i, \hat{\varphi}_j)_{\hat{\Omega}} = \int_{\hat{\Omega}} \hat{\varphi}_i \hat{\varphi}_j \, dx = \delta_{ij} \quad i, j \in \{1, \ldots, n_r\}, \tag{19}$$

n_r the number of degrees of freedom per element.

These basis polynomials are generated from the monomials using symbolic Gram-Schmidt orthogonalization with Mathematica. Evaluation of the basis polynomials uses Horner's scheme to avoid roundoff errors.

By mapping the basis functions $\hat{\varphi}_k$ from the reference element to the transformed element we obtain the global basis

$$\Phi_h = \{\varphi_1, \ldots, \varphi_m\}.$$

Inserting the basis representation $p_h = \sum_{i=1}^{m} \mathbf{x}_i \varphi_i$ into (17) we obtain a large system of linear equations

$$\boldsymbol{A}\mathbf{x} = \mathbf{b} \tag{20}$$

with

$$\boldsymbol{A}_{ij} = a_h(\varphi_j, \varphi_i), \qquad \mathbf{b}_i = l_h(\varphi_i). \tag{21}$$

The aim of this paper is the efficient solution of this linear system with a computation time proportional to m, the total number of degrees of freedom.

We implemented the DG method and the multigrid solver in the PDE software package UG, see [1]. Using the abstractions provided by UG the method can be used for all element types supported by UG: triangles, quadrilaterals, tetrahedra, prisms, (four-sided) pyramids and hexahedra.

2 Development of a Multigrid Algorithm

Multigrid methods, described e. g. in [4], consist of smoothing, prolongation and restriction operators. Subsequently we show how these operators are defined in our method.

2.1 Nested Finite Element Spaces

We assume that a set of *nested* triangulations $\mathcal{T}^0 \dots \mathcal{T}^L$, with n_l elements each, is given. On every mesh level l we have the discrete spaces $V_r^l = V_{h,r}(\mathcal{T}^l)$ equipped with the basis

$$\Phi^l = \{\varphi_1^l, \dots, \varphi_{m_l}^l\}.$$

φ_i^l has support in exactly one element. The indices of the basis functions associated with element $\Omega_e \in \mathcal{T}^l$ are

$$I_e^l = \{j \mid \text{supp } \varphi_j^l \subseteq \bar{\Omega}_e\}. \tag{22}$$

Note that we use polynomial degree r on every mesh level. The corresponding bilinear form and right hand side on level l are denoted by a^l and l^l.

Since the triangulations are nested, the spaces are also nested, i. e. $V_r^0 \subseteq \dots \subseteq V_r^L$, and every coarse grid basis function can be expanded in fine grid basis functions:

$$\varphi_i^l = \sum_{j=1}^{m_{l+1}} \omega_{ij}^{l+1} \varphi_j^{l+1}. \tag{23}$$

We now explain how to compute the factors ω_{ij}^{l+1}. Let φ_i^l be a basis function with support in $\Omega_e \in \mathcal{T}^l$. Let $\Omega_f \in \mathcal{T}^{l+1}$ be an element obtained from the subdivision of Ω_e, i. e. $\Omega_f \subseteq \Omega_e$. By $\hat{\varphi}_i$, $\hat{\varphi}_j^{l+1}$ we denote basis functions on the reference element that are mapped to the global basis functions φ_i and φ_j^{l+1}.

Function φ_i^l restricted to Ω_f is a polynomial of degree r and can be represented uniquely with the basis functions on Ω_f. The representation can be computed on the reference element $\hat{\Omega}$ using

$$\begin{aligned}
& \varphi_i^l|_{\Omega_f} = \sum_{j \in I_f^{l+1}} \omega_{ij}^{l+1} \varphi_j^{l+1} \\
\Leftrightarrow \quad & \hat{\varphi}_i^l \circ T_{\Omega_e}^{-1} \circ T_{\Omega_f} = \sum_{j \in I_f^{l+1}} \omega_{ij}^{l+1} \hat{\varphi}_j^{l+1} \\
\Leftrightarrow \quad & \forall k \in I_f^{l+1} : \quad (\hat{\varphi}_i^l \circ T_{\Omega_e}^{-1} \circ T_{\Omega_f}, \hat{\varphi}_k^{l+1})_{\hat{\Omega}} = \sum_{j \in I_f^{l+1}} \omega_{ij}^{l+1} (\hat{\varphi}_j^{l+1}, \hat{\varphi}_k^{l+1})_{\hat{\Omega}} \ .
\end{aligned} \tag{24}$$

Since our basis polynomials are L_2-orthonormal on the reference element, the mass matrix is the identity and we get

$$\omega_{ij}^{l+1} = (\hat{\varphi}_i^l \circ T_{\Omega_e}^{-1} \circ T_{\Omega_f}, \hat{\varphi}_j^{l+1})_{\hat{\Omega}}. \tag{25}$$

The transformation $T_{\Omega_e}^{-1} \circ T_{\Omega_f} : \hat{\Omega} \to \hat{\Omega}$ depends only on the type of refinement and not on the shape of the individual element and the factors ω_{ij}^{l+1} could be precomputed. However, since UG knows many refinement rules in 3d, we currently evaluate (25) for every element using numerical quadrature.

The factors ω_{ij} from (25) are the entries of the restriction and prolongation matrices ($\boldsymbol{R} = \boldsymbol{P}^T$) as can be seen from a straightforward application of the variational multigrid formulation.

2.2 Smoothing Iterations

The smoother should remove all algebraic errors that are not handled by the coarse grid correction. In that sense it is complementary to the coarse grid correction. Typically, smoothers are single grid iterative schemes for the solution of linear systems, see [5] for an introduction.

The point-wise Jacobi and Gauß-Seidel iterations are not applicable since the matrix $\boldsymbol{A}$ is only positive semi-definite ($\mathbf{x}^T \boldsymbol{A} \mathbf{x} \geq 0$). Zero diagonal elements are obtained for inserting into the bilinear form a function that is constant on an element.

For this reason one has to use block iterations. The most natural block structure is given by combining all degrees of freedom corresponding to one element, i. e.:

$$\boldsymbol{A}^{l,e,f} = \left\{ \boldsymbol{A}_{ij}^l \right\}_{i \in I_e^l,\ j \in I_f^l} \tag{26}$$

and $\boldsymbol{A}^l$ being the block matrix

$$\boldsymbol{A}^l = \left\{ \boldsymbol{A}^{l,e,f} \right\}_{\Omega_e, \Omega_f \in \mathcal{T}^l}. \tag{27}$$

Unfortunately, Jacobi and Gauß-Seidel iterations with respect to this block structure are not effective smoothers. This can be explained as follows: The diagonal block $\boldsymbol{A}^{l,e,e}$ corresponds to a discretization of the flow equation on element Ω_e *including* a weak form of continuity of pressure over $\partial\Omega_e$. Because neighboring elements fix the "boundary conditions" for the current element the error on the inter-element boundaries is only removed very slowly. The situation can be remedied by an overlapping patch smoother, where, for every element Ω_e, all degrees of freedom of Ω_e and some neighborhood are updated. This iteration is an effective smoother but has a high operation count.

However, it turns out that an incomplete LU-decomposition on the block structure (27) is a very effective smoother and has a low operation count.

The only draw-back of ILU is that an additional matrix has to be stored. Note that the ordering of the blocks is not important unless some robustness with respect to discontinuous or anisotropic permeability is required.

If the permeability $\boldsymbol{K}$ is very different from element to element we observe the following problem: Assume that element Ω_e has permeability $\boldsymbol{K} = \epsilon \boldsymbol{I}$ and all neighboring elements Ω_f have permeability $\boldsymbol{K} = \boldsymbol{I}$ where $\boldsymbol{I}$ is the identity. Then all entries of the diagonal block $\boldsymbol{A}^{l,e,e}$ will have size $O(\epsilon)$ and all offdiagonal blocks $\boldsymbol{A}^{l,e,f}$ will have entries of size $O(1)$. The patch-wise smoothers do not work in this case. The ILU-smoother works if *blocks corresponding to low permeability elements are ordered first.* Therefore, for general permeability fields, we order the blocks with respect to increasing permeability.

3 Numerical Results

We now illustrate the effectiveness of the proposed multigrid procedure with numerical experiments.

3.1 Model Problems

The first example is taken from [6], [8]. We solve $-\Delta p = f$, $p = g$ on $\partial\Omega$, in the unit square $\Omega = (0,1)^2$ where f and g are chosen such that the exact solution

$$p(x,y) = e^{-\left((x-1/2)^2+(y-1/2)^2\right)}$$

is obtained.

The unit square is discretized with a triangulation $\mathcal{T}^0$ consisting of four equidistant quadrilateral elements. Finer grids are obtained through regular subdivision of each element into four quadrilaterals. Figure 1 shows the convergence of the DG method in the H^1-norm in comparison to a standard vertex-centered finite volume scheme (for triangular elements this scheme is explained e. g. in [3]). It is clearly seen that optimal order convergenve $O(h^r)$ is obtained in this case. Note that the horizontal axis shows execution time for assembling and solving the linear systems and not just number of degrees of freedom. The most accurate solution (4096 elements with $r = 6$ had 114688 degrees of freedom.

The linear systems have been solved using a multigrid V-cycle with $\nu_1 = \nu_2 = 1$ pre- and post-smoothing steps with ILU on the element-wise block structured matrix. Iterations were stopped when the initial residual has been reduced by 10^{-8} in the Euclidean norm. Number of iterations for standard multigrid applied to the low-order finite volume discretization and the new method for DG are shown in Table 1. It can be observed that the method convergences independently of h and r.

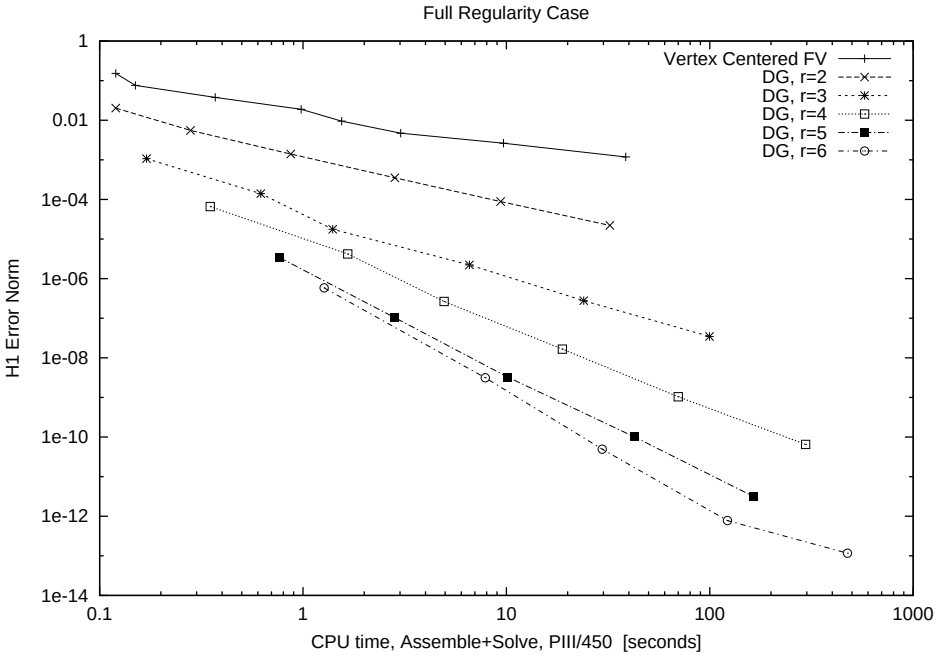

Figure 1: H^1-error for DG and vertex centered finite volume scheme in case of full regularity.

Table 1: Number of multigrid iterations for full regularity model problem.

h^{-1}	FV	$r=2$	$r=3$	$r=4$	$r=5$	$r=6$
4	3	5	5	5	5	4
8	4	7	6	6	5	6
16	4	7	6	6	5	6
32	4	7	6	6	5	6
64	4	7	6	6	5	6
128	4	6	6	6		
256	4					
512	4					

Table 2: Number of multigrid iterations for reentrant corner problem.

Triangles	FV	$r=2$	$r=3$	$r=4$	$r=5$	$r=6$
20	4	7	8	7	7	8
80	6	8	7	7	8	8
320	6	8	8	7	8	8
1280	6	9	8	7	8	8
5120	7	9	8	7		
20480	7	9				
81920	7					

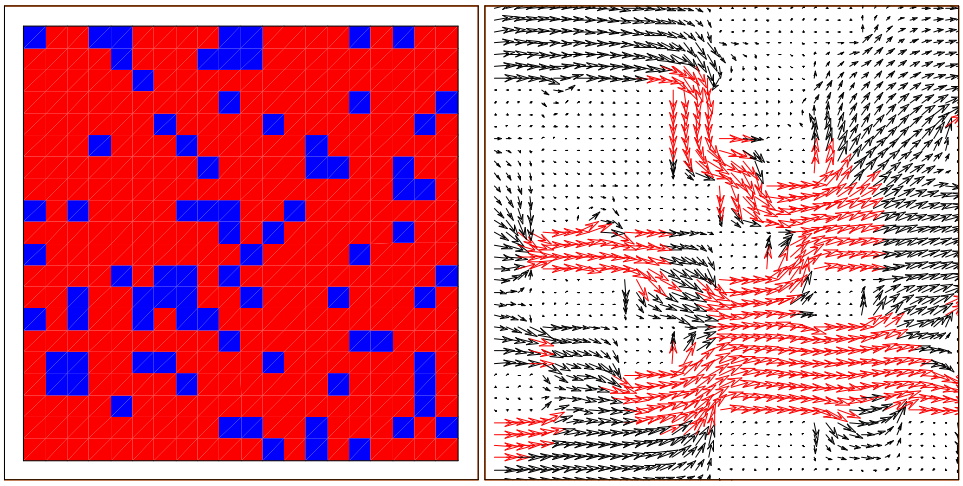

Figure 2: Permeability (left) and zoom of the flow field (right) for the discontinuous coefficient example.

In the second example we solve $-\Delta p = 0$, $p = g$ on $\partial\Omega$ in a domain with a reentrant corner, in this case 7/8 of a circle. Dirichlet boundary conditions are taken from the exact solution

$$p(r, \phi) = r^{\frac{4}{7}} \sin(\frac{4}{7}\phi)$$

in polar coordinates. Convergence order is now $O(h^{4/7})$ in the H^1-norm independent of the polynomial degree and the higher order method has no advantage in terms of error vs. computation time. However, the performance of the multigrid solver is of interest since it could also be used in combination with *hp*-adaptive refinement. Table 2 shows the number of multigrid iterations. Solver parameters were the same as in the full regularity problem. Similar asymptotic performance can be observed. The number of iterations is higher than in the full regularity case due to the fact that triangular elements are used and not because of a lack of regularity.

3.2 Discontinuous Coefficient Example

This example is taken from [3] and illustrates the accuracy of the method in the case of highly discontinuous coefficents.

We solve $-\nabla \cdot \{k(x, y)\boldsymbol{\nabla} p\} = 0$ in the unit square with $p = 1$ for $x = 0$, $p = 0$ for $x = 1$ and no flow boundary conditions for $y = 0$ and $y = 1$. The permeability field is defined on a 20×20 mesh and is shown in Figure 2 on the left. In dark areas the permeability is $k = 10^{-6}$ elsewhere it is $k = 1$.

The unit square is discretized with $20 \times 20 \times 2$ triangular elements such that the permeability field is resolved with coarse grid elements. Finer grids

Table 3: Total flux through the system for discontinuous coefficent example.

h^{-1}	FV	$r=2$	$r=3$	$r=4$	$r=5$	$r=6$	MFE
20	0.6991	0.5094	0.5152	0.5174	0.5232	0.5152	0.4508
40	0.6466	0.5179	0.5181	0.5208	0.5206		
80	0.6170	0.5194	0.5192	0.5201			
160	0.5998	0.5199	0.5198				
320	0.5890						
640	0.5816						

Table 4: Number of multigrid cycles in the discontinuous coefficient example.

l	h^{-1}	FV	$r=2$	$r=3$	$r=4$
1	40	6	14	14	16
2	80	7	14	12	15
3	160	7	13	12	
4	320	8			
5	640	9			

are obtained through regular refinement. The right half of Figure 2 shows a zoom of the flow field computed with $r = 3$ on the coarsest mesh.

In Table 3 we show results for the unknown total flux through the system. We compare the vertex centered finite volume method (which in this case is identical to P1 conforming finite elements), DG with $r = 2$ up to $r = 6$ and the lowest order mixed finite element method. The value for the mixed method is taken from [3]. The "exact" value has been given in [3] as 0.5205 which was obtained by computing approximations on a sequence of meshes up to 200×200 with a cell centered finite volume scheme and extrapolation to $h = 0$. The results clearly show the unsuitability of the standard finite element method for this type of problem. Moreover, the error in the mixed finite element solution on the coarsest mesh is about a factor six larger than the error in the DG result on the same mesh. However, the number of unknowns is also about 2.5 times larger for DG (4800 vs. 2000 in the non-hybridized version). This result clearly shows the suitability of the DG method since the MFE is considered optimal for this type of problem.

The performance of the multigrid method is shown in Table 4. For the discontinuous coefficent example we used a multigrid V-cycle with $\nu_1 = \nu_2 = 2$ ILU smoothing steps as a preconditioner in the BiCGSTAB-method, see [2]. The table shows the number of preconditioner evaluations needed to reduce the norm of the defect by 10^{-8}. Again the iteration numbers seem to be independent of h and r.

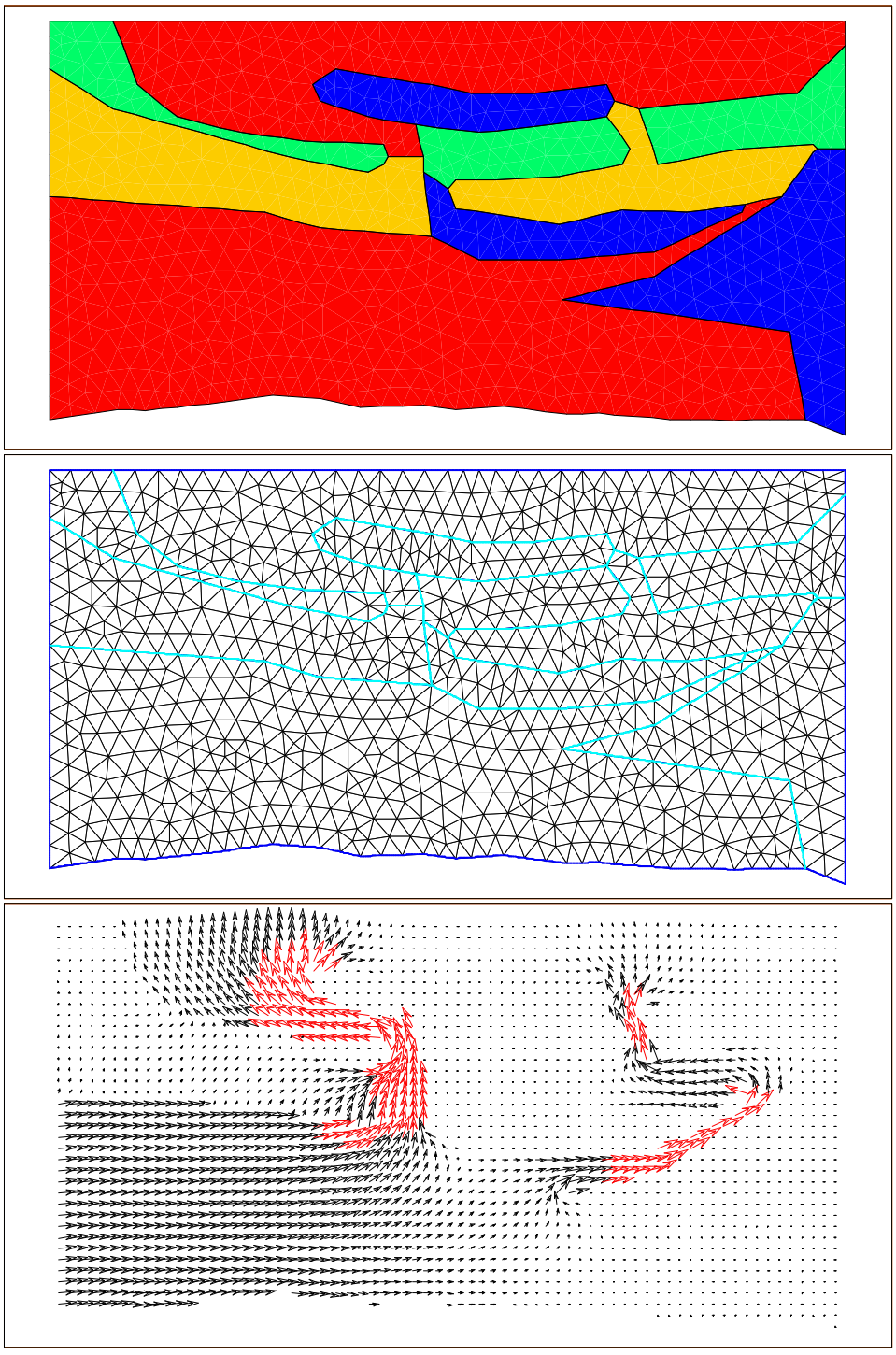

Figure 3: Permeability field (top), mesh and Darcy velocity (bottom) for the unstructured mesh example.

Table 5: Multigrid iterations and flux values for unstructured mesh example.

n_l (triangles)	1666	6664	26656	106624	426496
MG IT (FV)	1	5	5	6	6
MG IT (DG, $r = 2$)	1	10	13	13	
10^3·FLUX1, FV	1.4079	1.5497	1.6358	1.6926	1.6644
10^3·FLUX1, DG,$r = 2$	1.7307	1.7087	1.6967	1.6911	
10^4·FLUX2, FV	4.5173	4.7850	4.9394	5.0060	4.9450
10^4·FLUX2, DG,$r = 2$	4.7113	4.9171	4.9525	4.9598	

3.3 Unstructured Mesh Example

In order to asses the approximation quality and multigrid performance for more realistic groundwater applications we solve Eq. (1) in the domain shown in Figure 3. The permeability varies over four orders of magnitude and is shown in the top plot of Figure 3. The coarse mesh (middle plot) resolves the jumps in permeability and the Darcy velocity is shown in the bottom picture. In contrast to the previous examples this one is a vertical model including gravity forces.

Table 5 shows the number of multigrid cycles needed to reduce the residual norm by 10^{-6} and the fluxes through the left (FLUX1) and right narrow channels (FLUX2). We compare the vertex centered finite volume method and DG with $r = 2$. The multigrid V-cycle used $\nu_1 = \nu_2 = 2$ ILU smoothing steps and was employed as a preconditioner in BiCGSTAB. The flux values are again much more accurate for the DG method and show much better convergence.

4 Conclusions

In this paper we presented a multigrid method for higher order discontinuous Galerkin finite elements applied to the groundwater flow equation. We have shown experimentally that the method converges independent of the mesh size parameter and polynomial degree. The case of discontinuous permeability requires appropriate ordering of the unknowns and assumes that the jumps are resolved on the coarse grid.

We have also shown that the approximation using DG is considerably more accurate than with vertex centered finite volumes and it is even competitive with the mixed finite element method.

The code also implements the threedimensional case and can be used to do *hp* adaptive refinement. The DG method can also be applied to the solute transport equation and naturally yields higher order upwind and locally conservative discretizations. This has been implemented in the code will be the subject of a forthcoming paper.

Acknowledgement

The second author (V. R.) was supported by the BMBF-Project "Weiterentwicklung von Simulationstechniken für Gas-Wasser-Prozesse in geklüftet porösen Medien auf der Feldskala unter Berücksichtigung des Gaslösungsverhaltens."

References

[1] P. Bastian, K. Birken, S. Lang, K. Johannsen, N. Neuß, H. Rentz-Reichert, and C. Wieners. UG: A flexible software toolbox for solving partial differential equations. *Computing and Visualization in Science*, 1, 1997.

[2] H. A. Van der Vorst. BiCGSTAB: A fast and smoothly converging variant of Bi–CG for the solution of non–symmetric linear systems. *SIAM J. Sci. Stat. Comput.*, 13:631–644, 1992.

[3] L. J. Durlofsky. Accuracy of mixed and control volume finite element approximations to Darcy velocity and related quantities. *Water Resources Research*, 30(4):965–973, 1994.

[4] W. Hackbusch. *Multi–Grid Methods and Applications*. Springer–Verlag, 1985.

[5] W. Hackbusch. *Iterative Solution of Large Sparse Systems of Linear Equations*. Springer, 1994.

[6] J. T. Oden, I. Babuška, and C. E. Baumann. A discontinuous *hp* finite element method for diffusion problems. *Journal of Computational Physics*, 146:491–519, 1998.

[7] B. Rivière and M. F. Wheeler. Part I: Improved energy estimates for interior penalty, constrained and discontinuous Galerkin methods for elliptic problems. Technical Report 99-09, TICAM, 1999.

[8] B. Rivière and M. F. Wheeler. Part II: Discontinuous galerkin method applied to a single phase flow in porous media. Technical Report 99-10, TICAM, 1999.