Robust smoothers for high order discontinuous Galerkin discretizations of advection-diffusion problems

Guido Kanschat

Texas A&M University, Department of Mathematics, College Station, TX 77843-3368

Abstract

The multigrid method for discontinuous Galerkin discretizations of advection-diffusion problems is presented. It is based on a block Gauss-Seidel smoother with downwind ordering honoring the advection operator. The cell matrices of the DG scheme are inverted in this smoother in order to obtain robustness for higher order elements. Employing a set of experiments, we show that this technique actually yields an efficient preconditioner and that both ingredients, downwind ordering and blocking of cell matrices are crucial for robustness.

Key words: multigrid, discontinuous Galerkin, finite elements, advection, high order

1 Introduction

Discretizations of partial differential equations by higher order finite element methods or hp-methods gained popularity during the last years after computing power and memory became readily available. These methods yield much higher accuracy with similar degrees of freedom than standard schemes based on linear or bilinear polynomials. Nevertheless, the utility of these schemes depends very much on the availability of efficient solvers for the discrete problems. In this article, we continue previous work and demonstrate that multilevel methods with suitably chosen smoothers can provide such solvers, even in the presence of non self-adjoint advection terms and locally refined meshes.

In [8], we devised a multilevel preconditioner for the symmetric interior penalty method and proved it's optimality. These results were extended to other adjoint-

Email address: kanschat@tamu.edu (Guido Kanschat).
URL: http://www.math.tamu.edu/~kanschat (Guido Kanschat).

consistent discontinuous Galerkin (DG) methods for the Laplacian in [7] and to locally refined meshes with higher order elements in [11]. We also demonstrated in [8] that this preconditioner yields good results when applied to advection-diffusion problems with low order elements.

Here, we demonstrate that suitably chosen block Gauss-Seidel smoothers yield efficient multilevel methods for higher order discretizations of advection-diffusion problems. Since a sufficient theoretical framework for the analysis of these schemes is still missing, we demonstrate feasibility using generic model situations. While the computations in this article are all performed with the symmetric interior penalty method, they apply to all other adjoint-consistent DG methods for the Laplacian as well, as arguments in [7] and [1] show.

We consider the methods presented here not so much as an alternative to the domain decomposition methods developed in [6] and [1] than as complementing those schemes. While Schwarz methods provide efficient preconditioners by splitting the domain in subdomains, multilevel methods lend themselves as the tool of choice for the solution of the subproblems on each domain.

The methods described in this article were developed in the framework of the finite element library deal.II (see [3,4]) and have become part of it. deal.II is open source software and frealy available from `www.dealii.org`.

2 Description of the multilevel method

We consider the equation

$$-\nu\Delta u + w\cdot\nabla u = f \tag{1}$$

on Lipschitz-bounded polygonal domains $\Omega \subset \mathbb{R}^d$ for dimensions $d = 2, 3$ with Dirichlet boundary conditions $u = g$ on $\partial\Omega$. Let $\mathbb{T}_0$ be a coarse-level partition of Ω into quadrilaterals and hexahedra, respectively.

In order to define the multilevel preconditioner as introduced in [8], we assume that we have a hierarchy of meshes

$$\mathbb{T}_0 \sqsubseteq \mathbb{T}_1 \sqsubseteq \cdots \sqsubseteq \mathbb{T}_L,$$

where each cell of $\mathbb{T}_\ell$ with $\ell \geq 1$ is either obtained by regular refinement of a cell of $\mathbb{T}_{\ell-1}$ into 2^d children or is a cell of $\mathbb{T}_{\ell-1}$ itself (the latter only in case of local refinement). The set of faces between mesh cells and at the boundary is called $\mathbb{F}_\ell$, split into the sets $\mathbb{F}_\ell^i$ and $\mathbb{F}_\ell^D$ of faces in the interior and on the boundary, respectively.

Associated with these subdivisions is a hierarchy of nested finite element spaces

$$V_0 \subset \cdots \subset V_L.$$

These spaces are chosen such that the restriction of V_ℓ to a single cell $T \in \mathbb{T}_\ell$ is $\mathbb{Q}_k$, the space of mapped tensor product polynomials of degree k. Since we are considering discontinuous Galerkin methods, no further continuity between mesh cells is required.

Between the spaces V_ℓ, we define the prolongation operator $R^T_{\ell+1}$ from $V_{\ell-1}$ to V_ℓ as the canonical embedding and the restriction operator R_ℓ as the L^2-projection from V_ℓ to $V_{\ell-1}$. These are standard choices for finite element multilevel methods. We remark that the L^2-projection for DG discretizations is a local operation. Therefore, it is the identity on mesh cells that are not refined beyond level $\ell - 1$.

2.1 Discretization

On each level space V_ℓ, we define bilinear forms associated with DG discretizations of the Laplacian Δu and the advection operator $w{\cdot}\nabla u$. To facilitate the presentation, we introduce the following notation for jumps and mean values of discontinuous functions on faces:

$$\{\!\{u\}\!\} := \frac{u^+ + u^-}{2} \qquad\qquad [\![u\mathbf{n}]\!] := u^+\mathbf{n}^+ + u^-\mathbf{n}^-.$$

Furthermore, we abbreviate sums of integrals as

$$\Big(u, v\Big)_{\mathbb{T}_\ell} \equiv \sum_{T\in\mathbb{T}_\ell} \int_T u(x)v(x)\,dx, \qquad \Big\langle u, v\Big\rangle_{\mathbb{F}_\ell} \equiv \sum_{F\in\mathbb{F}_\ell} \int_F u(x)v(x)\,ds.$$

For the Laplacian, we choose the symmetric interior penalty method (see e. g. [2]) and write the associated bilinear form as

$$\begin{aligned} a_\ell(u,v) = \Big(\nabla u, \nabla v\Big)_{\mathbb{T}_\ell} &+ \Big\langle \kappa_F [\![u\mathbf{n}]\!], [\![v\mathbf{n}]\!]\Big\rangle_{\mathbb{F}^i_\ell} + \Big\langle \kappa_F u, v\Big\rangle_{\mathbb{F}^D_\ell} \\ &- \Big\langle \{\!\{\nabla u\}\!\}, [\![v\mathbf{n}]\!]\Big\rangle_{\mathbb{F}^i_\ell} - \Big\langle [\![u]\!], \{\!\{\nabla v\}\!\}\Big\rangle_{\mathbb{F}^i_\ell} - \Big\langle \partial_\mathbf{n} u, v\Big\rangle_{\mathbb{F}^D_\ell} - \Big\langle u, \partial_\mathbf{n} v\Big\rangle_{\mathbb{F}^D_\ell} \end{aligned} \tag{2}$$

Let $h_{T,F}$ be the length of the mesh cell T orthogonal to its face F and k_T the polynomial degree of the finite element on T. Then, $\kappa_F = 2k_T(k_T+1)/h_{T,F}$ for a boundary face F of the cell T and for a face between two mesh cells T_1 and T_2

$$\kappa_F = \frac{1}{2}\left(\frac{k_{T_1}(k_{T_1}+1)}{h_{T_1,F}} + \frac{k_{T_2}(k_{T_2}+1)}{h_{T_2,F}}\right).$$

Observe that this definition yields a stable bilinear form even on locally refined meshes or with hp-adaptivity (see for instance [12]).

The second form is the advection form with local Lax-Friedrichs flux studied in [13], namely

$$b_\ell(u,v) = \Bigl(-u, w\cdot\nabla v\Bigr)_{\mathbb{T}_\ell} + \Bigl\langle |w\cdot\mathbf{n}|u^\uparrow, v^\uparrow - v^\downarrow\Bigr\rangle_{\mathbb{F}_h}, \tag{3}$$

where $v^\uparrow$ and $v^\downarrow$ are the upwind and downwind value of v with respect to w on the face $F \in \mathbb{F}_h$; if any of the values in the face terms is taken from outside the domain, it is replaced by a zero value.

The discretization of (1) on level ℓ is then achieved by using the weighted sum of these bilinear forms, namely

$$\nu a_\ell(u,v) + b_\ell(u,v) = (f,v) + B_\ell(v), \tag{4}$$

where $B_\ell(v)$ is the linear form representing inhomogeneous boundary values:

$$B_\ell(v) = \nu\Bigl\langle \kappa g, v\Bigr\rangle_{\mathbb{F}_\ell^D} - \nu\Bigl\langle g, \partial_n v\Bigr\rangle_{\mathbb{F}_\ell^D} - \Bigl\langle [w\cdot\mathbf{n}]_- g, v\Bigr\rangle_{\mathbb{F}_\ell^D},$$

where $[w\cdot\mathbf{n}]_-$ denotes the negative part of $w\cdot\mathbf{n}$ (see e.g. [8]).

The aim of the preconditioner discussed here is solving (4) efficiently independent of the finest level L. To this end, we refer to it in matrix form

$$A_\ell u_\ell = b_\ell. \tag{5}$$

2.2 *The V-cycle*

We will consider the V-cycle multilevel method as the simplest multigrid cycle. The multigrid preconditioner P_ℓ^{-1} is defined recursively in the following way: on the coarsest level, we let $P_0^{-1} = A_0^{-1}$. For level ℓ, we assume that $P_{\ell-1}^{-1}$ is already defined and compute the action of P_ℓ^{-1} on a vector b_ℓ by:

(1) Since we do only a single smoothing step on V_ℓ, starting with $x_\ell^0 = 0$, the pre-smoothing reduces to:

$$x_\ell^1 = b_\ell$$

(2) Add a coarse grid correction by applying the V-cycle on level $\ell - 1$:

$$x_\ell^2 = x_\ell^1 + R_\ell^T P_{\ell-1}^{-1} R_\ell\Bigl(b_\ell - A_\ell x_\ell^1\Bigr).$$

(3) Perform a post smoothing step.

$$x_\ell^3 = x_\ell^2 + S_\ell\Bigl(b_\ell - A_\ell x_\ell^2\Bigr)$$

and let $P_\ell^{-1} b_\ell = x_\ell^3$.

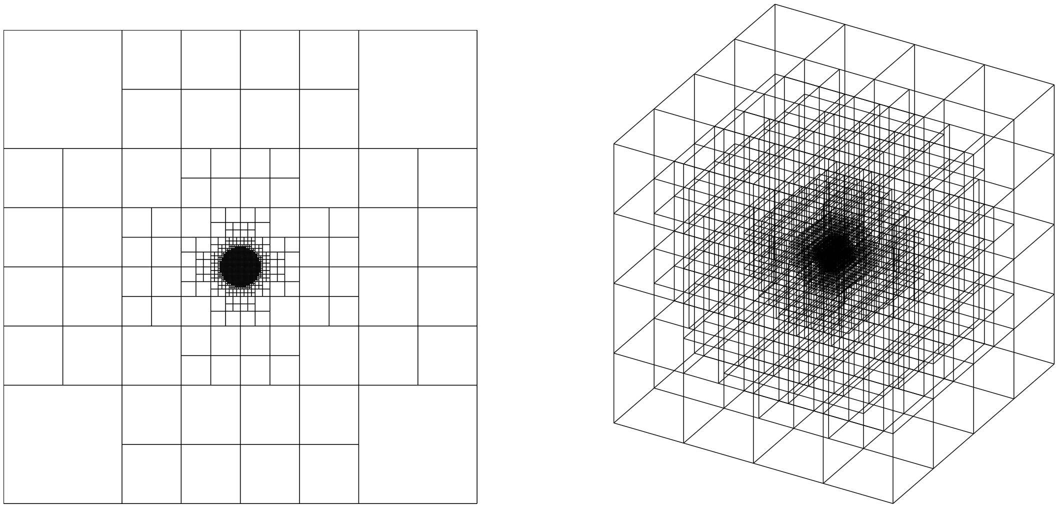

Fig. 1. Square and cube with local refinement into its center

In order to devise an efficient smoothing method S_ℓ, we note that a key feature of DG methods is the fact, that they constitute a well-posed boundary value problem on each mesh cell. Therefore, the cell matrices $A_{\ell,T}$, which are obtained by restricting the bilinear form in (4) to the functions with support on cell T and choosing a basis, is invertible. It is therefore possible to devise block Jacobi and block Gauss-Seidel smoothers using the inverses of these matrices. This will be the key to efficient smoothers for higher order discretizations.

2.3 Framework for numerical tests

In our tests, we solve equation (4) with $B_\ell(.) = 0$ and $f = 1$. The domain is the hypercube $[-1, 1]^d$. We compute either on meshes consisting of squares of size $h_\ell = 2^{1-\ell}$ globally or on meshes refined locally into a circle/ball at the center of the hypercube (see Figure 1). Therefore, the advection field will point into as well as out of the refined region. Reduction factors are always computed by running the Bi-CGSTAB [15] method as long as it takes to reduce the residual by a factor of 10^{10}.

We measure efficiency of the preconditioners by reduction of the Euclidean norm of the residual $\|r_n\|$ after n steps compared to the initial residual norm. Based on this we define average contraction rate $\bar{r}$ and the number n_{10} by

$$\bar{r} = \left(\frac{\|r_n\|}{\|r_0\|}\right)^{\frac{1}{n}} \qquad\qquad n_{10} = -10 \log_{10} \bar{r}.$$

We remark that n_{10} is the more intuitive measure, since it is immediately related to the amount of work required to solve a problem. Indeed, the smallest integer greater of less n_{10} is the number of steps required to reduce the residual by 10^{10},

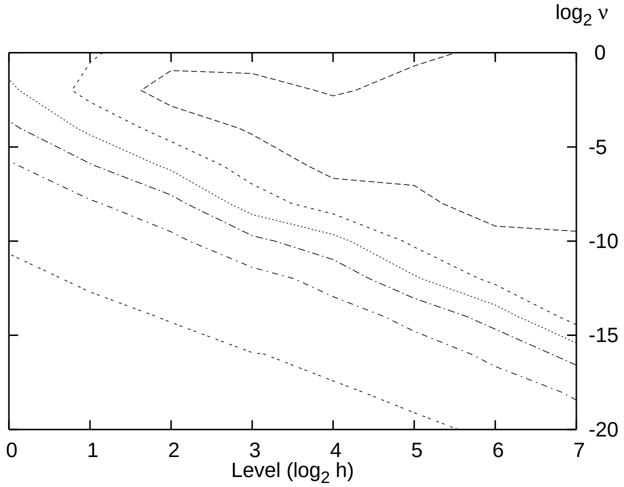

Fig. 2. Isolines for "iteration steps" n_{10} depending on refinement level and viscosity, $\mathbb{Q}_2$-elements. Lines correspond to values of 1 step (left) to 6 steps (right).

which is why we will refer to it as "iteration count" below. On the other hand, it is continuous and therefore gives a much finer measure of the quality of a method than true iteration counts, which at a size around 10 are very inaccurate.

3 Constant velocity

For a constant velocity and convex mesh cells, these cells can always be ordered such that a cell with higher index is downstream of a cell with lower index. With this ordering, the matrix associated with the bilinear form $b_\ell(.,.)$ becomes block-lower triangular (see e.g. [10]). Consequently, the block Gauss-Seidel method for this problem is a direct solver. Adding the bilinear form $\nu a_\ell(.,.)$ again, this property is lost. Nevertheless, the idea in [8] was that either the perturbation due to the elliptic form is small or the multigrid preconditioner will do the job.

The vector field used in our experiments is $w = (1.13, 2.13[, 3.13])$ such that the advection direction is neither aligned with the cell boundaries nor their diagonals.

In Figure 2 we show the "iteration counts" n_{10} for the Bi-CGSTAB method preconditioned by the V-cycle presented in the previous section, using this block Gauss-Seidel smoother. We display isolines on a plain of mesh refinement versus viscosity. The lower left and upper right lines correspond to a single step and to 6 steps, respectively. The results show that the method is indeed robust with respect to the viscosity ν in the following sense: while we recover the multigrid convergence rates of the elliptic problem for large ν, the scheme is a direct solver for the block triangular system with $\nu = 0$. The transition between the two is monotonous and occurs at a line corresponding to $h^2/\nu = const.$

	$\log_2 \nu$	0	-4	-8	-12	-16	-20
$\mathbb{Q}_2$	GS	5.5	5.9	5.7	4.8	2.9	1.5
	Jacobi	10.9	11.8	14.6	41.4	135	166

Table 1
Comparison of the block Gauss-Seidel and block Jacobi methods ("iteration steps" n_{10})

	$\mathbb{Q}_1$	$\mathbb{Q}_2$	$\mathbb{Q}_3$	$\mathbb{Q}_4$	$\mathbb{Q}_5$	$\mathbb{Q}_6$	$\mathbb{Q}_7$	$\mathbb{Q}_8$	$\mathbb{Q}_9$	$\mathbb{Q}_{10}$
$\nu = 1$										
block	5.3	5.5	6.0	6.1	7.3	7.5	8.5	9.0	10.8	10.8
point	8.1	6.9	7.9	11.3	20.4	32.6	72.3	182	387	—
3d-block	6.6	6.7	6.3	6.0	7.3	6.9	8.6	8.4		
$\nu = 1/16$										
block	6.3	5.6	6.8	6.5	7.6	8.1	9.5	9.8	11.4	11.1
point	15.3	32.2	21.2	14.6	19.2	35.8	70.6	173	—	—
$\nu = 1/256$										
block	5.4	6.0	6.2	6.9	8.1	8.8	9.6	10.2	10.7	11.0
point	no convergence on fine meshes									
3D, $\nu = 1$										
block	6.6	6.7	6.3	6.0	7.3	6.9	8.6	8.4		

Table 2
Comparison of block and point Gauss-Seidel smoothers for higher order elements ("iteration steps" n_{10})

In Table 1, we show the numbers of "iteration steps" n_{10} for the finest mesh and compare them to the results obtained with the block Jacobi method, which is clearly inferior and lacks robustness with respect to decreasing viscosity.

We present computations for representative diffusive and an advection dominated cases with higher order elements in Table 2. These numbers are complemented with those obtained by using a standard point Gauss-Seidel method instead of the blocked one. All results were obtained on fine meshes, where the iteration counts already remained constant (for the block method). We see that the use of the block Gauss-Seidel scheme is mandatory for higher order elements, independent of the diffusion parameter. Actually, for smaller diffusion, the point Gauss-Seidel method does not yield convergence anymore. Let us remark here that the choice of the block Gauss-Seidel method is mandated by the use of the local basis only, which in our case is the quite unsuited set of Lagrangian interpolating polynomials with equidistant grid points. By using a basis like suggested in [14], the cell matrices could be diagonal (or at least upper triangular if advection is added) and the point

	$\log_2 \nu$					
L	0	-4	-8	-12	-16	-20
2	4.6	4.6	2.8	2.0	2.0	2.0
3	5.0	4.9	3.1	2.0	2.0	2.0
4	5.0	5.2	4.1	2.3	2.0	2.0
5	5.0	5.2	4.1	2.5	2.0	2.0
6	5.0	5.2	4.3	3.1	2.5	2.2
7	5.0	5.2	4.4	3.5	2.8	2.7
8	5.0	5.2	4.4	3.6	3.1	3.1
three-dimensional						
6	5.3	5.1	3.8	3.3	3.2	3.2

Table 3
Block Gauss-Seidel smoother on locally refined meshes ($\mathbb{Q}_2$-elements)

smoother by definition would yield results similar to the block version.

In [11], we observed that the convergence rates for the multigrid method for the Laplacian do not deteriorate if local smoothing is used on locally refined meshes. This is different for the advection operator, since a Gauss-Seidel sweep applied only to the refined part of the mesh in Figure 1 is *not* a direct solver anymore. Therefore, convergence rates should not improve as dramatically as for the global refinement case. What the figures in Table 3 show though is, that the iteration counts for the advection-dominated case stay bounded by those for the diffusive case. As the last line of this table shows, this holds for the three-dimensional case as well.

4 Velocity fields with a vortex

If the velocity field contains a vortex, the advection operator exhibits closed characteristics and the situation becomes more complicated in two ways. First, the limit problem for $\nu = 0$ is not well-posed anymore. Therefore, we expect convergence of the solver only above a certain threshold for ν. Second, there is no downwind ordering of the mesh cells anymore. We study the behavior of the multilevel scheme with the advection field $w(x, y) = (-y, x)$.

Since the results for the block Jacobi smoother above suggest that the ordering is important, we do not expect optimal performance of a block Gauss-Seidel method with ordering according to a constant direction. In Table 4, the results for the "linear" smoother confirm this: already at a viscosity of 10^{-3}, convergence rates deteriorate dramatically and the scheme does not converge anymore on finer meshes for

		Level						
smoother	$\log_2 \nu$	2	3	4	5	6	7	8
linear	-4	5.4	5.6	5.8	5.7	5.8	5.9	6.0
	-6	7.2	6.8	6.9	7.0	7.1	7.2	7.2
	-8	12.0	11.4	11.1	11.0	11.1	11.6	11.7
	-10	23.7	26.3	25.8	25.0	24.5	26.6	28.3
	-12	45.2	60.1	75.1	73.7	81.4	84.3	87.6
	-14	67.4	—	—	—	—	—	—
circular	0	4.6	5.3	5.0	5.2	5.3	5.5	5.6
	-8	5.7	5.5	5.9	5.9	6.2	6.1	6.0
	-10	6.7	7.5	7.4	7.4	7.5	7.6	7.5
	-12	8.8	10.0	11.2	10.6	10.5	11.0	12.0
	-14	8.6	13.9	16.6	16.7	18.0	17.0	17.4
	-16	11.1	16.0	28.0	25.9	32.7	33.1	31.9
	-18	12.9	24.7	36.7	48.7	42.9	52.0	59.6
	-20	18.1	28.4	42.2	74.5	86.9	80.4	113
SSOR	0	3.9	4.1	4.6	4.8	4.6	4.6	4.6
	-8	6.4	6.6	6.7	6.4	6.4	6.4	6.4
	-10	9.3	10.8	10.6	10.3	10.1	10.0	9.8
	-12	13.4	19.7	21.8	22.3	22.6	21.6	20.9
	-14	21.1	32.8	50.9	59.9	60.4	59.1	58.3

Table 4
Smoothers for circular flow. “Iteration steps” n_{10} depending on mesh size and diffusion parameter ν.

approximately $\nu < 10^{-4}$. Therefore, a more sophisticated smoother is required. In Table 4, we explore the following options:

circular The mesh cells are sorted following the characteristics. Since these are closed, we apply a cut from the center of the vortex and start numbering from that cut (see Figure 3). In our situation, where we know the geometry of the vortex, this is a simple task. Automated numbering schemes for general two-dimensional vortices have been suggested in [9] and [5]. The results in Table 4 show that this smoother is actually much more robust than the previous one and we even have convergence for $\nu = 10^{-6}$. In particular, the method still converges very fast where the linear ordering already faltered.

Nevertheless, flow geometries in three dimensions may become quite compli-

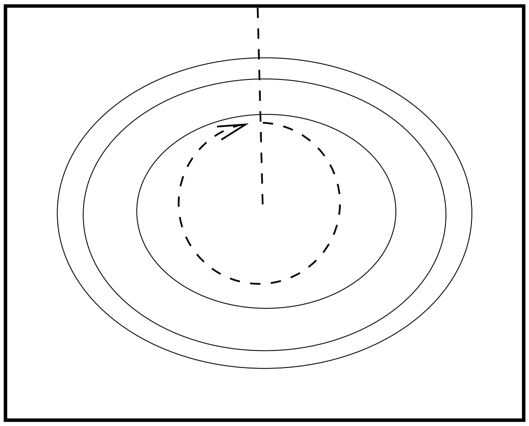

Fig. 3. Numbering scheme for a vortex

cated, such that this approach may not be suitable anymore.

SSOR Using the symmetric block Gauss-Seidel method, we cover one flow direction and its opposite. Note that there is no ordering according to the actual structure of the advection field involved, the ordering is the same as for constant advection. Still, the results in Table 4 suggest that this is a viable scheme, nearly as robust as the circular ordering. Down to diffusion parameters of 10^{-3}, it is as robust as the circular ordering.

It should be noted though that the effort for each smoothing step is about twice as high as for the circular smoothing. This is the price we have to pay for a much simpler setup of the method.

5 Conclusions

The results in this article show that block Gauss-Seidel smoothers with an ordering of the degrees of freedom reflecting the flow of the advection field yield efficient and robust multilevel preconditioners for advection-diffusion problems. Robustness is observed with respect to the diffusion parameter ν in the sense that the method converges with standard multigrid rates in the elliptic limit, in a single step in the advective limit and rates are monotonous in between. Furthermore, the method is robust with respect to the polynomial degree k in the sense that iteration rates deteriorate very slowly if the polynomial degree increases.

Our comparisons show that a block Jacobi smoother is not robust with respect to vanishing diffusion, while a point Gauss-Seidel method deteriorates with increasing polynomial degree. This leads us to the conclusion that the domain sweeps of the Gauss-Seidel method as well as the inversion of cell matrices are crucial for robustness. In particular, the Richardson smoother, which is sufficient —even if not efficient— for low order discretizations of the Laplacian, is not viable for these problems.

References

[1] P. F. Antonietti and B. Ayuso. Schwarz domain decomposition preconditioners for discontinuous Galerkin approximations of elliptic problems: Non-overlapping case. *Math. Model. Numer. Anal.*, to appear.

[2] D. N. Arnold, F. Brezzi, B. Cockburn, and D. Marini. Unified analysis of discontinuous Galerkin methods for elliptic problems. *SIAM J. Numer. Anal.*, 39(5):1749–1779, 2001.

[3] W. Bangerth, R. Hartmann, and G. Kanschat. deal.II — a general purpose object oriented finite element library. *ACM Trans. Math. Software*. to appear.

[4] W. Bangerth and G. Kanschat. Concepts for object-oriented finite element software – the `deal.II` library. Preprint 1999-43, SFB 359, Heidelberg, 1999.

[5] J. Bey and G. Wittum. Downwind numbering: Robust multigrid for convection-diffusion problems. *Appl. Numer. Math.*, 23(1):177–192, 1997.

[6] X. Feng and O. Karakashian. Two-level non-overlapping Schwarz methods for a discontinuous Galerkin method. *SIAM J. Numer. Anal.*, 39(4):1343–1365, 2001.

[7] J. Gopalakrishnan and G. Kanschat. Application of unified DG analysis to preconditioning DG methods. In K. J. Bathe, editor, *Computational Fluid and Solid Mechanics 2003*, pages 1943–1945. Elsevier, 2003.

[8] J. Gopalakrishnan and G. Kanschat. A multilevel discontinuous Galerkin method. *Numer. Math.*, 95(3):527–550, 2003.

[9] W. Hackbusch and T. Probst. Downwind Gauss-Seidel smoothing for convection dominated problems. *Numer. Linear Algebra Appl.*, 4(2):85–102, 1997.

[10] C. Johnson and J. Pitkäranta. An analysis of the discontinuous Galerkin method for a scalar hyperbolic equation. *Math. Comput.*, 46:1–26, 1986.

[11] G. Kanschat. Multi-level methods for discontinuous Galerkin FEM on locally refined meshes. *Comput. & Struct.*, 82(28):2437–2445, 2004.

[12] G. Kanschat. *Disctontinuous Galerkin Methods for Viscous Flow*. Teubner, Stuttgart, 2007. to appear.

[13] P. LeSaint and P.-A. Raviart. On a finite element method for solving the neutron transport equation. In C. de Boor, editor, *Mathematical aspects of finite elements in partial differential equations*, pages 89–123, New York, 1974. Academic Press.

[14] P. Šolin and T. Vejchodský. Higher-order finite elements based on generalized eigenfunctions of the Laplacian. *Math. Comp. sim.*, to appear.

[15] H. van der Vorst. Bi-CGSTAB: A fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems. *SIAM J. Sci. Stat. Comput.*, 13(2):631–644, 1992.