ICES REPORT 16-20

September 2016

An Adaptive DPG Method for High Frequency Time-harmonicWave Propagation Problems

by

Socratis Petrides, and Leszek F. Demkowicz

The Institute for Computational Engineering and Sciences
The University of Texas at Austin
Austin, Texas 78712

Reference: Socratis Petrides, and Leszek F. Demkowicz, "An Adaptive DPG Method for High Frequency Time-harmonicWave Propagation Problems," ICES REPORT 16-20, The Institute for Computational Engineering and Sciences, The University of Texas at Austin, September 2016.

An Adaptive DPG Method for High Frequency Time-harmonic Wave Propagation Problems

Socratis Petrides, Leszek F. Demkowicz

Institute for Computational Engineering and Sciences
The University of Texas at Austin, Austin, TX 78712, USA

Abstract

The Discontinuous Petrov-Galerkin (DPG) method for high frequency wave propagation problems is discussed. The DPG method offers uniform pre-asymptotic stability for any wavenumber. This allows for a fully automatic adaptive hp algorithm, that can start from very coarse meshes. Moreover DPG always delivers a Hermitian positive definite system, suggesting the use of the Conjugate Gradient algorithm for its solution. We present a new iterative solution scheme which benefits from these attractive properties of DPG. This novel solver is integrated within the adaptive procedure by constructing a two-grid-like preconditioner for the Conjugate Gradient method, that exploits information from previous meshes. The construction of such a preconditioner is discussed and an example for the 2D acoustics problem is presented. Our results show that the proposed iterative algorithm converges at a rate independent of the mesh and the wavenumber.

Keywords: discontinuous test functions, finite elements, adaptivity, Petrov-Galerkin method, multigrid, preconditioner, iterative solvers

1 Introduction

1.1 Motivation

Numerical simulation of high frequency wave propagation is one of the most studied fields in computational science. Applications in geomechanics, electromagnetic scattering, sonic tools and optics, require accurate but, at the same time, efficient discretization methods. In this paper, we are mostly interested in waves which are characterized by local behavior, such as non-diffracting pulses and beams [20], where the numerical simulation would benefit a lot from an adaptive method. These kind of waves are very essential in many applications such as modeling of lasers and LIDARs (Light Detection And Ranging), communications in free space, electromagnetic tweezers, etc.

1.2 Standard Finite Element method (FEM)

The standard FEM is a widely used numerical method for the solution of boundary value problems (BVPs). However, when it comes to the simulation of harmonic waves, especially for high frequency simulations in heterogeneous media and complex geometries, one has to deal with many challenges. First of all, standard FEM suffers from the so called "pollution" effect [1]. This is the phenomenon where, as the frequency

increases, the computed wave increasingly differs from the best approximation. Secondly, for highly oscillatory waves, a large number of grid points have to be used in order to resolve short wavelengths in a relatively large domain, and this results in a computationally intractable problem. Moreover, the resulting linear system is indefinite and that makes its solution more challenging. Lastly, due to the lack of pre-asymptotic stability, usual hp-adaptive algorithms have very poor performance until the mesh is sufficiently fine. These are the main challenges which DPG tries to overcome.

1.3 Properties of the DPG method

As in standard Galerkin methods, DPG starts from a BVP. The usual procedure is to multiply by a test function and integrate over the domain of interest. By integrating by parts, we then pass derivatives from the trial to the test functions and built the boundary conditions into the formulation (relaxation). In the case of the acoustics problem one has a first order system of equations; the conservation of mass and the conservation of linear momentum (see Eq. 3.12). Depending on which norm one seeks to measure convergence, the equations can be relaxed in different ways: a) the *trivial* or *strong* formulation where no equation is relaxed, b) the *classical* formulation where the momentum equation is relaxed , c) the *mixed* formulation where the conservation of mass equation is relaxed and finally d) the *ultraweak* formulation where both equations are relaxed. It can be shown that the four formulations are simultaneously well or ill posed [17]. The standard Galerkin method can only deal with a symmetric functional setting (where the trial and test spaces are the same), and this makes it suitable only for the classical and the mixed formulation. Moreover, standard FEM is only asymptotically stable, i.e., the mesh has to be fine enough to overcome the pollution effect. This brings us to the concept of the *optimal test functions*. Discretization of a well posed continuous problem doesn't always guarantee discrete stability. The discrete problem is well posed if and only if Babuška's *discrete inf-sup* condition holds [13]. An arbitrary choice of the discrete test space often leads to lack of stability. On the contrary, in the case of the DPG method, one computes the test functions which realize the supremum of the *inf-sup* condition on the fly. We then consider the Petrov-Galerkin problem with the optimal test space where discrete stability is guaranteed even in the pre-asymptotic region. DPG can also be viewed as a minimum residual method, where one minimizes the residual in the dual norm to the test space norm. Consequently, the resulting stiffness matrix is always Hermitian and positive definite, making the use of Conjugate Gradient (CG) ideal. Lastly, DPG can be interpreted as a mixed method where one solves simultaneously for the solution and the Riesz representation of the residual. The existence of a built-in error indicator and the aforementioned stability properties suggest the use of automatic hp-adaptivity, starting from very coarse meshes.

1.4 Organization of the paper

In the following section, we give a more detailed view on the DPG method. We then give an example of a high frequency wave propagation problem (scattering by a resonating cavity) and describe the proposed solver. We conclude our paper with some results on the solver and a discussion of future steps.

2 The DPG method

As discussed in the introduction, DPG can be interpreted in three different ways: a) a Petrov-Galerkin method with optimal test functions, b) a minimum residual method and c) a mixed method. We now introduce the concept of the ideal and the practical DPG method in these three different interpretations.

2.1 Petrov-Galerkin method with optimal test functions

Given a continuous bilinear (sesquilinear) form $b(\cdot,\cdot)$ defined on the product $U \times V$ of Hilbert spaces U (trial space) and V (test space), and a continuous linear (antilinear) form $l(\cdot)$ defined on V, we consider a variational problem

$$\begin{cases} u \in U \\ b(u,v) = l(v),\ \forall v \in V. \end{cases} \tag{2.1}$$

Given a finite-dimensional approximate trial space $U_h \subset U$, we then seek an approximate solution $u_h \in U_h$ to the problem:

$$\begin{cases} u_h \in U_h \subset U \\ b(u_h,v_h) = l(v_h),\ \forall v_h \in T(U_h) \end{cases} \tag{2.2}$$

where $T : U_h \to V$ is the *ideal trial-to-test operator* defined by

$$(T\delta u_h, \delta v)_V = b(\delta u_h, \delta v) \quad \delta u_h \in U_h,\ \delta v \in V, \quad \text{i.e.,}\ T = R_V^{-1} B$$

with $R_V : V \to V^{'}$ denoting the Riesz map defined by

$$(R_V y)(v) = (y,v)_V,\ \forall y, v \in V$$

and $B : U \to V^{'}$ being the operator generated by the bilinear form $b(\cdot,\cdot)$, i.e., for $u \in U$

$$(Bu)(v) = b(u,v)\ \forall\ v \in V.$$

While the above choice of the optimal test space guarantees discrete stability, it involves the inversion of Riesz operator R_V, i.e., solution of an additional infinite dimensional BVP (with multiple right-hand sides). This leads us to the *practical DPG* method where we approximate the *trial-to-test* operator by an approximate operator $T^r : U_h \to V^r \subset V$ defined by

$$(T^r \delta u_h, \delta v)_V = b(\delta u_h, \delta v), \quad \delta u_h \in U_h,\ \delta v \in V^r$$

We now seek the solution to the following problem :

$$\begin{cases} u_h \in U_h \subset U \\ b(u_h,v_h) = l(v_h),\ \forall v_h \in V^r = T^r(U_h) \end{cases} \tag{2.3}$$

where V^r is a finite dimensional subspace of V with $\dim(V^r) >> \dim(U_h)$, the so-called *enriched* test space. Typically, $r = p + \Delta p$ where p is the order of approximation of the trial space.
By introducing an appropriate Fortin operator J. Gopalakrishnan and W. Qiu [11] showed that the approximation maintains discrete stability. In [22], S. Nagaraj, S. Petrides and L. Demkowicz attempted to quantify the loss of stability by computing norms of relevant Fortin operators, defined on H^1 and $H(\mathrm{div})$ energy spaces in two space dimensions.

2.2 Minimum residual method

We can recast our problem as a minimum residual problem by reformulating the variational problem in the operator form,

$$\left\{\begin{array}{l} u \in U, \\ b(u,v) = l(v),\ v \in V, \end{array}\right. \quad \Leftrightarrow \quad \left\{\begin{array}{l} u \in U, \\ Bu = l\,, \end{array}\right. \tag{2.4}$$

where $B : U \to V^{'}$, $\langle Bu, v\rangle = b(u,v)$
In the ideal case, the minimum residual formulation reads

$$u_h = \arg\min_{w_h \in U_h} \|l - Bw_h\|_{V'} = \arg\min_{w_h \in U_h} \sup_{v \in V} \frac{|l(v) - b(w_h,v)|}{\|v\|_V}. \tag{2.5}$$

In the practical, case the minimum residual formulation becomes

$$u_h = \arg\min_{w_h \in U_h} \|l - Bw_h\|_{(V^r)'} = \arg\min_{w_h \in U_h} \sup_{v \in V^r} \frac{|l(v) - b(w_h,v)|}{\|v\|_V}. \tag{2.6}$$

2.3 Mixed method

By taking the Gâteaux derivative of (2.5), the minimization problem naturally leads to the ideal mixed problem:

$$\left\{\begin{array}{lll} u_h \in U_h,\ \psi \in V, & & \\ (\psi, v)_V + b(u_h, v) & = l(v),\ v \in V, & \\ b(w_h, \psi) & = 0, \qquad & w_h \in U_h \end{array}\right. \tag{2.7}$$

where $\psi = R_V^{-1}(l - Bu_h)$, is the error representation function , $\|\psi\|_V = \|(l - Bu_h)\|_{V'}$ is the norm of the residual, and $R_V : V \to V'$ is the Riesz map.
Again, for the practical DPG the mixed method becomes:

$$\left\{\begin{array}{lll} u_h \in U_h,\ \psi^r \in V^r, & & \\ (\psi^r, v)_V + b(u_h, v) & = l(v),\ v \in V^r, & \\ b(w_h, \psi^r) & = 0, \qquad & w_h \in U_h \end{array}\right. \tag{2.8}$$

where now $R_{V^r} : V^r \to (V^r)^{'}$, $\psi^r = R_{V^r}^{-1}(l - Bu_h)$ and $\|\psi^r\|_V = \|(l - Bu_h)\|_{(V^r)'}$ is the approximate residual. This built-in error indicator along with guaranteed stability, allows us to perform reliable adaptive refinements even in the pre-asymptotic region. We will demonstrate this attractive feature of DPG using an example of a 2D acoustics problem in a cavity in the next section. Before moving on, we emphasize that for both ideal and practical DPG methods, the three different formulations discussed above are equivalent [16].

2.4 Broken test spaces

In order to keep the additional computational cost of approximating the optimal test functions [16], to the minimum, we use discontinuous (broken) test spaces . This is where the word '*Discontinuous*' comes in to the picture for the DPG method, i.e., while the trial space remains conforming, the test space has no global

continuity conditions. However, this comes with an extra cost; the introduction of additional interface unknowns (Lagrange multipliers) that live on the whole mesh skeleton. Fortunately, stability is not lost here. Assuming the original variational formulation is well posed, the resulting variational formulation with broken test spaces remains well posed with a mesh-independent stability constant of the same order as the *inf-sup* constant of the original problem [5]. The use of broken test spaces is very critical for the efficiency of DPG method, since now the computation of the approximate optimal test functions can be performed on the element level.

2.5 Coding DPG

We conclude this section by a short tutorial on how the DPG method is coded [16]. Starting with the mixed method (2.7) we derive the corresponding 'broken formulation':

$$\begin{cases} u_h \in U_h,\ \hat{u}_h \in \hat{U}_h,\ \psi^r \in V^r, & \\ (\psi^r, v)_V + b(u_h, v) + \langle \hat{u}_h, v \rangle & = l(v),\ v \in V^r, \\ b(w_h, \psi^r) & = 0, \quad w_h \in U_h, \\ b(\hat{w}_h, \psi^r) & = 0, \quad \hat{w}_h \in \hat{U}_h, \end{cases} \tag{2.9}$$

where $\hat{u}_h$ corresponds to the new interface unknowns that live on the mesh skeleton.
This leads to the following matrix equation:

$$\begin{pmatrix} G & B_1 & B_2 \\ B_1^* & 0 & 0 \\ B_2^* & 0 & 0 \end{pmatrix} \begin{pmatrix} \psi^r \\ u_h \\ \hat{u}_h \end{pmatrix} = \begin{pmatrix} l \\ 0 \\ 0 \end{pmatrix}, \tag{2.10}$$

where now u_h, $\hat{u}_h$ and ψ^r are vectors of degrees of freedom for the original unknown u_h, Lagrange multiplier $\hat{u}_h$ and error representation function ψ^r. G is the square Gram matrix resulted from the test inner products of the shape functions for the enriched test space. The rectangular matrices B_1 and B_2 are obtained from the discretization of the bilinear form $b(u_h, v)$ and the pairing $\langle \hat{u}_h, v \rangle$ respectively. By solving the first equation for ψ^r and substituting to the next two equations we obtain the final DPG system:

$$\begin{pmatrix} B_1^* G^{-1} B_1 & B_1^* G^{-1} B_2 \\ B_2^* G^{-1} B_1 & B_2^* G^{-1} B_2 \end{pmatrix} \begin{pmatrix} u_h \\ \hat{u}_h \end{pmatrix} = \begin{pmatrix} B_1^* G^{-1} l \\ B_2^* G^{-1} l \end{pmatrix}. \tag{2.11}$$

The important message here is that the use of broken spaces allows us to perform the static condensation of the error representation function ψ^r, *element-wise*. We solve now for u_h and $\hat{u}_h$ using the standard FEM technology. Finally we perform local back substitution to compute the element contributions to the error representation function ψ^r, and use their norm as a local error indicator to drive adaptivity. Note that the DPG technology can be implemented within any standard FEM code that supports the discretization of all energy spaces forming the exact sequence, i.e, H^1, $H(\text{curl})$, $H(\text{div})$, L^2, with minimal modifications (mostly for the element matrices computations). We give a more detailed discussion on how the interface unknowns ($\hat{u}_h$) are discretized with an example in the next section.

3 Model problem - linear acoustics

3.1 Problem formulation

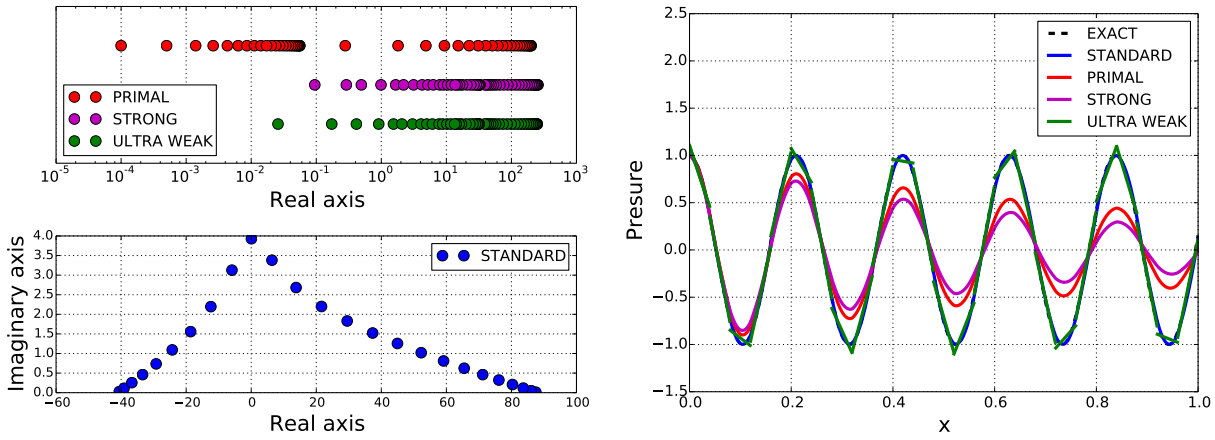
We consider the linear acoustics problem in a bounded domain Ω.

$$\begin{cases} i\omega u + \nabla p & = 0, \quad \text{in } \Omega \\ i\omega p + \operatorname{div} u & = 0, \quad \text{in } \Omega \\ p - u \cdot n & = g, \quad \text{on } \partial\Omega \end{cases} \tag{3.12}$$

where u is the velocity, p is the pressure and ω is the angular frequency.

As discussed in the introduction, one has different choices of relaxation. We refer the reader to the appendix for a detailed derivation of all different formulations of this problem. In Fig. 1 we show results of one dimensional simulations for all possible formulations for $\omega = 30$ and 25 quadratic elements. Figure (1a) shows the spectrum of the global stiffness matrix after static condensation of interior degrees of freedom. As expected, for all DPG formulations the spectrum lies on the positive real axis since the stiffness matrix is Hermitian. On the contrary, the standard Galerkin method (see A.26) delivers an indefinite stiffness matrix. Our study suggests that except for the case of the *primal* formulation (see A.30), where the eigenvalues form two separate clusters, the use of the Conjugate Gradient iterative solver would be preferable. In Fig. (1b) we demonstrate the different approximability properties for all formulations by showing all the corresponding numerical solutions for the pressure. As we can see, the primal and the strong (see A.24) formulations are diffusive. On the other hand, the standard and the ultraweak (see 3.13) formulations are more accurate. Note that both strong (least squares) and ultraweak DPG methods are robust, i.e., the stability constants are independent of ω, but they converge in different norms: least squares in the graph norm, and ultraweak in the L^2 norm. We will discuss the ultraweak formulation in more detail in sections (3.2, 3.3).

Figure 1: 1D linear acoustics



(a) Spectrum

(b) Numerical Pressure

3.2 The DPG ultraweak formulation

We focus now on the DPG ultraweak formulation for the two-dimensional case of problem (3.12), where $\Omega = (0,1)^2$. As usual, we multiply the equations by test functions v and q respectively, integrate over the domain Ω, and integrate by parts both equations. As mentioned earlier, due to the use of broken (discontinuous) test spaces, we need to introduce additional interface unknowns (Lagrange multipliers) between elements. These new unknowns, denoted by $\hat{p}$ and $\widehat{u \cdot n}$, are defined on the whole mesh skeleton $\Gamma_h = \bigcup_{K \in \mathcal{T}_h} \partial K$.

Finally, the DPG ultraweak formulation for problem (3.12) is

$$\begin{cases} u \in (L^2(\Omega))^d,\ p \in L^2(\Omega) \\ \widehat{u \cdot n} \in H^{-1/2}(\Gamma_h),\ \hat{p} \in H^{1/2}(\Gamma_h) \\ \hat{p} - \widehat{u \cdot n} = g,\ \text{on } \partial\Omega \\ \\ (i\omega u, v) - (p, \text{div}_h v) + \langle \hat{p}, v \cdot n \rangle \qquad = 0,\ v \in H(\text{div}, \Omega_h) \\ (i\omega p, q) - (u, \nabla_h q) + \langle \widehat{u \cdot n}, q \rangle \qquad = 0,\ q \in H^1(\Omega_h) \end{cases} \tag{3.13}$$

where $H^{-1/2}(\Gamma_h) := tr\, H(\text{div}, \Omega)$ on Γ_h, and $H^{1/2}(\Gamma_h) := tr\, H^1(\Omega)$ on Γ_h. Note that ∇_h and div_h denote element-wise operations. The duality pairings are defined by:

$$\langle \widehat{u \cdot n}, q \rangle = \sum_{K \in \mathcal{T}_h} \langle \widetilde{u}|_K \cdot n_K, q_K \rangle_{\partial K},\ \ q = \{q_K\}_{K \in \mathcal{T}_h} \in H^1(\Omega_h)$$

$$\langle \hat{p}, v \cdot n \rangle = \sum_{K \in \mathcal{T}_h} \langle \widetilde{p}_K, v_K \cdot n_K \rangle_{\partial K},\ \ v = \{v_K\}_{K \in \mathcal{T}_h} \in H(\text{div}, \Omega_h)$$

where $\widetilde{u} \in H(\text{div}, \Omega)$ is such that

$$\widehat{u \cdot n} = \widetilde{u}|_K \cdot n_K \text{ on } \partial K$$

with n_K denoting the outward unit normal vector on ∂K.
Similarly $\widetilde{p} \in H^1(\Omega)$ is such that

$$\hat{p} = \widetilde{p}|_K \text{ on } \partial K$$

3.3 Additional remarks on the ultraweak formulation

In the following paragraph, we switch to an abstract notation where $\mathfrak{u} = (u, p)$ denotes the group unknown, $\mathfrak{v} = (v, q)$ the group test function and $\hat{\mathfrak{u}} = (\widehat{u \cdot n}, \hat{p})$ the group trace.

3.3.1 Adjoint test norm

Let $A_\omega \mathfrak{u} = (i\omega u + \nabla p, i\omega p + \text{div} u)$. Then, in the ideal DPG case if we choose the *adjoint norm* for the test space, i.e., $\|\mathfrak{v}\|_V = \|A_\omega^* \mathfrak{v}\|$, then the method delivers L^2 projection. Indeed, it is well known that the

Petrov-Galerkin method with optimal test functions delivers an orthogonal projection in the so-called energy norm

$$\|\mathfrak{u}\|_E = \sup_{\mathfrak{v}\in V} \frac{|b(\mathfrak{u},\mathfrak{v})|}{\|\mathfrak{v}\|_V}. \tag{3.14}$$

Moreover, for the ultraweak formulation and the adjoint test norm, the energy norm coincides with the original (L^2) norm in V. Indeed,

$$\|\mathfrak{u}\|_E = \sup_{\mathfrak{v}\in V} \frac{|b(\mathfrak{u},\mathfrak{v})|}{\|\mathfrak{v}\|_V} = \sup_{\mathfrak{v}\in V} \frac{|(\mathfrak{u}, A^*_\omega \mathfrak{v})|}{\|A^*_\omega \mathfrak{v}\|} = \|\mathfrak{u}\|. \tag{3.15}$$

Unfortunately, the adjoint graph norm is not *localizable*, i.e., it stops being a norm for the larger, broken test space. In order to make it localizable, we augment it with an extra L^2 term, introducing a quasi-optimal test norm [30],

$$\|\mathfrak{v}\|^2_V = \|A^*_\omega \mathfrak{v}\|^2 + \alpha \|\mathfrak{v}\|^2, \quad \text{where} \quad \alpha = \mathcal{O}(1).$$

For the problem of interest, both A_ω and A^*_ω are bounded below with a frequency independent constant [6, 30],

$$\|A_\omega \mathfrak{u}\| \geq \gamma \|\mathfrak{u}\|, \quad \|A^*_\omega \mathfrak{v}\| \geq \gamma \|\mathfrak{v}\|. \tag{3.16}$$

Consequently, the original and modified test norms are robustly equivalent. Indeed,

$$\|A^*_\omega \mathfrak{v}\|^2 \leq \|\mathfrak{v}\|^2_V \text{ and } \|\mathfrak{v}\|^2_V \leq (1 + \frac{\alpha}{\gamma^2}) \|A^*_\omega \mathfrak{v}\|^2. \tag{3.17}$$

The robust stability constant is maintained for the broken spaces [5]. This implies that the approximation error is bounded by the corresponding best approximation uniformly in ω.

$$\|\mathfrak{u} - \mathfrak{u}_h\|^2 + \|\hat{\mathfrak{u}} - \hat{\mathfrak{u}}_h\|^2_Q \leq C \Big[\inf_{w_h} \|\mathfrak{u} - w_h\|^2 + \inf_{\hat{w}_h} \|\mathfrak{u} - \hat{w}_h\|^2 \Big] \tag{3.18}$$

The error in the trace $\hat{\mathfrak{u}}$ is measured in a special minimum energy extension norm. In 1D, the estimate implies that the DPG method is pollution free. Indeed, in this case traces are just numbers and the best approximation error (in any norm) is zero. Note that the L^2 best approximation is pollution free. Unfortunately, this is not true in the multidimensional case, where the best approximation error for the traces is not pollution free [6].

3.3.2 Adjusted adjoint test norm

Resolution of the optimal test functions corresponding to the quasi-optimal test norm is relatively easy for meshes that satisfy the Nyquist criterion (a fixed number of elements per wavelength). Satisfaction of the Nyquist criterion though, would require the use of very fine initial meshes for large wave numbers. We avoid this problem by adjusting the frequency in the quasi-optimal test norm to the element size h.

$$\|\mathfrak{v}\|^2_V = \|A^*_{\bar{\omega}} \mathfrak{v}\|^2 + \alpha \|\mathfrak{v}\|^2, \quad \text{where} \quad \bar{\omega} = \max(\omega, \frac{6}{h}) \tag{3.19}$$

Clearly, the new test norm is not longer quasi-optimal. Nevertheless, we will see in the presented numerical experiments that the resulting mesh refinements are correct. As the element size h decreases, the new test norm converges to the quasi-optimal test norm.

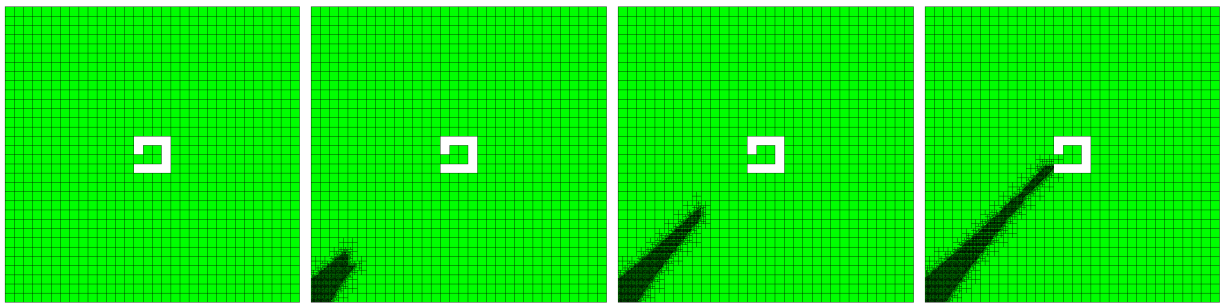
3.3.3 Static condensation

In (3.13), the field unknowns $u \in (L^2(\Omega))^d$ and $p \in L^2(\Omega)$ have no global continuity constraints. Therefore they can be element-wise condensed out and we solve only for the trace ($\hat{p}$) and the flux ($\widehat{u \cdot n}$). This reduces the size of the final linear system significantly.

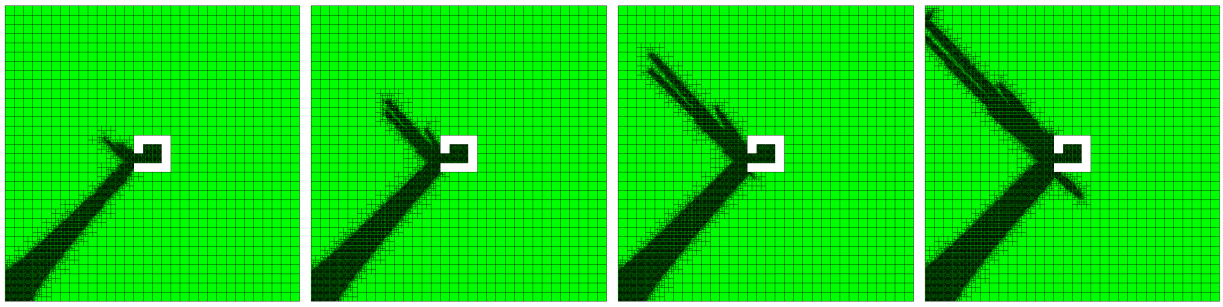
3.4 Gaussian beam simulation

We demonstrate the DPG adaptive technology by solving a problem that is characterized by a localized behavior of the solution. In particular we simulate a Gaussian beam source in free space with a cavity in the middle of the domain. The angular frequency is 500π (approximately 350 wavelengths along a 45° angle). In this example the adaptive DPG method turns out to be a very powerful tool, since it avoids unnecessary computations in areas of the domain where the wave doesn't exist. We mention here, that all the simulations are done using a Fortran package (hp2D), an extension of the code presented in [8], that supports the whole exact sequence. The code uses a recently developed shape functions package [10], and also a multi-frontal solver for shared memory architecture [19]. In Figures 2 and 3 below we show the evolution of the mesh and the corresponding numerical solution of the real part of the pressure, with the mesh refinements.

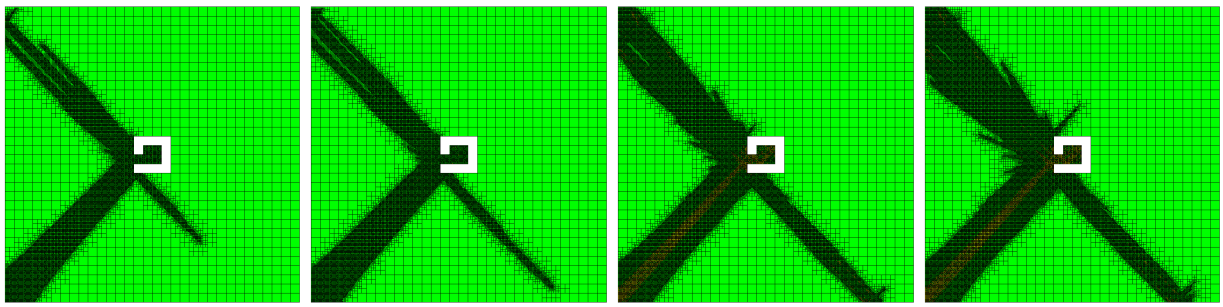
Figure 2: Meshes



(a) Refinement 0 (b) Refinement 20 (c) Refinement 40 (d) Refinement 60

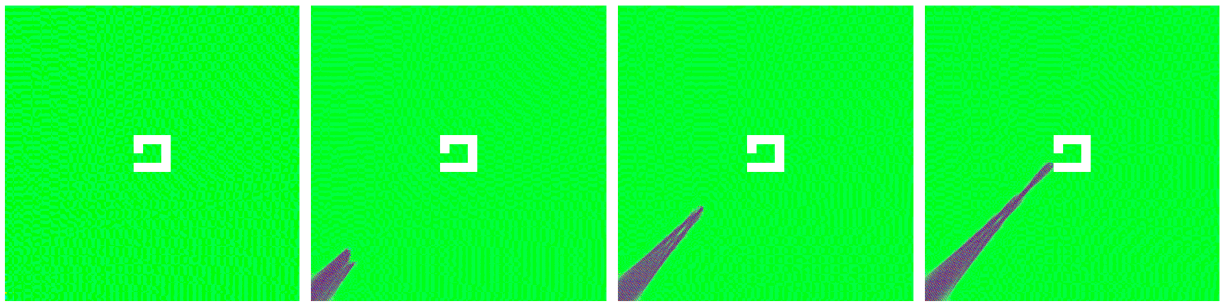


(e) Refinement 80 (f) Refinement 100 (g) Refinement 120 (h) Refinement 140

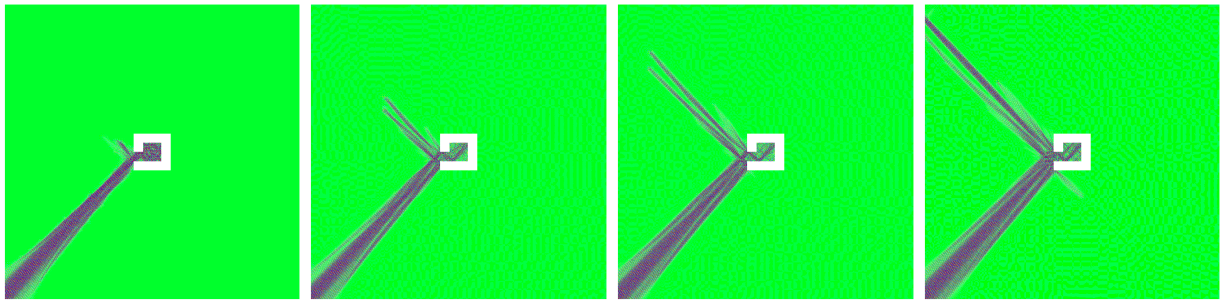


(i) Refinement 160 (j) Refinement 180 (k) Refinement 200 (l) Refinement 220

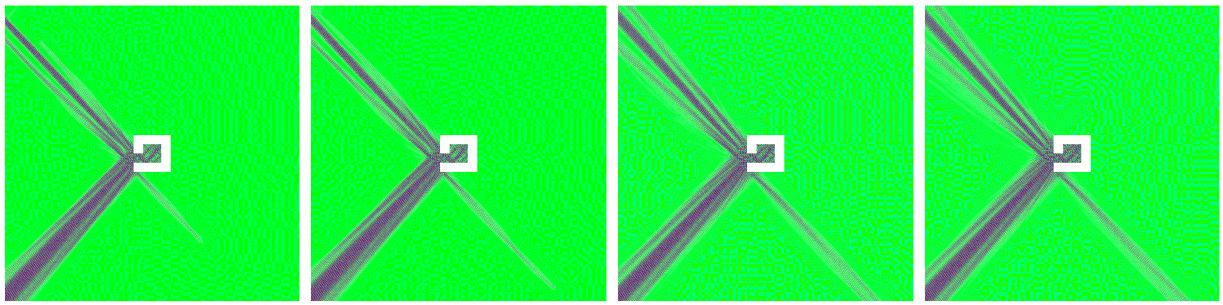
Figure 3: Numerical Pressure



(a) Refinement 0 (b) Refinement 20 (c) Refinement 40 (d) Refinement 60



(e) Refinement 80 (f) Refinement 100 (g) Refinement 120 (h) Refinement 140



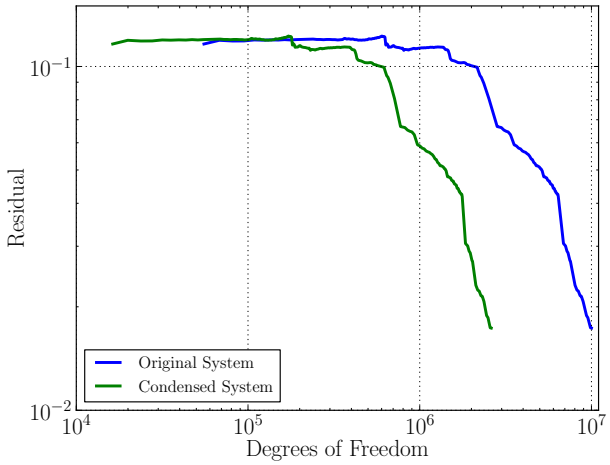
(i) Refinement 160 (j) Refinement 180 (k) Refinement 200 (l) Refinement 220

We start the simulation with a mesh of approximately 1000 cubic elements. The only reason we do this is to capture the geometry of the cavity with a uniform initial mesh. With this resolution, we are still very far away from satisfying the Nyquist criterion and as expected, the solution on that mesh is not meaningful. However, even at this stage we can start the adaptive process. The algorithm successfully manages to grow the mesh along with the solution, keeping the mesh very coarse at the areas where the solution is practically zero. We use a simple hp-strategy where we perform h-refinements until the size of the element reaches a half wavelength and then we switch to p-refinements. We also use a marking strategy to deal with the singularities at the corners of the cavity, i.e., we force h-refinements for elements adjacent to the corners.

3.5 Convergence

In Fig. 4 we show the convergence of the global residual against the number of degrees of freedom for both the original and the condensed system. Apart from some variations at the beginning of the simulation, the residual decreases, providing a reliable stopping criterion for the adaptivity process. Note that, condensing out the field unknowns (u and p), reduces the size of the final system significantly. This decreases memory requirements and saves a lot of computation time for the solution process of the linear system.

Figure 4: Residual Convergence



At this point we switch gears and focus on the solver for the linear system. In the following section, we present a two-grid-like preconditioner for DPG systems.

4 Solver

The adaptive algorithm used for the example in the previous section produces meshes that, in some sense, have a one dimensional structure, i.e., mesh refinements follow the wave in the direction of propagation. For this type of meshes the best choice for solving the linear system is a multi-frontal direct solver. However, the problem has to be solved many times (220 for the given example) before a meaningful solution is found.

Employing a direct solver so many times is far from optimal, especially in 3D computations. Additionally, one is not interested in a fully converged solution throughout the whole adaptive procedure, but only in a solution which is accurate enough to perform reliable adaptive refinements. As we mentioned earlier, the DPG global stiffness matrix is always Hermitian, even for indefinite wave operators. Consequently, the Conjugate Gradient (CG) method is the best candidate, provided that it is combined with a good preconditioner. On the contrary, for standard Galerkin discretizations one must use GMRES, with a preconditioner which shifts the spectrum to the positive half plane. The resulting solver is not always reliable, especially when the grid is not fine enough. In general, there are various approaches for designing fast and robust iterative solvers for high frequency wave propagation problems [9, 18]. Some of them are: the fast sweeping preconditioner [4, 3], shifted Laplacian techniques [18, 25], domain decomposition methods [24, 14, 28, 26] and multigrid methods [27, 15]. For the DPG method A. Barker et al. [2] analyzed a one-level additive Schwarz preconditioner for the Poison problem. Additionally, N. Roberts and J. Chan present numerical results for a geometric multigrid preconditioner applied to the DPG system for Poison, Stokes and Navier-Stokes equations [21]. We are aware of only one contribution relevant to the DPG method for high frequency wave propagation problems. Gopalakrishnan and Schöberl in [12] implemented a multiplicative Schwarz preconditioner for CG for the 2D primal formulation for Helmholtz equation and showed convergence at a rate independent of the polynomial degree and the wavenumber.

4.1 Two-grid-like preconditioner for CG

We now give a description of a new two-grid preconditioner by explicitly describing its steps. We emphasize here that this is not a typical multigrid strategy; the meshes used are far from the usual multigrid meshes.

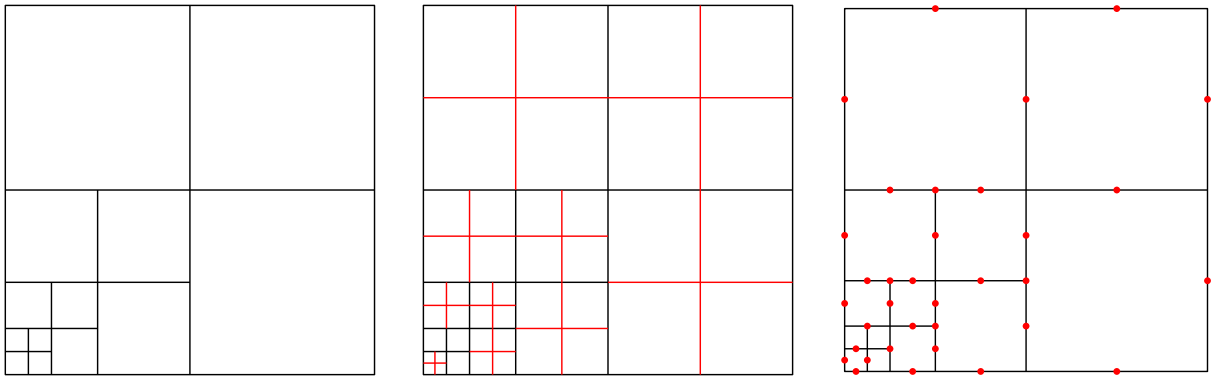
4.1.1 Step 1: Direct Solver

We fix any arbitrary mesh throughout the adaptive refinements and solve the problem using a direct solver. We define this to be our *coarse* grid, and we store the Cholesky decomposition of the global stiffness matrix. This decomposition will be used later in the *coarse-grid correction* step for forthcoming meshes produced after refinements.

4.1.2 Step 2: Macro-Element

Let the coarse grid be the n^{th} mesh. We denote by *fine* grid to be the $(n+k)^{\text{th}}$ mesh. The fine grid is the mesh resulted after k adaptive hp-refinements applied to the n^{th} mesh and it is the current mesh where we seek the solution to the problem. We define the *macro* grid to be the resulting mesh after we condense out all the new degrees of freedom which do not lie on the skeleton of the n^{th} (coarse) mesh. Notice that now the two meshes have the same topology. This step will be very essential for the construction of prolongation and restriction operators. We demonstrate this more clearly in Fig. (5).

Figure 5: Macro Grid Definition



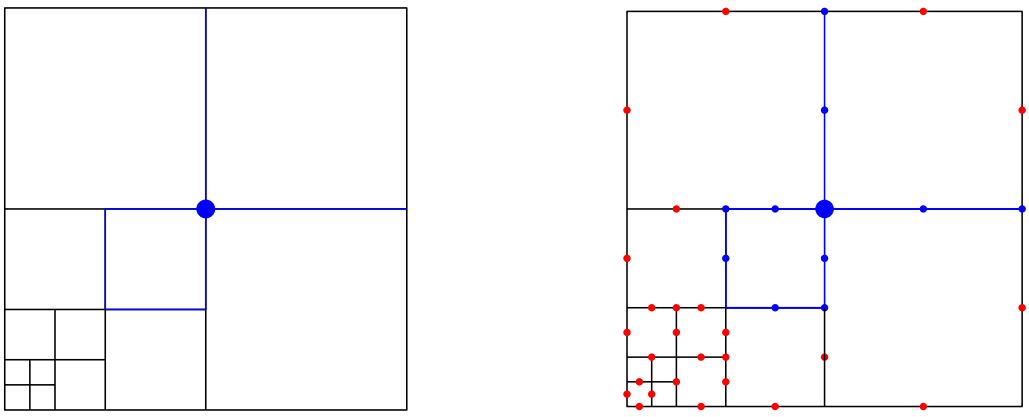
(a) n^{th} (Coarse) Grid (b) $(n+k)^{\text{th}}$ (Fine) Grid (c) $(n+k)^{\text{th}}$ (Macro) Grid

We note here two important points. First, the fine mesh can result from arbitrary and multiple hp-refinements of the coarse grid. This is not the usual global h-refined or p-refined multigrid fine mesh. Finally, because of the static condensation of the error representation function ψ, the interior (L^2) and the macro degrees of freedom, the usual subspace correction theory [29] can not be applied here.

4.1.3 Step 3: Additive Schwarz Smoother

We use the standard additive Schwarz method for smoothing the residual in the macro-grid, i.e., a block Jacobi iteration scheme with overlapping blocks. The additive smoother is preferred over the multiplicative one (Gauss Seidel with overlapping blocks) because of its local properties, i.e., it can be implemented in parallel. We define a patch to be the support of a coarse grid vertex basis function (see Fig. 6). A block is then constructed by the interaction of the macro degrees of freedom within a patch.

Figure 6: Construction of a vertex patch



(a) Vertex patch - coarse grid (b) Vertex patch - macro grid

4.1.4 Step 4: Coarse grid correction

In general, the coarse grid correction step requires the construction of a prolongation operator $I_{\mathrm{C}}^{\mathrm{F}}$ which will transfer a solution vector from the coarse to the fine mesh exactly and a restriction operator $I_{\mathrm{F}}^{\mathrm{C}}$ which restricts the fine residual vector to the coarse grid. In other words the prolongation maps a coarse basis function to its representation in terms of the fine basis functions. While such a construction in the standard FEM is straightforward, in case of DPG it is a bit more complicated. In DPG discretizations, new edges are created after an h-refinement, therefore new interface variables appear, which have no ancestors. Consequently, the usual prolongation operator based on constrained approximation [8] would not be well defined in this case. In order to overcome this difficulty, we introduced the concept of a macro-grid described in the previous section. Now, the two meshes have the same topology and the prolongation operator can be constructed edge-wise (face-wise in 3D). Practically, the construction reduces to a one dimensional interpolation problem [7]. In case of p-refinements with hierarchical shape functions, the new degrees of freedom are simply set to zero. Finally, the restriction operator is defined to be the adjoint of the prolongation operator.

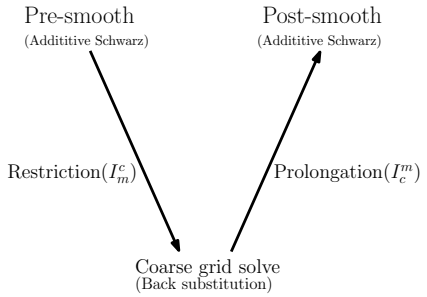
4.1.5 Step 5: Symmetric two-grid cycle

We implement the symmetric two-grid cycle between our coarse mesh and the macro mesh (see Fig. 7). Let $I_{\mathrm{C}}^{\mathrm{M}}$ denote the prolongation operator between the coarse and the macro grid. The restriction operator is defined by the transpose, $I_{\mathrm{M}}^{\mathrm{C}} = (I_{\mathrm{C}}^{\mathrm{M}})^*$, and the coarse grid correction operator is given by $Q = I_{\mathrm{M}}^{\mathrm{C}} A_{\mathrm{C}}^{-1} (I_{\mathrm{C}}^{\mathrm{M}})^*$, where A_{C}^{-1} denotes the inverse of the global stiffness matrix at the coarse level. Note that, in practice, we invoke the Cholesky decomposition of A_{C} which is already stored at step 1, therefore this is just a back substitution step. Finally, we denote by S the additive Schwarz smoothing operator.
An outline of the two-grid cycle is given below.

Let $Ax = b$ denote the linear system to be solved on the macro grid and let x_n be the solution to the n^{th} iteration of the CG algorithm. Then the preconditioning involves the following steps:

1. Compute initial residual: $r_n^1 = b - A\,x_n$
2. Pre-smooth: $z_n^1 = S\,r_n^1$
3. Compute new residual: $r_n^2 = r_n^1 - Az_n^1 = (I - AS)r_n^1$
4. Coarse grid correction: $z_n^2 = I_{\mathrm{C}}^{\mathrm{M}}\, A_{\mathrm{C}}^{-1}\, I_{\mathrm{M}}^{\mathrm{C}}\, r_n^2 = Q\, r_n^2$
5. Compute the updated residual: $r_n^3 = r_n^2 - Az_n^2 = (I - AQ)r_n^2$
6. Post-smooth: $z_n^3 = S\,r_n^3$
7. Compute the total correction to the solution: $z_n = z_n^1 + z_n^2 + z_n^3$

Figure 7: Two-grid cycle



4.1.6 Step 6: Preconditioned Conjugate Gradient (PCG)

The PCG [23] is given by Algorithm 1.

Algorithm 1 Preconditioned Conjugate Gradient

$r_0 = b - Ax_0$
$z_0 = Mr_0$
$p_0 = z_0$
do $j = 1, 2...$ *until convergence* :
$\alpha_j = \dfrac{(r_j, z_j)}{(Ap_j, p_j)}$
$x_{j+1} = x_j + \alpha_j p_j$
$r_{j+1} = r_j - \alpha_j A p_j$
$z_{j+1} = M r_{j+1}$
$\beta_j = \dfrac{(r_{j+1}, z_{j+1})}{(r_j, z_j)}$
$p_{j+1} = z_{j+1} + \beta_j p_j$
end do

The preconditioning operator M (see Algorithm 1) has to be Hermitian and positive definite. In order to guarantee this is true in our case, we have to perform the same number of pre and post smoothing steps. An explicit formula for M can be derived by:

$$\begin{aligned} z_n &= z_n^1 + z_n^2 + z_n^3 = Sr_n^1 + Qr_n^2 + Sr_n^3 \\ &= Sr_n^1 + Q(b - A(x_n + z_n^1)) + S(b - A(x_n + z_n^1 + z_n^2)) \\ &= Sr_n^1 + Q(r_n^1 - Az_n^1) + S(r_n^1 - Az_n^1 - Az_n^2) \\ &= Sr_n^1 + Qr_n^1 - QASr_n^1 + Sr_n^1 - SASr_n^1 - SAQr_n^2 \\ &= Sr_n^1 + Qr_n^1 - QASr_n^1 + Sr_n^1 - SASr_n^1 - SAQ(r_n^1 - ASr_n^1) \\ &= (Q + S(I - AQ) + (I - QA)S - S(I - AQ)AS)r_n^1. \end{aligned} \tag{4.20}$$

Therefore, the preconditioner is given by

$$M = Q + S(I - AQ) + (I - QA)S - S(I - AQ)AS \tag{4.21}$$

and the reduction of the residual is given by:

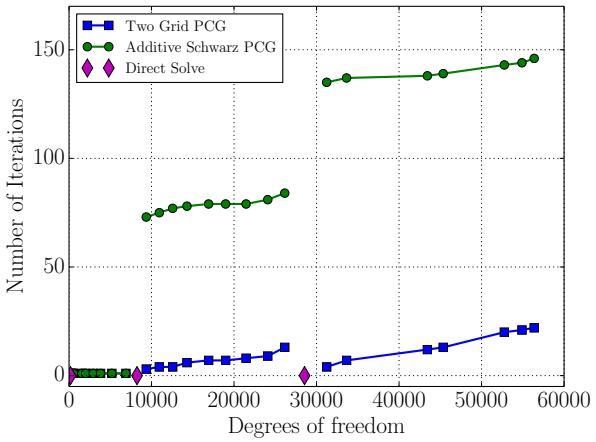
$$\begin{aligned} r_n^4 &= b - A(x_n^1 + z_n^1 + z_n^2 + z_n^3) \\ &= r_n^3 - Az_n^3 = r_n^3 - ASr_n^3 \\ &= (I - AS)r_n^3 = (I - AS)(I - AQ)r_n^2 \\ &= (I - AS)(I - AQ)(I - AS)r_n^1, \end{aligned} \tag{4.22}$$

We test our preconditioner (4.21) to solve the 2D linear acoustics problem for various values of the angular frequency ω. We summarize our results in the next section.

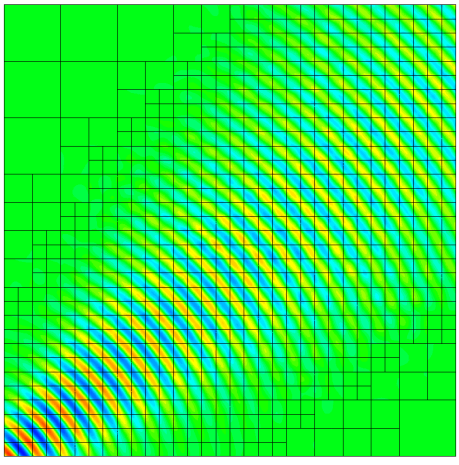
4.2 Solver - results

We run our solver for three different angular frequencies ($\omega = 40\pi$, 80π, 120π) and we summarize our results in Figures (8), (9) and (10) respectively. As a stopping criterion for CG iterations we use the L^2 discrete norm of the residual. Since we are interested only in a partially converged solution (enough to perform meaningful refinements), a tolerance of 10^{-2} was used. Additionally, we perform several smoothing steps (μ), and use a damping parameter (θ). These two parameters turned out to be very critical for the convergence of the solver. We present the results for $\theta = \frac{1}{3}$ and $\mu = 10$. These values were selected experimentally. We run all the simulations until the L^2 relative error reduces below 10%, when at this point the wave is resolved. Figures (8b), (9b) and (10b) show the final mesh and the solution of the numerical pressure. We can safely conclude that the tolerance used was small enough to produce optimal adaptive refinements. In figures (8a),(9a) and (10a) we compare two preconditioners, the additive Schwarz and the two-grid described in the previous section. For the two-grid preconditioner, we follow a simple strategy, where the coarse grid is redefined every 10 refinements. Although the additive Schwarz converges relatively fast, the number of iterations grows every time the coarse grid is reset, i.e., the patches are redefined. The coarse grid correction seems to be necessary in order for the iterations to remain bounded, and that makes the two-grid strategy superior. The number of iterations drop every time we reset the coarse grid and start growing very slowly as we proceed with refinements. Intuitively, this is expected because the coarse grid correction becomes less effective, since the macro grid increasingly differs from the coarse grid as we keep refining. However, the number of iterations needed until convergence for the two-grid preconditioner appears to be independent of the frequency and the mesh.

Figure 8: $\omega = 40\pi$

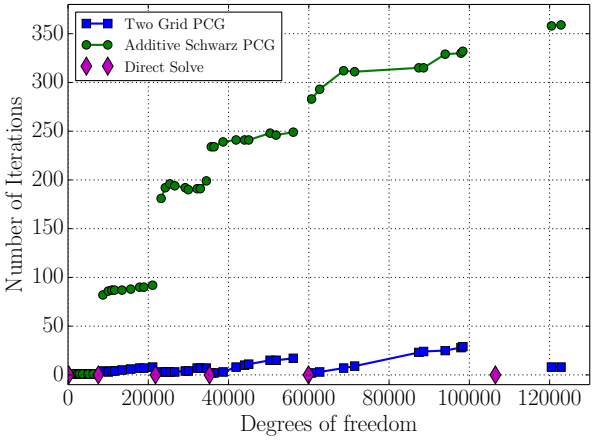


(a) Iterations vs dof

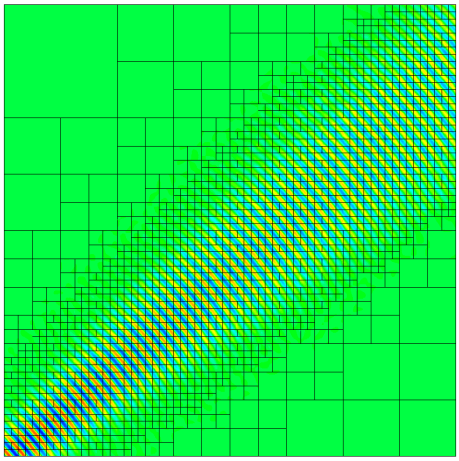


(b) Numerical pressure

Figure 9: $\omega = 80\pi$

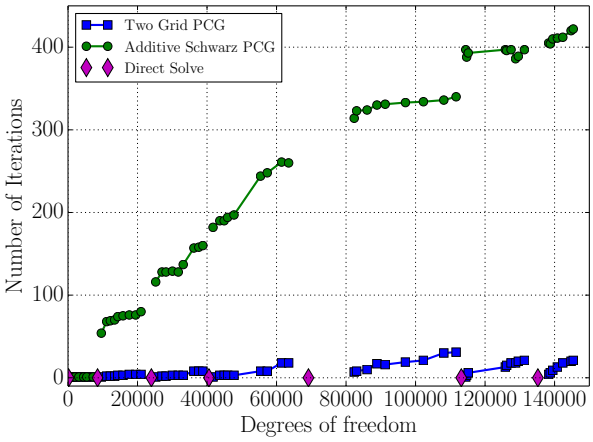


(a) Iterations vs dof

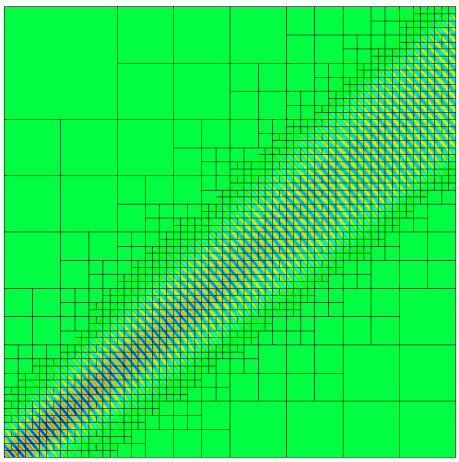


(b) Numerical pressure

Figure 10: $\omega = 120\pi$



(a) Iterations vs dof



(b) Numerical pressure

4.3 Additional comments/future work on the solver

The results on the solver presented above are very promising in the context of DPG. However, this is just a start and there are many issues we have to address in the future. First, a complete theoretical convergence analysis of the solver has to be developed. The usual subspace correction theory [29] can not be naively applied here because the solver operates on a statically condensed system. A theoretical analysis should give light for determining optimal damping parameters for the smoothing steps, which in our numerical experiments turned out to be very essential. In addition, despite the fact that the number of iterations of

the solver are independent of the problem size and the frequency, the overall computational cost obviously grows (smoother, static condensation, coarse grid correction). Fortunately, all these steps can be performed in parallel. A deeper study will also allow us to determine an optimal and efficient strategy with respect to when the coarse grid has to be reset and also how many smoothing steps have to be performed.

5 Conclusions/future work

In this paper, we presented the main ideas of the DPG methodology and demonstrated the important advantages it offers when applied to the simulation of high frequency wave propagation. The uniform pre-asymptotic stability properties of DPG along with the built in error indicator (the error representation function) makes it very attractive in simulating waves with localized solutions. Adaptivity works in a very optimal and efficient manner that allows us to allocate the computational resources correctly, without wasting computational power in unnecessary computations. For high frequency wave propagation, we saw that the mesh is built along with the solution, by solving the problem many times. However, we realized that for problems with even higher frequency especially for 3D computations, employing a direct solver for each solve is not efficient. This led us to the idea of constructing an iterative solver which exploits the information from previous meshes, in order to construct a partially converged solution to drive the adaptive refinements. Since the DPG stiffness matrix is always Hermitian positive definite, a preconditioner for CG was our obvious goal. We described the construction of the two-grid like preconditioner and showed convergence in terms of iterations, independent of the mesh and the wavenumber.
Our future work will focus on parallel implementations of the solver on shared and possibly distributed memory architectures. Ultimately, our plan is to built a solver which will adapt depending on the problem to be solved and according to the available computational resources. We will then be able to run larger simulations, such as 3D Maxwell's equations. Finally, we are aiming for some theoretical convergence analysis, not only to support our current numerical results, but also to help us optimize the future implementation of the solver.

Acknowledgments. The work has been partially supported with grants by AFOSR (FA9550-12-1-0484), NSF(DMS-1418822) and ONR (N00014-15-1-2496).

References

[1] Ivo M. Babuska and Stefan A. Sauter. Is the Pollution Effect of the FEM Avoidable for the Helmholtz Equation Considering High Wave Numbers? *SIAM J. Numer. Anal.*, 34(6):2392–2423, December 1997.

[2] Andrew T. Barker, Susanne C. Brenner, Eun-Hee Park, and Li-Yeng Sung. *A One-Level Additive Schwarz Preconditioner for a Discontinuous Petrov–Galerkin Method*, pages 417–425. Springer International Publishing, Cham, 2014.

[3] Björn Engquist and Lexing Ying. Sweeping preconditioner for the Helmholtz equation: Hierarchical Matrix Representation. *Commun. Pure Appl. Math.*, 64(5):697–735, 2011.

[4] Björn Engquist and Lexing Ying. Sweeping Preconditioner for the Helmholtz Equation: Moving Perfectly Matched Layers. *Multiscale Modeling & Simulation*, 9(2):686–710, 2011.

[5] C. Carstensen, L. Demkowicz, and J. Gopalakrishnan. Breaking spaces and forms for the DPG method and applications including Maxwell equations. *Computers & Mathematics with Applications*, 72(3):494 – 522, 2016.

[6] L. Demkowicz, J. Gopalakrishnan, I. Muga, and J. Zitelli. Wavenumber explicit analysis of a DPG method for the multidimensional Helmholtz equation. *Comput. Methods Appl. Mech. Engrg.*, 213/216:126–138, 2012.

[7] Leszek Demkowicz. *Computing with hp-Adaptive Finite Elements, Vol. 1: One and Two Dimensional Elliptic and Maxwell Problems*. Chapman & Hall/CRC, 1st edition, 2006.

[8] Leszek Demkowicz, Jason Kurtz, David Pardo, Maciej Paszynski, Waldemar Rachowicz, and Adam Zdunek. *Computing with hp-Adaptive Finite Elements, Vol. 2: Frontiers Three Dimensional Elliptic and Maxwell Problems with Applications*. Chapman & Hall/CRC, 1st edition, 2007.

[9] Yogi A. Erlangga. Advances in Iterative Methods and Preconditioners for the Helmholtz Equation. *Archives of Computational Methods in Engineering*, 15(1):37–66, 2008.

[10] Federico Fuentes, Brendan Keith, Leszek Demkowicz, and Sriram Nagaraj. Orientation embedded high order shape functions for the exact sequence elements of all shapes. *Computers & Mathematics with Applications*, 70(4):353 – 458, 2015.

[11] J. Gopalakrishnan and W. Qiu. An Analysis of the Practical DPG Method. *Math. Comp.*, 83:537–552, 2014.

[12] Gopalakrishnan, Jay and Schöberl, Joachim. *Degree and Wavenumber [In]dependence of Schwarz Preconditioner for the DPG Method*, pages 257–265. Springer International Publishing, Cham, 2015.

[13] I. Babuska. Error-Bounds for Finite Element Method. *Numer. Math.*, 16:322–333, 1971.

[14] J.Gopalakrishnan and J.E.Pasciak. Overlapping Schwarz Preconditioners for Indefinite Time Harmonic Maxwell Equations. *Math. Comput.*, 72(241):1–15, January 2003.

[15] J.Gopalakrishnan, J.E.Pasciak and L.F. Demkowicz. Analysis of a Multigrid Algorithm for Time Harmonic Maxwell Equations. *SIAM Journal on Numerical Analysis*, 42(1):90+, 2004.

[16] L. Demkowicz and J. Gopalakrishnan. Discontinuous Petrov-Galerkin (DPG) Method. *ICES Report*, 15-20, 2015.

[17] L.F.Demkowicz. Various Variational Formulations and Closed Range Theorem. *ICES Report*, 15-03, 2015.

[18] O. G. Ernst and M. J. Gander. Why it is Difficult to Solve Helmholtz Problems with Classical Iterative Methods. In Ivan G. Graham, Thomas Y. Hou, Omar Lakkis, and Robert Scheichl, editors, *Numerical Analysis of Multiscale Problems*, volume 83 of *Lecture Notes in Computational Science and Engineering*, pages 325–363. Springer Berlin Heidelberg, 2012.

[19] P. R. Amestoy and I. S. Duff and J. Koster and J.-Y. L'Excellent. A Fully Asynchronous Multifrontal Solver Using Distributed Dynamic Scheduling. *SIAM Journal on Matrix Analysis and Applications*, 23(1):15–41, 2001.

[20] Erasmo Recami, Michel Zamboni-Rached, and Leonardo A. Ambrosio. *Non-Diffracting Waves: An Introduction*. Wiley-VCH Verlag GmbH & Co. KGaA, 2013.

[21] Nathan V Roberts and Jesse Chan. A geometric multigrid preconditioning strategy for dpg system matrices. *arXiv preprint arXiv:1608.02567*, 2016.

[22] S. Nagaraj, S. Petrides and Leszek F. Demkowicz. Construction of DPG Fortin Operators For Second Order Problems. *ICES Report*, 15-22, 2015.

[23] Y. Saad. *Iterative Methods for Sparse Linear Systems*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2nd edition, 2003.

[24] Kim Seungil and Zhang Hui. Optimized Schwarz Method with Complete Radiation Transmission Conditions for the Helmholtz Equation in Waveguides. *SIAM Journal on Numerical Analysis*, 53(3):1537–1558, 2015.

[25] A. H. Sheikh, D. Lahaye, and C. Vuik. On the convergence of shifted Laplace preconditioner combined with multilevel deflation. *Numerical Linear Algebra with Applications*, 20(4):645–662, 2013.

[26] Christiaan C. Stolk. A rapidly converging domain decomposition method for the Helmholtz equation. *J. Comput. Phys.*, 241:240–252, 2013.

[27] Christiaan C. Stolk, Mostak Ahmed, and Samir Kumar Bhowmik. A Multigrid Method for the Helmholtz Equation with Optimized Coarse Grid Corrections. *SIAM J. Scientific Computing*, 36(6), 2014.

[28] Xiao-Chuan Cai and Olof B. Widlund. Domain Decomposition Algorithms for Indefinite Elliptic Problems. *SIAM J. Scientific Computing*, 13(1):243–258, 1992.

[29] Jinchao Xu. The method of subspace corrections. *Journal of Computational and Applied Mathematics*, 128(12):335 – 362, 2001. Numerical Analysis 2000. Vol. VII: Partial Differential Equations.

[30] J. Zitelli, I. Muga, L. Demkowicz, J. Gopalakrishnan, D. Pardo, and V. M. Calo. A Class of Discontinuous Petrov-Galerkin Methods. Part IV: The Optimal Test Norm and Time-harmonic Wave Propagation in 1D. *J. Comput. Phys.*, 230(7):2406–2432, April 2011.

A Appendix

A.1 Variational formulations - linear acoustics

We illustrate the concept of different variational formulations using the example of Linear Acoustics:

$$\begin{cases} i\omega u + \nabla p &= 0, \text{ in } \Omega \quad \text{(Linear Momentum)} \\ i\omega p + \text{div} u &= 0, \text{ in } \Omega \quad \text{(Conservation of mass)} \\ p - u \cdot n &= g, \text{ on } \partial\Omega \end{cases} \tag{A.23}$$

A.1.1 Strong formulation

By multiplying the first two equations by v and q respectively, and integrating over domain Ω, we end up with the *trivial* or *strong* formulation:

$$\begin{cases} u \in H(\text{div}, \Omega),\ p \in H^1(\Omega) \\ p - u \cdot n = g \text{ on } \partial\Omega \\ i\omega(u, v) + (\nabla p, v) \ = 0 \qquad v \in (L^2(\Omega))^d \\ i\omega(p, q) + (\text{div} u, q) = 0 \qquad q \in L^2(\Omega) \end{cases} \tag{A.24}$$

A.1.2 Classical formulation

If we relax (i.e. integrate by parts and build the boundary condition into the formulation) the conservation of mass equation, we have:

$$\begin{cases} u \in (L^2(\Omega))^d,\ p \in H^1(\Omega) \\ i\omega(u, v) + (\nabla p, v) &= 0, &v \in (L^2(\Omega))^d \\ i\omega(p, q) - (u, \nabla q) + \langle p, q\rangle_{\partial\Omega} &= \langle g, q\rangle_{\partial\Omega}, &q \in H^1(\Omega) \end{cases} \tag{A.25}$$

Eliminating velocity, we obtain the standard formulation for the Helmholtz equation,

$$\begin{cases} p \in H^1(\Omega) \\ (\nabla p, \nabla q) - \omega^2(p, q) + i\omega\langle p, q\rangle_\Gamma = i\omega\langle g, q\rangle_{\partial\Omega} \quad v \in H^1(\Omega) \end{cases} \tag{A.26}$$

A.1.3 Mixed formulation

If we relax the momentum equation, we have

$$\begin{cases} u \in V,\ p \in L^2(\Omega) \\ i\omega(u, v) - (p, \text{div} v) + \langle u \cdot n, v \cdot n\rangle_{\partial\Omega} &= \langle g, v \cdot n\rangle_{\partial\Omega}, &v \in V \\ i\omega(p, q) + (\text{div} u, q) &= 0, &q \in L^2(\Omega) \end{cases} \tag{A.27}$$

Note that the energy space for the velocity has now to incorporate an extra regularity assumption resulting from building in the impedance boundary condition,

$$V := \{v \in H(\text{div}, \Omega) \ :\ v \cdot n \in H^{1/2}(\partial\Omega)\}$$

We can eliminate now the pressure to obtain a variational formulation in terms of the velocity only,

$$\begin{cases} u \in V \\ (\text{div} u, \text{div} v) - \omega^2(u, v) + i\omega\langle u \cdot n, v \cdot n\rangle_\Gamma = i\omega\langle g, v \cdot n\rangle_{\partial\Omega}, \quad v \in V \end{cases} \tag{A.28}$$

A.1.4 Ultraweak formulation

Finally we can relax both equations:

$$\begin{cases} p \in L^2(\Omega), u \in (L^2(\Omega))^d \\ i\omega(u,v) - (p, \mathrm{div} v) + i\omega(p,q) - (u, \nabla q) = \langle g, q\rangle_{\partial\Omega}, \quad q \in H^1(\Omega),\ v \in H(\mathrm{div}, \Omega), \\ \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad q = -v\cdot n \text{ on } \partial\Omega \end{cases} \tag{A.29}$$

It can be shown that all formulations are simultaneously well or ill posed [17].

A.2 DPG formulations (broken test spaces)

In each of the discussed formulations, we can eliminate the boundary conditions on test functions and replace the test spaces with the corresponding broken test spaces

$$\begin{aligned} H^1(\Omega) &\longrightarrow H^1(\Omega_h) \\ H(\mathrm{curl}, \Omega) &\longrightarrow H(\mathrm{curl}, \Omega_h) \\ H(\mathrm{div}, \Omega) &\longrightarrow H(\mathrm{div}, \Omega_h)\,, \end{aligned}$$

i.e. we can "break test functions" at the expense of introducing additional unknowns (Lagrange multipliers) that are traces of dual energy spaces to mesh skeleton $\Gamma_h = \bigcup_{K\in\mathcal{T}_h} \partial K$,

$$\begin{aligned} H^1(\Omega_h) &\quad \text{introduces } n\cdot\hat{v} \in H^{-1/2}(\Gamma_h) := \mathrm{tr}_{\Gamma_h} H(\mathrm{div}, \Omega) \\ H(\mathrm{curl}, \Omega_h) &\quad \text{introduces } n\times\hat{H} \in H^{-1/2}(\mathrm{div}, \Gamma_h) := \mathrm{tr}^{\perp}_{\Gamma_h} H(\mathrm{curl}, \Omega) \\ H(\mathrm{div}, \Omega_h) &\quad \text{introduces } \hat{u} \in H^{1/2}(\Gamma_h) := \mathrm{tr}_{\Gamma_h} H^1(\Omega) \end{aligned}$$

We emphasize here that any well posed variational formulation can be reformulated with broken test spaces maintaining its stability properties. We refer the reader to [5] for further reading on well posedness of broken variational formulations. For example, the broken variational formulations for the classical and ultraweak formulations look as follows:

A.2.1 DPG primal (classical) formulation:

$$\begin{cases} p \in H^1(\Omega), \widehat{u\cdot n} \in H^{-1/2}(\Gamma_h),\ p - \widehat{u\cdot n} = g, \ \text{ on } \partial\Omega \\ (\nabla p, \nabla_h q) - \omega^2(p,q) + i\omega\langle p, q\rangle_{\partial\Omega} - \langle \widehat{u\cdot n}, q\rangle_{\Gamma_h} = 0, \quad q \in H^1(\Omega_h) \end{cases} \tag{A.30}$$

A.2.2 DPG ultraweak formulation:

$$\begin{cases} u \in (L^2(\Omega))^d,\ \ p \in L^2(\Omega) \\ \hat{p} \in H^{1/2}(\Gamma_h),\ \widehat{u\cdot n} \in H^{-1/2}(\Gamma_h), \hat{p} - \widehat{u\cdot n} \text{ on } \partial\Omega \\ -(p, i\omega q + \mathrm{div}_h v) - (u, i\omega v + \nabla_h q) + \langle \hat{p}, v\cdot n\rangle_{\Gamma_h} + \langle \widehat{u\cdot n}, q\rangle_{\Gamma_h} = 0 \\ \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad q \in H^1(\Omega_h),\ v \in H(\mathrm{div}, \Omega_h) \end{cases} \tag{A.31}$$

where ∇_h, div_h denote element-wise defined operators.
Note that the DPG strong formulation reduces to the usual least squares Method.