General notes on DPG

November 10, 2014

1 Least squares

We begin with a general variational formulation

$$b(u,v) = l(v)$$

DPG begins with the idea that you would like to do least squares on the operator equation

$$Bu = \ell, \quad Bu, \ell \in V'$$

where $\langle Bu, v\rangle_{V'\times V} = b(u,v)$ and $\langle \ell, v\rangle_{V'\times V} = l(v)$. Since $Bu - \ell \in V'$, we minimize the norm of this residual in V' over the finite dimensional space U_h, i.e.

$$\min_{u_h \in U_h} \left\| Bu_h - \ell \right\|_{V'}^2 .$$

This leads to the normal equations

$$\left(Bu - \ell, B\delta u\right)_{V'}, \quad \forall \delta u \in U_h.$$

The Riesz map gives us the equivalent definition

$$\left(R_V^{-1}\left(Bu - \ell\right), R_V^{-1}\left(B\delta u\right)\right)_V = 0, \quad \forall \delta u \in U_h.$$

Assuming we've specified the Riesz map through a test space inner product

$$\langle R_V v, \delta v\rangle_{V'\times V} = (v, \delta v)_V,$$

this leads to what I call a Dual Petrov-Galerkin method.

2 Algebraic perspective

In the above example, V is infinite dimensional. If we approximate V by V_h such that $\dim(V_h) > \dim(U_h)$, we get matrix representations of our operators

$$\begin{aligned} B_{ij} &= b(u_j, v_i), \quad u_j \in U_h, v_i \in V_h \\ R_V &= (v_i, v_j)_V \quad v_i, v_j \in V_h \\ \ell_i &= l(v_i), \quad v_i \in V_h. \end{aligned}$$

The resulting normal equations

$$\left(R_V^{-1}\left(Bu - \ell\right), R_V^{-1}\left(B\delta u\right)\right)_V, \quad \forall \delta u \in U_h.$$

can now be written as

$$\left(R_V^{-1}\left(Bu-\ell\right)\right)^T R_V \left(R_V^{-1}B\right) = 0,$$

or, after simplifying to $(Bu-\ell)^T R_V^{-1}B = 0$, we get the algebraic normal equations

$$B^T R_V^{-1} B u = B^T R_V^{-1} \ell.$$

This is just the solution to the algebraic least squares problem

$$\min_u \|Bu-\ell\|^2_{R_V^{-1}}.$$

Such problems can also be written using the augmented system for the least squares problem

$$\left[\begin{array}{cc} R_V & B \\ B^T & 0 \end{array}\right]\left[\begin{array}{c} e \\ u \end{array}\right] = \left[\begin{array}{c} l \\ 0 \end{array}\right].$$

This can be interpreted as the mixed form of the Dual Petrov-Galerkin method, which is used by Cohen, Welper, and Dahmen in their 2012 paper "Adaptivity and variational stabilization for convection-diffusion equations".

$$\begin{aligned}(e,v)_V + b(u,v) &= l(v)\\ b(\delta u, e) &= 0\end{aligned}$$

Eliminating e from the above system leads to the above algebraic normal equations.

Note: for general R_V, the algebraic normal equations are completely dense. Cohen, Welper, and Dahmen thus solve the augmented system to get solutions in this setting; however, this is a saddle point problem, and over 2x as large as the trial space, which makes preconditioning and solving more difficult.

3 Deriving the Discontinuous Petrov-Galerkin method

We want to avoid solving either a fully dense system or a saddle point problem, so we introduce Lagrange multipliers $\widehat{u}$ to enforce continuity weakly on e, which we will now approximate using discontinuous functions. These $\widehat{u}$ are defined on element edges only, similarly to hybrid variables or mortars in finite elements. This leads to the new system

$$\begin{aligned}\langle \widehat{u}, [\![v]\!]\rangle_{\Gamma_h} + (e,v)_V + b(u,v) &= l(v)\\ b(\delta u, e) &= 0\\ \langle \hat{\mu}, [\![e]\!]\rangle_{\Gamma_h} &= 0\end{aligned}$$

where Γ_h is the mesh skeleton (union of all element edges). The resulting algebraic system here is

$$\left[\begin{array}{ccc} R_V & B & \hat{B} \\ B^T & 0 & 0 \\ \hat{B}^T & 0 & 0 \end{array}\right]\left[\begin{array}{c} e \\ u \\ \widehat{u} \end{array}\right] = \left[\begin{array}{c} f \\ 0 \\ 0 \end{array}\right].$$

Because e is now discontinuous, R_V is made block-diagonal; eliminating e returns the (fairly) sparse symmetric positive-definite DPG system

$$A = \left[\begin{array}{cc} B^T R_V^{-1} B & B^T R_V^{-1}\hat{B} \\ \hat{B}^T R_V^{-1} B & \hat{B}^T R_V^{-1}\hat{B} \end{array}\right]\left[\begin{array}{c} u \\ \widehat{u} \end{array}\right] = \left[\begin{array}{c} B^T R_V^{-1} f \\ \hat{B}^T R_V^{-1} f \end{array}\right].$$

4 Fast evaluation of the Riesz inverse

Given that the test norm is typically defined

$$\|v\|_V \coloneqq \alpha \|v\|^2_{L^2(\Omega)} + \|v\|^2_{V_0},$$

where $\|v\|_{V_0}$ can be a seminorm. This leads the definition of the Riesz map R_V as

$$R_V = \alpha M + A.$$

Taking eigenvectors and eigenvalues of the generalized eigenvalue problem $AQ = MQ\Lambda$ and normalizing such that $Q^T MQ = I$, we can rewrite R_V as

$$R_V = Q^{-T}(\alpha I + \Lambda)Q^{-1}$$

which is then easy to invert. This technique is especially efficient when tensor-product techniques are applicable, reducing the generalized eigenvalue problem to one for a one-dimensional discretization.

4.1 Primal hybrid formulation: why it works for DPG

The primal hybrid formulation for a symmetric, positive-definite variational form is to enforce a weak continuity using mortar-like basis functions on element interfaces.

$$\sum_K (a_K(u,v) + \langle \lambda, v\rangle_{\partial K}) = (f,v)_{L^2(\Omega)}$$
$$\sum_K \langle \mu, u\rangle_{\partial K} = 0.$$

The weak continuity is enforced by noticing that $\sum_K \langle \mu, u\rangle_{\partial K} = \langle \mu, [\![u]\!]\rangle_{\Gamma_h}$, where Γ_h is the union of mesh interfaces.

The primal hybrid formulation gives Crouziex-Raviart elements for lowest order $N=1$, with flux order $N_f = N-1$. However, this does not work in defining elements for $N>1$; for N even and $N_f = N-1$, the Lagrange multiplier constraint set will not be linearly independent. Proof: on triangles, the jump constraints for N odd will imply that the solution must be an orthogonal polynomial of order N over the edge. Since N is odd, in 2D, the values at both vertices will be equal, and the trace on the edge will be continuous. Taking u to be any polynomial over T with this trace gives that $Bu = 0$, but $u \neq 0$. (I believe Jay Gopalakrishnan also noticed this independently.)

If you take N arbitrary and $N_f \leq N-2$, however, things work fine. This is roughly the situation with DPG - consider primal DPG, with N_{trial}, $N_{\text{test}} = N_{\text{trial}} + 2$, and $N_f = N_{\text{trial}} - 1$.

5 Block preconditioning

The DPG system can be reordered

$$A = \left[\begin{array}{cc} B^T R_V^{-1} B & B^T R_V^{-1}\hat{B} \\ \hat{B}^T R_V^{-1} B & \hat{B}^T R_V^{-1}\hat{B}\end{array}\right]\left[\begin{array}{c} u \\ \widehat{u}\end{array}\right] = \left[\begin{array}{cc} A_h & B_h \\ B_h^T & C_h\end{array}\right]\left[\begin{array}{c} u \\ \widehat{u}\end{array}\right] = \left[\begin{array}{c} f_h \\ g_h\end{array}\right]$$

where

$$\left[\begin{array}{c} f_h \\ g_h\end{array}\right] = \left[\begin{array}{c} B^T R_V^{-1} f \\ \hat{B}^T R_V^{-1} f\end{array}\right].$$

where A_h is the matrix corresponding to field (volume) variables, C_h is the submatrix corresponding to flux (surface/mortar) variables, and B_h couples the system together.

For a first pass, we consider the primal DPG method, which is probably the simplest DPG method available, where field variables are approximated with C_0-continuous piecewise polynomials and the flux variables are approximated using discontinuous mortar basis functions. Test functions are approximated with disjoint discontinuous polynomials of higher order than the trial field variables. (We may also wish to consider the DPG method with ultra-weak variational formulation (discontinuous field variables). In this case, A_h is also block diagonal.)

5.1 Block Jacobi

We can write the full DPG system in block form

$$\left[\begin{array}{cc} A_h & B_h \\ B_h^T & C_h \end{array}\right]\left[\begin{array}{c} u \\ \widehat{u} \end{array}\right] = \left[\begin{array}{c} f_h \\ g_h \end{array}\right].$$

We could naively break the system up into two blocks and use a fixed point iterative scheme

$$\begin{aligned} u^{n+1} &= A_h^{-1}(f_h - B_h\widehat{u}^n) \\ \widehat{u}^{n+1} &= C_h^{-1}(g_h - B_h^T u^{n+1}). \end{aligned}$$

or equivalently

$$\left[\begin{array}{c} u^{n+1} \\ \widehat{u}^{n+1} \end{array}\right] = \left[\begin{array}{cc} A_h^{-1} & \\ & C_h^{-1} \end{array}\right]\left(\left[\begin{array}{c} f_h \\ g_h \end{array}\right] - \left[\begin{array}{cc} & B_h \\ B_h^T & \end{array}\right]\left[\begin{array}{c} u^n \\ \widehat{u}^n \end{array}\right]\right)$$

which we can recognize as a block Jacobi iteration. This may be preferrable in a matrix-free environment, as both A_h and C_h can be computed in a matrix-free fashion ($C_h\widehat{u}$ can be computed using local inversions of the Riesz product with $\widehat{u}$ as boundary data). As far as I can tell, the Schur complement $S = C_h - B_h^T A_h^{-1} B_h$ for a general DPG system cannot, due to the non-explicit nature of $A_h = B_h^T R_V^{-1} B_h$.

This iteration was tested on pure convection under both the ultra-weak and primal formulations. As there was no observable difference in the primal formulation, we just show ultra-weak results here. Test norm was taken to be $\|v\|_V^2 = \alpha \|v\|_{L^2(\Omega)}^2 + \|\beta \cdot \nabla v\|_{L^2(\Omega)}^2$.

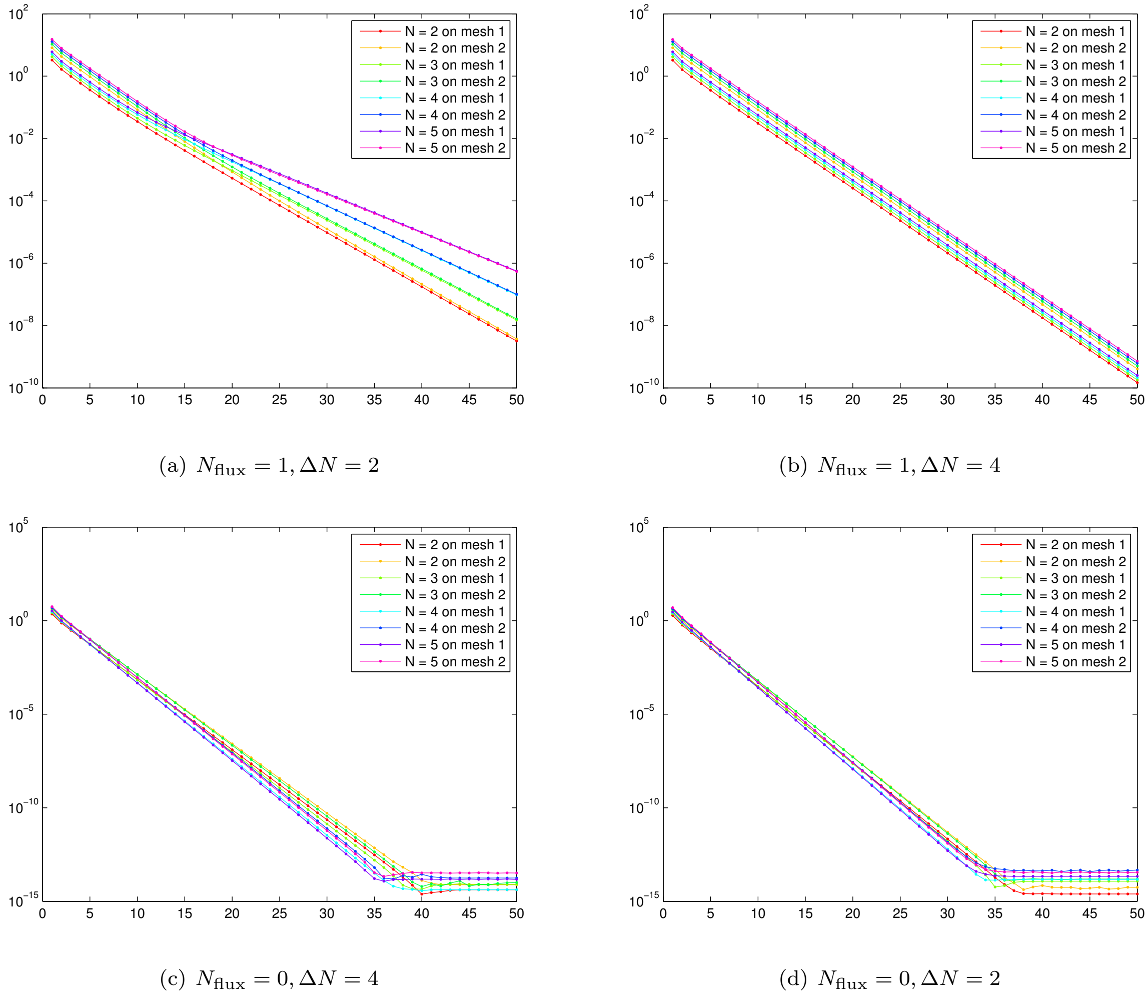
- Convergence depends on ΔN and N_{flux}.
- If $N_{\text{flux}} = 0$, convergence is roughly independent of ΔN. If $N_{\text{flux}} > 0$, the convergence is both slower than with $N_{\text{flux}} = 0$ and more sensitive to N and ΔN.

Convergence of the fixed point solution to the exact solution is shown here.

We note that these results are sensitive also to the regularization parameter α — smaller α results in faster convergence, but obviously worse conditioning of the local problem.

5.2 GMRES preconditioner

An issue with preconditioning using a fixed point iteration is that the preconditioner is no longer symmetric positive-definite, though we can still take advantage of the positive-definite nature of the blocks in the block Jacobi iteration. We therefore use the block Jacobi iteration as a GMRES preconditioner. We chart the dependence of the method on α and the number of fixed point iterations below. Counting the total number of solves in both inner and outer GMRES iterations, we can see that, while a larger number of inner iterations speeds up GMRES convergence, the most efficient method in terms of number of total solves is a single inner iteration. Thus, we abandon the idea of a fixed point iteration in favor of a single step preconditioner. Doing so will also allow us to construct a symmetric, positive-definite preconditioner for use in a conjugate gradient iteration.



(a) $N_{\text{flux}} = 1, \Delta N = 2$

(b) $N_{\text{flux}} = 1, \Delta N = 4$

(c) $N_{\text{flux}} = 0, \Delta N = 4$

(d) $N_{\text{flux}} = 0, \Delta N = 2$

Figure 1: Block Jacobi fixed point iterations for pure convection with $\beta = (1, 0)$, ultra-weak formulation.

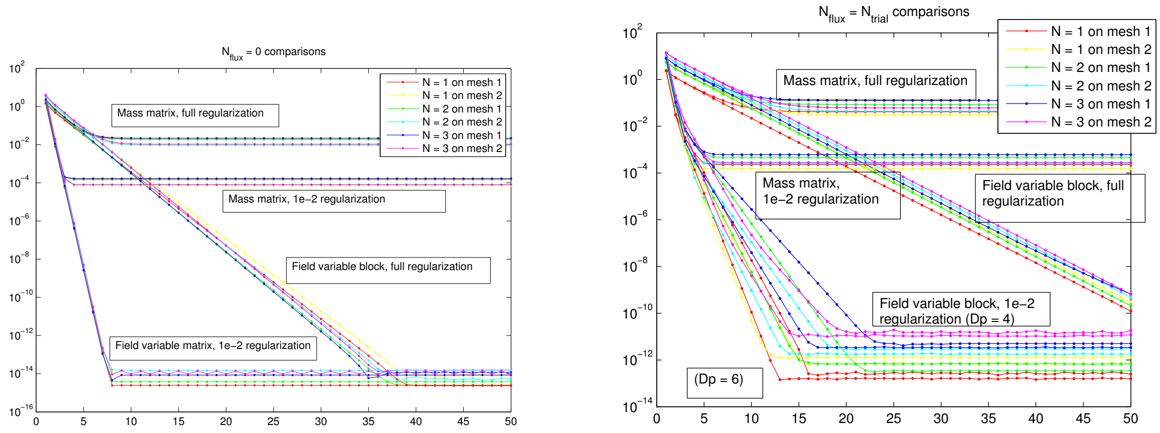
5.3 Approximate inverses

We assume that preconditioning the A_h submatrix can roughly be done using similar techniques to preconditioning standard elliptic equations. Initial numerical evidence appears to support this. Options include

- Two-level additive Schwarz. A coarse solve (P_1 elements with AGMG?) + (overlapping) block solves.
- Direct AGMG preconditioning.
- P_1 FEM preconditioning of nodal bases
- Geometric multigrid.

Preconditioning of the C_h submatrix is more difficult conceptually; however, similar ideas can be leveraged

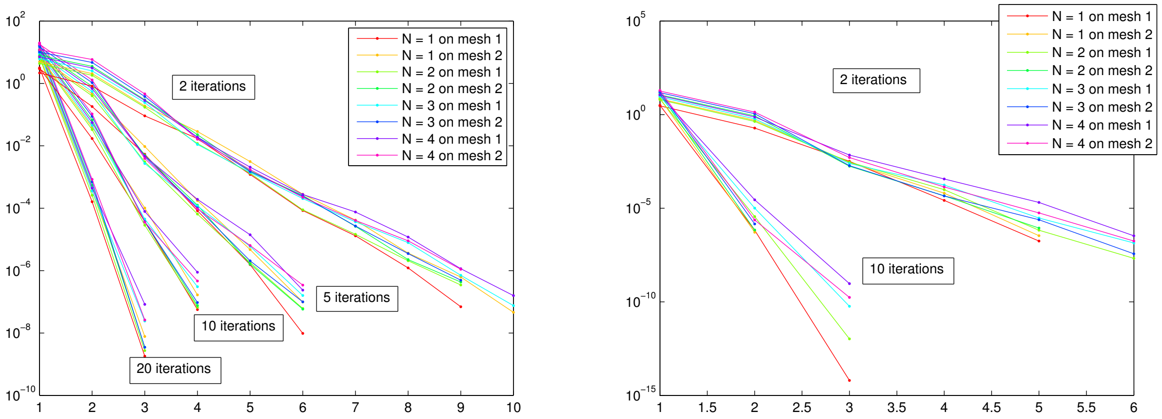
- Two-level additive Schwarz. A coarse solve (P_0 elements with AGMG?) + (overlapping) block solves.
- Geometric multigrid + AGMG preconditioning.
- Direct AGMG preconditioning.



(a) $N_{\text{flux}} = 0$

(b) $N_{\text{flux}} = N_{\text{trial}}$

Figure 2: Block Jacobi fixed point iterations for pure convection - dependence on α regularization parameter. "Mass matrix" means that the solution of $B^T R_V^{-1} B$ was replaced by the inversion of an $L^2(\Omega)$ mass matrix instead.



(a) $\alpha = 1$

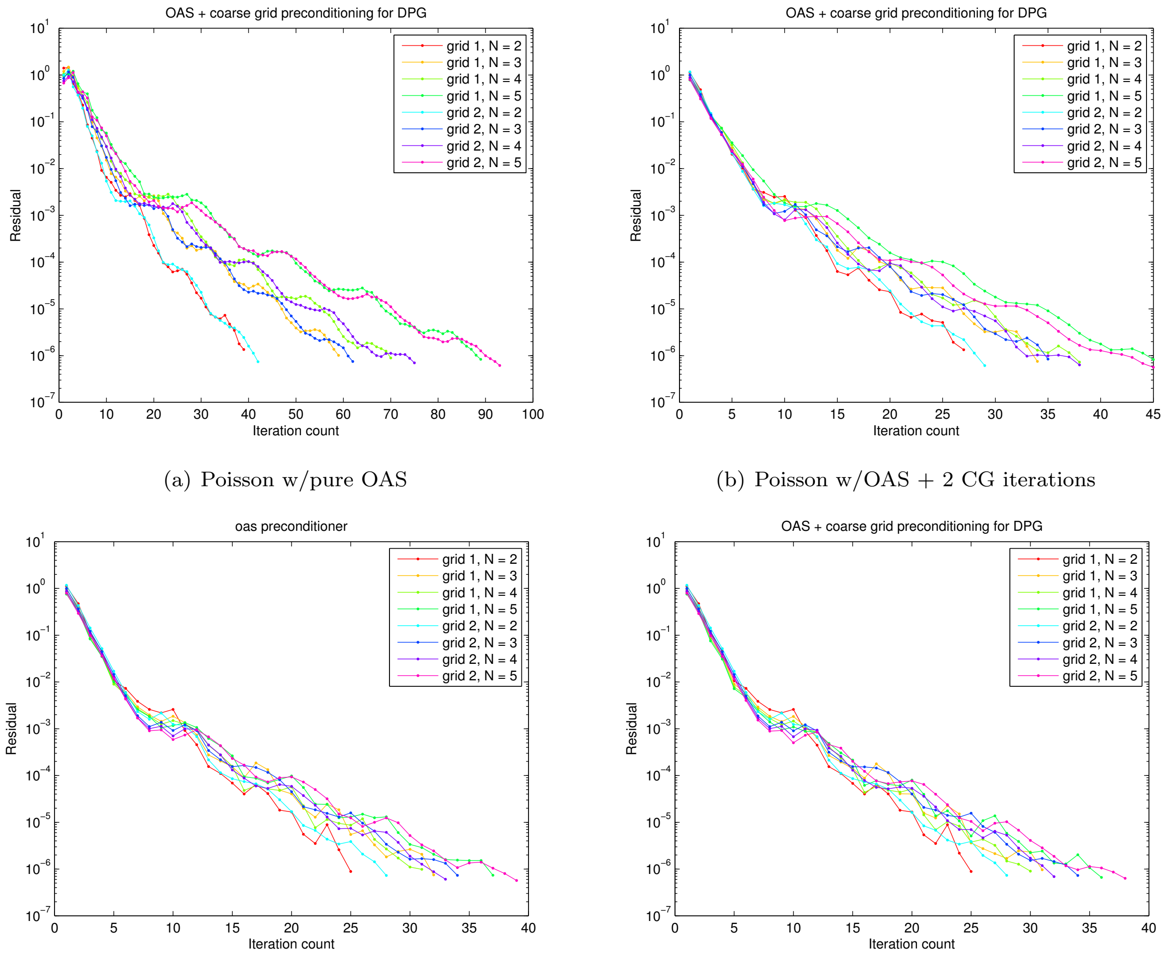
(b) $\alpha = .1$

Figure 3: Block Jacobi fixed point iterations for pure convection used as a preconditioner for GMRES.

P_1 FEM preconditioning is meant to sparsify the structure of the finite element matrix; however, since the C_h submatrix is constructed in a Schur complement manner, replacing a nodal polynomial for $\widehat{u}$ on an edge with piecewise-P_1 FEM at nodal points would not decrease sparsity. **Sparsity of the fluxes $\widehat{u}$ depends on the underlying test space: all fluxes bordering an element will be coupled together regardless of connectivity pattern so long as the test space over the element is coupled as well.**

5.4 OAS with coarse grid

- A flexible PCG method is needed when using an inner PCG iteration.
- The approximate preconditioner appears to be relatively h-independent,
- The N-dependency comes as a consequence of the triangular factorization involving A^{-1}, but can be



(a) Poisson w/pure OAS

(b) Poisson w/OAS + 2 CG iterations

(c) Poisson w/OAS + 5 CG iterations

(d) Poisson w/OAS + 10 CG iterations

controlled by a small number of PCG iterations in the inner iteration for A^{-1}.

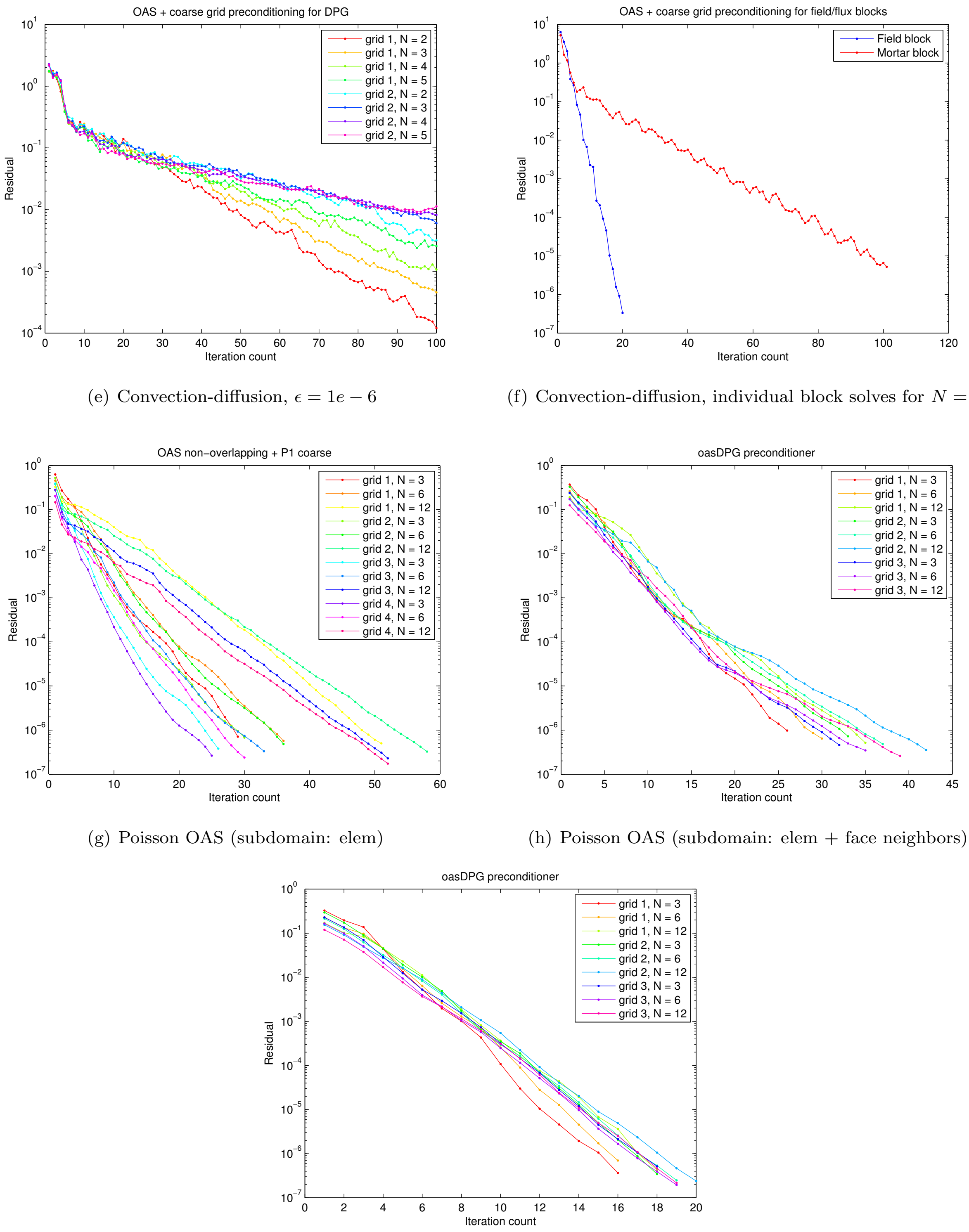
The block preconditioner is not robust for convection-diffusion, due to the fact that the C_h block corresponds to the mortar discretization of

$$u - \nabla \cdot ((\epsilon I + \beta\beta^T)\nabla(u))$$

for which mortar methods do poorly, since the coercivity constant of the above problem is ϵ, and there is only a streamline diffusion along the β direction. The field preconditioner converges as fast or faster than for Poisson.

6 OAS of Barker et. al

Only the patch OAS appears to deliver both h and N independent results.



(e) Convection-diffusion, $\epsilon = 1e - 6$

(f) Convection-diffusion, individual block solves for $N = 2$

(g) Poisson OAS (subdomain: elem)

(h) Poisson OAS (subdomain: elem + face neighbors)

(i) Poisson OAS (subdomain: vertex patches)

Figure 4: Behavior of OAS for different subdomains (element, neighbors, and one-ring patches). All cases include a P_1/P_0 coarse solver.

7 Tensor product approaches

Standard FEM/SEM on tensor product elements can take advantage of fast diagonalization using 1D operators. In other words, if M and K are mass and stiffness matrices on a quadrilateral grid, then

$$(M+K)^{-1} = (Q^T \otimes Q^T)(I \otimes I + \Lambda \oplus \Lambda)^{-1}(Q \otimes Q)$$

where Q and Λ are eigenvalues and eigenvectors resulting from the generalized eigenvalue problem $K_1 Q = M_1 Q \Lambda$ (K_1, M_1 are 1D operators), such that $Q^T M Q = I$. This inverts the operator exactly.

With DPG for Poisson, the stiffness matrix over one element is given by $B^T R_V^{-1} B$, where $B = K I_N^{N+\Delta N}$, and $I_N^{N+\Delta N}$ is an interpolation operator from order N polynomials to order $N + \Delta N$ polynomials. Under tensor product geometries, this reduces the system down to

$$B^T R_V^{-1} B = (I_N^{N+\Delta N} \otimes I_N^{N+\Delta N})^T (Q^{-1} \otimes Q^{-1})(\tilde{\Lambda})(Q^{-T} \otimes Q^{-T})(I_N^{N+\Delta N} \otimes I_N^{N+\Delta N})$$

where $\tilde{\Lambda} = (\Lambda \oplus \Lambda)(I \otimes I + \Lambda \oplus \Lambda)^{-1}(\Lambda \oplus \Lambda)$, or

$$\tilde{\Lambda}_{ij} = 1 + \lambda_i + \lambda_j.$$

Unfortunately, due to the presence of the $(I_N^{N+\Delta N} \otimes I_N^{N+\Delta N})$ matrices, it is difficult to invert this in closed form. Likewise, due to the ij-coupled nature of $\tilde{\Lambda}$, we can no longer utilize a tensor product structure for the resulting matrix $B^T R_V^{-1} B$.