General notes on DPG

August 13, 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.

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

This leads to the normal equations

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

The Riesz map gives us the equivalent definition

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

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

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

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

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

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

$$\begin{bmatrix} R_V & B \\ B^T & 0 \end{bmatrix}\begin{bmatrix} e \\ u \end{bmatrix} = \begin{bmatrix} l \\ 0 \end{bmatrix}.$$

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

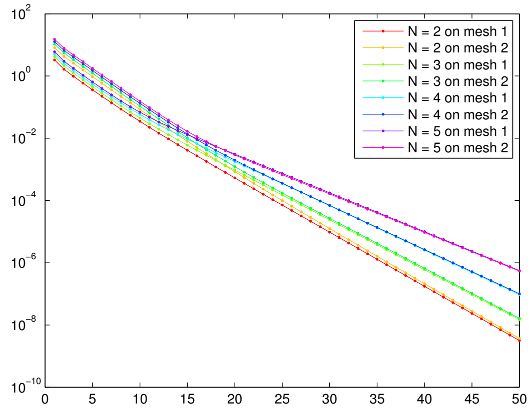
$$\begin{aligned}\langle \widehat{u}, [\![v]\!]\rangle_{\Gamma_h} + (e,v)_V + b(u,v) &= l(v)\\ b(\delta u, e) &= 0\\ \langle \hat{\mu}, [\![e]\!]\rangle_{\Gamma_h} &= 0\end{aligned}$$

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

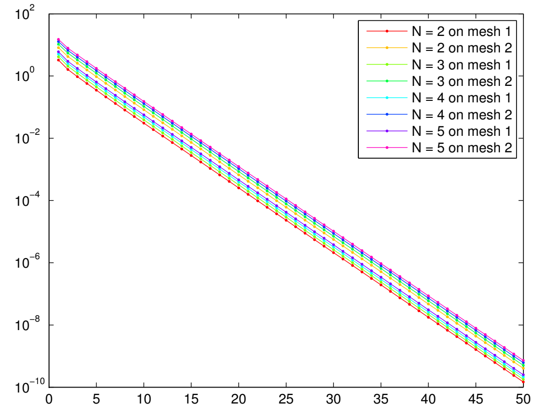
$$\begin{bmatrix} R_V & B & \hat{B} \\ B^T & 0 & 0 \\ \hat{B}^T & 0 & 0 \end{bmatrix}\begin{bmatrix} e \\ u \\ \widehat{u} \end{bmatrix} = \begin{bmatrix} f \\ 0 \\ 0 \end{bmatrix}.$$

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

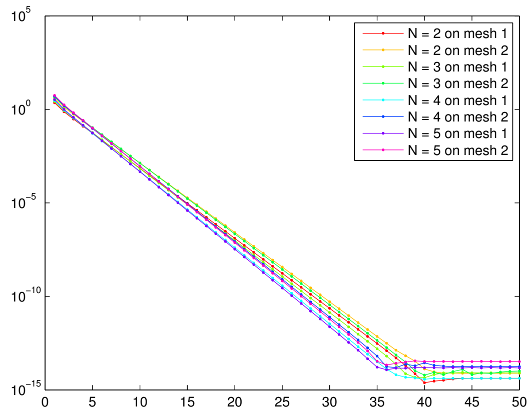
$$A = \begin{bmatrix} B^T R_V^{-1} B & B^T R_V^{-1}\hat{B} \\ \hat{B}^T R_V^{-1} B & \hat{B}^T R_V^{-1}\hat{B} \end{bmatrix}\begin{bmatrix} u \\ \widehat{u} \end{bmatrix} = \begin{bmatrix} B^T R_V^{-1} f \\ \hat{B}^T R_V^{-1} f \end{bmatrix}.$$



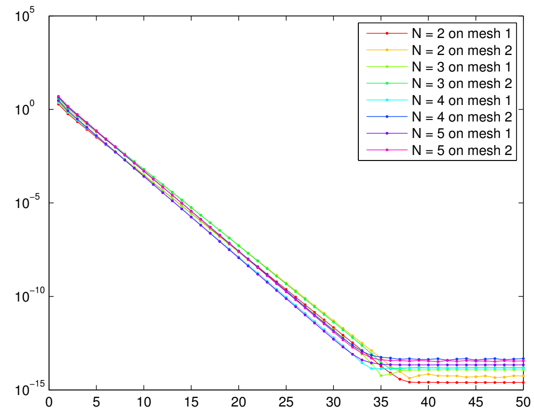
(a) $N_{\text{flux}} = 1, \Delta N = 2$



(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.

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

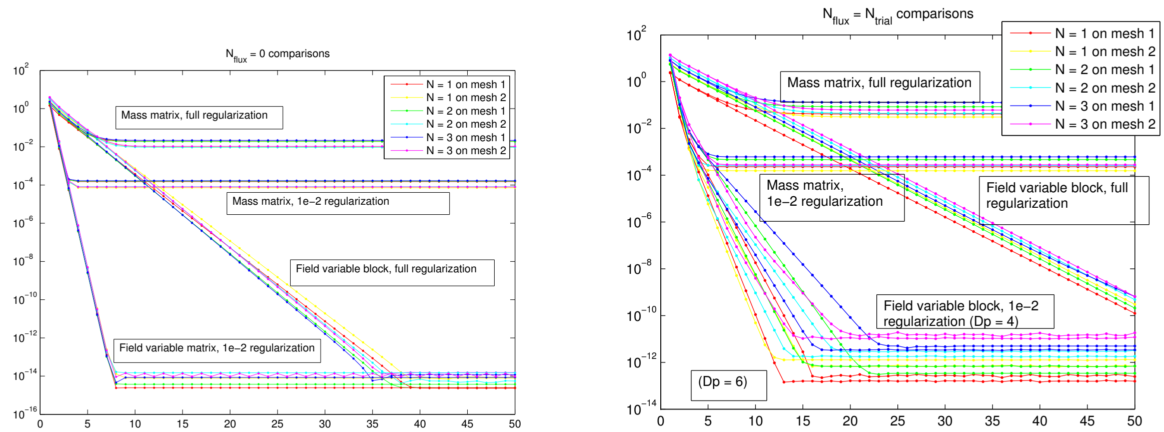
$$R_V = \alpha M + A.$$

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

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

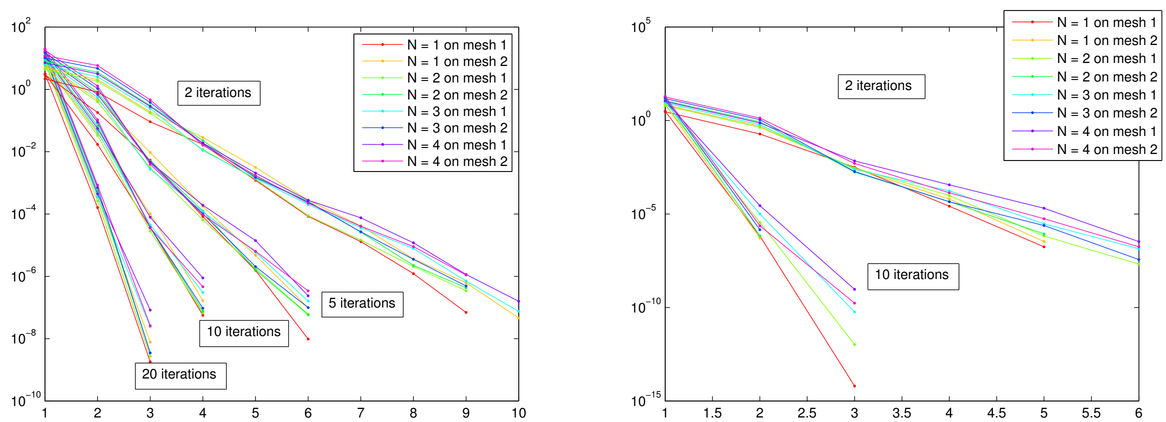
which is then easy to invert. This technique is especially efficient when tensor-product techniques are applicable, reducing the generalized eigenvalue problem to one for a one-dimensional discretization.

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.



(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.

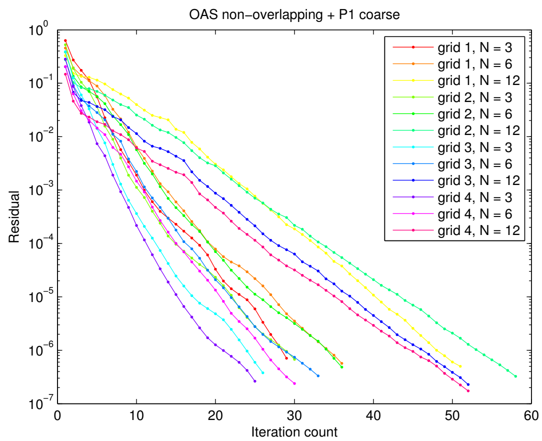


(a) $\alpha = 1$ (b) $\alpha = .1$

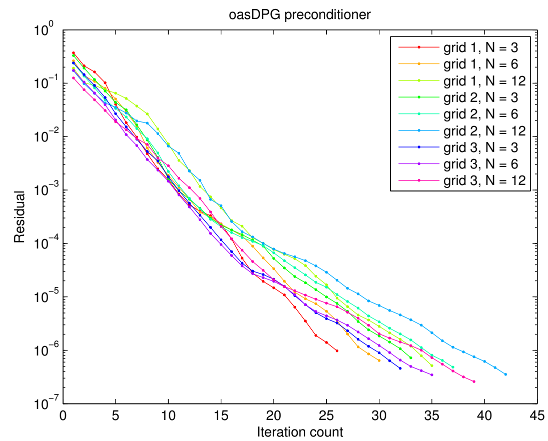
Figure 3: Block Jacobi fixed point iterations for pure convection used as a preconditioner for GMRES.

4.1 GMRES preconditioner

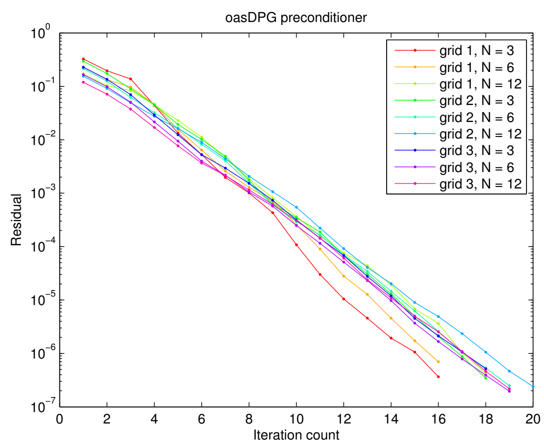
An issue with preconditioning using a fixed point iteration is that the preconditioner is no longer symmetric positive-definite, though we can still take advantage of the positive-definite nature of the blocks in the block Jacobi iteration. We therefore use the block Jacobi iteration as a GMRES preconditioner. 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) Poisson OAS (subdomain: elem)



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



(c) 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.

5 OAS of Barker et. al

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

6 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 $B^T R_V^{-1} B$.