Degree and wavenumber [in]dependence of Schwarz preconditioner for the DPG method

Jay Gopalakrishnan and Joachim Schöberl

Abstract This note describes an implementation of a discontinuous Petrov Galerkin (DPG) method for acoustic waves within the framework of high order finite elements provided by the software package NGSolve. A technique to impose the impedance boundary condition weakly is indicated. Numerical results from this implementation show that a multiplicative Schwarz algorithm, with no coarse solve, provides a p-preconditioner for solving the DPG system. The numerical observations suggest that the condition number of the preconditioned system is independent of the frequency k and the polynomial degree p.

1 A Petrov Galerkin formulation

We consider the Helmholtz equation modeling time harmonic acoustic waves in a homogenous medium,

$$-\Delta u - k^2 u = f \qquad \text{on } \Omega \tag{1a}$$

$$u = 0 \qquad \text{on } \partial\Omega, \tag{1b}$$

Here we have considered the simplest Dirichlet boundary condition (postponing the discussion of impedance boundary condition to later), and Ω is a polygonal (2D) or polyhedral (3D) domain, partitioned into a simplicial finite element mesh Ω_h. When k^2 is not an eigenvalue of $-\Delta$, this problem has a unique solution. We want to study its approximation by the so-called primal discontinuous Petrov Galerkin

Jay Gopalakrishnan
Portland State University, PO Box 751, Portland OR 97207, USA, e-mail: gjay@pdx.edu

Joachim Scöberl
Wiedner Hauptstraße 8-10, TU Wien, 1040 Wein, Austria, e-mail: joachim.schoeberl@tuwien.ac.at

(DPG) method [4] (also cf. [1] and [2]). This approximation is based on a Petrov Galerkin weak formulation.

The derivation of the formulation begins, as in other standard finite element formulations, by multiplying the equation by a smooth enough complex-valued test function v and integrating by parts. The difference in the DPG case is that v is allowed to be discontinuous across element interfaces. Hence the appearance of interelement fluxes is inevitable, i.e.,

$$\sum_{K\in\Omega_h}\left(\int_K \operatorname{grad} u\cdot\overline{\operatorname{grad} v}-\int_K k^2u\bar v-\int_{\partial K}(n\cdot\operatorname{grad} u)\bar v\right)=\sum_{K\in\Omega_h}\int_K f\bar v.$$

Here, n generically denotes the unit outward normal of any domain under consideration, f is assumed to be square integrable (although this can be relaxed), and as usual, the integral over ∂K must be interpreted as a duality pairing if u is not sufficiently regular. Letting $n\cdot\operatorname{grad} u$ be an independent unknown, denoted by $n\cdot q$, this leads to the following weak formulation: *Find $u\in U$ and $q\in Q$ such that*

$$(\operatorname{grad} u,\operatorname{grad} v)_{\Omega_h}-k^2(u,v)_{\Omega_h}-\langle n\cdot q,v\rangle_{\partial\Omega_h}=(f,v)_\Omega,\quad \forall v\in Y, \tag{2}$$

where $(r,s)_{\Omega_h}=\sum_{K\in\Omega_h}(r,s)_K$ and $(\cdot,\cdot)_D$, for any domain D, denotes the complex $L^2(D)$-inner product, $\langle\ell,w\rangle_{\partial\Omega_h}=\sum_{K\in\Omega_h}\langle\ell,w\rangle_{1/2,\partial K}$ where $\langle\ell,\cdot\rangle_{1/2,\partial K}$ denotes the action of a functional ℓ in $H^{-1/2}(\partial K)$,

$$U=H^1_0(\Omega),\qquad Y=\prod_{K\in\Omega_h}H^1(K),\qquad Q=H(\operatorname{div},\Omega)\Big/\prod_{K\in\Omega_h}H_0(\operatorname{div},K).$$

Formulation (2) is clearly of the Petrov-Galerkin kind as the trial space $X=U\times Q$ is different from the test space Y. Adapting the techniques in [3] and [4], it is possible to prove that this weak formulation has a unique solution whenever k^2 is not a cavity resonance. However, the focus of this note is on practical implementation.

The method we shall implement is not based on the above Petrov-Galerkin form, but rather on an equivalent mixed Bubnov-Galerkin form. To describe it, first let us set the sesquilinear form $b(\cdot,\cdot)$ by

$$b((u,q),v)=(\operatorname{grad} u,\operatorname{grad} v)_{\Omega_h}-k^2(u,v)_{\Omega_h}-\langle n\cdot q,v\rangle_{\partial\Omega_h}$$

and the Y-inner product by

$$(y,v)_Y=(\operatorname{grad} y,\operatorname{grad} v)_{\Omega_h}+k^2(y,v)_{\Omega_h}.$$

The equivalent mixed formulation is to find $(\varepsilon,u,q)\in Y\times X$ such that

$$(\varepsilon,y)_Y+b((u,q),y)=(f,y)_{\Omega_h} \tag{3a}$$

$$b((w,r),\varepsilon)=0, \tag{3b}$$

for all $(y,w,r)\in Y\times X$. One can show (see e.g., [5]) that the solution (u,q) of (2) together with $\varepsilon=0$ is the unique solution of (3).

2 A DPG method for the Helmholtz equation

The DPG method we want to study is a Galerkin method obtained directly from (3), i.e., the DPG approximation $(\varepsilon_h, u_h, q_h)$ is in a discrete subspace $Y_h \times U_h \times Q_h$ of $Y \times U \times Q$ and satisfies

$$(\varepsilon_h, y)_Y + b((u_h, q_h), y) = (f, y)_{\Omega_h} \tag{4a}$$

$$b((w, r), \varepsilon_h) = 0 \tag{4b}$$

for all $(y, w, r) \in Y_h \times U_h \times Q_h$. (A different DPG method for the Helmholtz equation based on an ultra-weak formulation can be found in [3].)

The discrete spaces are defined, for any degree $p \geq 0$, by

$$\begin{aligned} Y_h &= \{v \in Y : \ v|_K \in P_{p+2}(K), \forall K \in \Omega_h\}, \\ U_h &= \{w \in U : \ w|_K \in P_{p+1}(K), \forall K \in \Omega_h\}, \\ Q_h &= \{r \in Q : \ q|_K \in R_p^\partial(K), \forall K \in \Omega_h\}, \end{aligned}$$

where $P_p(K)$ denotes the space of polynomials of degree at most p on K and $R_p^\partial(K)$ is defined as follows. Recall that the Raviart-Thomas space in N space dimensions ($N = 2, 3$), namely $P_p(K)^N + xP_p(K)$ where $x \in \mathbb{R}^N$ is the coordinate vector, can be split into a subspace $R_p^0(K) = R_p(K) \cap H_0(\mathrm{div}, K)$ and a linearly independent remainder $R_p^\partial(K)$. The decomposition $R_p(K) = R_P^0(K) \oplus R_p^\partial(K)$ depends on the choice of the basis for $R_p(K)$, but since the sesquilinear form $b(\cdot, \cdot)$ uses only the trace $n \cdot q$ of function q in Q, its value is independent of the choice of the basis representation. The trace space of $R_p(K)$ and $R_p^\partial(K)$ coincide. Indeed, we may even use a space other than the Raviart-Thomas space, as long as its traces coincide with that of the Raviart-Thomas space of index p (i.e., polynomials of degree at most p on each subsimplex of K).

3 The matrix form of the method

Let $\{v_j\}, \{w_l\}, \{r_m\}$ denote some bases for Y_h, U_h, and Q_h, respectively. Then, defining the matrices A, B, C by

$$\begin{aligned} A_{ij} &= (v_j, v_i)_Y = \sum_K \left(\int_K \mathrm{grad}\, v_j \cdot \mathrm{grad}\, \overline{v}_i + k^2 \int_K v_j \overline{v}_i \right) \\ B_{lj} &= \overline{b((w_l, 0), v_j)} = \sum_K \left(\int_K \mathrm{grad}\, v_j \cdot \mathrm{grad}\, \overline{w}_l - k^2 \int_K v_j \overline{w}_l \right) \\ C_{mj} &= \overline{b((0, r_m), v_j)} = -\sum_K \left(\int_{\partial K} v_j \, n \cdot \overline{r}_m \right), \end{aligned}$$

we can write the matrix form of the DPG method as

$$\begin{bmatrix} A & B^* & C^* \\ B & 0 & 0 \\ C & 0 & 0 \end{bmatrix} \begin{bmatrix} x_\varepsilon \\ x_u \\ x_q \end{bmatrix} = \begin{bmatrix} F \\ 0 \\ 0 \end{bmatrix}, \tag{5}$$

where * denote conjugate transpose. Clearly, the system is Hermitian. It is possible to prove that this discrete system inherits invertibility from the well-posedness of the exact problem whenever Y_h is of sufficiently high degree, but in practice we choose Y_h to be of degree $p+2$ as already stated.

Since functions in Y_h have no continuity constraints across element interfaces, the matrix A is block diagonal (in addition to being Hermitian and positive definite) with one block per element, and is thus easy to invert. Therefore, the preferred matrix system for inversion is not (5), but rather its positive definite Schur complement computed as follows. With $L^* = [B^* \;\; C^*]$ and $x_{uq^*} = [x_u^* \;\; x_q^*]$, rewriting (5) as

$$\begin{bmatrix} A & L^* \\ L & 0 \end{bmatrix} \begin{bmatrix} x_\varepsilon \\ x_{uq} \end{bmatrix} = \begin{bmatrix} F \\ 0 \end{bmatrix}, \tag{6}$$

and eliminating x_ε, we obtain

$$(LA^{-1}L^*)x_{uq} = LA^{-1}F. \tag{7}$$

This is a Hermitian and positive definite system whenever (5) is invertible. Hence we are able to use the *preconditioned conjugate gradient method* as an iterative solver even though the original Helmholtz problem is indefinite. The remaining component x_ε can be recovered by $x_\varepsilon = A^{-1}(F - L^* x_{uq})$.

4 Implementation in NGSolve

We use several facilities provided by the package NGSolve [7, 8] to implement the above DPG method. First, the spaces Y_h and U_h are standard finite element spaces provided by the classes `L2HighOrderFESpace` and `H1HighOrderFESpace`, respectively. The space Q_h can be implemented by removing all interior degrees of freedom from the NGSolve class `HDivHighOrderFESpace`. A built-in facility for this removal is provided via the option `-orderinner` which allows one to restrict the degree of interior shape functions (those with zero normal traces on the element boundary). One then makes a compound space using these components. All of this can be done in the standard `pde`-input file format of NGSolve, as shown:

```
# Finite element spaces        (p = 2 case)
fespace fs1 -type=l2ho   -order=4 -complex                # Yh
fespace fs2 -type=h1ho   -order=3 -complex                # Uh
fespace fs3 -type=hdivho -order=2 -complex -orderinner=1  # Qh
fespace fs  -type=compound -spaces=[fs1,fs2,fs3] -complex  # Yh x Uh x Qh
```

Next, we must define all the sesquilinear forms in (4). The first form $(\cdot,\cdot)_Y$ in (4a) can be input in the `pde`-file using the built-in "integrator" classes `laplace` and `mass` provided in NGSolve. The $b(\cdot,\cdot)$ form however is nonstandard and is not available in NGSolve. We therefore exploit NGSolve's extensibility via shared library additions by writing new integrator classes. They use the dynamic polymorphism in NGSolve, inheriting properties from the abstract NGSolve class `BilinearFormIntegrator`. The new integrator classes are used to build a shared library of forms often needed in DPG methods. With the integrators for the $b(\cdot,\cdot)$ form made (subsumed under `[custom_integrators]` below) we can now define the sesquilinear form:

```
bilinearform dpg -fespace=fs -linearform=lf -nonsym -eliminate_internal
[custom_integrators]                    # b( (u,q), v)
laplace             (1.0)  --comp=1     # (grad e, grad v)
mass                (k*k)  --comp=1     # k*k* (e,v)
```

Of particular interest to us is the option `-eliminate_internal` above. Each degree of freedom in an NGSolve finite element space is marked if it is "inner" or not. An inner degree of freedom on one element does not interact with another inner degree of freedom on another element. By virtue of this stored information, the code can automatically perform static condensation of all inner degrees of freedom. In particular, all degrees of freedom of `L2HighOrderFESpace` within an element are marked to be inner. This means that the elimination of x_ε that allowed us to go from (5) to (7) is automatically performed by the code once the flag `-eliminate_internal` is given. To be precise, in addition to condensing (5) to (7), the code does a further condensation that eliminates all inner degrees of freedom of U_h.

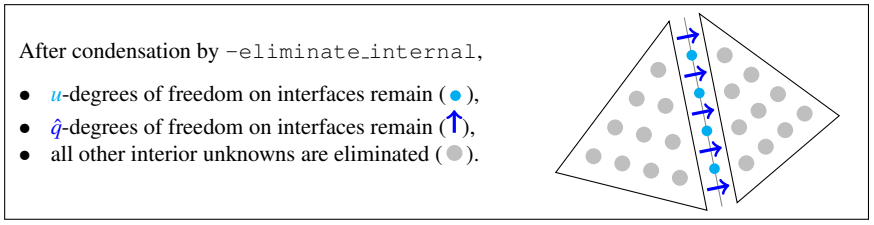

Fig. 1 Schematic of degrees of freedom left after condensation

Thus the *condensed system* consists only of degrees of freedom of Q_h (which by definition are associated only to element interfaces) and those degrees of freedom of U_h at the element interfaces (see Figure 1). This final system, being another Schur complement of the Hermitian positive definite Schur complement (7), is Hermitian and positive definite. We solve it by conjugate gradients, preconditioned by the Schwarz procedure discussed later.

This and other input files in their entirety as well as the code for the DPG shared library is publicly available at [https://github.com/jayggg/DPG].

5 The impedance boundary condition

Previously, we built the Dirichlet boundary condition (1b) into the weak formulation by essentially imposing it in U. Now suppose we are given, instead of (1b), the impedance condition

$$\frac{\partial u}{\partial n} - \hat{\imath} k u = 0, \qquad \text{on } \partial\Omega,$$

where $\hat{\imath}$ denotes the imaginary unit. Then instead of setting U to $H_0^1(\Omega)$, we now set $U = H^1(\Omega)$. Using the flux approximation given explicitly in the DPG formulations, the impedance boundary condition can be rewritten as

$$n \cdot q - \hat{\imath} k u = 0, \qquad \text{on } \partial\Omega. \tag{8}$$

Being a constraint tying two of the component spaces, a natural implementation would be by a Lagrange multiplier technique. However, this can result in loss of positive definiteness.

We pursue a different approach that imposes condition (8) weakly. The idea is to use the test function components w and r, i.e., we would like to impose the additional conditions $\int_{\partial\Omega}(n \cdot q_h - \hat{\imath} k u_h)\overline{n \cdot r} = 0$ and $\int_{\partial\Omega}(n \cdot q_h - \hat{\imath} k u_h)\overline{w} = 0$ without over-constraining the system. Since ε_h is an approximation to zero, we are motivated to build an approximate version of these conditions into the system by adding the term

$$\pm \int_{\partial\Omega} (n \cdot q_h - \hat{\imath} k u_h)\, \overline{(n \cdot r - \hat{\imath} k w)} \tag{9}$$

to the left hand side of (4b). This then perturbs the original system (6) to

$$\begin{bmatrix} A & L^* \\ L & D \end{bmatrix} \begin{bmatrix} x_\varepsilon \\ x_{uq} \end{bmatrix} = \begin{bmatrix} F \\ 0 \end{bmatrix}. \tag{10}$$

This system can also be condensed to get an analogue of (7):

$$(LA^{-1}L^* - D)x_{uq} = LA^{-1}F. \tag{11}$$

Now the choice of the sign in (9) becomes important: If we want (11) to be positive definite, we must choose the negative sign in (9) so that D is negative semidefinite.

6 The condensed Schwarz preconditioner

We now study a preconditioner for (11) constructed using a block Gauss-Seidel operator with overlapping blocks. The block Gauss-Seidel algorithm is standard, so we omit all details, except the specification of the blocks for our application. The blocks consists of all degrees of freedom after condensation, associated to a vertex patch. In 2D, one such block consists of all degrees of freedom of U_h and Q_h associated to the edges which meet at a single vertex (see Figure 2). The block corresponding

- Set up diagonal blocks from the *condensed* DPG matrix, with vertex patches blocks, as indicated.
- One preconditioner action:
  - Perform one block Gauss-Seidel relaxation sweep with these blocks.
  - Perform a reverse block Gauss-Seidel sweep, using the same blocks, but in the reverse order, for symmetrization.
  - The result is the action of the linear operator used as a preconditioner in conjugate gradients.

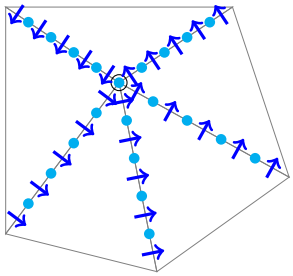

Fig. 2 Gauss-Seidel blocks

to a vertex in the 3D case consists of all degrees of freedom on all the mesh edges and the mesh faces containing that vertex. There are as many blocks as there are mesh vertices. The block Gauss-Seidel iteration multiplicatively updates an iterate by block inverses of certain residuals. These block inverses exist because they are principal submatrices of the positive definite matrix in (11). The action of our preconditioner consists simply of a block Gauss-Seidel relaxation algorithm followed by its adjoint given by the same relaxation done in the reverse block ordering.

7 Numerical results

We now report a result that is typical of our numerical experience with this method. We simulated a plane wave propagating in the x-direction on a uniform 4×4 triangular mesh of the unit square by providing the needed non-homogenous data to the impedance boundary condition. After assembling the condensed system (11), we used conjugate gradients, preconditioned by the above-mentioned block Gauss-Seidel algorithm, as an iterative solver. We stopped the iterations when successive iterates differed by less than 10^{-10}. The number of iterations are reported in Table 1. Each column of the table reports iteration counts obtained using a fixed wavenumber $k = 2\pi \times n_\lambda$ where n_λ (indicated atop the table) is the number of wavelengths that fit into the unit square.

The grayed out entries give iteration counts as well as indicate that computed solution did not resolve the wave. As is typical of all finite element type methods for wave problems, when meshes are too coarse, waves are not resolved. However, unlike many other methods, the DPG system remains solvable, no matter how coarse the mesh is. Moreover, the preconditioned conjugate gradient algorithm seems to converge at a degree-independent rate even on such coarse meshes. The bold entries also give the iteration numbers, but additionally indicate that in these cases the converged solution clearly showed the wave features. For example, in the $k = 2\pi \times 4$ case, it appears that we need at least $p = 8$ to resolve the wave. Note that we are able to go to polynomial degrees as high as 32 due to the good conditioning properties of the integrated Legendre shape functions implemented in NGSolve.

Schwarz preconditioner				
degree	number of waves			
p	2	4	8	16
1	16	14	12	11
2	22	13	12	10
4	**28**	27	12	12
8	**28**	**30**	32	11
16	**29**	**30**	**30**	32
32	**29**	**30**	**30**	**30**

Diagonal preconditioner				
degree	number of waves			
p	2	4	8	16
1	63	59	54	51
2	180	178	166	121
4	**261**	468	416	398
8	**328**	**612**	***	***
16	**662**	**894**	***	***
32	***	***	***	***

Table 1 Preconditioned conjugate gradient iteration counts

For comparison, we provide results from a simple diagonal preconditioning in a separate table. Clearly, the results from the block preconditioner are better. Entries marked "***" indicate that stopping criterion was not met even at 1000 iterations.

Our main conclusion from these observations is that the preconditioner seems to be uniform in p and k. (Similar observations were reported in [6] using an analogous preconditioner within GMRES for a different method. That method yields an indefinite system, while the current DPG method yields positive definite systems, so we may reliably use conjugate gradients on the latter.) Other (unreported) experiments in other wave directions in 2D, as well as in 3D tetrahedral meshes, all appear to confirm the uniformity of the preconditioner on k and p. Finally, we note that the preconditioner is not uniform in mesh size h. One usually needs to use a "coarse" solution to get h-uniformity. But for wave propagation, a good coarse problem is still a subject of debate.

Acknowledgements The authors wish to thank graduate student Lukas Kogler for his assistance in developing an initial version of the DPG code. This work was partially supported by the NSF under grant DMS-1318916 and by the AFOSR under grant FA9550-12-1-0484.

References

1. D. Broersen and R. Stevenson, *A Petrov-Galerkin discretization with optimal test space of a mild-weak formulation of convection-diffusion equations in mixed form*, IMA J. Numer. Anal., doi:10.1093/imanum/dru003, to appear in print (2014).
2. L. Demkowicz and J. Gopalakrishnan, *A class of discontinuous Petrov-Galerkin methods. Part II: Optimal test functions*, Numerical Methods for Partial Differential Equations, 27 (2011), pp. 70–105.
3. L. Demkowicz, J. Gopalakrishnan, I. Muga, and J. Zitelli, *Wavenumber explicit analysis for a DPG method for the multidimensional Helmholtz equation*, Computer Methods in Applied Mechanics and Engineering, 213/216 (2012), pp. 126–138.
4. L. Demkowicz and J. Gopalakrishnan, *A primal DPG method without a first-order reformulation*, Computers and Mathematics with Applications, 66 (2013), pp. 1058–1064.
5. J. Gopalakrishnan, *Five lectures on DPG methods*, arXiV: 1306.0557, (2013).
6. P. Monk, J. Schöberl, and A. Sinwel, *Hybridizing Raviart-Thomas elements for the Helmholtz equation*, Electromagnetics, 30 (2010), pp. 149–176.
7. J. Schöberl, *NETGEN – An advancing front 2D/3D-mesh generator based on abstract rules*, Comput. Visual. Sci, 1 (1997), pp. 41–52.
8. J. Schöberl, *NGSolve*, [http://sourceforge.net/projects/ngsolve/].