SIAM J. SCI. COMPUT.
Vol. 25, No. 3, pp. 815–831

PRECONDITIONING METHODS FOR LOCAL DISCONTINUOUS GALERKIN DISCRETIZATIONS*

GUIDO KANSCHAT†

Abstract. A multilevel interior penalty method is used as an efficient preconditioner for the Schur complement of the local discontinuous Galerkin (LDG) discretization of a Poisson problem. The method is then used in a block-triangular preconditioner of the LDG saddle point system. The block preconditioner is of the same efficiency as the Schur complement version. Finally, the block preconditioner is extended to the discretization of the Stokes problem by the LDG method. Again, the preconditioned saddle point problem can be solved in about as many steps as the Schur complement. The influence of several parameters on the performance of these methods is investigated.

Key words. finite elements, local discontinuous Galerkin, multigrid, preconditioning, saddle point problem

AMS subject classifications. 65N22, 65F10, 65N55

DOI. 10.1137/S1064827502410657

1. Introduction.

In [9, 11] and [14], the local discontinuous Galerkin (LDG) method was introduced and analyzed by Cockburn et al. for Poisson and Stokes equations, respectively, in primal mixed form. In this paper, we address the problem of solving the arising finite dimensional systems of equations efficiently. To this end, we extend to these systems the multilevel preconditioner for the interior penalty method developed in [20]. First, it is applied directly to the Schur complement of the LDG scheme for the Laplacian operator. Then, we show that the solution of the whole saddle point system can be achieved efficiently by using the block preconditioners suggested in [24, 26]. We restrict ourselves to the self-adjoint case, where the spectrum of the Schur complement is real and positive.

Since discontinuous Galerkin methods have been applied to elliptic second order partial differential equations (see, e.g., [28, 1, 5, 17], the review [15], and the collection [16]), efficient solution methods for arising discrete systems are required. If Krylov space solvers like the conjugate gradient method or GMRES are applied, their performance is limited on finer grids by the fact that the eigenvalue distribution of the matrix associated with the discretization causes the number of iteration steps to be of order $1/h$ at least, where h is the diameter of the smallest grid cell (cf. Saad [27]). Therefore, we are looking for uniform preconditioners, that is, preconditioners yielding a condition number independent of the mesh size.

Recently, a multilevel approach for the interior penalty method (see Arnold [1]) has been proposed by Gopalakrishnan and Kanschat [20]. In [2], Arnold et al. showed that those discontinuous Galerkin methods for elliptic problems, which can be expressed solely in the primal variable, lead to equivalent bilinear forms. These results are exploited here to obtain a uniform preconditioner for the Schur complement form of the standard LDG method.

The discrete system produced by the LDG method has saddle point form, where the upper left block is the mass matrix. Klawonn [24] and Murphy, Golub, and

*Received by the editors July 4, 2002; accepted for publication (in revised form) May 2, 2003; published electronically November 21, 2003.
http://www.siam.org/journals/sisc/25-3/41065.html
†Institut für Angewandte Mathematik, Universität Heidelberg, Im Neuenheimer Feld 293/294, 69120 Heidelberg, Germany (kanschat@dgfem.org).

Wather [26] consider a general framework for preconditioning saddle point problems, which has been applied successfully to stable, conforming discretizations of Stokes and Oseen equations by Elman, Silvester, and Wathen in [18]. We will use this approach to extend the multilevel preconditioner in [20] to the saddle point matrices arising from the LDG discretization of Poisson equations [17, 9, 11] and Stokes equations [14, 12]. In the Poisson case we obtain a sharp estimate on the spectrum of the preconditioned matrix.

Castillo [7, 8] uses ILU decomposition for preconditioning of the LDG method. In this case, the number of iteration steps cannot be bounded independently of the refinement level unless unreasonable fill-in is allowed during ILU decomposition. Therefore, the results here improve on his. Uniform domain decomposition preconditioners for the interior penalty method have been proposed in [19, 25]. These should yield uniform preconditioners for the LDG method as well as for the multilevel method we propose by the arguments of section 4.

This paper is organized as follows. After introducing some notation in section 2 and the LDG method in section 3, we investigate the preconditioner of the Schur complement in section 4. This is then extended to the LDG saddle point system in section 5. Finally, Stokes equations are considered in section 6.

Computational results were obtained using the `deal.II` library for finite element computations (see [4, 3]).

2. Notation. Let Ω be a domain in $\mathbb{R}^d (d = 2, 3)$ with boundary $\partial\Omega$. $\mathbb{T}_\ell, \ell = 0, \dots, L$, is a family of quadrilateral or hexahedral subdivisions of Ω such that the cells of $\mathbb{T}_{\ell+1}$ are the result of refinement of the cells of $\mathbb{T}_\ell$. For simplicity, we will assume that $\bar{\Omega} = \bigcup_{T\in\mathbb{T}_0} T$.

The grid parameter h_ℓ is the piecewise constant function defined on each grid cell $T \in \mathbb{T}_\ell$ as $h_\ell|_T = \operatorname{diam} T$. We will assume that the coarse grid $\mathbb{T}_0$ is quasi-uniform; that is, there exists a (moderate) constant c such that $\max h_0 \le c \min h_0$. Furthermore, we assume shape regularity. If refinement is local, we assume that the mesh is one-irregular; that is, the refinement levels of two adjacent grid cells differ at most by one.

For functions $u, v \in L^2(\Omega)$, we introduce the scalar product

$$\big(u, v\big)_{\mathbb{T}_\ell} := \sum_{T\in\mathbb{T}_\ell} \int_T u(x)v(x)\,dx$$

in order to define the broken Sobolev space

$$H^1(\mathbb{T}_\ell) := \big\{v \in L^2(\Omega) \big| \forall\, T \in \mathbb{T}_\ell : v|_T \in H^1(T)\big\}$$

and its seminorm

$$\big|v\big|_{H^1(\mathbb{T}_\ell)} := \big(\nabla v, \nabla v\big)_{\mathbb{T}_\ell}.$$

By $\mathbb{E}_\ell$, we denote the set of edges (*edges* will always mean surfaces in three dimensions) of the triangulation $\mathbb{T}_\ell$; $\mathbb{E}_\ell^i$, $\mathbb{E}_\ell^D$ are the subsets of interior edges and edges on the Dirichlet boundary, respectively. We define the bilinear form

$$\big\langle u, v\big\rangle_{\mathbb{E}_\ell} := \sum_{E\in\mathbb{E}_\ell} \int_E u(x)v(x)\,ds \quad \text{for } u, v \in L^2\Big(\bigcup E \in \mathbb{E}_\ell E\Big).$$

On the reference cell $T_0 = (-1,1)^d$, we introduce the space of tensor product polynomials of degree p,

$$\mathbb{Q}_p := \left\{ q(\mathbf{x}) = \prod_{\iota=1}^{d} \phi_\iota(x_\iota) \middle| \phi_\iota \text{ is a polynomial of degree at most } p \right\}.$$

Each grid cell T is represented as $T = \Psi_T(T_0)$, where Ψ_T is a d-linear mapping defined on the reference cell. The space of mapped tensor product polynomials on the cell T is $\mathbb{Q}_p(T) = \mathbb{Q}_p \circ \Psi_T^{-1}$. Using these local shape function spaces, we define discrete spaces $X_\ell^p \subset H^1(\mathbb{T}_\ell)$ by

$$X_\ell^p := \left\{ v \in L^2(\Omega) \middle| \forall\, T \in \mathbb{T}_\ell : v|_T \in \mathbb{Q}_p(T) \right\}. \tag{2.1}$$

For fixed p, we will drop the index p if no confusion arises. We choose a basis of X_ℓ^p by selecting the Lagrangian interpolation polynomials with respect to equidistant support points as a basis $\{\phi^{(i)}\}_{i=1,\dots,(p+1)^d}$ for $\mathbb{Q}_p$ on T_0 and letting

$$X_\ell^p = \operatorname{span} \bigcup_{T \in \mathbb{T}_\ell} \{\phi^{(T,i)}\}_{i=1,\dots,(p+1)^d}, \tag{2.2}$$

where $\phi^{(T,i)} = \phi^{(i)} \circ \Psi_T^{-1}$. The coordinate vector of a function $u \in X_\ell^p$ with respect to this basis will be denoted by $\vec{u}$ and $u(x) = \sum_{T,i} \vec{u}_{T,i} \phi^{(T,i)}(x)$. We remark here that the choice of a Lagrange basis is rather arbitrary. In fact, when we choose a Legendre basis, mass matrices are diagonal on the reference cell. Since this advantage is maintained on affine quadrilaterals only and our operators include mass matrices with weight functions, we did not pursue this option.

Functions in $H^1(\mathbb{T}_\ell)$ and X_ℓ^p can be considered multivalued on the interior edges $E \in \mathbb{E}_\ell^i$. In order to define the bilinear forms, we introduce single valued numerical fluxes depending on u on the edge. First, we fix an orientation for E by denoting the two normal vectors by $\mathbf{n}^+$ and $\mathbf{n}^-$. Accordingly, for any point $x \in E$,

$$\begin{aligned} u^+(x) &:= \lim_{\varepsilon \searrow 0} u(x + \varepsilon \mathbf{n}^+), \\ u^-(x) &:= \lim_{\varepsilon \searrow 0} u(x + \varepsilon \mathbf{n}^-). \end{aligned}$$

The numerical fluxes are defined in terms of mean values

$$\{\!\{u\}\!\} := \frac{u^+ + u^-}{2}$$

and of jump terms. For any function ζ and any operator $\odot$, we define the jump

$$\begin{aligned} [\![\zeta \odot \mathbf{n}]\!] &:= \zeta^+ \odot \mathbf{n}^+ + \zeta^- \odot \mathbf{n}^- \\ &= (\zeta^+ - \zeta^-) \odot \mathbf{n}^+. \end{aligned}$$

These operators are independent of the choice of $\mathbf{n}^+$ by their definition and therefore are uniquely defined on each edge. Furthermore, we define the scalar jump

$$[\![\zeta]\!] := \zeta^+ - \zeta^-,$$

which will only appear squared in bilinear forms.

3. The LDG method. In this section, we briefly review the LDG method introduced and analyzed in [17, 9, 10, 11]. It is derived from the mixed formulation of Poisson equation

$$\begin{array}{rcrcll} \sigma & - & \nu\nabla u & = & 0 & \\ -\nabla\cdot\sigma & & & = & f & \text{in } \Omega. \end{array} \tag{3.1}$$

We assume Dirichlet boundary conditions on the whole boundary of Ω,

$$u = g \qquad \text{on } \partial\Omega.$$

Testing with a smooth function, integrating by parts on each cell, and inserting numerical fluxes $(\hat\sigma, \hat u)$ yield the following discontinuous Galerkin formulation (see [9]): Find $(\sigma, u) \in H^1(\mathbb{T}_L)^d \times H^1(\mathbb{T}_L)$ such that, for all (τ, v) from the same space,

$$\begin{array}{lcllcll} \left(\frac{1}{\nu}\sigma,\tau\right)_{\mathbb{T}_L} & + & \left(u,\nabla\cdot\tau\right)_{\mathbb{T}_L} & - & \left\langle \hat u, [\![\tau\cdot\mathbf{n}]\!]\right\rangle_{\mathbb{E}_L} & = & 0, \\ \left(\sigma,\nabla v\right)_{\mathbb{T}_L} & & & - & \left\langle \hat\sigma, [\![v\mathbf{n}]\!]\right\rangle_{\mathbb{E}_L} & = & \left(f,v\right)_{\mathbb{T}_L}. \end{array} \tag{3.2}$$

In the standard LDG method, these fluxes are chosen as

$$\hat u = \{\!\{u\}\!\}, \quad \hat\sigma = \{\!\{\sigma\}\!\} - \tfrac{1}{h}[\![u\mathbf{n}]\!] \quad \text{on } \mathbb{E}_L^i, \tag{3.3}$$

$$\hat u = g, \qquad \hat\sigma = \sigma - \tfrac{1}{h}(u-g)\mathbf{n} \quad \text{on } \mathbb{E}_L^D. \tag{3.4}$$

We introduce the bilinear forms

$$\begin{aligned} a_L(\sigma,\tau) &:= \left(\tfrac{1}{\nu}\sigma,\tau\right)_{\mathbb{T}_L}, \\ b_L(\sigma,v) &:= \left(-\nabla\cdot\sigma, v\right)_{\mathbb{T}_L} + \left\langle [\![\sigma\cdot\mathbf{n}]\!], \{\!\{v\}\!\}\right\rangle_{\mathbb{E}_L^i}, \\ c_L(u,v) &:= \left\langle \tfrac{1}{h}[\![u]\!], [\![v]\!]\right\rangle_{\mathbb{E}_L^i} + \left\langle \tfrac{1}{h}u, v\right\rangle_{\mathbb{E}_L^D} \end{aligned}$$

so that we can write (3.2) in the form (after integrating by parts again)

$$\begin{array}{lcllcll} a_L(\sigma,\tau) & - & b_L(\tau,u) & = & F_L(\tau), \\ b_L(\sigma,v) & + & c_L(u,v) & = & G_L(v). \end{array} \tag{3.5}$$

The linear functionals on the right-hand side are given by

$$\begin{aligned} F_L(\tau) &= \left\langle g, \tau\cdot\mathbf{n}\right\rangle_{\mathbb{E}_L^D}, \\ G_L(v) &= \left(f,v\right)_\Omega + \left\langle g, v\right\rangle_{\mathbb{E}_L^D}. \end{aligned}$$

The discontinuous Galerkin discretization is achieved finally by choosing $\sigma, \tau \in \Sigma_L = (X_L^p)^d$ and $u, v \in V_L = X_L^{p\prime}$ in (3.5). The choice of $p\prime = p$ and $p\prime = p+1$ yields stable discretizations for any $p \geq 1$.

Developing σ and u with respect to the basis (2.2), this yields the finite dimensional, stabilized saddle point system

$$\mathcal{A}x = \mathcal{F} \tag{3.6}$$

with

$$x := \begin{pmatrix}\vec\sigma\\ \vec u\end{pmatrix}, \qquad \mathcal{A} := \begin{pmatrix} A & -B^T \\ B & C\end{pmatrix}, \qquad \mathcal{F} := \begin{pmatrix}\vec f\\ \vec g\end{pmatrix}. \tag{3.7}$$

Here, the entries of A, B, C, $\vec{f}$, and $\vec{g}$ are computed by applying the corresponding form to the basis functions, that is, e.g.,

$$a_{T,i;S,j} = a_L(\phi^{(T,i)}, \phi^{(S,j)}), \qquad f_{T,i} = F_L(\phi^{(T,i)}).$$

In the following, matrices without level index are always on the finest level L.

Remark 3.1. The matrix A consists of two diagonal blocks which are both scaled versions of the mass matrix. Since the discretization is discontinuous, these blocks decouple further into smaller blocks consisting of scaled cell mass matrices for each mesh cell. If we chose orthogonal polynomials on T (with respect to the weight $1/\nu$), A would be a diagonal matrix (see [8]). Finally, C is a positive semidefinite matrix achieving a stabilization of the system.

Remark 3.2. Due to the simple structure of A, it is possible to eliminate σ_L completely from the discretization. However, a drawback in a general finite element coding framework is that this involves second neighbor couplings between the cells and that computation of matrix entries in this case is more complicated. Therefore, we keep the stresses σ in our discrete formulation.

An alternative choice of the numerical fluxes $\hat{u}$ and $\hat{\sigma}$ on interior faces was analyzed in [11] on Cartesian grids, the *superconvergent* LDG,

$$\hat{u}_s = u_\uparrow, \quad \hat{\sigma}_s = \sigma_\downarrow - [\![u\mathbf{n}]\!] \quad \text{on } \mathbb{E}_L^i, \tag{3.8}$$

where $\uparrow$ and $\downarrow$ denote the upwind and downwind values, respectively, with respect to a certain vector field β, . The fluxes at the boundary are still chosen as in (3.4). This method yields an extra half-order of convergence for the stresses σ. The bilinear forms $b(.,.)$ and $c(.,.)$ in (3.5) are replaced by

$$\begin{aligned}
\tilde{b}(\sigma, v) &:= -\big(\nabla \cdot \sigma, v\big)_{\mathbb{T}_L} + \big\langle [\![\sigma \cdot \mathbf{n}]\!], v_\uparrow \mathrm{sign}(\beta \cdot n)\big\rangle_{\mathbb{E}_L^i}, \\
\tilde{c}(u, v) &:= \big\langle [\![u]\!], [\![v]\!]\big\rangle_{\mathbb{E}_L^i} + \big\langle \tfrac{1}{h} u, v\big\rangle_{\mathbb{E}_L^D}
\end{aligned}$$

with corresponding matrices $\tilde{B}$ and $\tilde{C}$.

4. Preconditioning the Schur complement. Taking the Schur complement of (3.6) yields the system

$$\begin{aligned}
S\vec{u} &= \vec{f}_s, \\
S = BA^{-1}B^T + C, \qquad \vec{f}_s &= \vec{g} - BA^{-1}\vec{f},
\end{aligned} \tag{4.1}$$

which is positive definite and symmetric for both choices of the fluxes (3.3) and (3.8). Its related bilinear form will be denoted by

$$s_L(u, v) := \vec{u}^T S \vec{v}. \tag{4.2}$$

Remark 4.1. The evaluation of the Schur complement is particularly simple since the matrix A can be inverted explicitly in a preprocessing step and does not require an iterative solution.

In the following, we will use a preconditioner for a discretization of Poisson's equation using the primal variable u only, namely the interior penalty method [1]. Its weak formulation on grid level ℓ, $\ell = 0, \ldots, L$, reads as follows: Find $u \in V_\ell$ such that

$$\begin{aligned}
\breve{a}_\ell(u, v) \equiv \big(\nu \nabla u, \nabla v\big)_{\mathbb{T}_\ell} + \big\langle \nu \tfrac{\kappa}{h_E} [\![u]\!], [\![v]\!]\big\rangle_{\mathbb{E}_\ell} - \big\langle \nu \{\!\{\nabla u\}\!\}, [\![v\mathbf{n}]\!]\big\rangle_{\mathbb{E}_\ell} - \big\langle \nu [\![u\mathbf{n}]\!], \{\!\{\nabla v\}\!\}\big\rangle_{\mathbb{E}_\ell} \\
= (f, v) \quad \forall\, v \in V_\ell.
\end{aligned}$$

This bilinear form is positive definite and symmetric as soon as κ is sufficiently large (by an inverse estimate), and its matrix will be denoted by $\breve{A}_\ell$. In [20], a multilevel preconditioner $\breve{P}_\ell$, $\ell = 0, \dots, L$, was developed for this discretization. Given a smoothing method G_ℓ on level ℓ and a restriction $R_\ell : X^p_{\ell+1} \to X^p_\ell$ (which is the L^2 projection here), this method reads as follows.

ALGORITHM 1. Set $G^{(i)}_\ell = G_\ell$ if i is odd and $G^{(i)}_\ell = G^T_\ell$ if i is even (G^T_ℓ is the transpose of G_ℓ). Also set $x^0 = 0$ and $\breve{P}_0 = \breve{A}_0^{-1}$. For $\ell \geq 1$ and any vector $d_\ell \in \mathbb{R}^{N_\ell}$, $\breve{P}_\ell d_\ell$ can be computed by the following four steps, assuming that $\breve{P}_{\ell-1}$ is already defined:

1. Compute x^l for $i = 1, \dots, m(\ell)$:
$$x^i = x^{i-1} + G^{(i+m(\ell))}_\ell (d_\ell - \breve{A}_\ell x^{i-1}).$$
2. Set $y^{m(\ell)} = x^{m(\ell)} + R^T_{\ell-1} \breve{P}_{\ell-1} R_{\ell-1} (d_\ell - \breve{A}_\ell x^{m(\ell)})$.
3. Compute y^i for $i = m(\ell) + 1, \dots, 2m(\ell)$:
$$y^i = y^{i-1} + G^{(i+m(\ell))}_\ell (d_\ell - \breve{A}_\ell y^{i-1}).$$
4. Set $\breve{P}_\ell d_\ell = y^{2m(\ell)}$.

Due to Theorem 3.1 in [20], the preconditioned matrix $\breve{P}_L \breve{A}_L$ is $\breve{A}_L$-symmetric, and its condition number is bounded independently of the grid level L if the smoothers G_ℓ are of Jacobi or Gauss–Seidel type and $m(\ell) = 2^{L-\ell}$ (variable V-cycle). This fact is established by the estimate

$$c_p \breve{a}_L(u,u) \leq \breve{a}_L(\breve{P}_L \breve{A}_L u, u) \leq C_p \breve{a}_L(u,u) \tag{4.3}$$

with constants c_p and C_p independent of the mesh size function h. Based on this result, we obtain the following.

LEMMA 4.1. *The multilevel interior penalty preconditioner $\breve{P}_L$ defined in Algorithm 1 is a uniform preconditioner for the Schur complement of the LDG method with standard fluxes* (3.3)*; that is, there are constants c and C independent of the refinement level L such that*

$$c s_L(u,u) \leq s_L(\breve{P}_L S u, u) \leq C s_L(u,u).$$

Furthermore, the preconditioned matrix $\breve{P}_L S$ is S-symmetric.

Proof. In [2] the spectral equivalence estimate

$$c_s \breve{a}_L(u,u) \leq s_L(u,u) \leq C_s \breve{a}_L(u,u) \tag{4.4}$$

was shown to hold for the standard flux (3.3). Inserting $\tilde{u} = \breve{A}_L^{-1} S u$ into (4.3) and combining the result with (4.4) yield (cf. [6, Lemma 3.1])

$$\frac{c_p}{C_s} s_L(u,u) \leq s_L(\breve{P}_L S u, u) \leq \frac{C_p}{c_s} s_L(u,u). \tag{4.5}$$

Therefore, the preconditioner is uniform. S-symmetry follows from the symmetry of $\breve{P}_L$ (cf. [20]). ∎

We now continue showing numerical experiments assessing the performance of this preconditioner depending on the parameters of the two methods.

Let us first describe the test framework. In all cases, the start vector of the iteration is the zero vector. Unless stated otherwise, we solve (3.1) on the unit square

TABLE 1
Convergence of cg iteration for the preconditioned standard LDG Schur complement, $Q_2, \kappa = 8$. The preconditioned interior penalty method is shown on the right for comparison.

	Variable V-cycle			V-cycle (2)			IP, var. V-cycle		
L	$\mathrm{cond}_2 \breve{P}_L S_L$	n_{10}	t_{10}	$\mathrm{cond}_2 \breve{P}_L S_L$	n_{10}	t_{10}	$\mathrm{cond}_2 \breve{P}_L \breve{A}_L$	n_{10}	t_{10}
1	8.36	19	–	6.45	18	–	2.15	10	–
2	7.94	27	–	7.01	25	–	2.13	18	–
3	8.47	29	–	7.50	27	–	2.14	19	–
4	8.70	29	–	7.69	28	–	2.14	19	–
5	8.72	29	–	7.65	28	–	2.14	19	–
6	8.75	30	32	7.70	28	40	2.14	20	23
7	8.75	30	148	7.71	28	172	2.14	20	101
8	8.75	29	607	7.71	28	728	2.14	21	433

$(-1,1)^2$ with $\nu \equiv 1$. The grid on level 0 consists of a single cell; therefore $h = 2^{1-L}$. The right-hand side f is chosen such that $u(x,y) = e^x e^y$. Therefore, the start residual is not in a special subset of eigenspaces of the operator. An outer conjugate gradient iteration is preconditioned by the multilevel preconditioner with either $2^{L-\ell}$ (variable V-cycle) or 2 (V-cycle) smoothing steps on level ℓ, where L is the level of the finest grid. We use a block Gauss–Seidel smoother in the symmetric way described in Algorithm 1. The blocks are the cell-matrices with lexicographic ordering of the grid cells. This block version of the smoother has been shown to be superior to the point version in [21] for higher order polynomials. In fact, the point version degenerates dramatically if p grows, while the block version still yields good preconditioners.

First, we present results for the standard fluxes (3.3) and (3.4). In Table 1, we show convergence results for biquadratic polynomials. Let us give some brief comments on the numbers given in the table:

- *The condition number* $\mathrm{cond}_2 \breve{P}_L S$ *of the preconditioned matrix, computed as the condition number of the Lanczos matrix after the last cg-step.* This number is mostly of theoretical interest since it is only indirectly related to the numerical effort. Its dependence on the penalty parameter κ of the preconditioner is very clear.
- n_{10}, *the number of steps needed to reduce the residual by* 10^{10}. This number is independent of the implementation, but it can be misleading if preconditioners with different complexity are used. Therefore, we include t_{10} described next.
- t_{10}, *the time to achieve this reduction.* This is the number that really matters, but it is highly dependent on the implementation. We use sparse matrices provided by the `deal.II` library (see [3]) for all operators involved (stiffness matrix, prolongation and restriction operators, smoothers); therefore the numerical efficiency in all the different steps of the algorithm should be fairly balanced. The results have been obtained on SUN SPARC Ultra III 750 MHz processors. Times smaller than 10 seconds are not displayed, since they are not sufficiently accurate.

In Table 1, we see that all condition numbers and iteration counts are indeed bounded independently of ℓ. Furthermore, the bounds are of moderate size. Therefore, this preconditioner actually can be recommended for real computations. Although the variable V-cycle with one smoothing step on the finest level needs slightly more iterations, it is superior to the simple V-cycle with two smoothing steps on each level if computation times are compared.

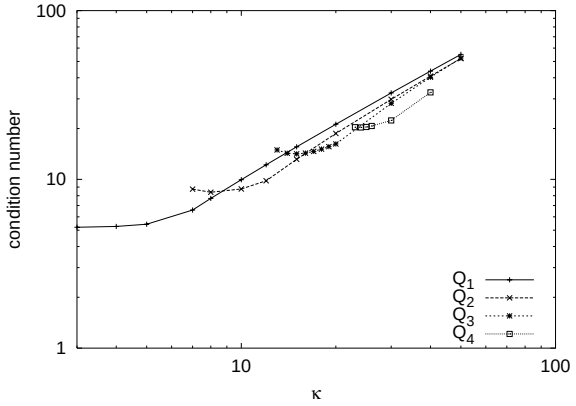

FIG. 1. *Condition numbers* $\mathrm{cond}_2 \check{P}S$ *of the preconditioned standard LDG Schur complement for different polynomial degrees.*

TABLE 2
Convergence of cg iteration for the preconditioned superconvergent LDG Schur complement $(Q_2, \kappa = 8)$*.*

	Variable V-cycle			V-cycle (2)		
L	$\mathrm{cond}_2 \check{P}_L \tilde{S}_L$	n_{10}	t_{10}	$\mathrm{cond}_2 \check{P}_L \tilde{S}_L$	n_{10}	t_{10}
1	18.24	22	–	16.96	22	–
2	20.94	38	–	18.56	37	–
3	23.42	49	–	20.40	44	–
4	24.98	50	–	22.11	46	–
5	25.90	51	–	22.97	46	10
6	26.37	51	52	23.40	46	62
7	26.62	51	246	23.62	46	282
8	26.74	50	1021	23.73	45	1155

In Figure 1, we show the condition numbers $\mathrm{cond}_2 \check{P}S$ (on a fine grid) for different polynomial degrees. The minimal possible stabilization parameter κ increases with the polynomial degree. On the other hand, the condition numbers depend only weakly on the polynomial degree if κ is fixed.

In Table 2, we show performance results for the Schur complement matrix $\tilde{S} = \tilde{B}A^{-1}\tilde{B}^T + \tilde{C}$ of the superconvergent LDG method with interior fluxes defined in (3.8). No spectral equivalence of $\tilde{S}$ with $\check{A}$ has been proven. Still, numerical evidence shown here suggests that $\check{P}_L$ is a uniform preconditioner in this case, too. This method requires about 60% more iteration steps than the standard method, while the complexity per step is the same (an optimal implementation could reduce the number of matrix entries slightly, due to the upwind fluxes).

Since the standard LDG already requires much more numerical effort than the interior penalty method (about 20 steps with only one matrix operation instead of seven for the LDG in two dimensions), we have to include the accuracy of the discretization in the comparison of these three methods. The following discussion should give a hint about which method to choose and when. For the results in Figure 2, the linear system was solved down to a residual norm of 10^{-13}. Then, for each method, solution time is plotted over the L^2-norm of the error of the solution u itself (left) and of its derivative σ (right), respectively. Here, the derivatives of the interior penalty solution are simply computed inside the cells. The figure shows that the interior

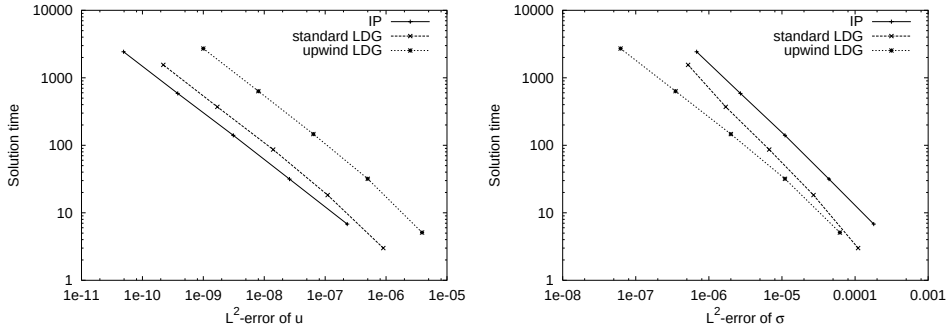

FIG. 2. *Performance of the different DG methods.*

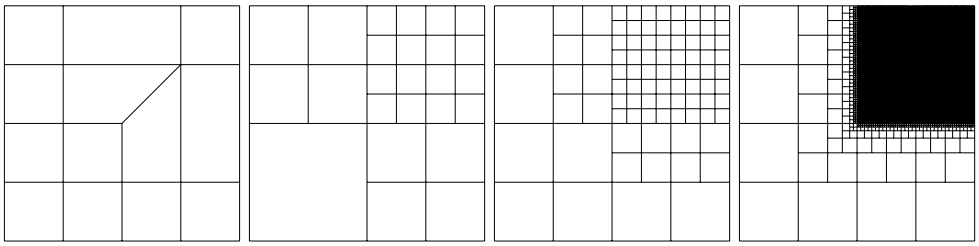

FIG. 3. *Distorted grid and locally refined meshes.*

penalty method is faster than the standard LDG by a factor of 2 if only the solution u is considered. In this case, the superconvergent LDG is slower by far. On the other hand, if derivatives are important, the picture is simply reversed. Superconvergent LDG is fastest due to its half-order gain in accuracy, followed by standard LDG. Therefore, a decision from among these three methods can only be made according to the quantities that have to be simulated accurately.

We finish this section by showing numerical experiments for the variable V-cycle on non-Cartesian grids. First, we generate meshes with hanging nodes by refining only the positive quadrant of the unit square. The remaining cells are refined such that each edge of a cell is divided at most once (hence asserting that the diameters of two adjacent cells differ only by a fixed factor, as required in [9]). Meshes with finest levels 2, 3, and 7 of this sequence are displayed in Figure 3. Smoothing on level ℓ is performed on cells belonging to this level in the grid hierarchy only (see the section on smoothing on subspaces in [6] and [22]). Table 3 shows that the condition numbers and iteration counts are hardly affected by the existence of hanging nodes.

Finally, we show results for the distorted grid on the left of Figure 3 and its globally refined versions in Table 4. This mesh was used in [23] to avoid superconvergence of the interior penalty method and to exhibit optimality of the derived L^∞-error estimate. Due to the distorted grid cells, the penalty parameter κ had to be chosen larger than on the Cartesian meshes, yielding higher condition numbers and iteration counts. Still, the preconditioner turns out to be uniform.

5. Preconditioning the LDG saddle point system. Instead of applying conjugate gradient iteration to the Schur complement of the LDG method, we can solve the original mixed system with GMRES or MINRES iteration. Then, we need a preconditioner for the mixed system (3.6). We construct this preconditioner using

TABLE 3
Performance of preconditioned cg on locally refined grids ($Q_2, \kappa = 8$, variable V-cycle).

	Standard		Superconvergent	
L	$\mathrm{cond}_2 \breve{P}_L S_L$	n_{10}	$\mathrm{cond}_2 \breve{P}_L \tilde{S}_L$	n_{10}
2	8.29	28	24.39	42
3	8.54	28	25.87	50
4	8.81	29	27.67	51
5	8.73	29	28.71	52
6	8.90	29	29.24	52
7	8.87	29	29.51	52
8	8.91	30	29.65	52

TABLE 4
Performance of preconditioned cg on distorted grids ($Q_2, \kappa = 14$, variable V-cycle).

	Standard		Superconvergent	
L	$\mathrm{cond}_2 \breve{P}_L S_L$	n_{10}	$\mathrm{cond}_2 \breve{P}_L \tilde{S}_L$	n_{10}
1	19.96	44	52.45	66
2	23.23	46	59.67	70
3	25.26	49	63.82	72
4	25.73	50	67.19	72
5	26.31	51	68.78	70
6	26.46	51	69.55	68

the result in [26] which states that

$$\begin{pmatrix} A & 0 \\ B & S \end{pmatrix}^{-1} \begin{pmatrix} A & -B^T \\ B & C \end{pmatrix} = \begin{pmatrix} I & -A^{-1}B^T \\ 0 & I \end{pmatrix}.$$

If we could compute the inverse on the left, the spectrum of the matrix would consist of the single value 1, and Krylov-space methods would converge in very few steps. Since the computational effort for this procedure is too high, we replace the inverse of the Schur complement by the preconditioner $\breve{P}_L$ of the previous section,

$$\mathcal{P} := \begin{pmatrix} A & 0 \\ B & \gamma_u^{-1} \breve{P}_L^{-1} \end{pmatrix}, \tag{5.1}$$

where $\gamma_u > 0$ is a tuning parameter.

LEMMA 5.1. *The matrix $\mathcal{A}\mathcal{P}^{-1}$ has only real eigenvalues, and its spectrum is*

$$\sigma(\mathcal{A}\mathcal{P}^{-1}) = \sigma(\gamma_u \breve{P} S) \cup \{1\}. \tag{5.2}$$

Proof. The proof is a modification of the proof of Lemma 3.1 in [24]. Instead of computing the spectrum of $\mathcal{A}\mathcal{P}^{-1}$, we consider the transpose system and solve the generalized eigenvalue problem

$$\mathcal{A}^T x = \lambda \mathcal{P}^T x,$$

or, splitting into components,

$$\begin{array}{rcrcrcl} A\vec{\sigma} & + & B^T\vec{u} & = & \lambda A\vec{\sigma} & + & \lambda B^T\vec{u}, \\ -B\vec{\sigma} & + & C\vec{u} & = & & & \lambda\gamma_u^{-1}\breve{P}_L^{-1}\vec{u}. \end{array} \tag{5.3}$$

If $\lambda = 1$, the first equation is void. In this case, we search for a nontrivial solution to

$$-B\vec{\sigma} + \underbrace{(C - \gamma_u^{-1}\breve{P}_L^{-1})}_{=:\hat{C}}\vec{u} = 0,$$

TABLE 5
Comparison of GMRES for the complete system and the Schur complement (Q_2, variable V-cycle, $\gamma_u = 1$).

	Standard version				Superconvergent				Standard version 3D			
	LDG system		Schur compl.		LDG system		Schur compl.		LDG system		Schur compl.	
ℓ	n_{10}	t_{10}	n_{10}	t_{10}	n_{10}	t_{10}	n_{10}	t_{10}	n_{10}	t_{10}	n_{10}	t_{10}
1	21	–	19	–	23	–	22	–	30	–	27	–
2	31	–	27	–	41	–	38	–	37	–	32	–
3	32	–	29	–	56	–	49	–	40	54.2	35	40.3
4	33	–	29	–	58	–	50	–	41	552.5	36	378.6
5	33	–	29	–	58	16.3	51	–				
6	33	46.7	29	33.6	58	85.2	51	52.3				
7	33	231.4	29	151.1	57	424.4	51	245.7				
8	33	1049.3	29	649.5	57	1935.2	50	1020.6				

which is an eigenvector to the eigenvalue 1. If $\hat{C}$ is singular, choose a nonzero pair $(\vec{\sigma}, \vec{u}) \in \ker B \times \ker \hat{C}$ arbitrarily. If $\hat{C}$ is regular, choose $\vec{\sigma} \neq 0$ arbitrarily and $\vec{u} := \hat{C}^{-1} B\vec{\sigma}$. Therefore, $\lambda = 1$ is an eigenvalue.

For $\lambda \neq 1$, we compute the Schur complement of (5.3), namely,

$$S\vec{u} = (BA^{-1}B^T + C)\vec{u} = \lambda \gamma_u^{-1} \breve{P}_L^{-1} \vec{u}.$$

Multiplying by $\gamma_u \breve{P}_L$ from the left, we see that λ is an eigenvalue of $\gamma_u \breve{P}_L S$ and $\vec{u}$ its associated eigenvector. Finally, the eigenvalues of $\mathcal{A}\mathcal{P}^{-1}$ are equal to those of $\mathcal{P}^{-T}\mathcal{A}^T$. ∎

Remark 5.1. Since the matrix $\mathcal{A}\mathcal{P}^{-1}$ is not normal, convergence results for Krylov-space methods cannot be applied solely based on the previous lemma. In [24], normality of the matrix is established with respect to another symmetric and positive definite matrix. Unfortunately, the theoretical results yield a number of iteration steps growing like $\mathcal{O}(2^L)$, a growth not observed in the numerical computations below.

Since the preconditioned matrix $\mathcal{A}\mathcal{P}^{-1}$ is nonsymmetric, we apply GMRES iteration and compare it to GMRES applied to the preconditioned Schur complement. The setup of the test calculations is the same as in the previous sections. The results in Table 5 show that solving the mixed system directly requires only a few more steps than solving for the Schur complement only. This observation holds even if we compare the results to those obtained with the conjugate gradient method in Table 1. Still, the computation time for the system is considerably longer due to the fact that vectors are three times as long for the system. The last column shows results for three-dimensional computations, where we obtain similar results. We remark again that the Schur complement iteration is competitive only, because the inverse of the mass matrix A in the Schur complement is very easily computed exactly in a preprocess such that $A^{-1}\vec{\sigma}$ is a simple matrix-vector multiplication.

The tuning parameter γ_u serves to shift the spectrum of $\breve{P}_L S$ (see the spectra in Figure 5) with respect to the additional eigenvalue 1 of the system. Figure 4 exhibits that the convergence speed of the preconditioned GMRES method is rather insensitive to γ_u within a wide range. This is expected from Lemma 5.1 since the single eigenvalue 1 does not critically influence the convergence of Krylov- space methods, even if isolated. Still, the introduction of a scaling will be important for solving Stokes equations below.

6. Stokes equations. In this section, we conclude by demonstrating that the preconditioning scheme of the previous section can be extended to a uniform

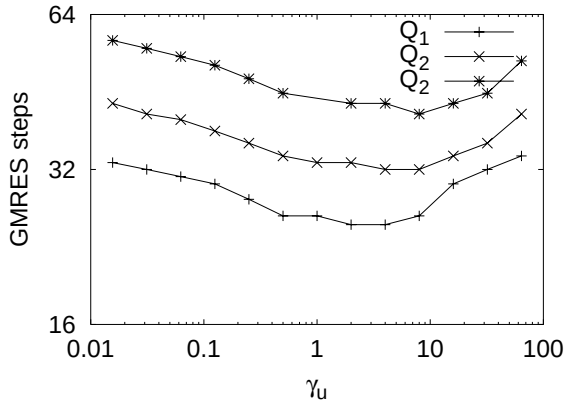

FIG. 4. *GMRES steps for the standard LDG system depending on the scaling factor* γ_u.

preconditioner for Stokes equations. Again, we consider the equations in mixed form

$$\begin{array}{rcrcrcll} \sigma_j & - & \nabla u_j & & & = & 0, & j=1,\dots,d, \\ \nabla\cdot\sigma & & & - & \nabla p & = & f, & \\ & & \nabla\cdot u & & & = & 0. & \end{array} \tag{6.1}$$

The "stress" variable σ is a tensor in the Stokes case and σ_j is its jth column. Furthermore, u_j is the jth component of the vector u.

The finite dimensional system of (6.1) considered here occurs from the discretization derived in [14]. It is a vector valued version of (3.2), augmented by a similar discretization of the incompressibility constraint. The bilinear form of the LDG discretization of Stokes equations reads

$$\begin{array}{rcrcrcrcll} a_L(\sigma_j,\tau_j) & - & b_L(\tau_j,u_j) & & & & = & F_{L;j}(\tau_j), & j=1,\dots,d, \\ b_L(\sigma_j,v_j) & + & c_L(u_j,v_j) & + & d_{L;j}(p,v_j) & = & G_{L;j}(v_j), & j=1,\dots,d, \\ d_L(q,u) & & & + & e_L(p,q) & = & H(q). & \end{array} \tag{6.2}$$

Here, $a_L(.,.)$, $b_L(.,.)$, and $c_L(.,.)$ are the forms in (3.5), and the additional forms are defined by

$$\begin{aligned} d_{L;j}(p,v) &:= \left(\partial_j v_j, p\right)_{\mathbb{T}_L} - \left\langle [\![v_j n_j]\!], \{\!\{ p \}\!\} \right\rangle_{\mathbb{E}_L^i}, \\ d_L(p,v) &:= \sum_{j=1}^d d_{L;j}(p,v_j), \\ e_L(p,q) &:= \left\langle h_E [\![p]\!], [\![q]\!] \right\rangle_{\mathbb{E}_L^i}. \end{aligned} \tag{6.3}$$

With matrices D and E being the matrices associated with the forms $d_L(.,.)$ and $e_L(.,.)$ as A is associated to $a_L(.,.)$ in (3.7) and

$$A_d := \begin{pmatrix} A & & \\ & \ddots & \\ & & A \end{pmatrix} \qquad \text{and} \qquad B_d := \begin{pmatrix} B & & \\ & \ddots & \\ & & B \end{pmatrix},$$

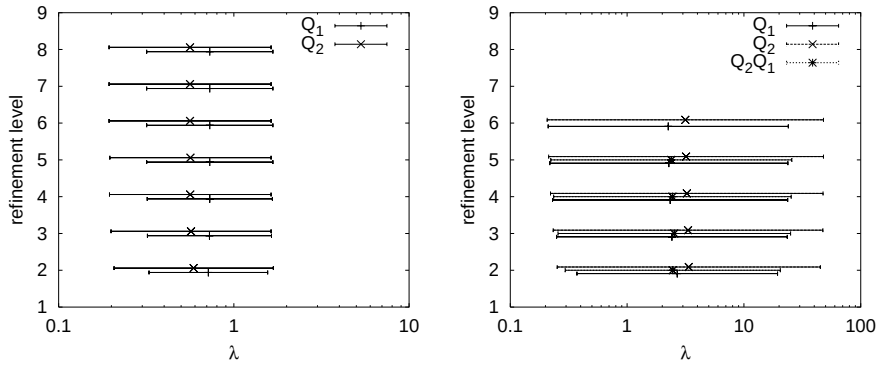

FIG. 5. *Envelopes of the spectra of the preconditioned LDG Schur complement for Poisson (left) and Stokes (right) equations.*

where the dots indicate d repetitions, the resulting linear system has the form

$$\underbrace{\begin{pmatrix} A_d & -B_d^T & & \\ B_d & C_d & -D^T & \\ & D & & E \end{pmatrix}}_{=:\mathcal{A}_S} \begin{pmatrix} \vec{\sigma} \\ \vec{u} \\ \vec{p} \end{pmatrix} = \begin{pmatrix} F \\ G \\ H \end{pmatrix}. \tag{6.4}$$

The condition number of the Stokes Schur complement $S_p = DS^{-1}D^T + E$ is independent of the mesh parameter (see [13]). Therefore, the mass matrix is a uniform preconditioner for the Schur complement of the system $\mathcal{A}_S$.

This proposition is verified in Figure 5 on the right for bilinear and biquadratic polynomials. The bars are envelopes of the spectrum of the Stokes Schur complement preconditioned by the mass matrix. On the finer grids, they exhibit a condition number in the order of magnitude of 100. This corresponds to about 110 conjugate gradient steps to reduce the error by a factor of 10^{10}. The bars on the left show the envelopes of the spectrum of the preconditioned LDG system for comparison. The test solution we employed here is again $u(x, y) = e^x e^y$ on the unit square, and the right-hand side f in (6.1) is chosen accordingly. Furthermore, we choose an inhomogeneous right-hand side in the third equation of (6.1) to account for the nonzero divergence of u.

Following [26] again, we derive the preconditioner

$$\mathcal{P}_S = \begin{pmatrix} A_d & & \\ B_d & \breve{P}_d^{-1} & \\ & D & \gamma^{-1}M \end{pmatrix} \tag{6.5}$$

for the complete system $\mathcal{A}_s$. Again, we introduced an additional tuning parameter γ. The parameter γ_u in the preconditioner $\mathcal{P}$ was set to 1 due to the results of the previous section.

Remark 6.1. A theoretical estimate of the convergence speed can be derived through Theorem 3.8 in [24]. Theorem 3.5 of that article can be applied to the preconditioner (6.5) and yields a condition number of the preconditioned system independent of L. Since the preconditioned matrix is not normal, the resulting number of GMRES iteration steps can only be bounded by $\mathcal{O}(1/h)$. As already remarked in [24], this increase is not observed in practice.

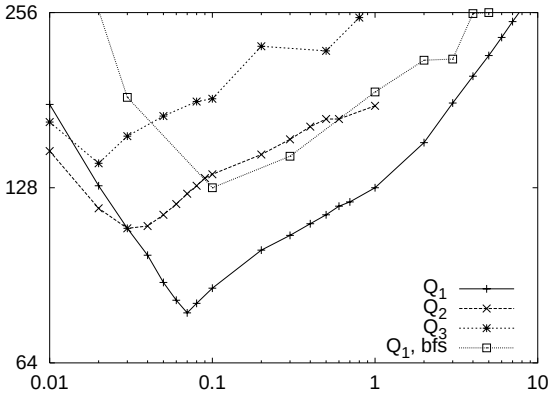

FIG. 6. *Iteration steps depending on the pressure scaling* γ.

TABLE 6
Performance for the preconditioned Stokes Schur complement (cg) and the Stokes system (GMRES).

	Schur-cg		System-GMRES	
L	n_{10}	t_{10}	n_{10}	t_{10}
2	37	–	70	–
3	62	33.8	79	–
4	71	195.6	84	23.8
5	76	1053.9	80	122.0
6	76	6038.4	81	633.3
7			82	3107.4

Figure 6 shows that the Stokes preconditioner is indeed sensitive to γ. Furthermore, $\gamma = 1$ is not optimal, and about 1/3 of the iteration steps can be saved by choosing γ appropriately for bilinear elements. Graphs are displayed for different polynomial degrees on the unit square as well as for a channel with backward facing step of length 6. While the optimal γ depends on the polynomial degree, it seems nearly independent of the geometry. In Table 6, we compare the performance of the conjugate gradient method for $M_p^{-1}S_p$ with GMRES for $\mathcal{P}_S^{-1}\mathcal{A}_S$. The results are for Q_2-polynomials and $\gamma = 0.03$. Computation times for the Schur complement are much higher, since the evaluation of S_p requires a complete solution of a Poisson problem. This must be performed with high accuracy in each step to obtain reliable results at all. Since the outer iteration count is not much larger for the preconditioned system, it beats the Schur complement clearly with only one variable V-cycle per iteration. Table 7 shows results for three-dimensional problems and polynomial spaces Q_1 and Q_2. Again, the iteration counts are bounded independent of the grid level.

The preconditioner of (6.5) can be modified by applying the velocity Schur complement preconditioner P several times to the mixed Poisson problem. This yields the following algorithm for $\mathcal{P}_m$.

ALGORITHM 2. Given an input vector (τ, v, q) and start vector 0,

1. perform a fixed number m of stationary iteration steps

$$\begin{pmatrix} \sigma_\iota^{k+1} \\ u_\iota^{k+1} \end{pmatrix} = \begin{pmatrix} \sigma_\iota^{k} \\ u_\iota^{k} \end{pmatrix} - \mathcal{P}\left(\begin{pmatrix} A^{-1} & B^T \\ -B & C \end{pmatrix} \begin{pmatrix} \sigma_\iota^{k} \\ u_\iota^{k} \end{pmatrix} - \begin{pmatrix} \tau_\iota \\ v_\iota \end{pmatrix} \right),$$

using the preconditioner $\mathcal{P}$ in (5.1) for $\iota = 1, \ldots, d$.

TABLE 7
Solution of the three-dimensional Stokes system with GMRES.

L	n_{10}, $Q_1, \kappa = 4$	n_{10}, $Q_2, \kappa = 8$
1	60	153
2	114	258
3	172	294
4	202	312
5	206	–

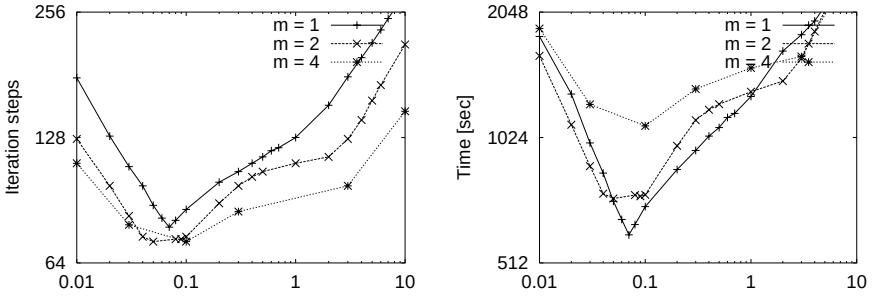

FIG. 7. *Iteration steps and time depending on the pressure scaling γ and number of velocity V-cycles.*

2. let the result of the preconditioning step be given by

$$\begin{aligned} \sigma &= \sigma^m, \\ u &= u^m, \\ p &= M^{-1}\left(q - \sum D_\iota u_\iota^m\right). \end{aligned}$$

An algorithm of this type is suggested for Oseen equations in [18]. Here, we only compare versions with different m for the simpler case of Stokes equations. Figure 7 shows the number of iteration steps and the computation times for different m in the algorithm above. For optimal γ, the version with $m = 1$ needs only a few steps more than the others. In fact, since the preconditioner for the Poisson part is much better than the one for the Stokes Schur complement (see Figure 5), we do not expect much effect from improving it. Due to the larger effort per preconditioning step, computation times are higher for the versions with higher m. On the other hand, increasing the iteration steps for the Poisson operator seems to diminish the dependency of the number of iteration steps on the pressure scaling γ. Therefore, increasing m may be advisable in more complex situations, where stability issues arise.

7. Conclusions. We applied the multilevel interior penalty method successfully as a preconditioner to the Schur complement of the LDG method. It was shown to be of optimal complexity as well as reasonably efficient for the standard version as well as for the superconvergent LDG method. Our results suggest that a spectral equivalence result similar to the one in [2] for the standard method may also hold for the superconvergent scheme; proof of this conjecture will be a subject of future work.

Using a general framework for preconditioners of saddle point systems, we showed that the same technique can be applied successfully to the saddle point system of the LDG method. Here, the preconditioned system has the same eigenvalues as the preconditioned Schur complement plus the additional eigenvalue 1.

Finally, the same ideas yielded a uniform preconditioner for Stokes equations. There, the performance is restricted mostly by the quality of the pressure Schur complement preconditioner, and more research has to be conducted in that direction.

REFERENCES

[1] D. N. Arnold, *An interior penalty finite element method with discontinuous elements*, SIAM J. Numer. Anal., 19 (1982), pp. 742–760.

[2] D. N. Arnold, F. Brezzi, B. Cockburn, and L. D. Marini, *Unified analysis of discontinuous Galerkin methods for elliptic problems*, SIAM J. Numer. Anal., 39 (2001), pp. 1749–1779.

[3] W. Bangerth, R. Hartmann, and G. Kanschat, `deal.II` *Differential Equations Analysis Library, Technical Reference*, 2002. Avaialbe online at http://www.dealii.org

[4] W. Bangerth and G. Kanschat, *Concepts for Object-Oriented Finite Element Software—the* `deal.II` *Library*, Preprint 43, SFB 359, Ruprecht-Karls University, Heidelberg, 1999. Available online at http://www.iwr.uni-heidelberg.de/sfb/Preprints1999.html

[5] F. Bassi and S. Rebay, *A high-order accurate discontinuous finite element method for the numerical solution of the compressible Navier-Stokes equations*, J. Comput. Phys., 131 (1997), pp. 267–279.

[6] J. H. Bramble, *Multigrid Methods*, Pitman Res. Notes Math. Ser. 294, Longman Scientific, Harlow, UK, 1993.

[7] P. Castillo, *Performance of discontinuous Galerkin methods for elliptic PDEs*, SIAM J. Sci. Comput., 24 (2002), pp. 524–547.

[8] P. Castillo, *Local Discontinuous Galerkin Methods for Convection-Diffusion and Elliptic Problems*, Ph.D. thesis, University of Minnesota, Minneapolis, MN, 2001.

[9] P. Castillo, B. Cockburn, I. Perugia, and D. Schötzau, *An a priori error analysis of the local discontinuous Galerkin method for elliptic problems*, SIAM J. Numer. Anal., 38 (2000), pp. 1676–1706.

[10] P. Castillo, B. Cockburn, D. Schötzau, and C. Schwab, *An optimal a priori error estimate for the hp-version of the local discontinuous Galerkin method for convection-diffusion problems*, Math. Comput., 71 (2002), pp. 455–478.

[11] B. Cockburn, G. Kanschat, I. Perugia, and D. Schötzau, *Superconvergence of the local discontinuous Galerkin method for elliptic problems on Cartesian grids*, SIAM J. Numer. Anal., 39 (2001), pp. 264–285.

[12] B. Cockburn, G. Kanschat, and D. Schötzau, *LDG methods for Stokes flow problems*, in Numerical Mathematics and Advanced Applications (ENUMATH 2001), F. Brezzi, A. Buffa, S. Corsaro, and A. Murli, eds., Springer Italia, Milan, 2003, pp. 755–764.

[13] B. Cockburn, G. Kanschat, and D. Schötzau,*The local discontinuous Galerkin method for the Oseen equations*, Math. Comput., to appear.

[14] B. Cockburn, G. Kanschat, D. Schötzau, and C. Schwab, *Local discontinuous Galerkin methods for the Stokes system*, SIAM J. Numer. Anal., 40 (2002), pp. 319–343.

[15] B. Cockburn, G. E. Karniadakis, and C.-W. Shu, *The development of discontinuous Galerkin methods*, in Discontinuous Galerkin Methods, B. Cockburn, G. E. Karniadakis, and C.-W. Shu, eds., Springer, Berlin, 2000, pp. 3–50.

[16] B. Cockburn, G. E. Karniadakis, and C.-W. Shu, eds., *Discontinuous Galerkin Methods*, Lecture Notes in Comput. Sci. Engrg. 11, Springer, Berlin, 2000.

[17] B. Cockburn and C.-W. Shu, *The local discontinuous Galerkin method for time-dependent convection-diffusion systems*, SIAM J. Numer. Anal., 35 (1998), pp. 2440–2463.

[18] H. C. Elman, D. J. Silvester, and A. J. Wathen, *Performance and analysis of saddle point preconditioners for the discrete steady-state Navier-Stokes equations*, Numer. Math., 90 (2002), pp. 665–688.

[19] X. Feng and O. A. Karakashian, *Two-level additive Schwarz methods for a discontinuous Galerkin approximation of second order elliptic problems*, SIAM J. Numer. Anal., 39 (2001), pp. 1343–1365.

[20] J. Gopalakrishnan and G. Kanschat, *A multilevel discontinuous Galerkin method*, Numer. Math., 95 (2003), pp. 527–550.

[21] J. Gopalakrishnan and G. Kanschat, *Multi-level preconditioners for the interior penalty method*, in Numerical Mathematics and Advanced Applications (ENUMATH 2001), F. Brezzi, A. Buffa, S. Corsaro, and A. Murli, eds., Springer Italia, Milan, 2003, pp. 795–804.

[22] G. Kanschat, *Practical Aspects of Multilevel Preconditioners for Discontinuous Galerkin Methods with Local Refinement*, Technical report, SFB 359, Ruprecht-Karls University, Heidelberg, in preparation.

[23] G. Kanschat and R. Rannacher, *Local error analysis of the interior penalty discontinuous Galerkin method for second order elliptic problems*, J. Numer. Math., 10 (2002), pp. 249–274.

[24] A. Klawonn, *Block-triangular preconditioners for saddle point problems with a penalty term*, SIAM J. Sci. Comput., 19 (1998), pp. 172–184.

[25] C. Lasser and A. Toselli, *Overlapping preconditioners for discontinuous Galerkin approximations of second order problems*, in Proceedings of the Thirteenth International Conference on Domain Decomposition Methods, N. Debit, M. Garbey, R. H. W. Hoppe, J. Périaux, D. Keyes, and Y. A. Kuznetsov, eds., DDM.org, 2001. Available online at http://ddm.org/DD13/homepage/

[26] M. F. Murphy, G. H. Golub, and A. J. Wathen, *A note on preconditioning for indefinite linear systems*, SIAM J. Sci. Comput., 21 (2000), pp. 1969–1972.

[27] Y. Saad, *Iterative Methods for Sparse Linear Systems*, 2nd ed., SIAM, Philadelphia, 2003.

[28] M. F. Wheeler, *An elliptic collocation-finite element method with interior penalties*, SIAM J. Numer. Anal., 15 (1978), pp. 152–161.