Application of unified DG analysis to preconditioning DG methods

Jayadeep Gopalakrishnan [a,*], Guido Kanschat [b]

[a] Department of Mathematics, University of Florida, Gainesville, FL 32611, USA
[b] Institut für Angewandte Mathematik, Universität Heidelberg, INF 293/294, D-69121, Heidelberg, Germany

Abstract

We point out an application of a recent unified analysis of several discontinuous Galerkin methods to accelerating the methods by a multigrid preconditioner.

Keywords: Discontinuous Galerkin; Finite element method; Multigrid

1. Background

A unified analysis of several discontinuous Galerkin (DG) methods appeared recently [2]. In this short note we show that this analysis permits analysis of multigrid preconditioners for several DG methods for elliptic problems. The multigrid analysis is a simple extension of the one in [7] for the interior penalty (IP) method [1]. We also provide results of some numerical studies.

We want to precondition DG methods on a mesh obtained by successive refinements. Let Ω be a convex polygon with a "coarse" quasiuniform triangulation $\mathcal{T}_1$. We obtain refinement $\mathcal{T}_j$ from $\mathcal{T}_{j-1}$ by connecting the midpoints of all edges of $\mathcal{T}_{j-1}$, for integers $2 \le j \le L$. Let Γ_k denote the union of boundaries of all triangles $\tau \in \mathcal{T}_k$. Let $V_k = \{v : v|_\tau$ is a polynomial of degree $\le d$ for all $\tau \in \mathcal{T}_k$, $v = 0$ on $\partial\Omega\}$, and $\boldsymbol{V}_k = (V_k)^2$.

Primal forms of various DG methods have been derived in [2]. Thus, efficient solution by these methods is possible by preconditioning the primal bilinear forms. We consider only those DG methods tabulated in Table 1. To describe the notation used therein, consider an edge e shared by triangles τ_1 and τ_2. For functions f defined on $\bar{\tau}_1 \cup \bar{\tau}_2$, let $\boldsymbol{n}_i$ denote the outward unit normal of τ_i and $f_i = f|_{\partial\tau_i}$. Define a jump vector on e: $[\![f]\!] = f_1\boldsymbol{n}_1 + f_2\boldsymbol{n}_2$. For vector functions $\boldsymbol{f}$, define $\{\boldsymbol{f}\} = \frac{1}{2}(\boldsymbol{f}_1 + \boldsymbol{f}_2)$ on e. Let $(\cdot,\cdot)_Z$ denote the $L_2(Z)$- (or $L_2(Z)^2$-) innerproduct, and $\langle u, v\rangle_k = \int_{\Gamma_k} uv\,\mathrm{d}s$. Following [2], define lifting operators $\boldsymbol{r} : (L_2(\Gamma_k))^2 \mapsto \boldsymbol{V}_k$ and $\boldsymbol{r}_e : (L_2(e))^2 \mapsto \boldsymbol{V}_k$ by $(\boldsymbol{r}(\phi), q)_\Omega = -\langle\phi, \{\boldsymbol{q}\}\rangle_k$, and $(\boldsymbol{r}_e(\phi), \boldsymbol{q})_\Omega = -(\phi, \{\boldsymbol{q}\})_e$, for all $\boldsymbol{q} \in \boldsymbol{V}_k, e \in \mathcal{E}_k$. Define forms

$$D_k(u,v) = \sum_{\tau\in\mathcal{T}_k} \int_\tau \nabla u \cdot \nabla v\,\mathrm{d}x,$$

$$\alpha_k^{\mathrm{r}}(u,v) = \sum_{e\in\mathcal{E}_k} \int_\Omega \eta_e \boldsymbol{r}_e([\![u]\!]) \cdot \boldsymbol{r}_e([\![v]\!])\,\mathrm{d}x,$$

$$\alpha_k^{\mathrm{j}}(u,v) = \int_{\Gamma_k} \frac{\eta_e}{h_e} [\![u]\!] \cdot [\![v]\!]\,\mathrm{d}s,$$

$$C_k(u,v) = \langle\{\nabla_k u\}, [\![v]\!]\rangle_k + \langle[\![v]\!], \{\nabla_k v\}\rangle_k,$$

where η_e is some positive number for each $e \in \mathcal{E}_k$, and h_e denotes the length of edge e. This defines all notation used in Table 1. Note that we have restricted ourselves to bilinear forms for solving the Poisson equation. Also, for simplicity, we have not considered the most general form of LDG.

All the methods in Table 1 are symmetric, and consistent [2]. We assume that η_e is chosen so that we always are in the regime of stability of these methods. Many estimates we need have already been proven in [2]. Let $\|\cdot\|_Z$ denote

Table 1
DG methods

Method	Definition of primal bilinear form $a_k(u,v)$
IP [1]	$D_k(u,v) - C_k(u,v) + \alpha^j(u,v)$
LDG [6]	$D_k(u,v) - C_k(u,v) + (\boldsymbol{r}([\![u]\!]), r([\![v]\!]))_\Omega + \alpha^j(u,v)$
BRMPS [3]	$D_k(u,v) - C_k(u,v) + \alpha^r(u,v)$
BMMPR [5]	$D_k(u,v) - C_k(u,v) + \big(r([\![u]\!]), r([\![v]\!])\big)_\Omega + \alpha^r(u,v)$

* Corresponding author. E-mail: jayg@math.ufl.edu

Computational Fluid and Solid Mechanics 2003
K.J. Bathe (Editor)

the $L^2(Z)$-norm. Define

$$|||v|||_k = \left(D_k(v,v) + \sum_{e\in\mathcal{E}_k} \frac{1}{h_e} \|[\![v]\!]\|_e^2 \right)^{1/2}.$$

For two real numbers A and B, we write $A \lesssim B$ to indicate that an inequality $A \le cB$ holds with some constant $c > 0$ independent of refinement level k and mesh sizes $\{h_k\}$. It follows from arguments in [2] that

$$a_k(u,v) \lesssim |||u|||_k |||v|||_k, \text{ and } |||u|||_k^2 \lesssim a_k(u,u), \tag{1}$$

for all $u, v \in V_k$. Moreover, the function $U_k \in V_k$ that satisfies $a_k(U_k, v) = (f, v)_\Omega$, for all $v \in V_k$, approximates the solution of $-\Delta U = f$ on Ω, with $U = 0$ on $\partial\Omega$, in the following sense [2]:

$$|||U - U_k|||_k \lesssim h_k \|f\|_\Omega, \quad \text{for all } f \in L^2(\Omega). \tag{2}$$

2. Multigrid results

Here we present uniform multigrid preconditioners for each $a_k(\cdot,\cdot)$ of Table 1. Smoothing iterations of the form

$$u^{(i+1)} = u^{(i)} + R_k(g - A_k u^{(i)}), \quad i = 0, 1, \ldots \tag{3}$$

are important ingredients of a multigrid algorithm. Here $u^{(0)} \in V_k$ is an initial guess, $g \in V_k, R_k$ is a "smoother", and $A_k : V_k \mapsto V_k$ is defined by $(A_k u, v)_\Omega = a_k(u,v)$, for all $u, v \in V_k$. Thanks to the now well known theory of subspace correction methods, smoothers based on $a_k(\cdot,\cdot)$ are easy to construct. E.g., decompose V_k by $V_k = V_{k,1} \oplus V_{k,2} \oplus \cdots \oplus V_{k,N}$ where $V_{k,j}$ is the space of restrictions of functions of V_k to jth triangle of $\mathcal{T}_k$. A scaled Jacobi smoothing iteration (with scaling factor γ) corresponding to this subspace decomposition is an iteration of the form (3), usually implemented as follows: Given $u^{(i)}$, set $\delta = g - A_k u^{(i)}$. Solve for $w_j \in V_{k,j}$ satisfying

$$a_k(w_j, v) = (\delta, v), \quad \text{for all } v \in V_{k,j},$$

by inverting small matrices. Then $u^{(i+1)}$ of (3) is given by $u^{(i+1)} = u^{(i)} + \gamma \sum_j w_j$. If one multiplicatively combines the local solves above, one gets a Gauss–Seidel smoother in a standard fashion. Note that we can compute smoothers so easily because $a_k(\cdot,\cdot)$ uses only local lifting operators. E.g., $\boldsymbol{r}([\![u]\!])$ for $u \in V_{k,j}$ is supported only on the jth triangle of $\mathcal{T}_k$ and the (at most three) triangles of $\mathcal{T}_k$ that share an edge with it.

Remark 2.1. Smoothers can be implemented without using the lifting operators. E.g., when LDG is implemented as a saddle point system, if all degrees of freedom involve values interior to elements, it is natural to group unknowns within elements into blocks. With this block partitioning, the standard block Jacobi method applied to the LDG Schur complement matrix is identical to the smoothing iteration given previously.

Algorithm 2.1. Let $R_k^{(l)} = R_k$ if l is odd, and $R_k^{(l)} = R_k^t$ if l is even (R_k^t is the L^2-adjoint of R_k). Let Q_k denote the L^2-orthogonal projection into V^k. Also set $x_0 = 0$ and $B_1 = A_1^{-1}$. For $k \ge 2$ and any function $d_k \in V_k$, $B_k d_k$ can be computed by the following four steps, assuming that B_{k-1} is already defined.

(1) Compute x^l for $l = 1, \ldots, m(k)$:

$$x^l = x^{l-1} + R_k^{(l+m(k))}(d_k - A_k x^{l-1}).$$

(2) Set $y^{m(k)} = x^{m(k)} + B_{k-1} Q_{k-1}(d_k - A_k x^{m(k)})$.

(3) Compute y^l for $l = m(k)+1, \ldots, 2m(k)$:

$$y^l = y^{l-1} + R_k^{(l+m(k))}(d_k - A_k y^{l-1}).$$

(4) Set $B_k d_k = y^{2m(k)}$.

We assume that the number of smoothings $m(k)$ increases as k decreases in such a way that $\beta_0 m(k) \le m(k-1) \le \beta_1 m(k)$, with $1 < \beta_0 \le \beta_1$. A typical choice is $m(k) = 2^{L-k}$. The following theorem shows that B_L is a good preconditioner for A_L. The spectral condition number of $B_L A_L$ is bounded independently of mesh-sizes.

Theorem 2.1. *For all* $u \in Vk$, $a_k(u,u) \lesssim a_k(B_k A_k u, u) \lesssim a_k(u,u)$.

Proof. The proof proceeds by verifying the conditions of the abstract multigrid theory of [4]. The locality of the lifting operators easily proves the "smoothing condition" for smoothers based on the above mentioned decomposition. It suffices to verify the "regularity and approximation condition". Define $P_{k-1} : V_k \mapsto V_{k-1}$ by $a_{k-1}(P_{k-1}u, v_{k-1}) = a_k(u, v_{k-1})$, for all $u \in V_k, v_{k-1} \in V_{k-1}$. The condition requires that there is an $0 < \alpha \le 1$ such that $|a_k(u - P_{k-1}u, u)| \lesssim h_k^{2\alpha} \|A_k u\|_\Omega^{2\alpha} a_k(u,u)^{1-\alpha}$ for all $u \in V_k$, and for all $k = 2, \ldots, J$. By (1), this condition follows with $\alpha = 1/2$ if we prove

$$|||u - P_{k-1}u|||_k \lesssim h_k \|A_k u\|_\Omega, \quad \text{for all } u \in V_k. \tag{4}$$

Proof of (4) proceeds exactly as the proof of Lemma 3.1 of [7], but using the more general error estimate (2) in place of IP error estimates. □

References

[1] Arnold DN. An interior penalty finite element method with discontinuous elements. SIAM J Numer Anal 1982;19(4): 742–760.

[2] Arnold DN, Brezzi F, Cockburn B, Marini LD. Unified analysis of discontinuous Galerkin methods for elliptic problems. SIAM J Numer Anal2001/2;39(5):1749–1779.

[3] Bassi F, Rebay S, Mariotti G, Pedinotti S, Savini M. A high-order accurate discontinuous finite element method for inviscid and viscous turbomachinery flows. In: 2nd Eur Conf on Turbomachinery Fluid Dynamics and Thermodynamics, 1997, pp. 99–108.

[4] Bramble JH, Pasciak JE, Xu J. The analysis of multigrid algorithms with nonnested spaces or noninherited quadratic forms. Math Comp 1991;56(193):1–34.
[5] Brezzi F, Manzini G, Marini D, Pietra P, Russo A. Discontinuous finite elements for diffusion problems. In: Atti Convegno in onore di F. Brioschi, 1999, pp. 197–217.
[6] Cockburn B, Shu C-W. The local discontinuous Galerkin method for time-dependent convection–diffusion systems. SIAM J Numer Anal 1998;35(6):2440–2463.
[7] Gopalakrishnan J, Kanschat G. A multilevel discontinuous Galerkin method. Numer Math (to appear).