Dissertations and Theses Dissertations and Theses

Spring 5-20-2016

Accuracy of Wave Speeds Computed from the DPG and HDG Methods for Electromagnetic and Acoustic Waves

Nicole Michelle Olivares
Portland State University

Recommended Citation

Olivares, Nicole Michelle, "Accuracy of Wave Speeds Computed from the DPG and HDG Methods for Electromagnetic and Acoustic Waves" (2016). *Dissertations and Theses.* Paper 2920.
https://doi.org/10.15760/etd.2916

Accuracy of Wave Speeds Computed from the DPG and HDG Methods for Electromagnetic and Acoustic Waves

by

Nicole Michelle Olivares

A dissertation submitted in partial fulfillment of the
requirements for the degree of

Doctor of Philosophy
in
Mathematical Sciences

Dissertation Committee:
Jay Gopalakrishnan, Chair
Jeffrey Ovall
Lisa Zurk
Panayot Vassilevski
Leszek Demkowicz

Portland State University
2016

Abstract

We study two finite element methods for solving time-harmonic electromagnetic and acoustic problems: the discontinuous Petrov-Galerkin (DPG) method and the hybrid discontinuous Galerkin (HDG) method.

The DPG method for the Helmholtz equation is studied using a test space normed by a modified graph norm. The modification scales one of the terms in the graph norm by an arbitrary positive scaling parameter. We find that, as the parameter approaches zero, better results are obtained, under some circumstances. A dispersion analysis on the multiple interacting stencils that form the DPG method shows that the discrete wavenumbers of the method are complex, explaining the numerically observed artificial dissipation in the computed wave approximations. Since the DPG method is a nonstandard least-squares Galerkin method, its performance is compared with a standard least-squares method having a similar stencil.

We study the HDG method for complex wavenumber cases and show how the HDG stabilization parameter must be chosen in relation to the wavenumber. We show that the commonly chosen HDG stabilization parameter values can give rise to singular systems for some complex wavenumbers. However, this failure is remedied if the real part of the stabilization parameter has the opposite sign of the imaginary part of the wavenumber. For real wavenumbers, results from a dispersion analysis for the Helmholtz case are presented. An asymptotic expansion of the dispersion relation, as the number of mesh elements per wave increase, reveal values of the stabilization parameter that asymptotically minimize the HDG wavenumber errors.

Finally, a dispersion analysis of the mixed hybrid RaviartThomas method shows that its wavenumber errors are an order smaller than those of the HDG method.

We conclude by presenting some contributions to the development of software tools for using the DPG method and their application to a terahertz photonic structure. We attempt to simulate field enhancements recently observed in a novel arrangement of annular nanogaps.

For CJM.

Acknowledgments

I wish to express sincere thanks to my advisor, Prof. Jay Gopalakrishnan, for his mentorship over the past four years. The approaches to computational mathematical problem solving and communication that he taught me are invaluable preparation for my future.

Thank you also to my dissertation committee for taking the time to read my work and provide useful feedback, and to Prof. Joachim Schöberl for helpful advice for using Netgen and NGSolve.

I would like to thank the donor of the Eugene Enneking Doctoral Fellowship, Dr. Fariborz Maseeh, and the faculty involved in making the fellowship possible. I am honored to have been a recipient, and I am very grateful for the freedom afforded by the fellowship to focus on my work.

I am also grateful for the opportunity to have had an internship at INRIA Sophia Antipolis Méditerranée, where discussions leading to some of this work originated. This work was also partially supported by the AFOSR under grant FA9550-12-1-0484 and by the NSF under grant DMS-1318916.

Finally, I thank my family for their encouragement, and my dear friend Joshua for always being up for a bike ride.

Table of Contents

List of Tables

List of Figures

1

Introduction

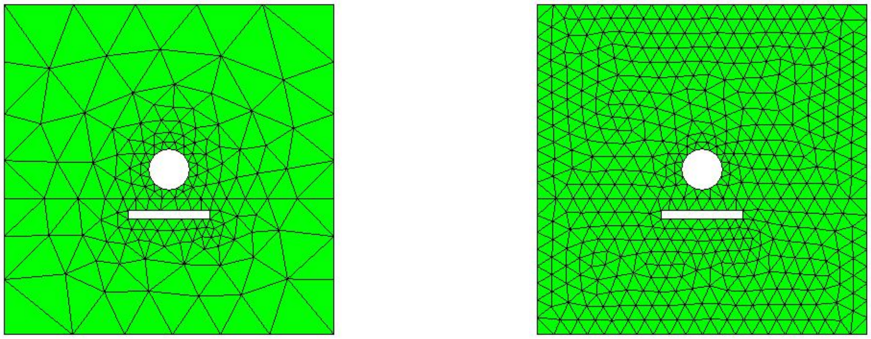
Waves consist of localized oscillations that continually transfer energy from one spatial location to another. Of the many types of waves, we focus on acoustic waves and electromagnetic (EM) waves. Understanding how these waves propagate is central to the development of myriads of applications in science and engineering. For example, EM waves are manipulated at ever-smaller scales in the design and production of microelectronics. Other domains of application include medical imaging and wireless communication.

To understand how a wave will propagate though a given region of space, one must solve a system of partial differential equations (PDE). We consider the PDE under the assumption that the wave is time-harmonic, that is, it oscillates at a single temporal frequency at every point in space. The simplest example of a time-harmonic wave is a plane wave, which is completely determined by its wavelength, amplitude, propagation direction, and phase. Note that any one of the quantities of wavelength, wave speed, and wavenumber determines the other two quantities.

Acoustic and EM waves share an underlying mathematical similarity, even though they differ from each other in the sense that acoustic waves propagate via oscillations of a material medium, whereas EM waves can exist without a material medium– the oscillating quantities of EM waves are the EM fields. The Helmholtz equation and time-harmonic Maxwell's equations are the two related PDE that describe time-harmonic acoustic and EM wave propagation, respectively. Note that, in order to solve Helmholtz equation or Maxwell's equations, additional knowledge about the materials and energy sources within the region of interest, as well as knowledge about the boundary of this region, must be incorporated into the problem. The resulting PDE and boundary conditions comprise a boundary value problem (BVP). The solution of a time-harmonic wave BVP completely characterizes the wave within the specified region, either by determining the material medium's velocity at every point in the case of an acoustic wave, or by determining the value of the EM field at every point for an EM wave.

Generally, it is not possible to find an exact solution to the Helmholtz equation or Maxwell's equations, so an approximate, numerical solution is sought instead, using some method to discretize the PDE. There are many methods for doing this, including the finite element method (FEM), the boundary element method (BEM), and the finite difference method. In the case of FEM, the region of interest is spatially discretized with a mesh of small shapes called elements– see Figure 1.1. For each of the various methods, a discretization determines a linear system of equations associated with the original problem, which is solved to obtain the numerical solution. Although solving linear systems is a very common computational problem for which extensive libraries of code have been written, it may still require significant amounts of time and/or memory resources. One factor that affects the computation time and memory requirements is the number of unknowns in the discretization (i.e., the size of the linear system). For example, the number of unknowns for FEM increases when a finer mesh is used. Another factor affecting the resource requirements of a method is whether the mathematical structure of the linear system has certain desirable properties that lend themselves to speedy solutions.

For applications that require many, many wave propagation problems to be solved, such as those involving optimization, even a slight reduction in the amount of time needed to compute a solution would significantly speed up the pace of development,



(A) A 2D mesh.

(B) A finer mesh.

Figure 1.1

since the time savings accumulate with each iteration of the linear solve step. For example, to produce an integrated circuit using lithography, the design of a photomask is optimized by solving for the EM waves diffracted by many candidate photomasks. If an optimal photomask could be found faster, it would reduce the costs of developing integrated circuits.

The difficulty of solving wave problems quickly is widely known, and is partially attributed to the fact that, for higher frequencies, the number of unknowns required to guarantee a given level of accuracy becomes very large. Intuitively, it is understandable that finer meshes (and, hence, more unknowns) are needed for higher frequencies: higher frequencies correspond to shorter wavelengths, so smaller elements are needed to capture the smaller variations of such waves. From this, one might hope that fixing a minimum number of elements per wavelength would be sufficient for maintaining a given level of accuracy at high frequencies. However, careful mathematical analysis has shown that this is not sufficient– the number of elements per wavelength would have to increase without bound as the wavenumber increases [**30**]. This phenomenon, known as the “pollution effect”, is directly related to the dispersion of a numerical method (which is not the same as physical dispersion).

This research focuses on contributing to the understanding of the performance of two finite element methods for solving time-harmonic wave propagation problems, the Discontinuous Petrov-Galerkin (DPG) method and the Hybrid Discontinuous-Galerkin (HDG) method.

The remainder of this chapter includes background about previous work on the DPG and HDG methods for wave propagation problems, and previous work studying the pollution effect and dispersion for FEM. We then summarize the contributions of this dissertation, and give precise statements of the Helmholtz equation and time-harmonic Maxwell's equations for the case of Dirichlet boundary conditions.

1.1. Background

The classical FEM uses the Ritz-Galerkin method to obtain a discretization from a variational (weak) formulation of a PDE [**43**]. Here, the FEM solution is sought in a finite dimensional space of continuous piecewise polynomial functions, called the trial space. Ritz-Galerkin FEM sets the test space to be the same as the trial space.

The HDG method also sets the trial and test spaces to be the same, but it is a discontinuous Galerkin (DG) method, a broad class of methods that use various techniques to enrich the trial and test spaces with discontinuous functions. The HDG method for elliptic problems was invented in [**10**], and first used to solve the Helmholtz equation with Dirichlet boundary conditions in [**27**]. It was extended to the Maxwell case [**38**], the mild slope equation [**21**], and the Helmholtz case with impedance boundary conditions [**11**]. A Schwarz algorithm for the Maxwell case was also developed in [**34**].

The DPG method was introduced in the series of papers [**12**], [**13**], [**15**], and [**45**]. This method uses discontinuous trial and test function spaces that are generally different, hence its characterization as a Petrov-Galerkin method. It minimizes a residual norm, so can be considered to belong to the class of least-squares Galerkin methods [**4**], [**7**], [**20**], but in a nonstandard functional setting. Analysis of the DPG method and optimal error estimates for the Helmholtz equation appeared in [**14**]. The DPG method for Maxwell's equations was analyzed in [**8**].

The pollution effect, already mentioned as a primary difficulty for solving the Helmholtz equation numerically, was proven to exist for Galerkin FEM in [**30**] and shown to be unavoidable in two or more dimensions for a large class of methods, called generalized FEM, in [**3**]. Efforts have been made to compare the severity of the pollution effect among different methods in order to minimize it. One way to measure the pollution effect is to quantify the dispersive errors of a method [**29**]. A numerical technique to measure the dispersion and dissipation of classical Galerkin FEM and related methods was presented in [**18**]. Explicit forms of dispersion relations for Galerkin FEM, including higher order schemes, have also been found [**31**], [**1**].

1.2. Contribution of this work

We perform the first dispersion analyses of the DPG and HDG finite element methods. This leads to studies of the effects of certain parameters used in these methods, and to comparisons of the dispersive errors of the DPG method and HDG method with other finite element methods.

For the DPG method, the method for the Helmholtz equation introduced in [**14**] is modified to include a positive parameter ε in the definition of the test space norm. When $\varepsilon = 1$, the method here reduces to that in [**14**]. The use of such scaling parameters was advocated in [**15**] based on numerical experience. Here, we provide a theoretical basis for its use with an error estimate that shows explicitly the dependence of the coefficient on ε. The dispersion analysis uncovers several important properties of the method as ε is varied.

For the HDG method, one focus of our study begins with the observation that, for both Helmholtz equation and Maxwell's equations, the methods are not always stable if the wavenumber is complex valued. Since complex valued wavenumbers do arise in important applications, we present results on how to choose the HDG stabilization parameter to ensure stability.

Another focus of study for the HDG method is the dispersion analysis in the case of real wavenumbers. Analytic computation of the dispersion relation is feasible in the lowest order case. We are thus able to study the influence of the stabilization parameter on the discrete wavenumber and offer recommendations on choosing good stabilization parameters. The optimal stabilization parameter values are found not to depend on the wavenumber. In the higher order case, since analytic calculations pose difficulties, we conduct a dispersion analysis numerically.

We also include an application of the DPG method to a real-world problem of 3D EM wave propagation. The numerical computations required substantial work on a shared library add-on to be used with the NGSolve [**42**] finite element software package.

Chapter 2 presents the theoretical results for the DPG method, and Chapter 2 presents the theoretical results for the HDG method. In Chapter 4, we present the results of dispersion analyses for both the DPG and HDG methods for the two dimensional (2D) Helmholtz equation. In Chapter 5, we present the numerical work involving NGSolve.

Some material in this dissertation first appeared in these publications:

[**23**] J. Gopalakrishnan, S. Lanteri, N. Olivares, and R. Perrussel, *Stabilization in relation to wavenumber in HDG methods*, Advanced Modeling and Simulation in Engineering Sciences, 2 (2015), p. 13

[**24**] J. Gopalakrishnan, I. Muga, and N. Olivares, *Dispersive and dissipative errors in the DPG method with scaled norms for Helmholtz equation*, SIAM Journal on Scientific Computing, 36 (2014), pp. A20–A39

In particular, Chapter 2 and part of Chapter 4 is based on [**24**]. Similarly, Chapter 3 and part of Chapter 4 is based on [**23**].

1.3. Boundary value problems

Throughout, all function spaces are over the complex field $\mathbb{C}$, and $\hat{\imath}$ denotes the imaginary unit. Domains (usually denoted D or Ω) in $\mathbb{R}^N$ are always assumed to be bounded, open, and connected with Lipschitz boundary.

1.3.1. The Helmholtz equation. The first order Helmholtz system on Ω is,

$$\hat{\imath}k\vec{u} + \vec{\nabla}\phi = \vec{0}, \qquad \text{in } \Omega, \tag{1a}$$

$$\hat{\imath}k\phi + \vec{\nabla}\cdot\vec{u} = f, \qquad \text{in } \Omega. \tag{1b}$$

where $f \in L^2(\Omega)$. In different contexts we will specify whether the wavenumber k is taken to be real or complex valued. As an acoustics model, the Helmholtz equation relates the linearized velocity $\vec{u}$ and the linearized pressure ϕ. The quantities represented by these variable in certain electromagnetic models will be described in Subsection 1.3.2.

It will be useful to write Equation (1) using operator notation. Let $A : H(\text{div}, \Omega) \times H^1(\Omega) \to L^2(\Omega)^N \times L^2(\Omega)$ denote the Helmholtz wave operator defined by

$$A(\vec{v}, \eta) = (\hat{\imath}k\vec{v} + \vec{\nabla}\eta, \hat{\imath}k\eta + \vec{\nabla}\cdot\vec{v}). \tag{2}$$

Then Equation (1) takes the form $A(\vec{u}, \phi) = \mathsf{f}$, with $\mathsf{f} = (\vec{0}, f)$. If we eliminate the vector component $\vec{u}$ from the system, we recover the usual second order form of the

Helmholtz equation,

$$-\Delta\phi - k^2\phi = \hat{\imath} k f, \qquad \text{on } \Omega. \tag{3}$$

This must be supplemented with boundary conditions. In the derivations of the DPG and HDG methods, we specify the Dirichlet boundary condition

$$\phi = 0, \qquad \text{on } \partial\Omega. \tag{4}$$

Defining the space

$$R = H(\text{div}, \Omega) \times H_0^1(\Omega), \tag{5}$$

the boundary value problem can be stated as:

$$\text{Find } (\vec{u}, \phi) \in R \text{ satisfying } A(\vec{u}, \phi) = f. \tag{6}$$

It is well known [**28**] that, except for k in an isolated countable set of real values Σ, this problem has a unique solution. We assume henceforth that k is not in Σ. A quantitative form of this assumption is that there exists a constant $C(k) > 0$, possibly depending on k, such that the solution of (6) satisfies

$$\|(\vec{u}, \phi)\| \le C(k)\|f\|. \tag{7}$$

Here and throughout this work, $\|\cdot\|$ denotes the L^2 norm, or the natural norm in the Cartesian product of several L^2 component spaces. One expects the constant $C(k)$ to become large as k approaches any of the resonances in Σ.

	Quantity	SI units
$\vec{x}$	Position	m
t	Time	s
$\vec{\mathcal{E}}$	Electric field intensity	$\mathrm{V{\cdot}m^{-1} = kg{\cdot}m{\cdot}s^{-3}{\cdot}A^{-1}}$
$\vec{\mathcal{D}}$	Electric displacement	$\mathrm{C{\cdot}m^{-2} = s{\cdot}A{\cdot}m^{-2}}$
$\vec{\mathcal{H}}$	Magnetic field intensity	$\mathrm{A{\cdot}m^{-1}}$
$\vec{\mathcal{B}}$	Magnetic induction	$\mathrm{T = kg{\cdot}s^{-2}{\cdot}A^{-1}}$
$\vec{\mathcal{J}}$	Electric current density	$\mathrm{A{\cdot}m^{-2}}$
ρ	Electric charge density	$\mathrm{C{\cdot}m^{-3} = s{\cdot}A{\cdot}m^{-3}}$
ϵ	Permittivity	$\mathrm{F{\cdot}m^{-1} = s^{4}{\cdot}A^{2}{\cdot}kg^{-1}{\cdot}m^{-3}}$
μ	Permeability	$\mathrm{H{\cdot}m^{-1} = kg{\cdot}m{\cdot}A^{-2}{\cdot}s^{-2}}$
σ	Conductivity	$\mathrm{S{\cdot}m^{-1} = s^{3}{\cdot}A^{2}{\cdot}kg^{-1}{\cdot}m^{-3}}$
f	Frequency	$\mathrm{s^{-1}}$
ω	Angular frequency	$\mathrm{rad{\cdot}s^{-1}}$
k	Wavenumber	$\mathrm{rad{\cdot}m^{-1}}$

Table 1.1. Variables used to express Maxwell's equations, with units.

1.3.2. The 3D and 2D Maxwell systems. The time-harmonic 3D Maxwell system is derived from the time-dependent Maxwell system

$$\frac{\partial \vec{\mathcal{B}}}{\partial t} + \vec{\nabla}\times\vec{\mathcal{E}} = \vec{0}, \qquad \text{(Faraday's law)} \tag{8a}$$

$$\frac{\partial \vec{\mathcal{D}}}{\partial t} - \vec{\nabla}\times\vec{\mathcal{H}} = -\vec{\mathcal{J}}, \qquad \text{(Ampère's law)} \tag{8b}$$

$$\nabla\cdot\vec{\mathcal{D}} = \rho, \qquad \text{(Gauss's law)} \tag{8c}$$

$$\nabla\cdot\vec{\mathcal{B}} = 0. \tag{8d}$$

Here, $\vec{\mathcal{E}}$, $\vec{\mathcal{D}}$, $\vec{\mathcal{H}}$, and $\vec{\mathcal{B}}$ are vector fields dependent on $\vec{x}\in\mathbb{R}^3$ and $t\in\mathbb{R}$. Units for these and other quantities used in the derivation are given in Table 1.1. The fundamental fields $\vec{\mathcal{E}}$ and $\vec{\mathcal{H}}$ are the electric and magnetic field intensities, respectively. The fields $\vec{\mathcal{D}}$ and $\vec{\mathcal{B}}$ are the electric displacement and the magnetic induction, respectively. The sources ρ and $\vec{\mathcal{J}}$ are the electric charge density and electric current density, respectively. Assuming conservation of charge, which is quantified by

$$\nabla \cdot \vec{\mathcal{J}} + \frac{\partial \rho}{\partial t} = 0,$$

equations (8c) and (8d) can be derived from equations (8a) and (8b).

We also assume that all materials are isotropic and time invariant within the region of space that we are interested in modeling. Although it is certainly possible to model materials that have time-dependent and/or non-isotropic properties, we do not need these features for our purposes. With this simplification, then, the constitutive (material) parameters ϵ and μ, called the electric permittivity and magnetic permeability, respectively, are scalar functions of $\vec{x}$. The constitutive relations

$$\vec{\mathcal{D}} = \epsilon \vec{\mathcal{E}} \qquad \text{and} \qquad \vec{\mathcal{B}} = \mu \vec{\mathcal{H}}$$

quantify the dependence of $\vec{\mathcal{D}}$ and $\vec{\mathcal{B}}$ on the fundamental fields $\vec{\mathcal{E}}$ and $\vec{\mathcal{H}}$. It is convenient to scale the constitutive parameters by their values for vacuum, which are

$$\epsilon_0 = 8.8541878176 \times 10^{-12} \qquad \text{and} \qquad \mu_0 = 4\pi \times 10^{-7}$$

(see Table 1.1 for units). We may then consider the relative permittivity of a material, ϵ_r, and the relative permeability, μ_r, with

$$\epsilon = \epsilon_0 \epsilon_r \qquad \text{and} \qquad \mu = \mu_0 \mu_r.$$

A third constitutive relation is required for conductive materials, which have non-zero conductivity σ. If the field strengths are not large, this relation is

$$\vec{\mathcal{J}} = \sigma \vec{\mathcal{E}} + \vec{\mathcal{J}}_a,$$

where $\vec{\mathcal{J}}_a$ is the applied current density. Then, equations (8a) and (8b) can be written

$$\mu\frac{\partial\vec{\mathcal{H}}}{\partial t} + \vec{\nabla}\times\vec{\mathcal{E}} = \vec{0}, \tag{9a}$$

$$\epsilon\frac{\partial\vec{\mathcal{E}}}{\partial t} + \sigma\vec{\mathcal{E}} - \vec{\nabla}\times\vec{\mathcal{H}} = -\vec{\mathcal{J}}_a. \tag{9b}$$

The derivation of the time-harmonic equations follows from assuming that the fields are of the form

$$\vec{\mathcal{E}}(\vec{x},t) = \mathrm{Re}\left(c_1\vec{E}(\vec{x})e^{\hat{\imath}\omega t}\right), \tag{10a}$$

$$\vec{\mathcal{D}}(\vec{x},t) = \mathrm{Re}\left(c_1\vec{D}(\vec{x})e^{\hat{\imath}\omega t}\right), \tag{10b}$$

$$\vec{\mathcal{H}}(\vec{x},t) = \mathrm{Re}\left(c_2\vec{H}(\vec{x})e^{\hat{\imath}\omega t}\right), \tag{10c}$$

$$\vec{\mathcal{B}}(\vec{x},t) = \mathrm{Re}\left(c_2\vec{B}(\vec{x})e^{\hat{\imath}\omega t}\right). \tag{10d}$$

Here, $\omega = 2\pi f$ is a given angular frequency, and c_1 and c_2 are scaling constants. For consistency, the source $\vec{\mathcal{J}}_a$ as well is taken to be of the form

$$\vec{\mathcal{J}}_a(\vec{x},t) = \mathrm{Re}\left(c_2\vec{J}_a(\vec{x})e^{\hat{\imath}\omega t}\right). \tag{11}$$

In this dissertation, it will be convenient to use different scaling constants under different circumstances, so we leave c_1 and c_2 unspecified for the moment. We extend the relative permittivity to have an imaginary part,

$$\hat{\epsilon}_r = \epsilon_r - i\frac{\sigma}{\omega\epsilon_0}, \tag{12}$$

and set $\hat{\epsilon} = \epsilon_0\hat{\epsilon}_r$. The wavenumber

$$k = \omega\sqrt{\mu\hat{\epsilon}}$$

is a *potentially complex-valued* function of $\vec{x}$. From now on, for simplicity, we will omit the hats above $\hat{\epsilon}$ and $\hat{\epsilon}_r$, as well as the subscript for $\vec{J}_a$. Equations (9) become

$$\hat{\imath}\omega\mu\left(\frac{c_2}{c_1}\right)\vec{H} + \vec{\nabla}\times\vec{E} = \vec{0}, \tag{13a}$$

$$\hat{\imath}\omega\epsilon\left(\frac{c_1}{c_2}\right)\vec{E} - \vec{\nabla}\times\vec{H} = -\vec{J}. \tag{13b}$$

For all work involving the HDG method, we take μ and ϵ (and, hence, k) to be constant. Under this assumption, we set the scaling constants as

$$c_1 = \frac{1}{\sqrt{\epsilon}} \qquad \text{and} \qquad c_2 = \frac{1}{\sqrt{\mu}}.$$

Our HDG method is then based on

$$\hat{\imath}k\vec{E} - \vec{\nabla}\times\vec{H} = -\vec{J}, \qquad \text{in } \Omega, \tag{14a}$$

$$\hat{\imath}k\vec{H} + \vec{\nabla}\times\vec{E} = \vec{0}, \qquad \text{in } \Omega, \tag{14b}$$

$$\hat{\nu}\times\vec{E} = \vec{0}, \qquad \text{on } \partial\Omega, \tag{14c}$$

where $\vec{J} \in (L^2(\Omega))^3$ and $k \in \mathbb{C}$. Note that we use the notation $\hat{\nu}$ throughout to generically denote the outward unit normal on various domains – the specific domain will be clear from context – e.g., in (14c), it is $\partial\Omega$. The Dirichlet boundary condition is used to simplify the presentation of the HDG method for the purpose of the dispersion analysis which, as we will see in Chapter 4, only involves the local matrix of the method and not the boundary conditions. It is of course possible to use the HDG method with other boundary conditions. Similarly, although the HDG method is easily applicable for varying ϵ and μ, our assumption that they are constant is made for simplifying the development of the dispersion analysis later.

For the numerical work using the DPG method in Chapter 5, we shall modify the above derivation of Maxwell's equations in order to be consistent with certain references. The modifications are addressed in Subsection 5.0.1.

It is interesting to consider the 2D Maxwell system as well. In fact, when we define an HDG method for solving Maxwell's equations in Chapter 3, we will see that an HDG method for the 2D Maxwell system can be determined from the HDG method for the 2D Helmholtz system, and this will guide us in making the 2D Helmholtz and 3D Maxwell formulations consistent.

The 2D time-harmonic Maxwell system is obtained from (14a)–(14b) by imposing cylindrical symmetry, with $\vec{H}$ confined to the $x-y$ plane and $\vec{E}$ having a single nonzero component in the z-direction. This gives

$$\hat{\imath}kE - \nabla\times\vec{H} = -J, \tag{15a}$$

$$\hat{\imath}k\vec{H} + \vec{\nabla}\times E = 0. \tag{15b}$$

Here, the two-dimensional scalar curl $\nabla\times\cdot$ and the vector curl $\vec{\nabla}\times\cdot$ are defined by

$$\nabla\times\vec{H} = \partial_1 H_2 - \partial_2 H_1 = \vec{\nabla}\cdot\mathrm{Rot}(\vec{H}), \qquad \vec{\nabla}\times E = (\partial_2 E, -\partial_1 E) = \mathrm{Rot}(\vec{\nabla}E),$$

where $\mathrm{Rot}(v_1, v_2) = (v_2, -v_1)$ is the operator that rotates vectors clockwise by $+\pi/2$ in the plane. If we set $\vec{r} = -\mathrm{Rot}(\vec{H})$, then (15) becomes

$$\hat{\imath}kE + \vec{\nabla}\cdot\vec{r} = -J,$$

$$-\hat{\imath}k\vec{r} + \mathrm{Rot}(\mathrm{Rot}(\vec{\nabla}E)) = 0,$$

which, since $\mathrm{Rot}(\mathrm{Rot}(\vec{v})) = -\vec{v}$ (rotation by π), coincides with (1) with $N = 2$ and

$$\phi = E, \tag{16a}$$

$$\vec{u} = \vec{r}, \tag{16b}$$

$$f = -J. \tag{16c}$$

It is this equivalence to which we will later refer when formulating the HDG method for 3D Maxwell's equations.

2

The Discontinuous Petrov-Galerkin method for the Helmholtz equation

We begin this chapter by defining the DPG$_\varepsilon$ method for the Helmholtz equation, which augments the original DPG method of [**14**] by introducing a positive parameter ε in the definition of the test space norm. We then present an analysis of the DPG$_\varepsilon$ method that shows explicitly how the error of the DPG$_\varepsilon$ method depends on ε. The chapter concludes with a description of the lowest order stencil for the case of square two-dimensional elements, which will be used for the dispersion analysis in Section 4.4.

2.1. Derivation of the method

Let Ω_h be a disjoint partitioning of $\Omega \subset \mathbb{R}^N$ into open elements K such that $\overline{\Omega} = \cup_{K\in\Omega_h}\overline{K}$. The shape of the mesh elements in Ω_h is unimportant for now, except that we require their boundaries ∂K to be Lipschitz so that traces make sense. Let

$$V = H(\mathrm{div}, \Omega_h) \times H^1(\Omega_h), \tag{17}$$

where

$$H(\mathrm{div}, \Omega_h) = \{\vec{\tau} : \vec{\tau}|_K \in H(\mathrm{div}, K),\ \forall K \in \Omega_h\},$$

$$H^1(\Omega_h) = \{v : v|_K \in H^1(K),\ \forall K \in \Omega_h\}.$$

Let $A_h : V \to L^2(\Omega)^N \times L^2(\Omega)$ be defined in the same way as A in (2), except the derivatives are taken element by element, i.e., on each $K \in \Omega_h$, we have $A_h(\vec{v}, \eta)|_K = (\hat{\imath}k\vec{v}|_K + \vec{\nabla}\eta|_K, \hat{\imath}k\eta|_K + \vec{\nabla}\cdot\vec{v}|_K)$.

2.1.1. Integration by parts. The following basic formula that we shall use is obtained by integrating by parts each of the derivatives involved:

$$\int_D A(\vec{w}\,,\psi)\cdot\overline{(\vec{v},\eta)} = -\int_D (\vec{w}\,,\psi)\cdot\overline{A(\vec{v},\eta)} + \int_{\partial D}(\vec{w}\,\cdot\hat{\nu})\,\overline{\eta} + \int_{\partial D}\psi\,\overline{(\vec{v}\cdot\hat{\nu})}, \tag{18}$$

for smooth functions $(\vec{w}\,,\psi)$ and $(\vec{v},\eta)$ and domains D with Lipschitz boundary. Above, overlines denote complex conjugations and the integrals use the appropriate Lebesgue measure. Introducing the following abbreviated notations for tuples $\mathsf{w} = (\vec{w}\,,\psi)$ and $\mathsf{v} = (\vec{v},\eta)$,

$$\langle \mathsf{w},\mathsf{v}\rangle_h = \sum_{K\in\Omega_h}\int_K \vec{w}\,\cdot\overline{\vec{v}} + \psi\,\overline{\eta},$$

$$\langle\!\langle \mathsf{w},\mathsf{v}\rangle\!\rangle_h = \sum_{K\in\Omega_h}\int_{\partial K}(\vec{w}\,\cdot\hat{\nu})\,\overline{\eta} + \int_{\partial K}\psi\,\overline{(\vec{v}\cdot\hat{\nu})}\,,$$

we can rewrite (18), applied element by element, as

$$\langle A\mathsf{w},\mathsf{v}\rangle_h = -\langle \mathsf{w}, A_h\mathsf{v}\rangle_h + \langle\!\langle \mathsf{w},\mathsf{v}\rangle\!\rangle_h. \tag{19}$$

By density, (19) holds for all $\mathsf{w}\in H(\mathrm{div},\Omega)\times H^1(\Omega)$ and all $\mathsf{v}\in V$. Then, $\langle\!\langle\cdot,\cdot\rangle\!\rangle_h$ must be interpreted using the appropriate duality pairing as the last term in (19) contains interelement traces on $\partial\Omega_h = \{\partial K : K\in\Omega_h\}$.

It will be convenient to introduce notation for such traces. Let Z denote the space of all functions of the form $\xi\hat{\nu}$ where ξ is in $H^{1/2}(\partial K)$, normed by $\|\xi\hat{\nu}\|_Z = \|\xi\|_{H^{1/2}(\partial K)}$. Let Z' denote the dual space of Z. Now, consider the map $M\vec{q} = (\vec{q}\,\cdot\hat{\nu})\hat{\nu}|_{\partial K}$, defined for smooth functions $\vec{q}$ on $\bar{K}$. Since

$$\int_{\partial K} M\vec{q}\,\cdot\xi\hat{\nu} = \int_{\partial K}(\vec{q}\,\cdot\hat{\nu})\xi$$

(the left and right hand sides extend to duality pairings in Z and $H^{1/2}(\partial K)$, respectively), the standard trace theory implies that M can be extended to a continuous linear operator $M : H(\mathrm{div}, K) \to Z'$. We denote the range of M by $H^{-1/2}(\partial K)\hat{\nu}$ and define

$$\mathrm{tr}_h : H(\mathrm{div}, \Omega) \times H^1(\Omega) \to \prod_K H^{-1/2}(\partial K)\hat{\nu} \times H^{1/2}(\partial K)$$

such that, for any $(\vec{w}, \psi) \in H(\mathrm{div}, \Omega) \times H^1(\Omega)$, the restriction of $\mathrm{tr}_h(\vec{w}, \psi)$ on the boundary of any mesh element ∂K takes the form $((\vec{w} \cdot \hat{\nu})\hat{\nu}|_{\partial K}, \psi|_{\partial K}) \in H^{-1/2}(\partial K)\hat{\nu} \times H^{1/2}(\partial K)$. Throughout this work, functions in $H^{-1/2}(\partial K)\hat{\nu}$ appear together with a dot product with $\hat{\nu}$, so we could equally well consider the standard space $H^{-1/2}(\partial K)$, but the notation simplifies with the former. In particular, with this notation, $\mathrm{tr}_h(\vec{w}, \psi)$ *is a single-valued function on the element interfaces* for smooth $(\vec{w}, \psi)$ on Ω.

2.1.2. An ultraweak formulation. The boundary value problem we wish to approximate is (6). To deal with the Dirichlet boundary condition, we recall the definition of R in (5) and denote the trace space

$$Q = \mathrm{tr}_h(R). \tag{20}$$

To derive the DPG method we use the integration parts by formula (19) to rewrite (6) as

$$-\langle (\vec{u}, \phi), A_h(\vec{v}, \eta)\rangle_h + \langle\!\langle \mathrm{tr}_h(\vec{u}, \phi), (\vec{v}, \eta)\rangle\!\rangle_h = \langle \mathsf{f}, (\vec{v}, \eta)\rangle_h$$

for all $(\vec{v}, \eta) \in V$. Now we let the trace $\mathrm{tr}_h(\vec{u}, \phi)$ be an independent unknown $(\hat{u}, \hat{\phi})$ in Q. Defining the bilinear form $b((\vec{u}, \phi, \hat{u}, \hat{\phi}), (\vec{v}, \eta)) = -\langle (\vec{u}, \phi), A_h(\vec{v}, \eta)\rangle_h + \langle\!\langle (\hat{u}, \hat{\phi}), (\vec{v}, \eta)\rangle\!\rangle_h$, we obtain the ultraweak formulation of [**14**]: Find $\mathsf{u} = (\vec{u}, \phi, \hat{u}, \hat{\phi})$ in

$$U = L^2(\Omega)^N \times L^2(\Omega) \times Q \tag{21}$$

satisfying

$$b(\mathsf{u}, \mathsf{v}) = \langle \mathsf{f}, \mathsf{v}\rangle_h, \qquad \forall \mathsf{v} \in V. \tag{22}$$

The wellposedness of this formulation was proved in [**14**] for the case of impedance boundary conditions. We refer to the solution component $\hat{u}$ as the *numerical flux* and $\hat{\phi}$ as the *numerical trace.*

2.1.3. The DPG$_\varepsilon$ method. Let $U_h \subset U$ be a finite dimensional trial space. The *DPG method* finds u_h in U_h satisfying

$$b(u_h, v_h) = \langle f, v_h \rangle_h, \tag{23}$$

for all v_h in the test space V_h, defined by

$$V_h = TU_h, \tag{24}$$

where $T : U \to V$ is defined by

$$\langle Tw, v \rangle_V = b(w, v), \qquad \forall v \in V, \tag{25}$$

and the V-inner product $\langle \cdot, \cdot \rangle_V$ is the inner product associated with the norm

$$\|v\|_V^2 = \|A_h v\|^2 + \varepsilon^2 \|v\|^2. \tag{26}$$

Here, $\varepsilon > 0$ is an arbitrary scaling parameter. Note that when $\varepsilon = 1$, (26) defines a *graph norm* on V. The case $\varepsilon = 1$, analyzed in [**14**], is the standard DPG method. The general case, which we refer to as the DPG$_\varepsilon$ method, will be analyzed in the next section.

The DPG$_\varepsilon$ method can be reformulated as a residual minimization problem. (All DPG methods with test spaces as in (25) minimize a residual as already pointed out in [**13**].) Letting V' denote the dual space of V, normed with $\|\cdot\|_{V'}$, we define $F \in V'$ by $F(v) = \langle f, v \rangle_h$. Then letting $B : U \to V'$ denote the operator generated by the above-defined $b(\cdot, \cdot)$, i.e., $Bw(v) = b(w, v)$ for all $w \in U$ and $v \in V$, one can immediately see that u_h solves (23) if and only if

$$u_h = \arg\min_{w_h \in U_h} \|B w_h - F\|_{V'}.$$

The norm in the minimization highlights the difference between the DPG method and the standard L^2-based least-squares method.

2.1.4. Inexactly computed test spaces. A basis for the test space V_h defined in (24) can be obtained by applying T to a basis of U_h. One application of T requires solving (25), which although local (calculable element by element), is still an infinite dimensional problem. Accordingly a practical version of the DPG method uses a finite dimensional subspace $V^r \subset V$ and replaces T by $T^r : U \to V^r$ defined by

$$(27) \qquad \langle T^r w, v \rangle_V = b(w, v), \qquad \forall v \in V^r.$$

In computations, we then use, in place of V_h, the inexactly computed test space $V_h^r \equiv T^r U_h$, i.e., the practical DPG method finds u_h^r in U_h satisfying

$$(28) \qquad b(u_h^r, v) = \langle f, v \rangle_h, \qquad \forall v \in V_h^r.$$

For the Helmholtz example with square elements in $\mathbb{R}^2$, which we will use for the dispersion analysis in Section 4.4, we set V^r as follows: Let $\mathcal{Q}_{l,m}$ denote the space of polynomials of degree at most l and m in x_1 and x_2, resp. Let $RT_r \equiv \mathcal{Q}_{r,r-1} \times \mathcal{Q}_{r-1,r}$ denote the Raviart-Thomas subspace of $H(\text{div}, K)$. We set

$$V^r = \{v : v|_K \in RT_r \times \mathcal{Q}_{r,r}\}.$$

Clearly, $V^r \subseteq H(\text{div}, \Omega_h) \times H^1(\Omega_h)$. Using the Fortin operators developed in [**25**], it can be shown that T^r is injective for $r \geq 2$, which implies that (28) yields a positive definite system. However, a complete analysis using [**25**] tracking k and r dependencies remains to be developed, and is not the subject of this dissertation.

2.2. Analysis of the DPG$_\varepsilon$ method

The purpose of this section is to study how the stability constant of the DPG$_\varepsilon$ method (23) depends on ε. The analysis in this section provides the theoretical motivation to introduce the scaling by ε into the DPG setting.

2.2.1. Assumption. The analysis is under the already placed assumption that the boundary value problem (6) is uniquely solvable. For any $(\vec{r}, \psi) \in R$, choosing $f = A(\vec{r}, \psi)$ and applying the inequality (7), we obtain

$$\|(\vec{r}, \psi)\| \leq C(k)\|A(\vec{r}, \psi)\|, \qquad \forall(\vec{r}, \psi) \in R. \tag{29}$$

This is the form in which we will use the assumption.

Note that in the case of the impedance boundary condition, the unique solvability assumption can be easily verified [**36**] for all k. Furthermore, when that boundary condition is imposed, for instance, on the boundary of a convex domain, the estimate (29) is proved in [**14**, Lemmas 4.2 and 4.3] using a result of [**36**] . The resulting constant $C(k)$ is bounded *independently of* k. However, we cannot expect this independence to hold for the Dirichlet boundary condition (4) we are presently considering.

Finally, let us note that the ensuing analysis applies equally well to the impedance boundary condition: We only need to replace the space R considered here by that in [**14**] and assume (29) for all functions in the revised R.

2.2.2. Quasioptimality. It is well-known that if there are positive constants C_1 and C_2 such that

$$C_1 \| v \|_V \le \sup_{w \in U} \frac{|b(w,v)|}{\| w \|_U} \le C_2 \| v \|_V, \qquad \forall v \in V, \tag{30}$$

then a quasioptimal error estimate

$$\| u - u_h \|_U \le \frac{C_2}{C_1} \inf_{w \in U_h} \| u - w \|_U \tag{31}$$

holds. This follows from [**14**, Theorem 2.1], or from the more general result of [**25**, Theorem 2.1], after noting that the following uniqueness condition holds: Any $w \in U$ satisfying $b(w,v) = 0$ for all $v \in V$ vanishes. (Since this uniqueness condition can be proved as in [**14**, Lemma 4.1], we shall not dwell on it here.)

Accordingly, the remainder of this section is devoted to proving (30), tracking the dependence of constants with ε, and using the U-norm we define below. First, let

$$\| (\vec{r}, \psi) \|_R = \frac{1}{\varepsilon} \| A(\vec{r}, \psi) \|.$$

By virtue of (29), this is clearly a norm under which the space R, defined in (5), is complete. The space Q in (20) is normed by the quotient norm, i.e., for any $\hat{q} \in Q$,

$$\| \hat{q} \|_Q = \inf \{ \| r \|_R : \text{ for all } r \in R \text{ such that } \mathrm{tr}_h r = \hat{q} \}.$$

The function in R which achieves the infimum above defines an extension operator $E : Q \to R$ that is a continuous right inverse of tr_h and satisfies

$$\| E \hat{q} \|_R = \| \hat{q} \|_Q. \tag{32}$$

With these notations, we can now define the norm on the trial space by

$$\|(w,\psi,\hat{w},\hat{\psi})\|_U^2 = \|(w,\psi)\|^2 + \|(\hat{w},\hat{\psi})\|_Q^2.$$

The following theorem is proved by extending the ideas in [**14**] to the DPG$_\varepsilon$ method.

Theorem 2.2.1. *Suppose* (29) *holds and let* $c = C(k)\left(C(k)\varepsilon/2 + \sqrt{1+C(k)^2\varepsilon^2/4}\right)$, *where* $C(k)$ *is the constant defined by* (29). *Then the inf-sup condition in* (30) *holds with* $C_1 = 1/\sqrt{1+c\varepsilon}$ *and the continuity condition in* (30) *holds with* $C_2 = \sqrt{1+c\varepsilon}$. *Hence, the DPG solution admits the error estimate*

$$\|\mathsf{u}-\mathsf{u}_h\|_U \le (1+c\varepsilon)\inf_{\mathsf{w}\in U_h}\|\mathsf{u}-\mathsf{w}\|_U.$$

Proof. We first prove the continuity estimate. Let $(\mathsf{w},\hat{\mathsf{q}}) \in U$ and let $\mathsf{v} \in V$. We use the abbreviated notations $\hat{\mathsf{q}} = (\hat{w},\hat{\psi})$, $\mathsf{w} = (w,\psi)$, and $\mathsf{v} = (\vec{v},\eta)$. By (29) and (32),

$$\|E\hat{\mathsf{q}}\| \le C(k)\varepsilon\|\hat{\mathsf{q}}\|_Q, \qquad \|AE\hat{\mathsf{q}}\| = \varepsilon\|\hat{\mathsf{q}}\|_Q. \tag{33}$$

The extension E can be used to rewrite $b((\mathsf{w},\hat{\mathsf{q}}),\mathsf{v}) = -\langle \mathsf{w}, A_h\mathsf{v}\rangle_h + \langle E\hat{\mathsf{q}}, A_h\mathsf{v}\rangle_h + \langle AE\hat{\mathsf{q}},\mathsf{v}\rangle_h$. Then, applying the Cauchy-Schwarz inequality, and using (33), we have

$$\begin{aligned}|b((\mathsf{w},\hat{\mathsf{q}}),\mathsf{v})| &\le \|\mathsf{w}\|\|A_h\mathsf{v}\| + C(k)\varepsilon\|\hat{\mathsf{q}}\|_Q\|A_h\mathsf{v}\| + \varepsilon\|\hat{\mathsf{q}}\|_Q\|\mathsf{v}\| \\ &\le \left(\|\mathsf{w}\|^2 + \|\hat{\mathsf{q}}\|_Q^2\right)^{1/2} t, \qquad (34)\end{aligned}$$

where $t^2 = \|A_h\mathsf{v}\|^2 + \left(C(k)\varepsilon\|A_h\mathsf{v}\| + \varepsilon\|\mathsf{v}\|\right)^2$. With $a = C(k)\varepsilon\|A_h\mathsf{v}\|$ and $b = \varepsilon\|\mathsf{v}\|$ we apply the inequality $(a+b)^2 \le (1+\alpha^2)a^2 + (1+\alpha^{-2})b^2$ to obtain

$$t^2 \le \left(1 + (1+\alpha^2)C(k)^2\varepsilon^2\right)\|A_h\mathsf{v}\|^2 + (1+\alpha^{-2})\varepsilon^2\|\mathsf{v}\|^2,$$

for any $\alpha > 0$. Setting $\alpha^2 = -1/2 + \sqrt{1/4 + C(k)^{-2}\varepsilon^{-2}}$, so that

$$(1 + \alpha^2)C(k)^2\varepsilon^2 = \alpha^{-2} = c\,\varepsilon \tag{35}$$

with c as in the statement of the theorem. Hence, $t^2 \le (1+c\varepsilon)\|v\|_V^2$. Returning to (34),

$$|b((w, \hat{q}), v)| \le C_2\|(w, \hat{q})\|_U\|v\|_V.$$

with $C_2 = \sqrt{1 + c\varepsilon}$. This verifies the upper inequality of (30).

To prove the lower inequality of (30), let r be the unique function in R satisfying $Ar = v$ for any given $v \in V$. Then, by (29),

$$\|r\| \le C(k)\|v\|. \tag{36}$$

Also, since $\|Ar\| = \|v\|$, letting $\hat{r} = \mathrm{tr}_h r$, we have

$$\|\hat{r}\|_Q = \frac{1}{\varepsilon}\|AE\hat{r}\| \le \frac{1}{\varepsilon}\|Ar\| = \frac{1}{\varepsilon}\|v\|. \tag{37}$$

By (19), we have $\langle Ar, v\rangle_h = -\langle r, A_h v\rangle_h + \langle\!\langle \hat{r}, v\rangle\!\rangle_h$, so

$$\|v\|_V^2 = \varepsilon^2\|v\|^2 + \|A_h v\|^2 = \varepsilon^2\, b((z, \hat{r}), v), \tag{38}$$

where $z = r - \varepsilon^{-2}A_h v$, a function that can be bounded using (36), as follows:

$$\begin{aligned}\|z\|^2 &\le (1 + \alpha^2)\|r\|^2 + (1 + \alpha^{-2})\varepsilon^{-4}\|A_h v\|^2\\ &\le (1 + \alpha^2)C(k)^2\|v\|^2 + (1 + \alpha^{-2})\varepsilon^{-4}\|A_h v\|^2,\end{aligned}$$

for any $\alpha > 0$. Choosing α as in (35) and using (36)–(37),

$$\begin{aligned}\varepsilon^4 \|(z, \hat{r})\|_U^2 &= \varepsilon^4 \|z\|^2 + \varepsilon^4 \|\hat{r}\|_Q^2 \\ &\le \left(1 + (1+\alpha^2) C(k)^2 \varepsilon^2\right) \varepsilon^2 \|v\|^2 + (1 + \alpha^{-2}) \|A_h v\|^2 \\ &\le (1 + c\varepsilon) \left(\varepsilon^2 \|v\|^2 + \|A_h v\|^2\right). \qquad (39)\end{aligned}$$

Returning to (38), we now have

$$\|v\|_V^2 = \frac{b((z,\hat{r}), v)}{\|(z,\hat{r})\|_U} \varepsilon^2 \|(z,\hat{r})\|_U \le \left(\sup_{x \in U} \frac{|b(x,v)|}{\|x\|_U} \right) \sqrt{1 + c\varepsilon}\, \|v\|_V$$

by virtue of (39), verifying the lower inequality of (30) with $C_1 = 1/\sqrt{1 + c\varepsilon}$. □

Remark 2.2.2. Although we presented the above result only for the Helmholtz equation, the ideas apply more generally. It seems possible to prove a similar result abstractly, e.g., using the abstract setting in [**6**], for any DPG application that uses a scaled graph norm analogous to (26) (with the wave operator A_h replaced by suitable others).

2.2.3. Discussion. Theorem 2.2.1 shows that the use of the ε-scaling in the test norm can ameliorate some stability problems, e.g., those that can arise from large $C(k)$.

Observe that the best possible value for the constant C_2/C_1 in (31) is 1. Indeed, if C_2/C_1 equals 1, then the computed solution u_h coincides with the best approximation to u from U_h. Theorem 2.2.1 shows that the quasioptimality constant of the DPG$_\varepsilon$ method approaches the ideal value of 1 as $\varepsilon \to 0$.

However, since the norms depend on ε, we must further examine the components of the error separately, by defining

$$e^2 = \|\vec{u} - \vec{u}_h\|^2 + \|\phi - \phi_h\|^2, \tag{40a}$$

$$\hat{e}^2 = \|AE(\hat{u} - \hat{u}_h, \hat{\phi} - \hat{\phi}_h)\|^2. \tag{40b}$$

The estimate of Theorem 2.2.1 implies that

$$e^2 + \frac{\hat{e}^2}{\varepsilon^2} \le (1 + c\,\varepsilon)^2 \left(a^2 + \frac{\hat{a}^2}{\varepsilon^2}\right) \tag{41}$$

where a and $\hat{a}$ are the best approximation errors defined by

$$\begin{aligned} a^2 &= \inf_{(\vec{w},\psi,0,0)\in U_h} \|\vec{u} - \vec{w}\|^2 + \|\phi - \psi\|^2, \\ \hat{a}^2 &= \inf_{(0,0,\hat{w},\hat{\psi})\in U_h} \|AE(\hat{u} - \hat{w}, \hat{\phi} - \hat{\psi})\|^2. \end{aligned} \tag{42}$$

Note that E is independent of ε.

We want to compare the error bounds for the numerical fluxes and traces in the $\varepsilon = 1$ case with the case of $0 < \varepsilon \ll 1$. To distinguish these cases we will denote the error defined in (40b) by $\hat{e}_1$ when $\varepsilon = 1$. Clearly, (41) implies

$$\hat{e}_1^2 \le (1 + c)^2 \left(a^2 + \hat{a}^2\right). \tag{43}$$

For the other case, (41) implies, after multiplying through by ε^2,

$$\hat{e}^2 \le (1 + c\,\varepsilon)^2 \left(\varepsilon^2 a^2 + \hat{a}^2\right).$$

Comparing this with (43), and noting that a and $\hat{a}$ remain the same for different ε, we find that the DPG$_\varepsilon$ errors for fluxes and traces admit a *better bound for smaller* ε in an ε-independent norm. Whether the actually observed numerical error improves,

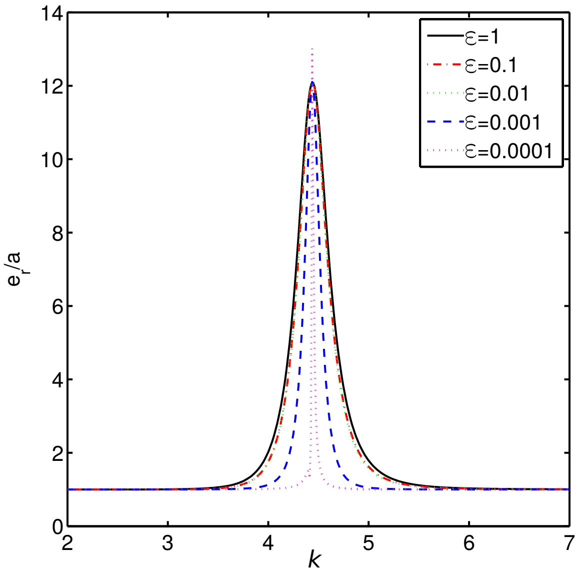


FIGURE 2.1. The regularizing effect of DPG$_\varepsilon$ method as seen from a plot of the ratio e_r/a near a resonance.

will be investigated through the dispersion analysis presented in a later section, as well as in the next subsection.

2.2.4. Numerical illustration. Theorem 2.2.1 partially explains a numerical observation we now report. We implemented the DPG$_\varepsilon$ method by setting the parameter $r = 3$ (see § 2.1.4) and computed $\boldsymbol{u}_h^r = (\vec{u}_h^r, \phi_h^r, \hat{u}_h^r, \hat{\phi}_h^r)$. In analogy with (40), define the discretization errors e_r and $\hat{e}_r$ by $e_r^2 = \|\vec{u} - \vec{u}_h^r\|^2 + \|\phi - \phi_h^r\|^2$ and $\hat{e}_r^2 = \|AE(\hat{u} - \hat{u}_h^r, \hat{\phi} - \hat{\phi}_h^r)\|^2$. Although Theorem 2.2.1 suggests an investigation of

$$\frac{\|\boldsymbol{u} - \boldsymbol{u}_h^r\|_U}{\inf_{\boldsymbol{w} \in U_h} \|\boldsymbol{u} - \boldsymbol{w}\|_U} = \left(\frac{e_r^2 + (\hat{e}_r/\varepsilon)^2}{a^2 + (\hat{a}/\varepsilon)^2} \right)^{1/2},$$

due to the difficulty of applying the extension operator E in practice, we have investigated the ratio e_r/a as a function of k. Recall that a is the $L^2(\Omega)$ best approximation error defined in (42), so e_r/a measures how close the discretization errors are to the best possible.

For a range of wavenumbers k, we chose the data $\mathsf{f} = (\vec{0}, f)$ so that the exact solution to (6) on the unit square would be $(\vec{u}\,, \phi) = (\frac{i}{k}\vec{\nabla}\phi, \phi)$, with $\phi = x(1-x)y(1-y)$. Each resulting boundary value problem was then solved using the DPG$_\varepsilon$ method with $\varepsilon = 10^{-n}, n = 0, 1, 2, 3, 4$, on a fixed mesh of $h = 1/16$ and the corresponding discretization errors e_r were collected.

The resulting ratios e_r/a are plotted as a function of k in Figure 2.1 for a few ε values. First of all, observe that the graph of the ratio begins close to the optimal value of one for all ε values in the figure. Next, observe that the ratio spikes up as k approaches the exact resonance value $k = \pi\sqrt{2} \approx 4.44$, where $C(k)$ is infinity. It is interesting to look at the points near (but not at) the resonance. Observe that as ε is decreased, the DPG$_\varepsilon$ method exhibits a "regularizing" effect at points near the resonance: E.g., at $k = 5$, the values of e_r/a are closer to 1 for smaller ε. It therefore seems advantageous to use smaller ε for problems near resonance.

The theoretical explanation for this numerical observation would be complete (by virtue of Theorem 2.2.1), if we had computed using the exact DPG test spaces ($r = \infty$), instead of the inexactly computed spaces ($r = 3$). Certain discrete effects arising due to this inexact computation of test spaces will be presented in Section 4.4.

2.3. Lowest order stencil for the DPG$_\varepsilon$ method

We consider the example of square two-dimensional elements, which will be used for the dispersion analysis in Section 4.4. The lowest order case of the DPG method is obtained using $Q(\partial K) = \{(\hat{w}, \hat{\psi}) : \hat{w}$ is constant on each edge of ∂K, $\hat{\psi}$ is linear on each edge of ∂K, and $\hat{\psi}$ is continuous on $\partial K\}$. Let $S(K) = \{(\vec{w}, \psi) :\ \vec{w}$ and ψ are constant (vector and scalar, resp.) functions on $K\}$. We consider the DPG$_\varepsilon$ method using the lowest order global trial space

$$U_h = S_h \times Q_h,$$

where $Q_h = \{\hat{\mathsf{r}} : \hat{\mathsf{r}}|_{\partial K} \in Q(\partial K)$ for all mesh elements $K\}$ and $S_h = \{\mathsf{w} : \mathsf{w}|_K \in S(K)$ for all mesh elements $K\}$.

Let $\hat{\chi}_e$ denote the indicator function of an edge e. If a denotes a vertex of the square element K, let ϕ_a denote the bilinear function that equals one at a and equals zero at the other three vertices of K. Let $\hat{\phi}_a = \phi_a|_{\partial K}$. The collection of eight functions of the form $(0, \hat{\phi}_a)$ and $(\hat{\chi}_e, 0)$, one for each vertex, and one for each edge of K, forms a basis for $Q(\partial K)$. We distinguish between the horizontal and vertical edges, because the unknowns there approximate different components of the velocity $\vec{u}$. Accordingly, we will denote by $\hat{\chi}_e^h$ the indicator function of a horizontal edge and by $\hat{\chi}_e^v$ the indicator function of a vertical edge.

We now define the local 11×11 DPG matrix for a single element using the basis for $S(K) \times Q(\partial K)$ obtained by supplementing the basis for $Q(\partial K)$ described above with the basis for $S(K)$ consisting of three indicator functions. Enumerating these basis functions as e_i, $i = 1, \ldots 11$, the local DPG matrix $B \equiv B(k, \varepsilon)$ is defined by

$$B_{ij} = b(e_j, T^r e_i), \tag{44}$$

where T^r is as defined in (27). The basis for the space V^r is chosen such that each basis function is supported on one element. In our computations, we did not specialize the basis for V^r any further so, to overcome round-off problems due to ill-conditioned local matrices, we resorted to high precision arithmetic for these local computations.

Given a square element with sides of length h parallel to the axes, B can be computed by mapping to the reference element $\breve{K} = [0, 1]^2$. For any function v on

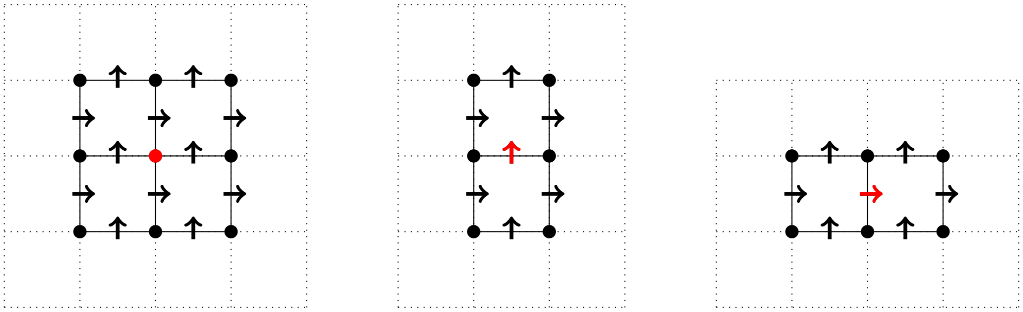
K (resp., ∂K), let the mapped function $\breve{\mathsf{v}}$ on $\breve{K}$ (resp., $\partial\breve{K}$) be defined by $\breve{\mathsf{v}}(\breve{x}) = \mathsf{v}(h\breve{x}+\vec{b}_K)$, with $\vec{b}_K$ such that that $K-\vec{b}_K = h\breve{K}$. The mapped functions $\breve{e}_i$ are precisely the basis vectors of $S(\breve{K})\times Q(\partial\breve{K})$ used when applying (44) to compute $\breve{B}(k,\varepsilon)$, the local DPG matrix for $\breve{K}$. By a change of variables, it is easy to see that

$$B(k,\varepsilon) = h^2\breve{B}(kh,\varepsilon h). \tag{45}$$

Thus we may compute local DPG matrices by scaling the local DPG matrix for the fixed reference element $\breve{K}$ obtained using the *normalized wavenumber* kh and scaling parameter εh. It is enough to compute the element matrix $\breve{B}$ using high precision arithmetic for the ensuing dispersion analysis.

Next, we eliminate the three interior variables of $S(K)$ and consider the *condensed* 8×8 local stiffness matrix for the variables in $Q(\partial K)$. At this stage it will be useful to classify these eight variables (unknowns) into three categories: (1) Unknowns at vertices a (which are the coefficients multiplying the basis function $\hat{\phi}_a$) denoted by "●", (2) unknowns on horizontal edges (coefficients multiplying $\hat{\chi}_e^h$) denoted by "↑", and (3) unknowns on vertical edges (coefficients multiplying the corresponding $\hat{\chi}_e^v$) denoted by "→". The normal vectors on all horizontal and vertical edges are fixed to be $(0,1)$ and $(1,0)$, respectively, corresponding to the direction of the above-indicated arrows.

Now suppose the mesh is a uniform mesh of congruent square elements. Assembling the above-described condensed 8×8 element matrices on such a mesh, we obtain a global system where the interior variables are all condensed out. The resulting equations can be represented using the stencils in Figure 2.2. A row of the matrix system corresponding to an unknown of the type "●" has 21 nonzero entries corresponding to unknowns of all three types, as shown in Figure 2.2a. Similarly, the unknowns of



(A) 21-point stencil

(B) 13-point stencil

(C) 13-point stencil

FIGURE 2.2. Stencils

the type "↑" and "→" connect to other unknowns in the 13-point stencils depicted in Figures 2.2b and 2.2c, respectively.

3

The Hybrid Discontinuous Galerkin method

In this chapter, we begin by describing the HDG methods. We set the stage for our study by showing that the commonly chosen HDG stabilization parameter values for elliptic problems are not appropriate for all complex wavenumbers. This has practical implications, since complex wavenumbers arise when modeling absorbing materials with complex refractive indices as in, for example, the nanogap problem of Chapter 5. In Section 3.2, we discover a constraint on the stabilization parameter, dependent on the wavenumber, that guarantees unique solvability of both the global and the local HDG problems.

3.1. The Helmholtz and Maxwell formulations

We borrow the basic methodology for constructing HDG methods from [**10**] and apply it to the time-harmonic Helmholtz and Maxwell equations (written as first order systems). While doing so, we set up the notations used throughout, compare the formulation we use with other existing works, and show that for complex wavenumbers there are stabilization parameters that will cause the HDG method to fail.

3.1.1. Undesirable stabilization parameters for the Helmholtz system. We begin by considering the lowest order HDG system for Helmholtz equation. Let k be a complex number. Consider Equation (1), the Helmholtz system with homogeneous Dirichlet boundary conditions, on $\Omega \subset \mathbb{R}^2$. Let $\mathcal{T}_h$ denote a square or triangular mesh of disjoint elements K, so $\overline{\Omega} = \cup_{K \in \mathcal{T}_h} \overline{K}$, and let $\mathcal{F}_h$ denote the collection of edges. The HDG method produces an approximation $(\vec{u}_h, \phi_h, \hat{\phi}_h)$ to the exact solution $(\vec{u}, \phi, \hat{\phi})$ of Equation (1), where $\hat{\phi}$ denotes the trace of ϕ on the collection of element boundaries $\partial\mathcal{T}_h$. The HDG solution $(\vec{u}_h, \phi_h, \hat{\phi}_h)$ is in the finite dimensional space $\mathcal{V}_h \times \mathcal{W}_h \times \mathcal{M}_h$ defined by

$$\begin{aligned}
\mathcal{V}_h &= \{\vec{v} \in (L^2(\Omega))^2 \ : \ \vec{v}|_K \in \mathcal{V}(K), \ \forall K \in \mathcal{T}_h\} \\
\mathcal{W}_h &= \{\psi \in L^2(\Omega) \ : \ \psi|_K \in \mathcal{W}(K), \ \forall K \in \mathcal{T}_h\} \\
\mathcal{M}_h &= \{\hat{\psi} \in L^2(\bigcup_{F \in \mathcal{F}_h} F) \ : \ \hat{\psi}|_F \in \mathcal{M}(F), \ \forall F \in \mathcal{F}_h \text{ and } \hat{\psi}|_{\partial\Omega} = 0\},
\end{aligned}$$

with polynomial spaces $\mathcal{V}(K)$, $\mathcal{W}(K)$, and $\mathcal{M}(F)$ specified differently depending on element type:

Triangles	Squares
$\mathcal{V}(K) = (\mathcal{P}_p(K))^2$	$\mathcal{V}(K) = (\mathcal{Q}_p(K))^2$
$\mathcal{W}(K) = \mathcal{P}_p(K)$	$\mathcal{W}(K) = \mathcal{Q}_p(K)$
$\mathcal{M}(F) = \mathcal{P}_p(F)$	$\mathcal{M}(F) = \mathcal{P}_p(F).$

Here, for a given domain D, $\mathcal{P}_p(D)$ denotes polynomials of degree at most p, and $\mathcal{Q}_p(D)$ denotes polynomials of degree at most p in each variable.

The HDG solution solves

$$\sum_{K \in \mathcal{T}_h} \hat{\imath}k(\vec{u}_h, \vec{v})_K - (\phi_h, \vec{\nabla} \cdot \vec{v})_K + \langle \hat{\phi}_h, \vec{v} \cdot \hat{\nu} \rangle_{\partial K} = 0, \tag{46a}$$

$$\sum_{K \in \mathcal{T}_h} -(\vec{\nabla} \cdot \vec{u}_h, \psi)_K + \langle \tau \hat{\phi}_h, \psi \rangle_{\partial K} - \langle \tau \phi_h, \psi \rangle_{\partial K} - \hat{\imath}k(\phi_h, \psi)_K = -(f, \psi)_\Omega, \tag{46b}$$

$$\sum_{K \in \mathcal{T}_h} \langle \vec{u}_h \cdot \hat{\nu} + \tau(\phi_h - \hat{\phi}_h), \hat{\psi} \rangle_{\partial K} = 0, \tag{46c}$$

for all $\vec{v} \in \mathcal{V}_h$, $\psi \in \mathcal{W}_h$, and $\hat{\psi} \in \mathcal{M}_h$. The last equation enforces the conservativity of the numerical flux

$$\hat{u}_h \cdot \hat{\nu} = \vec{u}_h \cdot \hat{\nu} + \tau(\phi_h - \hat{\phi}_h). \tag{47}$$

The stabilization parameter τ is assumed to be constant on each ∂K. We are interested in how the choice of τ in relation to k affects the method, especially when k is complex valued. Comparisons of this formulation with other HDG formulations for Helmholtz equations in the literature are summarized in Table 3.1.

One of the main reasons to use an HDG method is that all interior unknowns $(\vec{u}_h, \phi_h)$ can be eliminated to get a global system for solely the interface unknowns $(\hat{\phi}_h)$. This is possible whenever the local system

$$\hat{\imath}k(\vec{u}_h, \vec{v})_K - (\phi_h, \vec{\nabla}\cdot \vec{v})_K = -\langle \hat{\phi}_h, \vec{v} \cdot \hat{\nu}\rangle_{\partial K}, \qquad \forall \vec{v} \in \mathcal{V}(K), \tag{48a}$$

$$-(\vec{\nabla}\cdot \vec{u}_h, \psi)_K - \langle \tau\phi_h, \psi\rangle_{\partial K} - \hat{\imath}k(\phi_h, \psi)_K = -\langle \tau\hat{\phi}_h, \psi\rangle_{\partial K}, \qquad \forall \psi \in \mathcal{W}(K), \tag{48b}$$

is uniquely solvable. (For details on this elimination and other perspectives on HDG methods, see [**10**].) In the lowest order ($p = 0$) case, on a square element K of side length h, if we use a basis in the following order

$$\vec{u}_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad \vec{u}_2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad \phi_1 = 1, \qquad \text{on } K,$$

then the element matrix for the system (48) is

$$M = \begin{bmatrix} \hat{\imath}k\,h^2 & 0 & 0 \\ & \hat{\imath}k\,h^2 & 0 \\ 0 & 0 & -4\,h\tau - \hat{\imath}k\,h^2 \end{bmatrix}.$$

This shows that if

$$4\tau = -\hat{\imath}kh, \tag{49}$$

then M is singular, and so the HDG method will fail. *The usual recipe of choosing $\tau = 1$ is therefore inappropriate when k is complex valued.*

3.1.2. Intermediate case of the 2D Maxwell system. Recalling (16), the HDG method for Helmholtz equation immediately gives an HDG method for the 2D Maxwell system (15). We thus conclude that there exist stabilization parameters that will cause the HDG system for 2D Maxwell system to fail.

To examine this 2D HDG method, if we let $\vec{H}_h$ and E_h denote the HDG approximations for $R\vec{r}$ and $\mathcal{E}$, respectively, then the HDG system (46) with $\vec{u}_h$ and ϕ_h replaced by $-R\vec{H}_h$ and E_h, respectively, gives

$$\begin{aligned}
\sum_{K\in\mathcal{T}_h} \qquad & -(E_h, \vec{\nabla}\times\vec{w})_K + \langle \hat{E}_h, \hat{\nu}\times\vec{w}\rangle_{\partial K} - \hat{\imath}k(\vec{H}_h, \vec{w})_K = 0,\\
\sum_{K\in\mathcal{T}_h} \qquad & \hat{\imath}k(E_h, \psi)_K - (\vec{\nabla}\times\vec{H}_h, \psi)_K + \langle \tau(E_h - \hat{E}_h),\ \psi\rangle_{\partial K} = -(\vec{J}, \psi)_\Omega,\\
\sum_{K\in\mathcal{T}_h} \qquad & \langle \widehat{R\vec{H}_h}\cdot\hat{\nu},\ \hat{\psi}\rangle_{\partial K} = 0,
\end{aligned}$$

for all $\vec{w} \in R(V_h), \psi \in W_h$ and $\hat{\psi} \in M_h$. We have used the fact that $-(R\vec{H})\cdot\hat{\nu} = \vec{H}\cdot\vec{t}$, where $\vec{t} = R\hat{\nu}$ the tangent vector, and we have used the 2D cross product defined by $\vec{v}\times\hat{\nu} = \vec{v}\cdot\vec{t}$. In particular, the numerical flux prescription (47) implies

$$-\widehat{R\vec{H}_h}\cdot\hat{\nu} = -R\vec{H}_h\cdot\hat{\nu} + \tau(E_h - \hat{E}_h),$$

where $\widehat{R\vec{H}_h}$ denotes the numerical trace of $R\vec{H}_h$. We rewrite this in terms of $\vec{H}_h$ and E_h, to obtain

$$\hat{H}_h\cdot\vec{t} = \vec{H}_h\cdot\vec{t} + \tau(E_h - \hat{E}_h).$$

One may rewrite this again, as

$$\hat{H}_h \times \hat{\nu} = \vec{H}_h \times \hat{\nu} + \tau(E_h - \hat{E}_h). \tag{50}$$

This expression is notable because it will help us consistently transition the numerical flux prescription from the Helmholtz to the full 3D Maxwell case discussed next. A comparison of this formula with those in the existing literature is included in Table 3.1.

3.1.3. The 3D Maxwell formulation. For this problem, $\Omega \subset \mathbb{R}^3$, $\mathcal{T}_h$ denotes a cubic or tetrahedral mesh, and $\mathcal{F}_h$ denotes the collection of mesh faces. The HDG method approximates the exact solution $(\vec{E}, \vec{H}, \hat{E})$ of Equation (14), where $\hat{E}$ denotes the tangential component of the trace of $\vec{E}$ on element boundaries. The HDG approximation is $(\vec{E}_h, \vec{H}_h, \hat{E}_h) \in \mathcal{Y}_h \times \mathcal{Y}_h \times \mathcal{J}_h$. The discrete spaces are defined by

$$\mathcal{Y}_h = \{\vec{v} \in (L^2(\Omega))^2 \ : \ \vec{v}|_K \in \mathcal{Y}(K), \ \forall K \in \mathcal{T}_h\}$$

$$\mathcal{J}_h = \{\hat{\eta} \in (L^2(\mathcal{F}_h))^3 \ : \ \hat{\eta}|_F \in \mathcal{J}(F), \ \forall F \in \mathcal{F}_h \text{ and } \hat{\eta}|_{\partial\Omega} = \vec{0}\},$$

with polynomial spaces $\mathcal{Y}(K)$ and $\mathcal{J}(F)$ specified by:

Tetrahedra	Cubes
$\mathcal{Y}(K) = (\mathcal{P}_p(K))^3$	$\mathcal{Y}(K) = (\mathcal{Q}_p(K))^3$
$\mathcal{J}(F) = \{\hat{\eta} \in (\mathcal{P}_p(F))^3 \ : \ \hat{\eta} \cdot \hat{\nu} = 0\}$	$\mathcal{J}(F) = \{\hat{\eta} \in (\mathcal{P}_p(F))^3 \ : \ \hat{\eta} \cdot \hat{\nu} = 0\}$

Reference	Their notations and equations	Connection to our formulation
[**11**] Helmholtz case	$\vec{q}_{[\mathbf{11}]} + \vec{\nabla} u_{[\mathbf{11}]} = \vec{0}$ $\vec{\nabla}\cdot \vec{q}_{[\mathbf{11}]} - k^2 u_{[\mathbf{11}]} = 0$ $\hat{q}_{[\mathbf{11}]} \cdot \vec{n} = \vec{q}_{[\mathbf{11}]} \cdot \vec{n} + \hat{\imath}\tau_{[\mathbf{11}]}(u_{[\mathbf{11}]} - \hat{u}_{[\mathbf{11}]})$	$\tau_{[\mathbf{11}]} = k\,\tau$ $\hat{\imath} k u_{[\mathbf{11}]} = \phi$ $\vec{q}_{[\mathbf{11}]} = \vec{u}$
[**27**] Helmholtz case	$\hat{\imath} k \vec{q}_{[\mathbf{27}]} + \vec{\nabla} u_{[\mathbf{27}]} = \vec{0}$ $\hat{\imath} k u_{[\mathbf{27}]} + \vec{\nabla}\cdot \vec{q}_{[\mathbf{27}]} = 0$ $\hat{q}_{[\mathbf{27}]} \cdot \vec{n} = \vec{q}_{[\mathbf{27}]} \cdot \vec{n} + \tau_{[\mathbf{27}]}(u_{[\mathbf{27}]} - \hat{u}_{[\mathbf{27}]})$	$\tau_{[\mathbf{27}]} = \tau$ $u_{[\mathbf{27}]} = \phi$ $\vec{q}_{[\mathbf{27}]} = \vec{u}$
[**33**] 2D Maxwell case	$\hat{\imath}\omega_{[\mathbf{33}]}\varepsilon_r E_{[\mathbf{33}]} - \nabla\times \vec{H}_{[\mathbf{33}]} = 0$ $\hat{\imath}\omega_{[\mathbf{33}]}\mu_r \vec{H}_{[\mathbf{33}]} + \vec{\nabla}\times E_{[\mathbf{33}]} = \vec{0}$ $\hat{H}_{[\mathbf{33}]} = \vec{H}_{[\mathbf{33}]} + \tau_{[\mathbf{33}]}(E_{[\mathbf{33}]} - \hat{E}_{[\mathbf{33}]})\vec{t}$	$\tau_{[\mathbf{33}]} = \sqrt{\frac{\varepsilon_r}{\mu_r}}\tau$ $\omega_{[\mathbf{33}]} = \omega\sqrt{\varepsilon_0\mu_0}$ $E_{[\mathbf{33}]} = \frac{1}{\sqrt{\varepsilon_r}} E$ $\vec{H}_{[\mathbf{33}]} = \frac{1}{\sqrt{\mu_r}} \vec{H}$
[**38**] Maxwell case	$\mu \vec{w}_{[\mathbf{38}]} - \vec{\nabla}\times \vec{u}_{[\mathbf{38}]} = \vec{0}$ $\vec{\nabla}\times \vec{w}_{[\mathbf{38}]} - \varepsilon\omega^2 \vec{u}_{[\mathbf{38}]} = \vec{0}$ $\hat{w}_{[\mathbf{38}]} = \vec{w}_{[\mathbf{38}]} + \tau_{[\mathbf{38}]}(\vec{u}_{[\mathbf{38}]} - \hat{u}_{[\mathbf{38}]}) \times \hat{\nu}$	$\tau_{[\mathbf{38}]} = \hat{\imath}\sqrt{\frac{\varepsilon\omega^2}{\mu}}\,\tau$ $\mu\vec{w}_{[\mathbf{38}]} = -\hat{\imath} k \vec{H},$ $\vec{u}_{[\mathbf{38}]} = \vec{E}$

Table 3.1. Comparison with some HDG formulations in the literature. Notations in the indicated external references are used after subscripting them by the reference number. Notations without subscripts are those defined in this work.

Our HDG method for (14) is

$$
\begin{array}{lrl}
\displaystyle\sum_{K\in\mathcal{T}_h} & \hat{\imath} k(\vec{E}_h, \vec{v})_K - (\vec{\nabla}\times \vec{H}_h, \vec{v})_K + \langle (\hat{H} - H) \times \hat{\nu},\ \vec{v}\rangle_{\partial K} = -(\vec{J}, \vec{v})_\Omega, & \forall \vec{v} \in \mathcal{Y}_h, \\
\displaystyle\sum_{K\in\mathcal{T}_h} & -(\vec{E}_h, \vec{\nabla}\times \vec{w})_K + \langle \hat{E}_h, \hat{\nu} \times \vec{w}\rangle_{\partial K} - \hat{\imath} k(\vec{H}_h, \vec{w})_K = 0, & \forall \vec{w} \in \mathcal{Y}_h, \\
\displaystyle\sum_{K\in\mathcal{T}_h} & \langle \hat{H} \times \hat{\nu},\ \hat{w}\rangle_{\partial K} = 0, & \forall \hat{w} \in \mathcal{J}_h,
\end{array}
$$

where, in analogy with (50), we now set numerical flux by

$$\hat{H} \times \hat{\nu} = \vec{H}_h \times \hat{\nu} + \tau(\vec{E}_h - \hat{E}_h)_t, \tag{51}$$

where $(\vec{E}_h - \hat{E}_h)_t$ denotes the tangential component, or equivalently

$$\hat{H} \times \hat{\nu} = \vec{H}_h \times \hat{\nu} + \tau(\hat{\nu} \times (\vec{E}_h - \hat{E}_h)) \times \hat{\nu}.$$

We noted that the 2D system (15) is obtained from the 3D Maxwell system (14) by assuming symmetry in z-direction. Hence, for consistency between 2D and 3D formulations, we should have the same form for the numerical flux prescriptions in 2D and 3D.

The HDG method is then equivalently written as

$$\sum_{K \in \mathcal{T}_h} \hat{\imath}k(\vec{E}_h, \vec{v})_K - (\vec{\nabla}\times \vec{H}_h, \vec{v})_K + \langle \tau(\vec{E}_h - \hat{E}_h) \times \hat{\nu},\ \vec{v} \times \hat{\nu} \rangle_{\partial K} = -(\vec{J}, \vec{v})_\Omega, \tag{52a}$$

$$\sum_{K \in \mathcal{T}_h} -(\vec{E}_h, \vec{\nabla}\times \vec{w})_K + \langle \hat{E}_h, \hat{\nu} \times \vec{w} \rangle_{\partial K} - \hat{\imath}k(\vec{H}_h, \vec{w})_K = 0, \tag{52b}$$

$$\sum_{K \in \mathcal{T}_h} \langle \vec{H}_h + \tau\, \hat{\nu} \times (\vec{E}_h - \hat{E}_h),\ \hat{w} \times \hat{\nu} \rangle_{\partial K} = 0, \tag{52c}$$

for all $\vec{v}, \vec{w} \in \mathcal{Y}_h$, and $\hat{w} \in \mathcal{J}_h$. For comparison with other existing formulations, see Table 3.1.

Again, let us look at the solvability of the *local element problem*

$$\hat{\imath}k(\vec{E}_h, \vec{v})_K - (\vec{\nabla}\times \vec{H}_h, \vec{v})_K + \langle \tau \vec{E}_h \times \hat{\nu},\ \vec{v} \times \hat{\nu} \rangle_{\partial K} = \langle \tau \hat{E}_h \times \hat{\nu},\ \vec{v} \times \hat{\nu} \rangle_{\partial K}, \tag{53a}$$

$$-(\vec{E}_h, \vec{\nabla}\times \vec{w})_K - \hat{\imath}k(\vec{H}_h, \vec{w})_K = -\langle \hat{E}_h, \hat{\nu} \times \vec{w} \rangle_{\partial K}, \tag{53b}$$

for all $\vec{v}, \vec{w} \in Y(K)$. In the lowest order ($p = 0$) case, on a cube element K of side length h, if we use a basis in the following order

$$(54) \qquad \vec{E}_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \quad \vec{E}_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \quad \vec{E}_3 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \quad \vec{H}_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \quad \vec{H}_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \quad \vec{H}_3 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix},$$

then the 6×6 element matrix for the system (53) is

$$M = \begin{bmatrix} (4h^2\tau + \hat{\imath}kh^3)I_3 & 0 \\ 0 & -(\hat{\imath}kh^3)I_3 \end{bmatrix},$$

where I_3 denotes the 3×3 identity matrix. Again, exactly as in the Helmholtz case – cf. (49) – we find that if

$$(55) \qquad 4\tau = -\hat{\imath}kh,$$

then the local static condensation required in the HDG method will fail in the Maxwell case also.

3.1.4. Behavior on tetrahedral meshes. For the lowest order ($p = 0$) case on a tetrahedral element, just as for the cube element described above, there are bad stabilization parameter values. Consider, for example, the tetrahedral element of size h defined by

$$(56) \qquad K = \{\vec{x} \in \mathbb{R}^3 \; : \; x_j \geq 0 \; \forall j, \; x_1 + x_2 + x_3 \leq h\},$$

with a basis ordered as in (54). The element matrix for the system (53) is then

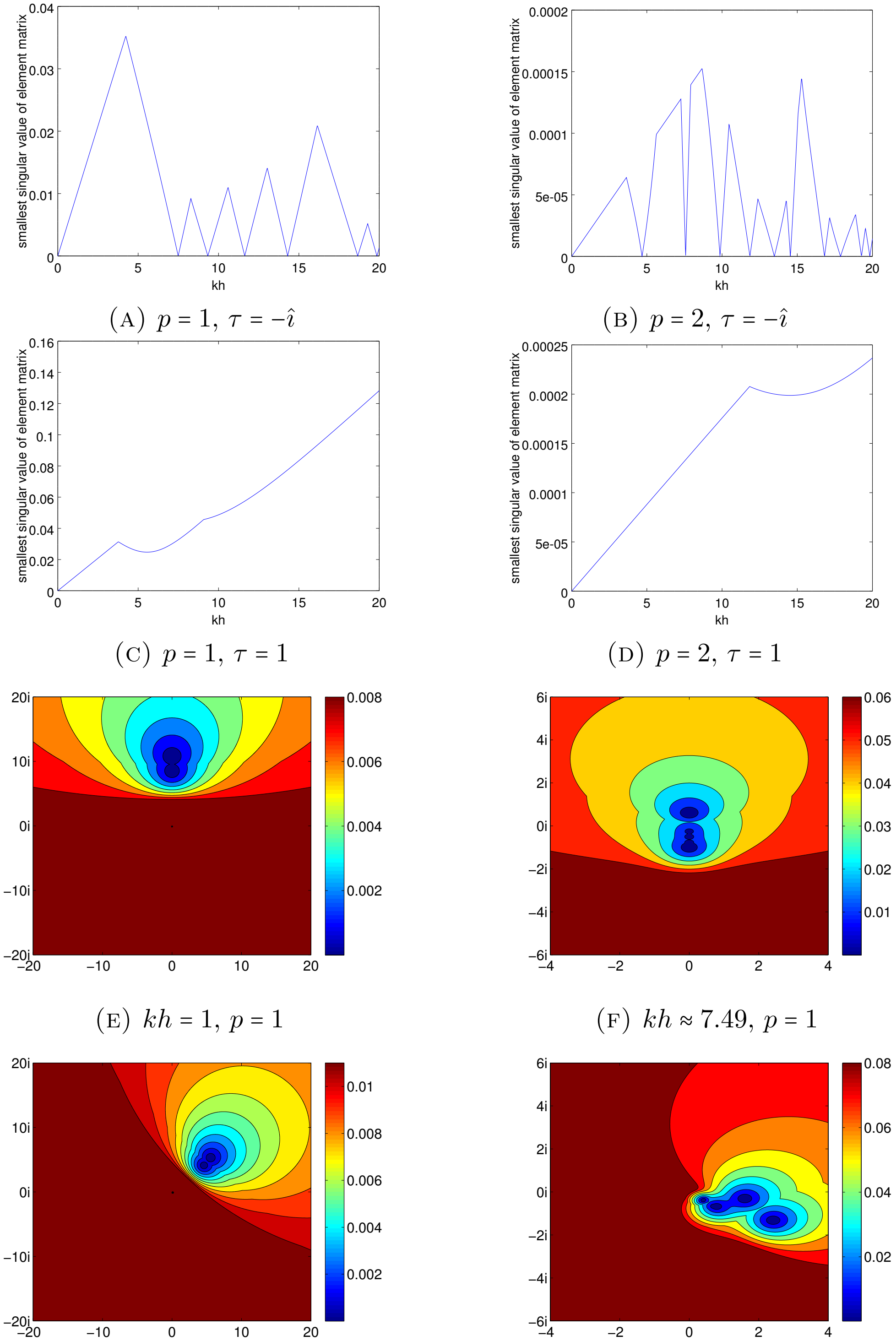
$$M = \frac{1}{6}\begin{bmatrix} (2\sqrt{3}+6)h^2\tau + \hat{\imath}kh^3 & -\sqrt{3}h^2\tau & -\sqrt{3}h^2\tau & 0 & 0 & 0 \\ -\sqrt{3}h^2\tau & (2\sqrt{3}+6)h^2\tau + \hat{\imath}kh^3 & -\sqrt{3}h^2\tau & 0 & 0 & 0 \\ -\sqrt{3}h^2\tau & -\sqrt{3}h^2\tau & 4h^2\tau + \hat{\imath}kh^3 & 0 & 0 & 0 \\ 0 & 0 & 0 & -\hat{\imath}kh^3 & 0 & 0 \\ 0 & 0 & 0 & 0 & -\hat{\imath}kh^3 & 0 \\ 0 & 0 & 0 & 0 & 0 & -\hat{\imath}kh^3 \end{bmatrix}.$$

We immediately see that the rows become linearly dependent if

$$(3\sqrt{3}+6)\tau = -\hat{\imath}kh.$$

Hence, for this τ-value – cf. (55) – the HDG method will fail on tetrahedral meshes.

For orders $p \geq 1$, the element matrices are too complex to find bad parameter values so simply. Instead, we experiment numerically. Setting $\tau = -\hat{\imath}$, which is equivalent to the choice made in [**38**] (see Table 3.1), we compute the smallest singular value of the element matrix M (the matrix of the left hand side of (53) with K set by (56)) for a range of normalized wavenumbers kh. Figures 3.1a and 3.1b show that, for orders $p = 1$ and $p = 2$, there are values of kh for which $\tau = -\hat{\imath}$ results in a singular value very close to zero. Taking a closer look at the first nonzero local minimum in Figure 3.1a, we find that the local matrix corresponding to normalized wavenumber $kh \approx 7.49$ has an estimated condition number exceeding 3.9×10^{15}, i.e., for all practical purposes, the element matrix is singular. To illustrate how a different choice of stabilization parameter τ can affect the conditioning of the element matrix, Figures 3.1c and 3.1d show the smallest singular values for the same range of kh, but with $\tau = 1$. Clearly the latter choice of τ is better than the former.



(A) $p = 1,\ \tau = -\hat{\imath}$

(B) $p = 2,\ \tau = -\hat{\imath}$

(C) $p = 1,\ \tau = 1$

(D) $p = 2,\ \tau = 1$

(E) $kh = 1,\ p = 1$

(F) $kh \approx 7.49,\ p = 1$

(G) $kh = 1 + \hat{\imath},\ p = 1$

(H) $kh \approx 7.49(1 + \hat{\imath}),\ p = 1$

FIGURE 3.1. The smallest singular values of a tetrahedral HDG element matrix.

From another perspective, Figure 3.1e shows the smallest singular value of the element matrix as τ is varied in the complex plane, while fixing kh to 1. Figure 3.1f is similar except that we fixed kh to the value discussed above, approximately 7.49. In both cases, we find that the worst values of τ are along the imaginary axis. Finally, in Figures 3.1g and 3.1h, we see the effects of multiplying these real values of kh by $1 + \hat{\imath}$. The region of the complex plane where bad values of τ are found changes significantly when kh is complex.

3.2. Results on unisolvent stabilization

We now turn to the question of how we can choose a value for the stabilization parameter τ that will guarantee that the local matrices are not singular. The answer, given by a condition on τ, surprisingly also guarantees that the global condensed HDG matrix is nonsingular. These results are based on a tenuous stability inherited from the fact nonzero polynomials are never waves, stated precisely in the ensuing lemma. Then we give the condition on τ that guarantees unisolvency, and before concluding the section, present some caveats on relying solely on this tenuous stability.

As is standard in all HDG methods, the unique solvability of the element problem allows the formulation of a condensed global problem that involves only the interface unknowns. We introduce the following notation to describe the condensed systems. First, for Maxwell's equations, for any $\eta \in N_h$, let $(\vec{E}^\eta, \vec{H}^\eta) \in Y_h \times Y_h$ denote the fields such that, for each $K \in \mathcal{T}_h$, the pair $(\vec{E}^\eta|_K, \vec{H}^\eta|_K)$ satisfies the local problem (53) with data $\eta|_{\partial K}$. That is,

$$\hat{\imath}k(\vec{E}^\eta, \vec{v})_K - (\vec{\nabla}\times \vec{H}^\eta, \vec{v})_K + \langle \tau \vec{E}^\eta \times \hat{\nu},\ \vec{v} \times \hat{\nu}\rangle_{\partial K} = \langle \tau\eta \times \hat{\nu},\ \vec{v} \times \hat{\nu}\rangle_{\partial K}, \tag{57a}$$

$$-(\vec{E}^\eta, \vec{\nabla}\times \vec{w})_K - \hat{\imath}k(\vec{H}^\eta, \vec{w})_K = -\langle \eta, \hat{\nu} \times \vec{w}\rangle_{\partial K}, \tag{57b}$$

for all $\vec{v} \in Y(K)$, $\vec{w} \in Y(K)$. If all the sources in (52) vanish, then the *condensed global problem* for $\hat{E} \in N_h$ takes the form

$$a(\hat{E}, \eta) = 0, \quad \forall \eta \in N_h, \tag{58}$$

where

$$a(\Lambda, \eta) = \sum_{K \in \mathcal{T}_h} \langle \hat{H}^\Lambda \times \hat{\nu}, \eta \rangle_{\partial K}.$$

By following a standard procedure [**10**] we can express $a(\cdot, \cdot)$ explicitly as follows:

$$\begin{aligned}
a(\Lambda, \eta) &= \sum_{K \in \mathcal{T}_h} \langle \vec{H}^\Lambda \times \hat{\nu}, \eta \rangle_{\partial K} + \langle (\hat{H}^\Lambda - \vec{H}^\Lambda) \times \hat{\nu}, \eta \rangle_{\partial K} \\
&= \sum_{K \in \mathcal{T}_h} \hat{\imath}\bar{k}(\vec{H}^\Lambda, \vec{H}^\eta)_K - (\vec{\nabla} \times \vec{H}^\Lambda, \vec{E}^\eta)_K + \langle \tau \hat{\nu} \times (\hat{\nu} \times (\Lambda - \vec{E}^\Lambda)), \eta \rangle_{\partial K} \\
&= \sum_{K \in \mathcal{T}_h} \hat{\imath}\bar{k}(\vec{H}^\Lambda, \vec{H}^\eta)_K - \hat{\imath}k(\vec{E}^\Lambda, \vec{E}^\eta)_K + \langle \tau \hat{\nu} \times (\Lambda - \vec{E}^\Lambda), \hat{\nu} \times \vec{E}^\eta \rangle_{\partial K} \\
&\qquad - \langle \tau \hat{\nu} \times (\Lambda - \vec{E}^\Lambda), \hat{\nu} \times \eta \rangle_{\partial K} \\
&= \sum_{K \in \mathcal{T}_h} \hat{\imath}\bar{k}(\vec{H}^\Lambda, \vec{H}^\eta)_K - \hat{\imath}k(\vec{E}^\Lambda, \vec{E}^\eta)_K - \tau \langle \hat{\nu} \times (\Lambda - \vec{E}^\Lambda), \hat{\nu} \times (\eta - \vec{E}^\eta) \rangle_{\partial K}.
\end{aligned}$$

Here we have used the complex conjugate of (57b) with $\vec{w} = \vec{H}^\Lambda$, along with the definition of $\hat{H}^\Lambda$, and then used (57a).

Similarly, for the Helmholtz equation, let $(\vec{u}^{\,\eta}, \phi^\eta) \in V_h \times W_h$ denote the fields such that, for all $K \in \mathcal{T}_h$, the functions $(\vec{u}^{\,\eta}|_K, \phi^\eta|_K)$ solve the element problem (48) for given data $\hat{\phi} = \eta$. If the sources in (46) vanish, then the *condensed global problem* for $\hat{\phi} \in M_h$ is written as

$$b(\hat{\phi}, \eta) = 0, \quad \forall \eta \in M_h, \tag{59}$$

where the form is found, as before, by the standard procedure:

$$\begin{aligned} b(\Lambda,\eta) &= \sum_{K\in\mathcal{T}_h} \langle \hat{u}^\Lambda \cdot \hat{\nu}, \eta\rangle_{\partial K} \\ &= \sum_{K\in\mathcal{T}_h} \hat{\imath}\bar{k}(\vec{u}^{\,\Lambda}, \vec{u}^{\,\eta})_K - \hat{\imath}k(\phi^\Lambda, \phi^\eta)_K - \tau\langle \Lambda - \phi^\Lambda, \eta - \phi^\eta\rangle_{\partial K}. \end{aligned}$$

The sesquilinear forms $a(\cdot,\cdot)$ and $b(\cdot,\cdot)$ are used in the main result, which gives sufficient conditions for the solvability of the local problems (53), (48) and the global problems (58), (59).

Before proceeding to the main result, we give a simple lemma, which roughly speaking, says that nontrivial harmonic *waves are not polynomials.*

Lemma 3.2.1. *Let $p \geq 0$ be an integer, $0 \neq k \in \mathbb{C}$, and D an open set. Then, there is no nontrivial $\vec{E} \in (\mathcal{P}_p(D))^3$ satisfying*

$$\vec{\nabla}\times(\vec{\nabla}\times\vec{E}) - k^2\vec{E} = 0$$

and there is no nontrivial $\phi \in \mathcal{P}_p(D)$ satisfying

$$\Delta\phi + k^2\phi = 0.$$

Proof. We use a contradiction argument. If $E \not\equiv \vec{0}$, then we may assume without loss of generality that at least one of the components of $\vec{E}$ is a polynomial of degree exactly p. But this contradicts $k^2\vec{E} = \vec{\nabla}\times(\vec{\nabla}\times\vec{E})$ because all components of $\vec{\nabla}\times(\vec{\nabla}\times\vec{E})$ are polynomials of degree at most $p-2$. Hence $\vec{E} \equiv \vec{0}$. An analogous argument can be used for the Helmholtz case as well. □

Theorem 3.2.2. *Suppose*

$$\text{(60a)} \qquad \mathrm{Re}(\tau) \neq 0, \qquad \textit{whenever } \mathrm{Im}(k) = 0, \textit{ and}$$

$$\text{(60b)} \qquad \mathrm{Im}(k)\,\mathrm{Re}(\tau) \leq 0, \qquad \textit{whenever } \mathrm{Im}(k) \neq 0.$$

Then, in the Maxwell case, the local element problem (53) *and the condensed global problem* (58) *are both unisolvent. Under the same condition, in the Helmholtz case, the local element problem* (48) *and the condensed global problem* (59) *are also unisolvent.*

Proof. We first prove the theorem for the local problem for Maxwell's equations. Assume (60) holds and set $\hat{E}_h = \vec{0}$ in the local problem (53). Unisolvency will follow by showing that $\vec{E}_h$ and $\vec{H}_h$ must equal $\vec{0}$. Choosing $\vec{v} = \vec{E}_h$, and $\vec{w} = \vec{H}_h$, then subtracting (53b) from (53a), we get

$$\hat{\imath}k\left(\|\vec{E}_h\|_K^2 + \|\vec{H}_h\|_K^2\right) + 2\hat{\imath}\mathrm{Im}(\vec{E}_h, \vec{\nabla}\times\vec{H}_h)_K + \tau\|\vec{E}_h \times \hat{\nu}\|_{\partial K}^2 = 0,$$

whose real part is

$$-\mathrm{Im}(k)\left(\|\vec{E}_h\|_K^2 + \|\vec{H}_h\|_K^2\right) + \mathrm{Re}(\tau)\|\vec{E}_h \times \hat{\nu}\|_{\partial K}^2 = 0.$$

Under condition (60b), we immediately have that the fields $\vec{E}_h$ and $\vec{H}_h$ are zero on K. Otherwise, (60a) implies $\vec{E}_h \times \hat{\nu}|_{\partial K} = 0$, and then (53) gives

$$\hat{\imath}k\vec{E}_h - \vec{\nabla}\times\vec{H}_h = 0,$$

$$\hat{\imath}k\vec{H}_h + \vec{\nabla}\times\vec{E}_h = 0,$$

implying

$$\vec{\nabla}\times(\vec{\nabla}\times\vec{E}_h) = k^2\vec{E}_h.$$

By Lemma 3.2.1 this equation has no nontrivial solutions in the space $Y(K)$. Thus, the element problem for Maxwell's equations is unisolvent.

We prove that the global problem for Maxwell's equations is unisolvent by showing that $\hat{E}_h = \vec{0}$ is the unique solution of equation (58). This is done in a manner almost identical to what was done above for the local problem: First, set $\eta = \hat{E}_h$ in equation (58) and take the real part to get

$$\sum_{K\in\mathcal{T}_h} \mathrm{Im}(k)\left(\|\vec{H}_h\|_K^2 + \|\vec{E}_h\|_K^2\right) - \mathrm{Re}(\tau)\|\hat{\nu}\times(\hat{E}_h - \vec{E}_h)\|_{\partial K}^2 = 0. \tag{61}$$

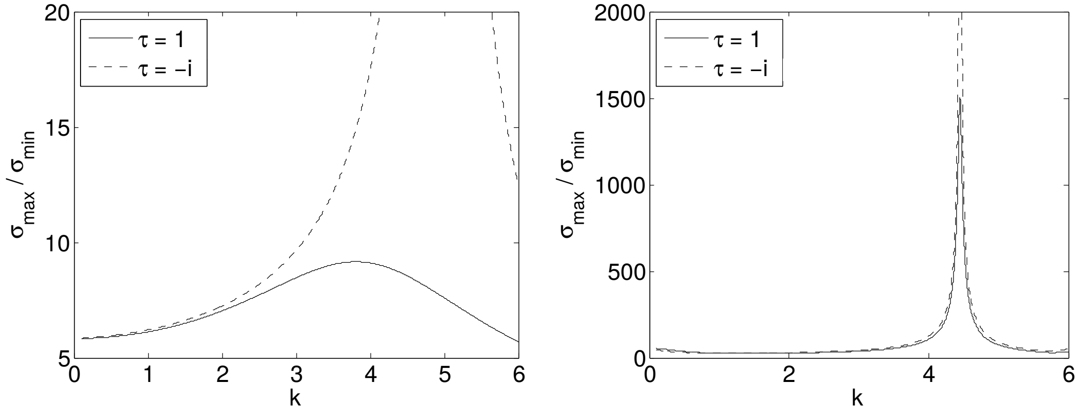
This immediately shows that if condition (60b) holds, then the fields $\vec{E}_h$ and $\vec{H}_h$ are zero on $\Omega\subset\mathbb{R}^3$ and the proof is finished. In the case of condition (60a), we have $\hat{\nu}\times(\hat{E}_h - \vec{E}_h|_{\partial K}) = \vec{0}$ for all K. Using equations (52), this yields

$$[\vec{\nabla}\times(\vec{\nabla}\times\vec{E}_h)]|_K = k^2\vec{E}_h|_K,$$

so Lemma 3.2.1 proves that the fields on element interiors are zero, which in turn implies $\hat{E}_h = \vec{0}$ also. Thus, the theorem holds for the Maxwell case.

The proof for the Helmholtz case is entirely analogous. □

Note that even with Dirichlet boundary conditions and real k, the theorem asserts the existence of a unique solution for the Helmholtz equation. However, the exact Helmholtz problem (1) is well-known to be *not* uniquely solvable when k is set to one of an infinite sequence of resonance values in $\Sigma\subset\mathbb{R}$. The fact that the discrete system is uniquely solvable even when the exact system is not, suggests the presence of artificial dissipation in HDG methods. We will investigate this issue more thoroughly in the next section.

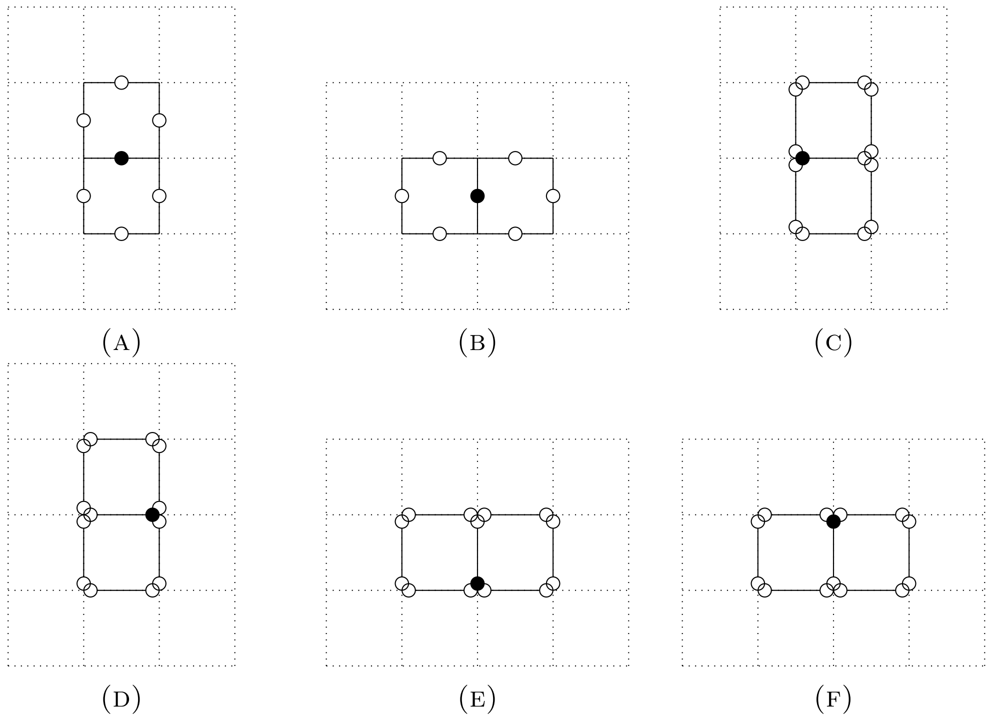


(A) Degree $p = 0$

(B) Degree $p = 1$

FIGURE 3.2. Conditioning of the HDG matrix for the Helmholtz equation near the first resonance $k = \pi\sqrt{2} \approx 4.44$.

However, we do not advocate relying on this discrete unisolvency near a resonance where the original boundary value problem is not uniquely solvable. The discrete matrix, although invertible, can be poorly conditioned near these resonances. Consider, for example, the Helmholtz equation on the unit square with Dirichlet boundary conditions. The first resonance occurs at $k = \pi\sqrt{2}$. In Figure 3.2, we plot the condition number $\sigma_{max}/\sigma_{min}$ of the condensed HDG matrix for a range of wavenumbers near the resonance $k = \pi\sqrt{2}$, using a small fixed mesh of mesh size $h = 1/4$, and a value of $\tau = 1$ that satisfies (60). We observe that although the condition number remains finite, as predicted by the theorem, it peaks near the resonance for both the $p = 0$ and the $p = 1$ cases. We also observe that a parameter setting of $\tau = -\hat{\imath}$ that does not satisfy the conditions of the theorem produce much larger condition numbers, e.g., the condition numbers that are orders of magnitude greater than 10^{10} (off axis limits of Figure 3.2b) for k near the resonance were obtained for $p = 1$ and $\tau = -\hat{\imath}$. To summarize the caveat, we note that, in general, *even though the condition number is always bounded for values of* τ *that satisfy* (60)*, it may still be practically infeasible to find the HDG solution near a resonance.*


(A)

(B)

(C)

(D)

(E)

(F)

FIGURE 3.3. Stencils corresponding to the shaded node types. (A)–(B): Two node types of the lowest order ($p = 0$) method; (C)–(F): Four node types of the first order ($p = 1$) method.

3.3. Lowest order and first order HDG stencils

The lowest order and first order HDG stencils for the Helmholtz equation using square elements are compared in Figure 3.3. They are used for the dispersion analysis in Section 4.3. Note that the figure only shows the interactions of the degrees of freedom corresponding to the $\hat{\phi}$ variable—the only degrees of freedom involved after elimination of the $\vec{u}$ and ϕ degrees of freedom via static condensation. The lowest order method has two node types (shown in Figures 3.3a–3.3b), while the first order method has four node types (shown in Figures 3.3c–3.3f).

4

Dispersion analyses for the DPG$_\varepsilon$ and HDG methods

The pollution effect and numerical dispersion, which we recall from the beginning of Chapter 1, are reintroduced in Section 4.1. Then, in Section 4.2 we describe the approach of [**18**] for computing discrete wavenumbers, which allow us to quantify dispersive and dissipative errors. This technique is adapted to the DPG and HDG methods for the 2D Helmholtz equation on meshes of square elements. Results for the lowest order and first order HDG methods are in Section 4.3. For the lowest order HDG method, we begin with the one dimensional (1D) case, and we obtain exact dispersion relations in 1D and 2D. We also compare the lowest order and first order HDG methods with the Hybrid Raviart-Thomas (HRT) method, for which we have also obtained an exact dispersion relation for the lowest order case. Dispersion analysis results for the lowest order DPG method are presented in Section 4.4. Finally, the DPG and HDG methods are compared in Section 4.5. Throughout this chapter, k is assumed to be real valued.

4.1. Numerical dispersion and dissipation

Suppose we use a given numerical method to solve the Helmholtz equation (3) for ϕ. Since the approximation ϕ_h is contained in a finite dimensional function space, the total pointwise error $e(\vec{x}) = \phi(\vec{x}) - \phi_h(\vec{x})$ is bound to include interpolation error $e_I(\vec{x}) = \phi(\vec{x}) - \phi_I(\vec{x})$, where ϕ_I denotes the interpolant. However, in practice one observes total errors that are larger than the interpolation error. The additional contribution to the total error that cannot be attributed to interpolation is known as "pollution", that is, $e_{\text{pol}}(\vec{x}) = e(\vec{x}) - e_I(\vec{x}) = \phi_I(\vec{x}) - \phi_h(\vec{x})$. The pollution is due to dispersive and dissipative error [**29**], and has been shown to be inevitable for a class of generalized finite element methods in two or more dimensions [**3**]. The effect is worse for large wavenumbers.

Numerical dispersion and dissipation relate to the fact that, if the exact solution ϕ is a plane wave (restricted to the computational domain) with wavenumber $k > 0$, traveling in the θ-direction, then the computed solution will have a discrete wavenumber, $k^h(\theta) \in \mathbb{C}$, that is not equal to k. A precise definition of $k^h(\theta)$ is given in the next section– here, we just illustrate the main idea. If

$$\phi(\vec{x}) = e^{\hat{\imath}k(\cos(\theta)x_1+\sin(\theta)x_2)},$$

for any $\vec{x}$ in the domain, then, for some $a \neq 0$, the discrete wavenumber $k^h(\theta)$ satisfies

$$\begin{aligned}\phi_h(\vec{x}) &= ae^{\hat{\imath}k^h(\theta)(\cos(\theta)x_1+\sin(\theta)x_2)}\\ &= ae^{-\operatorname{Im}(k^h(\theta))(\cos(\theta)x_1+\sin(\theta)x_2)}e^{\hat{\imath}\operatorname{Re}(k^h(\theta))(\cos(\theta)x_1+\sin(\theta)x_2)},\end{aligned}$$

for values of $\vec{x}$ that coincide with stencil nodes that are associated with ϕ_h. This expression shows that $\operatorname{Re}(k^h(\theta))$ introduces error into the wavelength of ϕ_h, and $\operatorname{Im}(k^h(\theta))$ introduces error into the amplitude. In general, if $k^h(\theta)$ can be computed, either by using an exact dispersion relation or a numerical technique, then the dispersive, dissipative, and total errors, defined by

$$\rho_{\text{disp}} = \max_\theta |\operatorname{Re}(k^h(\theta)) - k|, \tag{62a}$$

$$\rho_{\text{dissip}} = \max_\theta |\operatorname{Im}(k^h(\theta))|, \tag{62b}$$

$$\rho_{\text{total}} = \max_\theta |k^h(\theta) - k|, \tag{62c}$$

respectively, can be used to evaluate the method's performance.

For further illustration, as well as motivation for the DPG$_\varepsilon$ dispersion analysis, consider the L^2 least-squares Galerkin method for (6). Set $R_h \subset R$ to the Cartesian product of the lowest order Raviart-Thomas and Lagrange spaces, together with the boundary condition in R. The method finds $(\vec{u}_h^{\text{ls}}, \phi_h^{\text{ls}}) \in R_h$ such that

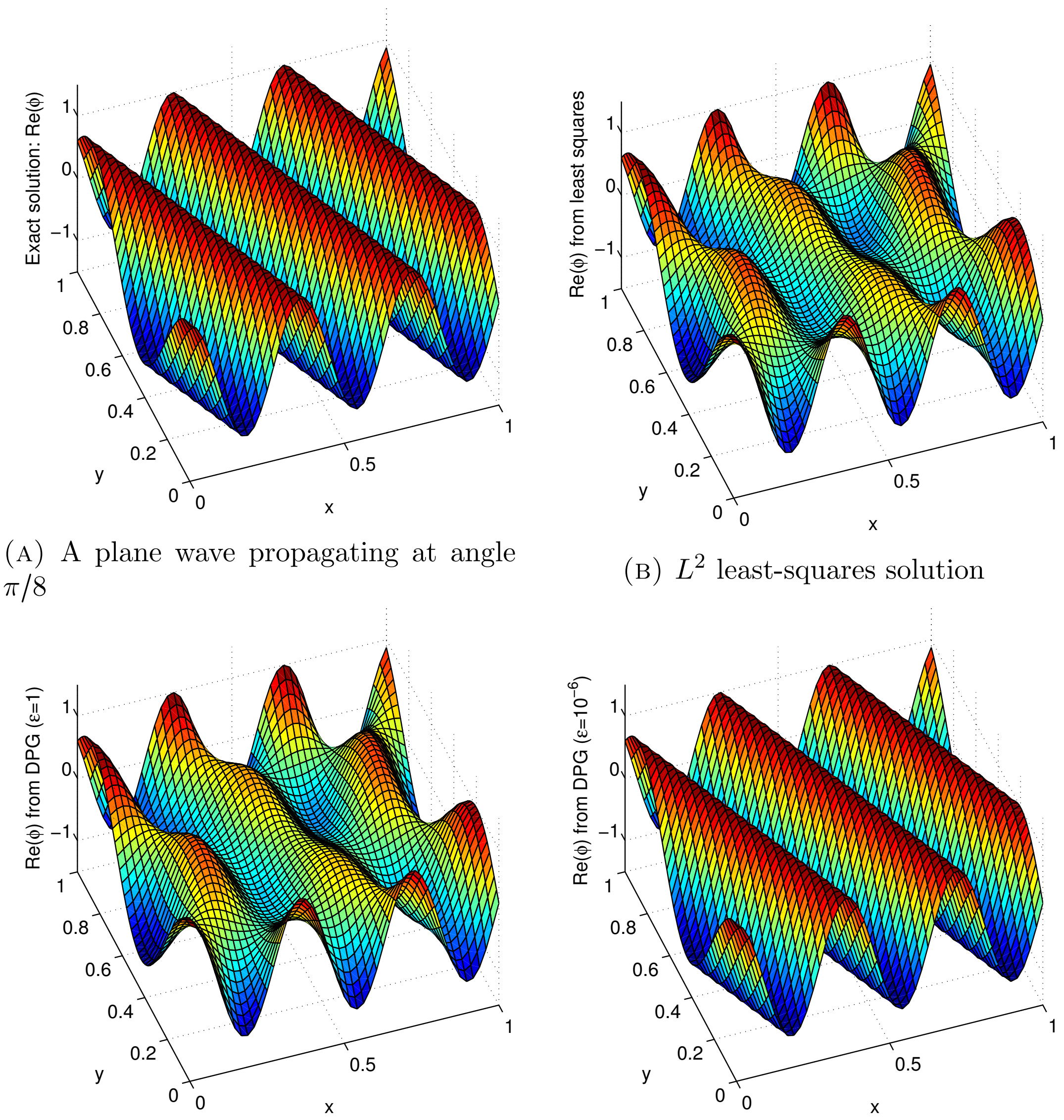
$$(\vec{u}_h^{\text{ls}}, \phi_h^{\text{ls}}) = \arg\min_{w \in R_h} \| f - Aw \|. \tag{63}$$

The method (63) belongs to the so-called FOSLS [**7**] class of methods.

Although (63) appears at first sight to be a reasonable method, computations yield solutions with artificial dissipation. For example, suppose we use (63), appropriately modified to include nonhomogeneous boundary conditions, to approximate a plane wave propagating at angle $\theta = \pi/8$ in the unit square. A comparison between the real parts of the exact solution (in Figure 4.1a) and the computed solution (in Figure 4.1b) shows that the computed solution dissipates at interior mesh points. The same behavior is observed for the lowest order DPG method (which has $\varepsilon = 1$) in Figure 4.1c. The DPG$_\varepsilon$ method with $\varepsilon = 10^{-6}$ however gave a solution (in Figure 4.1d) that is visually indistinguishable from the exact solution.

4.2. An approach to compute discrete wavenumbers

To briefly adapt the approach of [**18**] to fit our context, we consider a general method for the homogeneous Helmholtz equation on an infinite uniform lattice of $h \times h$ square elements, with vertices $(h\mathbb{Z})^2$. Suppose the method has S different types of nodes on this lattice, some falling in between the lattice points, each corresponding to a different type of variable, with its own stencil (and hence its own equation). All nodes of the t^{th} type ($t = 1, 2, \ldots, S$) are assumed to be of the form $\vec{\jmath}h$ where $\vec{\jmath}$ lies in an infinite subset of $(\mathbb{Z}/2)^2$. The solution value at a general node $\vec{\jmath}h$ of the t^{th}



(A) A plane wave propagating at angle $\pi/8$

(B) L^2 least-squares solution

(C) Numerical traces from the lowest order DPG$_\varepsilon$ method with $\varepsilon = 1$

(D) Numerical traces from the lowest order DPG$_\varepsilon$ method with $\varepsilon = 10^{-6}$

FIGURE 4.1. Approximations to a plane wave computed using a uniform mesh of square elements of size $h = 1/48$ (about sixteen elements per wavelength). Artificial dissipation is visible in Figures 4.1b and 4.1c. (The parameter r in the DPG$_\varepsilon$ method, defined in §2.1.4 is set to 3 for these computations.)

type is denoted by $\psi_{t,\vec{\jmath}}$. Note that methods with multiple solution components are accommodated using the above mentioned node types.

The t^{th} stencil, centered around a node of the t^{th} type at position $\vec{\jmath}h$, consists of a finite number of nodes of potentially all types. Suppose we have finite index sets $J_s \subset (\mathbb{Z}/2)^2$, for each $s = 1, 2, \ldots, S$, such that all the nodes of the t^{th} stencil centered around $\vec{\jmath}h$ can be listed as $N_{\vec{\jmath},t} = \{(\vec{\jmath} + \vec{l}\,)h : \vec{l} \in J_s \text{ for some } s = 1, 2, \ldots, S\}$ with the understanding that $(\vec{\jmath} + \vec{l}\,)h$ is a node of s^{th} type whenever $\vec{l} \in J_s$. This allows interaction between variables of multiple types. Every node $(\vec{\jmath} + \vec{l}\,)h$ in $N_{\vec{\jmath},t}$ has a corresponding stencil coefficient (or weight) denoted by $D_{t,s,\vec{l}}$, which are the linear combinations of the entries of the local matrix (i.e., (44) for the DPG method or (48) for the HDG method) that likewise arise as entries of the global matrix. Due to translational invariance, these weights do not change if we center the stencil at another node at position $\vec{\jmath}\,'h$. Hence, the numbers $D_{t,s,\vec{l}}$ do not depend on the center index $\vec{\jmath}$.

We obtain the method's equation at a general node $\vec{\jmath}h$ of the t^{th} type by applying the t^{th} stencil centered around $\vec{\jmath}h$ to the solution values $\{\psi_{t,\vec{\jmath}}\}$, namely

$$\sum_{s=1}^{S} \sum_{l \in J_s} D_{t,s,\vec{l}}\; \psi_{s,\vec{\jmath}+\vec{l}} = 0. \tag{64}$$

Note that we have set all sources to zero to get a zero right hand side in (64).

Plane waves, $\psi(\vec{x}) \equiv e^{\hat{\imath}\vec{k}\cdot\vec{x}}$, are exact solutions of the Helmholtz equation with zero sources (and are often used to represent other solutions). Here the wavevector $\vec{k}$ is of the form $\vec{k} = k(\cos(\theta), \sin(\theta))$ for some $0 \le \theta < 2\pi$ representing the direction of propagation. The objective of dispersion analysis is to find similar solutions of the discrete homogeneous system. Accordingly, we set in (64), the ansatz

$$\psi_{t,\vec{\jmath}} = a_t e^{\hat{\imath}\vec{\kappa}_h\cdot\vec{\jmath}h}, \tag{65}$$

where $\vec{\kappa}_h = k^h(\cos(\theta), \sin(\theta))$ and a_t is an arbitrary complex number associated to the t^{th} variable type. We want to find such discrete wavenumbers k^h satisfying (64).

To this end, we must solve (64) after substituting (65) therein, namely

$$\sum_{s=1}^{S} a_s \sum_{l \in J_s} D_{t,s,\vec{l}}\, e^{\hat{\imath}\vec{\kappa}_h \cdot (\vec{\jmath}+\vec{l})h} = 0, \tag{66}$$

for all $t = 1, 2, \ldots S$. Multiplying by $e^{-\hat{\imath}\vec{\kappa}_h \cdot \vec{\jmath}h}$, we remove any dependence on $\vec{\jmath}$. Defining the $S \times S$ matrix $F \equiv (F_{ts}(k^h))$ by

$$F_{ts}(k^h) = \sum_{\vec{l} \in J_s} D_{t,s,\vec{l}}\, e^{\hat{\imath}(k^h(\cos\theta, \sin\theta)\cdot \vec{l})h},$$

we observe that solving (66) is equivalent to solving

$$\det F(k^h) = 0. \tag{67}$$

This is the nonlinear equation we solve to obtain the discrete wavenumber k^h corresponding to any given θ and k. Note that, for the DPG method, k^h will also depend on the value of the scaling parameter ε and, for the HDG method, k^h will depend on the value of the stabilization parameter τ.

4.3. Dispersion analysis for the HDG method

When the wavenumber k is *complex*, we have seen that it is important to choose the stabilization parameter τ such that (60b) holds. We have also seen that when k is real, the stability obtained by (60a) is so tenuous that it is of negligible practical value. For real wavenumbers, it is safer to rely on stability of the (undiscretized) boundary value problem, rather than the stability obtained by a choice of τ.

The focus of this section is on *real* k and the Helmholtz equation (1). In this case, having already separated the issue of stability from the choice of τ, we are now free to optimize the choice of τ for other goals. By means of a dispersion analysis, we now proceed to show that some values of τ are better than others for minimizing discrepancies in wavenumber. Since dispersion analyses are limited to the study of plane-wave propagation, we will not explicitly consider the Maxwell HDG system in this section. However, since we have written the Helmholtz and Maxwell system consistently with respect to the stabilization parameter (cf. the transition from (47) to (51) via (50)), we anticipate our results for the Helmholtz case to be useful for the Maxwell case also.

4.3.1. The dispersion relation in the one-dimensional case. Consider the HDG method (46) in the lowest order ($p = 0$) case in 1D – after appropriately interpreting the boundary terms in (46). We follow the techniques of [**1**] for performing a dispersion analysis. Using a basis on a segment of size h in this order $u_1 = 1, \quad \phi_1 = 1, \quad \hat{\phi}_1 = 1, \quad \hat{\phi}_2 = 1$, the HDG element matrix takes the form $M = \left[\begin{smallmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{smallmatrix}\right]$ where

$$M_{11} = \begin{bmatrix} \hat{\imath}kh & 0 \\ 0 & -\hat{\imath}kh - 2\tau \end{bmatrix} \qquad M_{12} = \begin{bmatrix} -1 & +1 \\ \tau & \tau \end{bmatrix}$$

$$M_{21} = M_{12}^t \qquad M_{22} = \begin{bmatrix} -\tau & 0 \\ 0 & -\tau \end{bmatrix}.$$

The Schur complement for the two endpoint basis functions $\{\hat{\phi}_1, \hat{\phi}_2\}$ is then

$$S = -\begin{bmatrix} \dfrac{1}{\hat{\imath}kh} - \dfrac{\tau^2}{\hat{\imath}kh + 2\tau} + \tau & -\dfrac{1}{\hat{\imath}kh} - \dfrac{\tau^2}{\hat{\imath}kh + 2\tau} \\ -\dfrac{1}{\hat{\imath}kh} - \dfrac{\tau^2}{\hat{\imath}kh + 2\tau} & \dfrac{1}{\hat{\imath}kh} - \dfrac{\tau^2}{\hat{\imath}kh + 2\tau} + \tau \end{bmatrix}.$$

Applying this matrix on an infinite uniform grid (of elements of size h), we obtain the stencil at an arbitrary point. If $\hat{\psi}_j$ denotes the solution (trace) value at the jth point ($j \in \mathbb{Z}$), then the jth equation reads

$$2\hat{\psi}_j \left(\frac{1}{\hat{\imath} kh} - \frac{\tau^2}{\hat{\imath} kh + 2\tau} + \tau \right) + (\hat{\psi}_{j-1} + \hat{\psi}_{j+1}) \left(-\frac{1}{\hat{\imath} kh} - \frac{\tau^2}{\hat{\imath} kh + 2\tau} \right) = 0.$$

In a dispersion analysis, we are interested in how this equation propagates plane waves on the infinite uniform grid. Hence, substituting $\hat{\psi}_j = \exp(\hat{\imath} k^h jh)$, we get the following dispersion relation for the unknown discrete wavenumber k^h:

$$\cos(k^h h) \left(\frac{1}{\hat{\imath} kh} + \frac{\tau^2}{\hat{\imath} kh + 2\tau} \right) = \left(\frac{1}{\hat{\imath} kh} - \frac{\tau^2}{\hat{\imath} kh + 2\tau} + \tau \right)$$

Simplifying,

$$k^h h = \cos^{-1} \left(1 - \frac{(kh)^2}{2 + \hat{\imath} kh(\tau + \tau^{-1})} \right). \tag{68}$$

This is the *dispersion relation* for the HDG method in 1D. Even when τ and k are real, the argument of the arccosine is not. Hence

$$\operatorname{Im}(k^h) \neq 0, \tag{69}$$

in general, indicating the *presence of artificial dissipation in HDG methods.*

Let us now study the case of small kh (i.e., large number of elements per wavelength). As $kh \to 0$, using the approximation $\cos^{-1}(1 - x^2/2) \approx x + x^3/24 + \cdots$ valid for small x, and simplifying (68), we obtain

$$k^h h - kh \approx -\frac{(\tau^2 + 1)\hat{\imath}}{4\tau} (kh)^2 + O((kh)^3). \tag{70}$$

Comparing this with the discrete dispersion relation of the standard finite element method in one space dimension (see [**1**]), namely $k^h h - kh \approx O((kh)^3)$, we find that wavenumber discrepancies from the HDG method can be larger depending on the value of τ. In particular, we conclude that *if we choose $\tau = \pm\hat{\imath}$, then the error $k^h h - kh$ in both methods are of the same order $O((kh)^3)$.*

Before concluding this discussion of the one-dimensional case, we note an alternate form of the dispersion relation suitable for comparison with later formulas. Using the half-angle formula, equation (68) can be rewritten as

$$c^2 = 1 - \left(\frac{(kh)^2}{2}\right)\left(\frac{\tau}{\hat{\imath}kh(\tau^2+1)+2\tau}\right), \tag{71}$$

where $c = \cos(k^h h/2)$.

4.3.2. Lowest order two-dimensional case. In the 2D case, we use an infinite grid of square elements of side length h. The HDG element matrix associated to the lowest order ($p = 0$) case of (46) is now larger, but the Schur complement obtained after condensing out all interior degrees of freedom is only 4×4 because there is one degree of freedom per edge. Note that horizontal and vertical edges represent two distinct types of degrees of freedom, as shown in Figures 3.3a and 3.3b. Hence there are two types of stencils.

For conducting dispersion analysis with multiple stencils, we follow the approach in [**18**] (already described more generally in Section 4.2). Accordingly, let C_1 and C_2 denote the infinite sets of stencil centers for the two types of stencils present in our case. Then, we get an infinite system of equations for the unknown solution (numerical trace) values $\hat{\psi}_{1,\vec{p}_1}$ and $\hat{\psi}_{1,\vec{p}_2}$ at all $\vec{p}_1 \in C_1$ and $\vec{p}_2 \in C_2$, respectively. We are interested in how this infinite system propagates plane wave solutions in every angle θ. Therefore, with the ansatz $\hat{\psi}_{j,\vec{p}_j} = a_j \exp(\hat{\imath}\vec{\kappa}_h \cdot \vec{p}_j)$ for constants a_j ($j = 1$ and 2), where the discrete wavevector is given by

$$\vec{\kappa}_h = k^h \begin{bmatrix} \cos\theta \\ \sin\theta \end{bmatrix},$$

we proceed to find the relation between the discrete wavenumber k^h and the exact wavenumber k.

Substituting the ansatz into the infinite system of equations and simplifying, we obtain a 2×2 system $F\left[\begin{smallmatrix} a_1 \\ a_2 \end{smallmatrix}\right] = 0$ where

$$F = \begin{bmatrix} 2\,kh\tau^2 c_1\,c_2 & d_1\,(4\,\tau + \hat{\imath}kh) + 2\,kh\tau^2 {c_1}^2 \\ d_2\,(4\,\tau + \hat{\imath}kh) + 2\,kh\tau^2 {c_2}^2 & 2\,kh\tau^2 c_1 c_2 \end{bmatrix}$$

and, for $j = 1, 2$,

$$c_j = \cos\left(\frac{1}{2} h k_j^h\right), \qquad d_j = 2\hat{\imath}(1 - c_j^2) - \tau kh, \qquad k_1^h = k^h \cos\theta, \qquad k_2^h = k^h \sin\theta.$$

Hence the *2D dispersion relation* relating k^h to k in the HDG method is

$$\det(F) = 0. \tag{72}$$

To formally compare this to the 1D dispersion relation, consider these two sufficient conditions for $\det(F) = 0$ to hold:

$$2(kh)^2\tau^2 c_j^2 + d_j\left(2\tau kh + \hat{\imath}(k_j h)^2\right) = 0, \qquad \text{for } j = 1, 2, \tag{73}$$

where $k_1 = k\cos\theta$ and $k_2 = k\sin\theta$. (Indeed, multiplying $(73)_j$ by d_j and summing over $j = 1, 2$, one obtains $kh\det(F)$.) The equations in (73) can be simplified to

$$c_j^2 = 1 - \frac{(k_j h)^2}{2\hat{\imath}}\left(\frac{kh\,\tau}{(k_j h)^2 + (kh)^2\,\tau^2 - 2\,\hat{\imath}\,kh\,\tau}\right), \qquad j = 1, 2, \tag{74}$$

which are relations that have a form similar to the 1D relation (71). Hence we use asymptotic expansions of arccosine for small kh, similar to the ones used in the 1D case, to obtain an expansion for k_j^h, for $j = 1, 2$.

The final step in the calculation is the use of the simple identity

$$k^h = \left((k_1^h)^2 + (k_2^h)^2 \right)^{1/2}. \tag{75}$$

Simplifying the above-mentioned expansions for each term on the right hand side above, we obtain

$$k^h h - kh = \frac{\hat{\imath}(\cos(4\,\theta) + 3 + 4\,\tau^2)}{16\,\tau} (kh)^2 + O((kh)^3) \tag{76}$$

as $kh \to 0$. Thus, we conclude that if we want dispersion errors to be $O((kh)^3)$, then we must choose

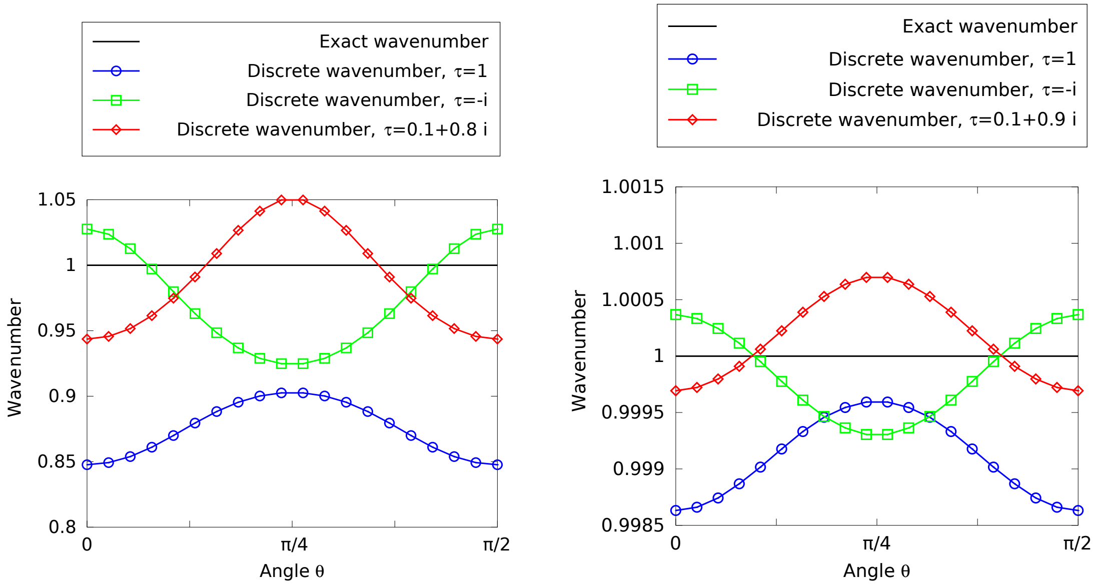
$$\tau = \pm\frac{1}{2}\hat{\imath}\sqrt{\cos(4\theta) + 3}, \tag{77}$$

a prescription that is not very useful in practice because it depends on the propagation angle θ. However, we can obtain a more practically useful condition by setting τ to be the constant value that best approximates $\pm\frac{1}{2}\hat{\imath}\sqrt{\cos(4\theta) + 3}$ for all $0 \leq \theta \leq \pi/2$, namely

$$\tau = \pm\hat{\imath}\frac{\sqrt{3}}{2}. \tag{78}$$

These values of τ *asymptotically minimize errors in discrete wavenumber over all angles for the lowest order 2D HDG method.*

We now report results of numerical computation of $k^h = k^h(\theta)$ by directly applying a nonlinear solver to the 2D dispersion relation (72) (for a set of propagation angles θ). The obtained values of the real part $\operatorname{Re} k^h(\theta)$ are plotted in Figure 4.2a, for a few fixed values of τ. The discrepancy between the exact and discrete curves quantifies the difference in the real parts of the wavenumber for the computed and the exact wave. Next, analyzing the computed $k^h(\theta)$ for values of τ on a uniform grid in the



(A) $p = 0$ (B) $p = 1$

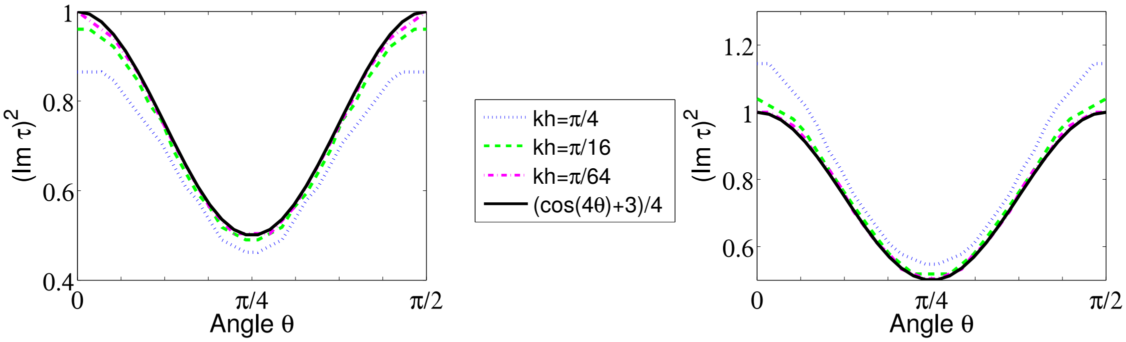
FIGURE 4.2. Real part of the numerical wavenumber $\text{Re}(\vec{k}^h(\theta))$ as a function of θ for various choices of τ. Here, $k = 1$ and $h = \pi/4$.

complex plane, we found that the values of τ that minimize $|kh - k^h(\theta)h|$ are purely imaginary. As shown in Figure 4.3, these τ-values approach the asymptotic values determined analytically in (77). A second validation of our analysis is performed by considering the maximum error over all θ for each value of τ and then determining the practically optimal value of τ. The results, given in Table 4.1, show that the optimal τ values do approach the analytically determined value (see (78)) of $\pm\hat{\imath}\frac{\sqrt{3}}{2} \approx 0.8660$. Further numerical results for the $p = 0$ case are presented together with a higher order case in the next subsection.

4.3.3. Higher order case. To go beyond the $p = 0$ case, we again use the approach of Section 4.2. Using a higher order HDG stencil, we want to obtain an analogue of (72), which can be numerically solved for the discrete wavenumber $k^h = k^h(\theta)$.

kh	Optimal τ, $\text{Im}(\tau) > 0$	Optimal τ, $\text{Im}(\tau) < 0$
$\pi/4$	$0.807\hat{\imath}$	$-0.931\hat{\imath}$
$\pi/8$	$0.837\hat{\imath}$	$-0.898\hat{\imath}$
$\pi/16$	$0.851\hat{\imath}$	$-0.882\hat{\imath}$
$\pi/32$	$0.859\hat{\imath}$	$-0.874\hat{\imath}$
$\pi/64$	$0.863\hat{\imath}$	$-0.871\hat{\imath}$
$\pi/128$	$0.865\hat{\imath}$	$-0.868\hat{\imath}$
$\pi/256$	$0.866\hat{\imath}$	$-0.867\hat{\imath}$

TABLE 4.1. Numerically found values of τ that minimize $|kh - k^h(\theta)h|$ for all θ in the $p = 0$ case.



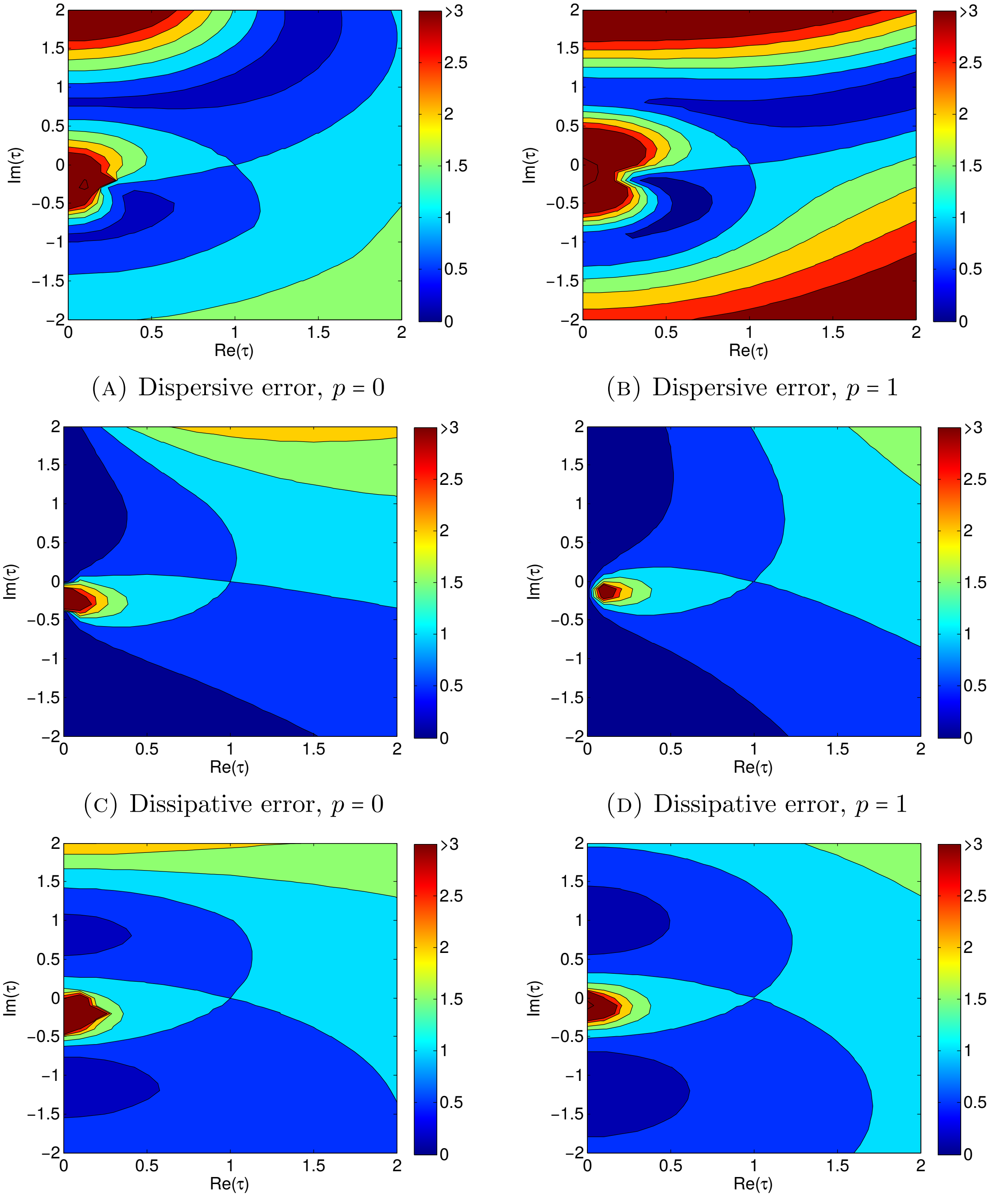
(A) $\text{Im}(\tau) > 0$

(B) $\text{Im}(\tau) < 0$

FIGURE 4.3. The values of τ that locally minimize $|kh - k^h h|$ are purely imaginary. Here, $(\text{Im}(\tau))^2$ is compared with asymptotic values (solid lines).

Again, we consider an infinite lattice of $h \times h$ square elements with the ansatz that the HDG degrees of freedom interpolate a plane wave traveling in the θ direction with wavenumber k^h.

Results of the dispersion analysis are shown in Figures 4.2 and 4.4. These figures combine the results from previously discussed $p = 0$ case and the $p = 1$ case to facilitate comparison. Here, we set $k = 1$ and $h = \pi/4$, i.e., 8 elements per wavelength. Figure 4.4 shows the dispersive, dissipative, and total errors for various values of $\tau \in \mathbb{C}$. For both the lowest order and first order cases, although the dispersive error is minimized at a



(A) Dispersive error, $p = 0$

(B) Dispersive error, $p = 1$

(C) Dissipative error, $p = 0$

(D) Dissipative error, $p = 1$

(E) Total error, $p = 0$

(F) Total error, $p = 1$

FIGURE 4.4. Dispersive error ϵ_{disp}, dissipative error ϵ_{dissip}, and total error ϵ_{total} for various $\tau \in \mathbb{C}$. Here, $k = 1$, $h = \pi/4$, and all errors are normalized by their respective values at $\tau = 1$.

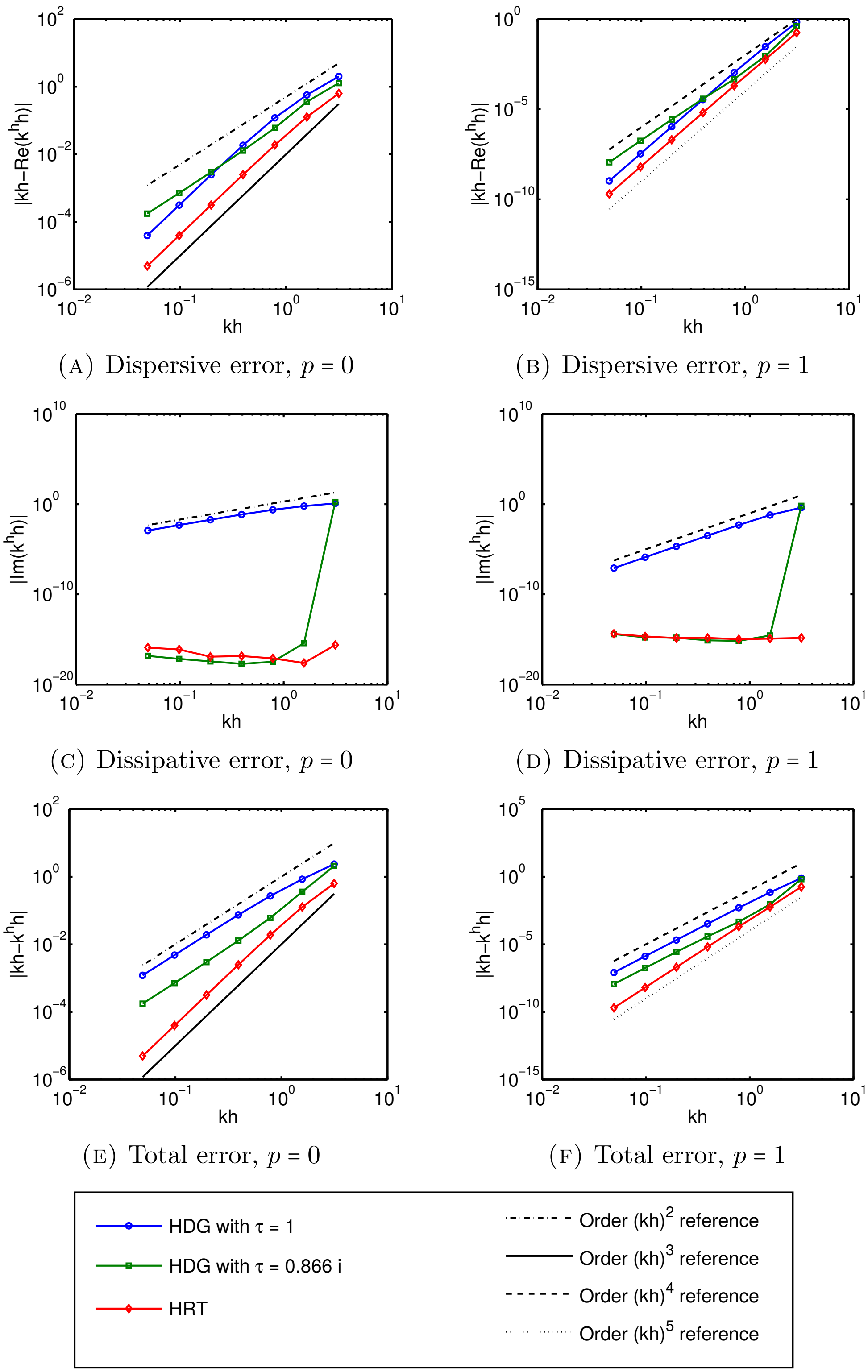
value of τ having nonzero real part, the total error is minimized at a purely imaginary value of τ. This is attributed to the small dissipative errors for such τ. Specifically, the total error is minimized when $\tau = 0.87\hat{\imath}$ in the $p = 1$ case. This is close to the optimal value of τ found (both analytically and numerically) for $p = 0$. This value of τ reduces the total error by 90% in the $p = 1$ case, relative to the total error when using $\tau = 1$.

We remark that Figure 4.2b corrects a mistake that appeared in Figure 5b of [**23**]. There, the figure resulted from setting $h = \pi/2$, not $h = \pi/4$. The conclusions of the paper are not affected.

4.3.4. Comparison with the Hybrid Raviart-Thomas method. The HRT (Hybrid Raviart-Thomas) method is a classical mixed method [**2, 9, 40**] which has a similar stencil pattern, but uses different spaces. Namely, the HRT method for the Helmholtz equation is defined by exactly the same equations as (46) but with these choices of spaces on square elements: $V(K) = \mathcal{Q}_{p+1,p}(K) \times \mathcal{Q}_{p,p+1}(K)$, $W(K) = \mathcal{Q}_p(K)$, and $M(F) = \mathcal{P}_p(F)$. Recall that $Q_{l,m}(K)$ denotes the space of polynomials which are of degree at most l in the first coordinate and of degree at most m in the second coordinate. The general method of dispersion analysis described in the previous subsection can be applied for the HRT method. We proceed to describe our new findings, which in the lowest order case includes an exact dispersion relation for the HRT method.

In the $p = 0$ case, after statically condensing the element matrices and following the procedure leading to (72), we find that the discrete wavenumber k^h for the HRT method satisfies the 2D dispersion relation

$$(c_1^2 + c_2^2)\left(2(hk)^2 - 12\right) + c_1^2 c_2^2 \left(4(hk)^2 + 48\right) + (hk)^2 - 24 = 0, \tag{79}$$



(A) Dispersive error, $p = 0$

(B) Dispersive error, $p = 1$

(C) Dissipative error, $p = 0$

(D) Dissipative error, $p = 1$

(E) Total error, $p = 0$

(F) Total error, $p = 1$

FIGURE 4.5. Convergence rates as $kh \to 0$.

where c_j, as defined in (65), depends on k_j^h, which in turn depends on k^h. Similar to the HDG case, we now observe that the two equations

$$\big(2(hk_j)^2 + 12\big)\, c_j^2 + (hk_j)^2 - 12 = 0, \qquad j = 1, 2, \tag{80}$$

are sufficient conditions for (79) to hold. Indeed, if l_j is the left hand side above, then $l_1(2c_2^2 + 1) + l_2(2c_1^2 + 1)$ equals the left hand side of (79). The equations of (80) can immediately be solved:

$$hk_j^h = 2\cos^{-1}\left(\frac{12 - (hk_j)^2}{2\,(hk_j)^2 + 12}\right)^{1/2}$$

Hence, using (75) and simplifying using the same type of asymptotic expansions as the ones we previously used, we obtain

$$k^h h - kh = -\frac{\cos(4\,\theta) + 3}{96}\,(kh)^3 + O((kh)^5) \tag{81}$$

as $kh \to 0$. Comparing with (76), we find that in the lowest order case, *the HRT method has an error in wavenumber that is asymptotically one order smaller than the HDG method* for any propagation angle, irrespective of the value of τ.

To conclude this discussion, we report the results from numerically solving the nonlinear equation (79) for $k^h(\theta)$ for $\theta = 0$, as kh approaches zero. We have also calculated the analogue of (79) for the $p = 1$ case (following the procedure described in the previous subsection). Recall the dispersive, dissipative, and total errors in the wavenumbers, as defined in (62). After scaling by the mesh size h, these errors for both the HDG and the HRT methods are graphed in Figure 4.5 for $p = 0$ and $p = 1$. We find that the dispersive errors decrease at the same order for the HRT method and the HDG method with $\tau = 1$. While (81) suggests that the dissipative errors for the HRT method should be of higher order, our numerical results found them to be zero (up to machine accuracy). The dissipative errors also quickly fell to machine zero for the HDG method with the previously discussed “best” value of $\tau = \hat{\imath}\sqrt{3}/2$, as seen from Figure 4.5.

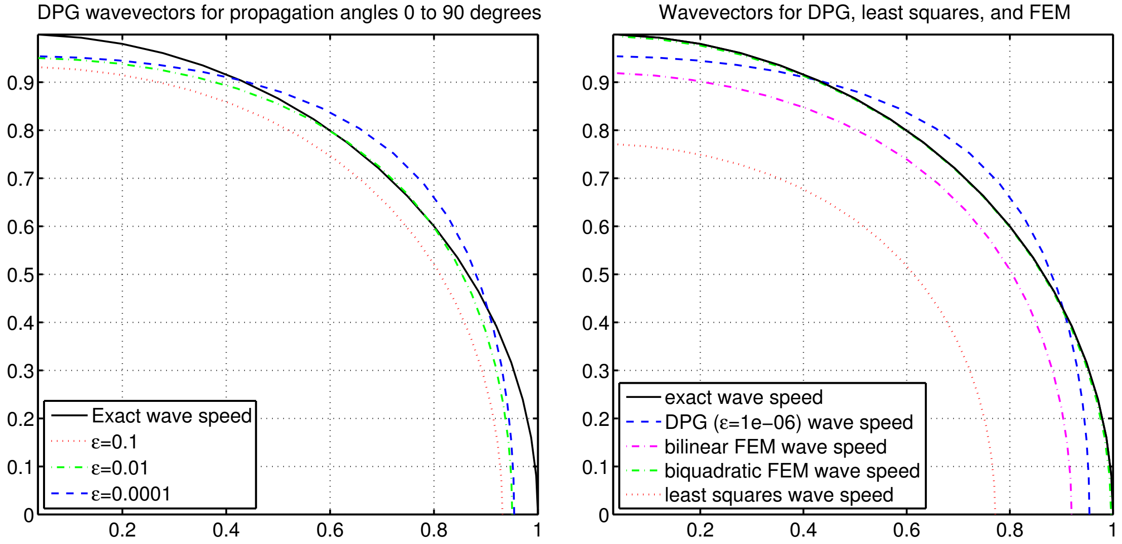
4.4. Dispersion analysis for the DPG$_\varepsilon$ method

Next, we apply the framework described in Section 4.2 to the lowest order DPG stencil discussed in Section 2.3. Since there are three different types of stencils (see Figure 2.2), we have $S = 3$. Unknowns of the first type, denoted by “●” in Figure 2.2, represent the DPG method’s approximation to the value of ϕ at the nodes $\vec{\jmath}h$ for all $\vec{\jmath} \in \mathbb{Z}^2$. The stencil of the first type is the one shown in Figure 2.2a. The unknowns of the second type represent the method’s approximation to the vertical components of $\vec{u}$ on the midpoints of horizontal edges, i.e., at all points in $(h\mathbb{Z} + h/2) \times h\mathbb{Z}$. These unknowns are denoted by “🡡” and have the stencil portrayed in Figure 2.2b. Similarly, the third type of stencil and unknown are as in Figure 2.2c. To summarize, (65) in the lowest order DPG case, becomes

$$\begin{aligned}
\psi_{1,\vec{\jmath}} &= \hat{\phi}_h(\vec{x}_{\vec{\jmath}}) = a_1 e^{\hat{\imath}\vec{\kappa}_h\cdot\vec{x}_{\vec{\jmath}}} && \forall \vec{x}_{\vec{\jmath}} \in (h\mathbb{Z})^2,\\
\psi_{2,\vec{\jmath}} &= \hat{u}_h(\vec{x}_{\vec{\jmath}}) = a_2 e^{\hat{\imath}\vec{\kappa}_h\cdot\vec{x}_{\vec{\jmath}}} && \forall \vec{x}_{\vec{\jmath}} \in (h\mathbb{Z} + h/2) \times h\mathbb{Z},\\
\psi_{3,\vec{\jmath}} &= \hat{u}_h(\vec{x}_{\vec{\jmath}}) = a_3 e^{\hat{\imath}\vec{\kappa}_h\cdot\vec{x}_{\vec{\jmath}}} && \forall \vec{x}_{\vec{\jmath}} \in h\mathbb{Z} \times (h\mathbb{Z} + h/2).
\end{aligned}$$

The condensed 8×8 DPG matrices, discussed in Section 2.3, can be used to compute the stencil weights $D_{t,s,\vec{l}}$ in each of the three cases, which in turn lead to the nonlinear equation (67) for any given propagation angle θ.

We numerically solved the nonlinear equation for k^h, for various choices of θ (propagation angle), r (enrichment degree), ε (scaling factor in the V-norm), and h (mesh size). The first important observation from our computations is that the



(A) Dependence of $\mathrm{Re}(\vec{\kappa}_h)$ on ε

(B) Comparison of methods

FIGURE 4.6. The curves traced out by the discrete wavevectors $\vec{\kappa}_h$ as θ goes from 0 to $\pi/2$. These plots were obtained using $k = 1$ and $h = 2\pi/4$.

computed wavenumbers k^h are complex numbers. They lie close to k in the complex plane. The small but nonzero imaginary parts of k^h indicate that the DPG$_\varepsilon$ method has dissipation errors, in addition to dispersion errors. The results are described in more detail below.

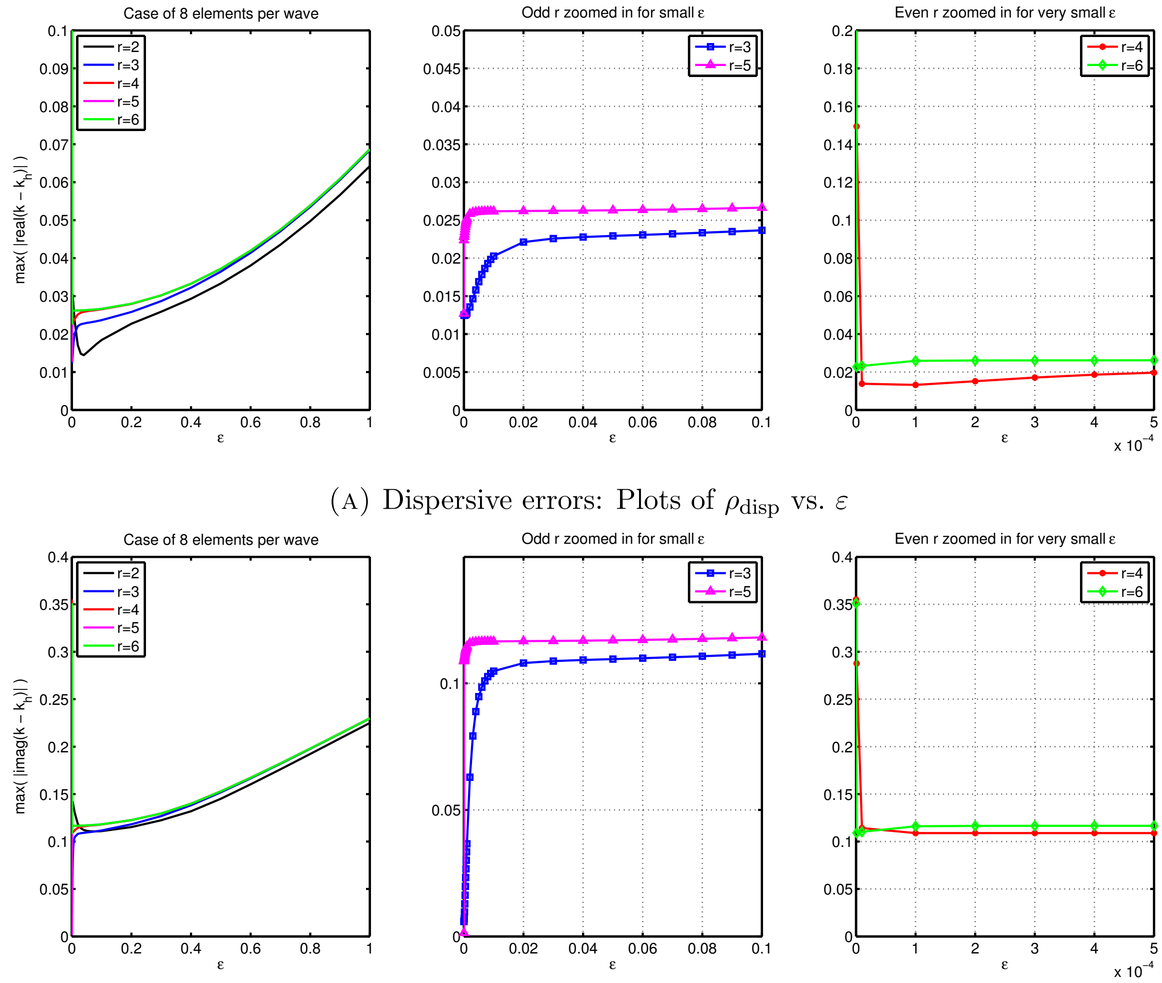
4.4.1. Dependence on θ. To understand how dispersion errors vary with propagation angle θ, we fix the exact wavenumber k appearing in the Helmholtz equation to 1 (so the wavelength is 2π) and examine the computed $\mathrm{Re}(k^h)$ for each θ.

One way to visualize the results is through a plot of the corresponding discrete wavevectors $\mathrm{Re}(\vec{\kappa}_h)$ vs. $\vec{k}$ for every propagation direction θ. Due to symmetry, we only need to examine this plot in the region $0 \le \theta \le \pi/2$. We present these plots for the case $r = 3$ in Figure 4.6. We fix $h = 2\pi/4$. (This corresponds to four elements per wavelength if the propagation direction is aligned with a coordinate axis.) In Figure 4.6a, we plot the curve traced out by the endpoints of the discrete wavevectors $\vec{\kappa}_h$. We see that as ε decreases, the curve gets closer to the (solid) circle traced out by the exact wavevector $\vec{k}$. This indicates better control of dispersive errors with decreasing ε (cf. Theorem 2.2.1).

In Figure 4.6b, we compare the $\vec{\kappa}_h$ obtained using the lowest order DPG method with the discrete wavenumbers of the standard lowest order (bilinear) finite element method (FEM). Clearly the wavenumbers obtained from the DPG method are closer to the exact $k = 1$ than those obtained by bilinear FEM. However, since the lowest order DPG method has a larger stencil than bilinear FEM, one may argue that a better comparison is with methods having the same stencil size. We therefore compare the DPG method with two other methods which have stencils of exactly the same shape and size: (i) The biquadratic FEM (which after condensation has three stencils of the same size as the lowest order DPG method), and (ii) the conforming first order $L^2(\Omega)$ least-squares method using the lowest order Raviart-Thomas and Lagrange spaces (which has no interior nodes to condense out). While the wavenumbers from the DPG method did not compare favorably with the biquadratic FEM of (i), we found that the DPG method performs better than the least-squares method in (ii).

4.4.2. Dependence on ε and r. We have seen in Figure 4.6 that the discrete wavenumber k^h is a function of the propagation angle θ. We now examine the maximum discrepancy between real and imaginary parts of k^h and k over all angles. Recall the definitions of dispersive error ρ_{disp} and dissipative error ρ_{dissip} of (62). Fixing $k = 1$ and $h = 2\pi/8$ (so that there are about eight elements per wavelength), we examine these quantities as a function of r and ε in Figure 4.7. The first of the plots in Figures 4.7a and 4.7b show that the errors decrease as ε decreases from 1 to about 0.1. In view of Theorem 2.2.1, we expected this decrease.

However, the behavior of the method for smaller ε is curious. In the remaining plots of Figure 4.7 we see that when r is odd, the errors continue to decrease for



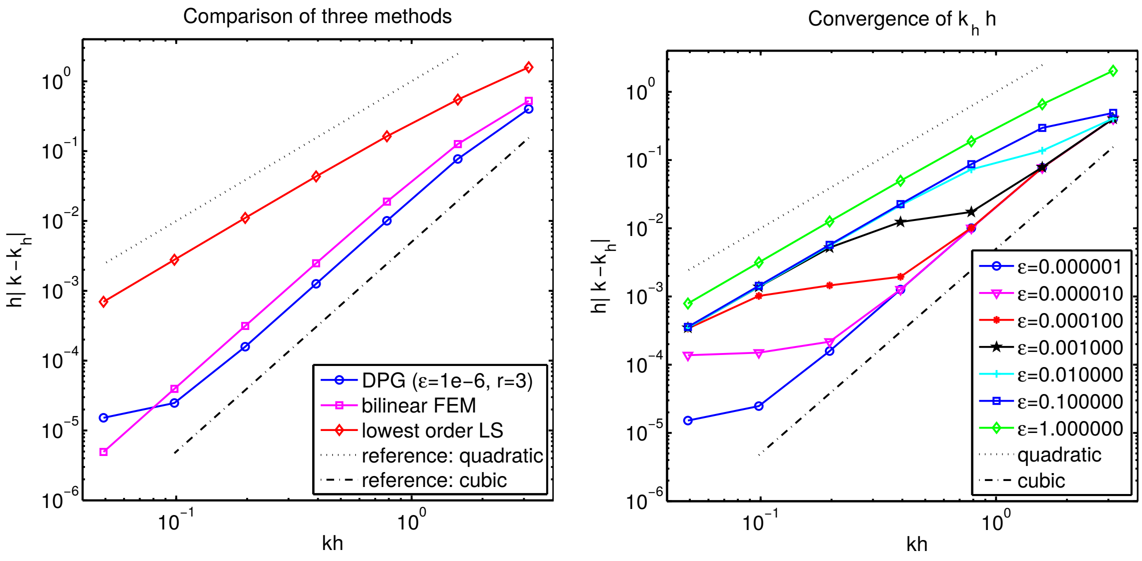
(A) Dispersive errors: Plots of ρ_{disp} vs. ε

(B) Dissipative errors: Plots of ρ_{dissip} vs. ε

FIGURE 4.7. The discrepancies between exact and discrete wavenumbers as a function of ε, when $k = 1$ and $h = 2\pi/8$.

smaller ε, while for even r, the errors start to increase as $\varepsilon \to 0$. This suggests the presence of discrete effects due to the inexact computation of test functions. We do not yet understand it enough to give a theoretical explanation.

4.4.3. Dependence on k. Now we examine how k^h depends on k. First, let us note that the matrix F in (67) only depends on kh. (This can be seen, for instance, from (45) and noting how the stencil weights depend on the entries of B.) Hence, we



(A) Plot of $|k^h h - kh|$ for three methods (B) Case of DPG with $r = 3$ and various ε

FIGURE 4.8. Rates of convergence of $|k^h h - kh|$ to zero for small kh, in the case of propagation angle $\theta = 0$.

will study how $k^h h$ depends on the normalized wavenumber kh, restricting ourselves to the case of $\theta = 0$.

In Figure 4.8a, we plot (in logarithmic scale) the absolute value of $k^h h - kh$ vs. kh for the standard bilinear FEM, the lowest order L^2 least-squares method (marked LS), and the DPG$_\varepsilon$ method with $\varepsilon = 10^{-6}, r = 3$. We observe that while $|k^h h - kh|$ appears to decrease at $O(kh)^2$ for the least squares method, it appears to decrease at the higher rate of $O(kh)^3$ for the FEM and DPG$_\varepsilon$ cases considered in the same graph. For easy reference, we have also plotted lines indicating slopes corresponding to $O(kh)^2$ and $O(kh)^3$ decrease, marked "quadratic" and "cubic", resp., in the same graph.

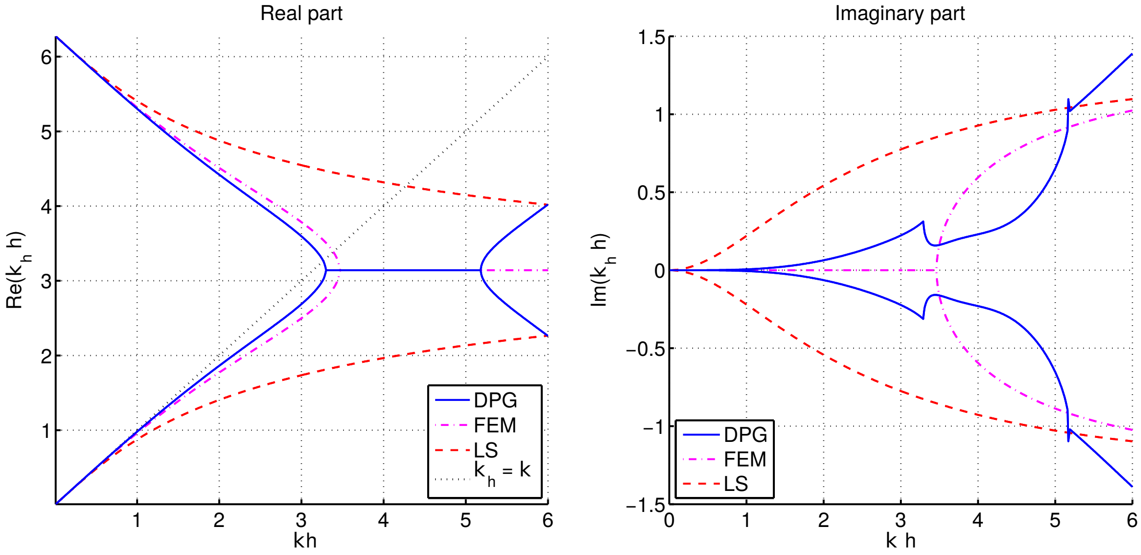
Note that a convergence rate of $|k^h h - kh| = O(kh)^3$ implies that the difference between discrete and exact wavenumbers goes to zero at the rate

$$|k_h - k| = k\, O(kh)^2.$$

This shows the presence of the so-called [**3**] pollution errors: For instance, as k increases, even if we use finer meshes so as to maintain kh fixed, the error in wavenumbers will continue to grow at the rate of $O(k)$. Our results show that pollution errors are present in all the three methods considered in Figure 4.8a. The difference in convergence rates, e.g., whether $|k_h - k|$ converges to zero at the rate $k\,O(kh)^2$ or at the rate $k\,O(kh)$, becomes important, for example, when trying to answer the following question: What h should we use to obtain a fixed error bound for $|k^h - k|$ for all frequencies k? While methods with convergence rate $kO(kh)$ would require $h \approx k^{-2}$, methods with convergence rate $kO(kh)^2$ would only require $h \approx k^{-3/2}$.

Next, consider Figure 4.8b, where we observe interesting differences in convergence rates within the DPG$_\varepsilon$ family. While the DPG$_\varepsilon$ method for $\varepsilon = 1$ exhibits the same quadratic rate of convergence as the least-squares method, we observe that a transition to higher rates of convergence progressively occur as ε is decreased by each order of magnitude. The $\varepsilon = 10^{-6}$ case shows a rate virtually indistinguishable from the cubic rate in the considered range. The convergence behavior of the DPG$_\varepsilon$ method thus seems to vary "in between" those of the least-squares method and the standard FEM as ε is decreased. The values of kh considered in these plots are $2\pi/2^l$ for $l = 1, 2, \ldots, 7$, which cover the numbers of elements per wavelength in usual practice.

Next, we consider a wider range of kh following [**44**], where such a study was done for standard finite elements, separating the real and imaginary parts of $k^h h$. Our results for the case of $\theta = 0$ are collected in Figure 4.9. To discuss these results, let us first recall the behavior of the standard bilinear finite element method (whose discrete wavenumbers are also plotted in dash-dotted curve in Figure 4.9). From its well-known dispersion relation (see e.g, [**1**]), we observe that if $k^h h$ solves the dispersion relation, then $2\pi - k^h h$ also solves it. Accordingly, the plot in Figure 4.9a is symmetric about the horizontal line at height π. Furthermore, as already shown



(A) $\mathrm{Re}(k^h h)$ as a function of kh

(B) $\mathrm{Im}(k^h h)$ as a function of kh

FIGURE 4.9. A comparison of discrete wavenumbers obtained by three lowest order methods in the case of propagation angle $\theta = 0$.

in [**44**], $k^h h$ is real-valued in the range $0 < kh < \sqrt{12}$. The threshold value $kh = \sqrt{12}$ was called the "cut-off" frequency. (Note that in the regime $kh > \pi$, we have less than two elements per wavelength. Note also that $\sqrt{12} > \pi$.) As can be seen from Figures 4.9a and 4.9b, in the range $\sqrt{12} < kh \le 6$, the bilinear finite elements yield $k^h h$ with a constant real part of π and nonzero imaginary parts of increasing magnitude.

We observed a somewhat similar behavior for the DPG$_\varepsilon$ method – see the solid curves of Figure 4.9, which were obtained after calculating F explicitly using the computer algebra package Maple, for the lowest order DPG$_\varepsilon$ method, setting $r = 3$ and $\varepsilon = 0$. The major difference between the DPG$_\varepsilon$ and FEM results is that $k^h h$ from the DPG$_\varepsilon$ method was not real-valued even in the regime where FEM wavenumbers were real. It seems difficult to define any useful analogue of the cut-off frequency in this situation. Nonetheless, we observe from Figures 4.9a and 4.9b that there is a segment of constant real part of value π, before which the imaginary part of $k^h h$ is relatively small. As the number of mesh elements per wavelength increases (i.e., as kh becomes smaller), the imaginary part of $k^h h$ becomes small. We therefore expect the diffusive errors in the DPG$_\varepsilon$ method to be small when kh is small. Finally, we also conclude from Figure 4.9 that both the dispersive and dissipative errors are better behaved for the DPG$_\varepsilon$ method when compared to the L^2 least-squares method.

4.5. Comparison of the DPG$_\varepsilon$ and HDG methods

We now collate selected results from the previous sections to directly compare the dispersive errors of the DPG$_\varepsilon$ and HDG methods. We set the parameters (ϵ and r for the DPG$_\varepsilon$ method, τ for the HDG method) to values that have already been shown to perform well.

Figure 4.10 shows the real parts of the computed wavenumbers $\text{Re}(k^h(\theta))$ for both methods. We see that, for fixed $k = 1$ and $h = \pi/4$, the error in the computed wavenumber of the lowest order DPG$_\varepsilon$ method is smaller than that of the order $p = 0$ HDG method, but larger than that of the order $p = 1$ HDG method. It is more appropriate to compare the lowest order DPG$_\varepsilon$ method with the order $p = 1$ HDG method, since both methods have local matrices of size 8×8. The order $p = 0$ HDG method has a local matrix of size 4×4, so lower accuracy is to be expected.

Figure 4.11, which shows the rates of convergence of $|kh - k^h h|$ as $k^h h$ approaches zero for a fixed angle θ, indicates that the order $p = 1$ HDG method not only has smaller error than the lowest order DPG$_\varepsilon$ method, but also converges at a higher rate.

From this we conclude that, with regard to the error in the computed wavenumber k^h for the methods we have considered here, the HDG method with an optimal stabilization parameter performs better than the DPG$_\varepsilon$ method.

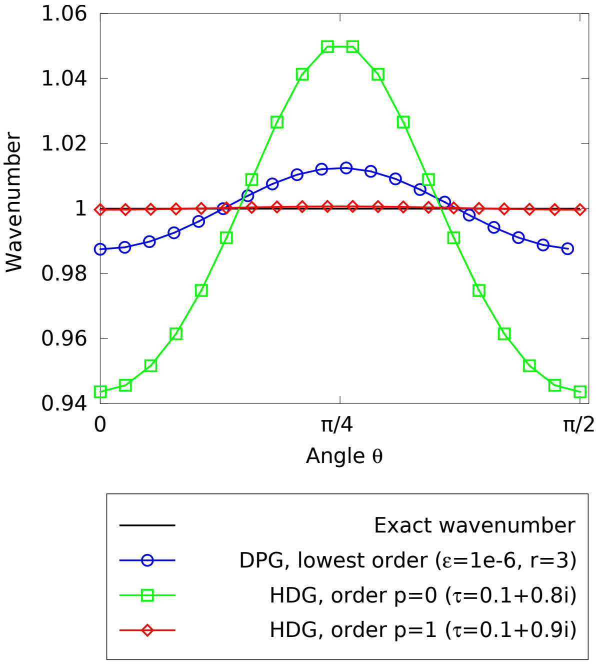


FIGURE 4.10. Real part of the numerical wavenumber $\mathrm{Re}(\vec{k}^h(\theta))$ as a function of θ. Here, $k = 1$ and $h = \pi/4$.

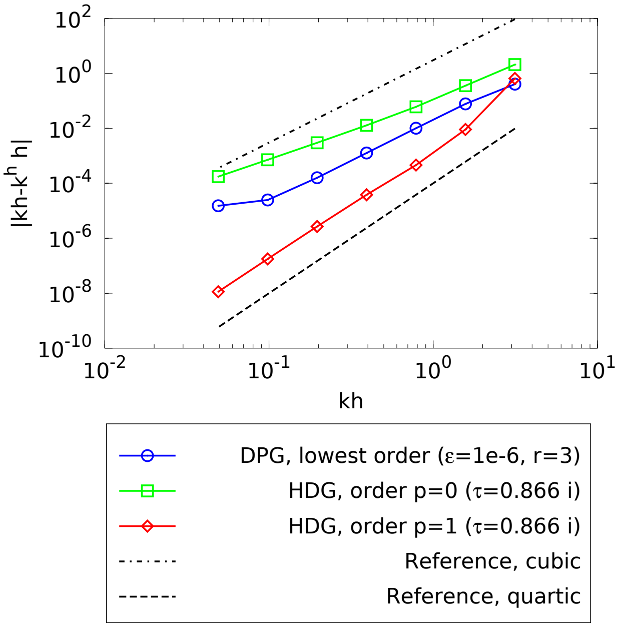


FIGURE 4.11. Rates of convergence of $|kh - k^h h|$ to zero for small $k^h h$, in the case of propagation angle $\theta = 0$.

5

Modeling an array of annular nanogaps

In this chapter, we present numerical work using the DPG method to solve a 3D Maxwell system for a realistic problem: the nanogap array studied in [**39**]. This structure is designed to facilitate extraordinary optical transmission (EOT) [**19, 35**], which refers to the transmission of light through sub-wavelength apertures. EOT is possible via the excitation of surface plasmon polaritons (SPPs), which are electromagnetic surface waves that occur at the interface of a conductor and a dielectric material when the EM fields are coupled with the oscillations of the conductor's electrons. EOT has emerging applications in sensing and in the design of optical switches.

After addressing some notational conventions below, we state the problem in Section 5.1. The DPG method we use is similar to the method analyzed in Section 6.2 of [**8**]. There, however, Dirichlet boundary conditions were used. The DPG method for the nanogap problem, which involves periodic and radiation boundary conditions, is described in Section 5.2. This method, however, suffers instability for small wavenumbers. In Section 5.3, we demonstrate the instability using a simplified two-layer geometry for which an exact solution is known. A modification to the method that somewhat ameliorates the instability is then presented in Subsection 5.3.2.

To implement the DPG method using the NGSolve finite element software package, new integrators were written for the DPG shared library add-on [**26**]. This and other implementation details are discussed in Section 5.4. Results of the simulation are presented in Section 5.5.

5.0.1. Time-harmonic sign convention. The references [**39**] and [**8**] use the $-\hat{\imath}\omega t$ sign convention for the time-harmonic assumption, instead of the $+\hat{\imath}\omega t$ convention that we assumed in (10). To facilitate the comparison of our work with these references, we shall modify the derivation of Maxwell's equations presented in Subsection 1.3.2. Throughout this chapter, the time-harmonic fields are defined by

$$\begin{aligned}
\vec{\mathcal{E}}(\vec{x},t) &= \mathrm{Re}\left(\vec{E}(\vec{x})e^{-\hat{\imath}\omega t}\right),\\
\vec{\mathcal{D}}(\vec{x},t) &= \mathrm{Re}\left(\vec{D}(\vec{x})e^{-\hat{\imath}\omega t}\right),\\
\vec{\mathcal{H}}(\vec{x},t) &= \mathrm{Re}\left(\vec{H}(\vec{x})e^{-\hat{\imath}\omega t}\right),\\
\vec{\mathcal{B}}(\vec{x},t) &= \mathrm{Re}\left(\vec{B}(\vec{x})e^{-\hat{\imath}\omega t}\right),\\
\vec{\mathcal{J}}_a(\vec{x},t) &= \mathrm{Re}\left(\vec{J}_a(\vec{x})e^{-\hat{\imath}\omega t}\right).
\end{aligned}$$

instead of (10) and (11). The definition (12) is likewise replaced by

$$\hat{\epsilon}_r = \epsilon_r + i\frac{\sigma}{\omega\epsilon_0},$$

resulting in the Maxwell system

$$-\hat{\imath}\omega\mu\vec{H} + \vec{\nabla}\times\vec{E} = \vec{0}, \tag{82a}$$

$$-\hat{\imath}\omega\hat{\epsilon}\vec{E} - \vec{\nabla}\times\vec{H} = -\vec{J}. \tag{82b}$$

For the nanogap problem, μ is constant (i.e., $\mu_r \equiv 1$). The DPG method is then based on the second order equation,

$$\vec{\nabla}\times(\vec{\nabla}\times\vec{E}) - k^2\vec{E} = \vec{F}, \tag{83}$$

obtained by eliminating $\vec{H}$ from (82b) and setting $\vec{F} = -\hat{\imath}\omega\mu\vec{J}$. The boundary conditions will be discussed in the next section.

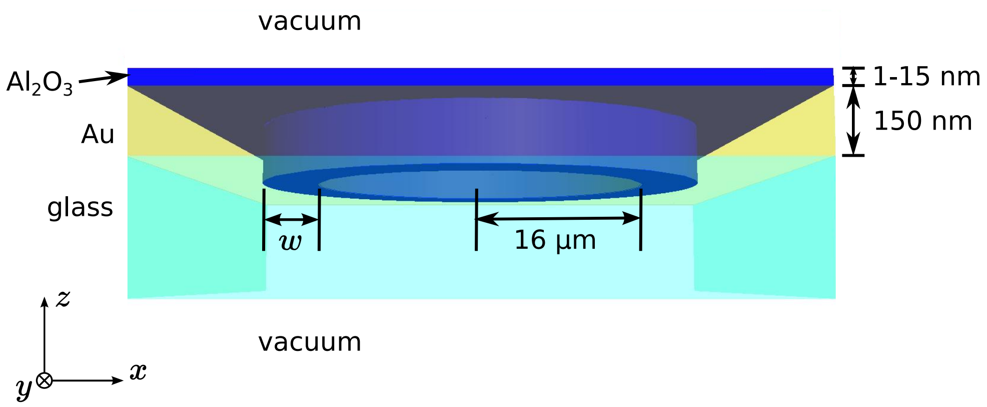


FIGURE 5.1. One period of the nanogap array in the domain $\{(x, y, z) : |x| < 25 \text{ µm}, |y| < 25 \text{ µm}\}$. (Not to scale.)

5.1. The nanogap problem

In [**39**], arrays of annular nanogaps are fabricated and experiments are conducted to detect the thickness of dielectric films. For this purpose, the nanogap array is designed to exhibit ring resonances when plane waves with frequencies in the 0.1 to 1.0 THz range, traveling in the positive z-direction, are transmitted through the structure. The authors of [**39**] compare their experimental results to simulations using the HDG method. We attempt to reproduce their simulations using the DPG method as a first step towards the longer term goal of designing efficient iterative techniques for such realistic simulations.

The nanogap array is periodic in x and y, with a period of 50 µm in each direction. Figure 5.1 shows one period of the array. This figure is for illustration purposes and is not to scale. A 150 nm thin gold (Au) film on a glass substrate has an annular aperture of diameter 32 µm etched out to form a nanogap. The gap size w is in the 1 to 10 nm range. The gold film is covered with a (dielectric) layer of aluminum oxide (Al_2O_3), which also fills the inside of the nanogap. The thickness of the Al_2O_3 layer is 1 to 15 nm. The thickness of the glass substrate, which is not specified in [**39**], is set to 100 nm. Vacuum regions above and below the structure extend to $z = \pm\infty$.

All lengths in our DPG computation are in units of micrometers, not meters. In converted units, then,

$$\epsilon_0 = 8.845 \times 10^{-30}\ \mathrm{s}^4\cdot\mathrm{A}^2\cdot\mathrm{kg}^{-1}\cdot\mathrm{\mu m}^{-3}$$

and

$$\mu_0 = 4\pi \times 10^{-1}\ \mathrm{kg}\cdot\mathrm{\mu m}\cdot\mathrm{A}^{-2}\cdot\mathrm{s}^{-2}.$$

The permeabilities of the glass, the Al_2O_3, and the Au are all assumed to equal the vacuum permeability. That is, $\mu \equiv \mu_0$. The permittivities of the materials, as specified in [**39**] based on references cited therein, are

$$\begin{aligned}
\epsilon_{\text{vacuum}} &= \epsilon_0, \\
\epsilon_{\text{glass}} &= 1.95\epsilon_0, \\
\epsilon_{\mathrm{Al_2O_3}} &= \begin{cases} 2.34\epsilon_0, & \text{for } w = 10 \text{ nm}, \\ 2.12\epsilon_0, & \text{for } w = 5 \text{ nm}, \\ 1.73\epsilon_0, & \text{for } w = 2 \text{ nm}, \end{cases} \\
\epsilon_{\text{Au}}(\omega) &= \left(1 - \frac{\omega_p^2}{\omega(\omega + \hat{\imath}\gamma)}\right)\epsilon_0,
\end{aligned}$$

where $\omega_p = 1.37 \times 10^4$ THz and $\gamma = 40.7$ THz. The resulting wavenumber k is in units of $\mathrm{\mu m}^{-1}$, and is presented in Table 5.1 for the lowest and highest frequencies f considered.

The computational domain

$$\Omega = \{(x, y, z) \in \mathbb{R}^3 \ :\ |x| < 25,\ |y| < 25,\ \text{and } -200 < z < 100\}$$

is a properly scaled version of Figure 5.1, with the glass-Au interface contained within the plane $\{z = 0\}$. The boundary of Omega has a periodic boundary condition in the

	f = 0.1 THz	f = 1 THz
$k_0 = k_{\text{vacuum}}$	0.002096	0.02096
k_{glass}	0.002927	0.02927
$k_{\text{Al}_2\text{O}_3}$	0.003206	0.03206
k_{Au}	$0.7054\hat{\imath}$	$6.972\hat{\imath}$

TABLE 5.1. Wavenumber k in units of µm^{-1}, for frequencies 0.1 THz and 1 THz, to indicate the range of wavenumbers considered. Within the Al_2O_3 region, $k_{\text{Al}_2\text{O}_3}$ depends on the nanogap size w. Here, w = 10 nm.

x-direction, identifying $\partial\Omega \cap \{x = -25\}$ with $\partial\Omega \cap \{x = 25\}$, and a periodic boundary condition in the y-direction identifying $\partial\Omega \cap \{y = -25\}$ with $\partial\Omega \cap \{y = 25\}$. To impose these conditions concisely, denote $\Gamma_x = \partial\Omega \cap \{x = -25\}$ and $\Gamma_y = \partial\Omega \cap \{y = -25\}$. At the remaining boundary components, which we denote $\Gamma_z = \partial\Omega \cap \{z = -200 \text{ or } 100\}$, a Silver-Müller boundary condition is used to approximate a non-reflecting boundary. That is, we solve

$$\vec{\nabla}\times(\vec{\nabla}\times\vec{E}) - k^2\vec{E} = \vec{0}, \qquad \text{in } \Omega, \tag{84a}$$

$$\vec{E}(\vec{x}) = \vec{E}(\vec{x} + 50\hat{a}_x), \qquad \text{on } \Gamma_x, \tag{84b}$$

$$\vec{E}(\vec{x}) = \vec{E}(\vec{x} + 50\hat{a}_y), \qquad \text{on } \Gamma_y, \tag{84c}$$

$$\hat{\nu}\times(\vec{\nabla}\times\vec{E}) + \hat{\imath}k\ \hat{\nu}\times(\vec{E}\times\hat{\nu}) = \hat{\nu}\times(\vec{\nabla}\times\vec{E}^{\text{inc}}) + \hat{\imath}k\ \hat{\nu}\times(\vec{E}^{\text{inc}}\times\hat{\nu}), \qquad \text{on } \Gamma_z, \tag{84d}$$

where $\hat{a}_x, \hat{a}_y$, and $\hat{a}_z$ denote the standard unit vectors, and

$$\vec{E}^{\text{inc}} = \begin{bmatrix} e^{\hat{\imath}k_0 z} \\ 0 \\ 0 \end{bmatrix}$$

is the (time-harmonic) incident plane wave with wavenumber $k_0 = 2\pi f\sqrt{\mu_0\epsilon_0}$, traveling in the $+z$-direction and polarized in the x-direction.

It is convenient to reformulate the problem and solve for the scattered field $\vec{E}^{\mathrm{s}} = \vec{E} - \vec{E}^{\mathrm{inc}}$ instead of $\vec{E}$. Subtracting $\vec{\nabla}\times(\vec{\nabla}\times\vec{E}^{\mathrm{inc}}) - k_0^2\vec{E}^{\mathrm{inc}} = \vec{0}$ from (84a) and rewriting (84d), the scattered field satisfies

$$\text{(85a)} \qquad \vec{\nabla}\times(\vec{\nabla}\times\vec{E}^{\mathrm{s}}) - k^2\vec{E}^{\mathrm{s}} = (k^2 - k_0^2)\vec{E}^{\mathrm{inc}}, \qquad \text{in } \Omega,$$

$$\text{(85b)} \qquad \hat{\nu}\times(\vec{\nabla}\times\vec{E}^{\mathrm{s}}) + \hat{\imath}k\ \hat{\nu}\times(\vec{E}^{\mathrm{s}}\times\hat{\nu}) = \vec{0}, \qquad \text{on } \Gamma_z,$$

as well as the periodic boundary conditions in the x- and y-directions. We note that $\vec{E}^s$ satisfies the periodic boundary conditions because both $\vec{E}$ and $\vec{E}^{\mathrm{inc}}$ satisfy them. (In general, an incident wave traveling in a direction other than the z-direction would not satisfy the periodic boundary conditions. In the scattered wave formulation, Bloch coefficients would be introduced to modify each periodic boundary condition with a phase correction. We do not require this generalization, however.)

5.2. The DPG method for Maxwell's equations

The method we use to solve the nanogap problem is the primal DPG method, based on the second order formulation (83). In order to state the method, we first define the function spaces and derive a variational formulation. Let Ω_h be a mesh Ω. Define the periodic trial space $X = X_0 \times \hat{X}$, where

$$X_0 = \{\vec{w} \in H(\mathrm{curl},\Omega) : \vec{w}(\vec{x}) = \vec{w}(\vec{x} + 50\hat{a}_x),\ \text{on } \Gamma_x,\ \text{and } \vec{w}(\vec{x}) = \vec{w}(\vec{x} + 50\hat{a}_y),\ \text{on } \Gamma_y\},$$

$$\hat{X} = \{\hat{\nu}\times\hat{w} \in H^{-1/2}(\mathrm{div},\partial\Omega_h) : \hat{\nu}\times\hat{w}(\vec{x}) = \hat{\nu}\times\hat{w}(\vec{x} + 50\hat{a}_x), \text{ on } \Gamma_x, \text{ and}$$
$$\hat{\nu}\times\hat{w}(\vec{x}) = \hat{\nu}\times\hat{w}(\vec{x} + 50\hat{a}_y), \text{ on } \Gamma_y\},$$

and define the discontinuous test space $Y = H(\text{curl}, \Omega_h)$. In this chapter, we use inner products with h subscripts defined by

$$\begin{aligned}
(\vec{w}, \vec{v})_h &= \sum_{K \in \Omega_h} (\vec{w}, \vec{v})_K, \\
\langle \vec{w}, \vec{v} \rangle_h &= \sum_{K \in \Omega_h} \langle \vec{w}, \vec{v} \rangle_{\partial K \cap \partial \Omega}, \quad \text{and} \\
\langle\!\langle \vec{w}, \vec{v} \rangle\!\rangle_h &= \sum_{K \in \Omega_h} \langle \vec{w}, \vec{v} \rangle_{\partial K}.
\end{aligned}$$

Multiplying (85a) by $\vec{v} \in Y$ and integrating by parts element-by-element, we have

$$(\vec{\nabla} \times \vec{E}^{\text{s}}, \vec{\nabla} \times \vec{v})_h - (k^2 \vec{E}^{\text{s}}, \vec{v})_h + \langle\!\langle \hat{\nu} \times (\vec{\nabla} \times \vec{E}^{\text{s}}), \vec{v} \rangle\!\rangle_h = ((k^2 - k_0^2) \vec{E}^{\text{inc}}, \vec{v})_h. \tag{86}$$

Now set $\hat{\nu} \times \hat{M} = -\hat{\imath}\, \hat{\nu} \times (\vec{\nabla} \times \vec{E}^{\text{s}})$ to be an independent unknown in $\hat{X}$. Substituting this in equation (86) leads to

$$b((\vec{E}^{\text{s}}, \hat{\nu} \times \hat{M}), \vec{v}) = l(\vec{v}), \qquad \forall \vec{v} \in Y,$$

where the bilinear form $b : X \times Y \to \mathbb{C}$ is defined by

$$b((\vec{E}^{\text{s}}, \hat{\nu} \times \hat{M}), \vec{v}) = (\vec{\nabla} \times \vec{E}^{\text{s}}, \vec{\nabla} \times \vec{v})_h - (k^2 \vec{E}^{\text{s}}, \vec{v})_h + \hat{\imath} \langle\!\langle \hat{\nu} \times \hat{M}, \vec{v} \rangle\!\rangle_h, \tag{87}$$

and the linear form $l : Y \to \mathbb{C}$ is

$$l(\vec{v}) = ((k^2 - k_0^2) \vec{E}^{\text{inc}}, \vec{v})_h.$$

Similarly, the radiation boundary condition (85b) leads to

$$c((\vec{E}^{\text{s}}, \hat{\nu} \times \hat{M}), (\vec{F}, \hat{\nu} \times \hat{W})) = 0, \qquad \forall (\vec{F}, \hat{\nu} \times \hat{W}) \in X \tag{88}$$

where $c : X \times X \to \mathbb{C}$ is defined by

$$(89)\quad c((\vec{E}^{\mathrm{s}}, \hat{\nu} \times \hat{M}), (\vec{F}, \hat{\nu} \times \hat{W})) = -\langle \hat{\nu} \times \hat{M} + k\hat{\nu} \times (\vec{E}^{\mathrm{s}} \times \hat{\nu}), \hat{\nu} \times \hat{W} + k\hat{\nu} \times (\vec{F} \times \hat{\nu}) \rangle_h.$$

Using the mixed Galerkin formulation [**17**], [**22**] and imposing (88) weakly, we obtain the DPG variational formulation: Find $(\vec{e}, \vec{E}^{\mathrm{s}}, \hat{\nu} \times \hat{M}) \in Y \times X_0 \times \hat{X}$ such that

$$(90a)\qquad (\vec{e}, \vec{v})_Y + b((\vec{E}^{\mathrm{s}}, \hat{\nu} \times \hat{M}), \vec{v}) = l(\vec{v}), \qquad \forall \vec{v} \in Y,$$

$$(90b)\quad \overline{b((\vec{F}, \hat{\nu} \times \hat{W}), \vec{e})} + c((\vec{E}^{\mathrm{s}}, \hat{\nu} \times \hat{M}), (\vec{F}, \hat{\nu} \times \hat{W})) = 0, \qquad \forall (\vec{F}, \hat{\nu} \times \hat{W}) \in X_0 \times \hat{X}.$$

In order to state the discrete problem, recall that, for a given domain D, $\mathcal{P}_p(D)$ denotes polynomials of degree at most p. Let $\mathcal{N}_p(D) = \mathcal{P}_{p-1}(D)^3 + \vec{x} \times \mathcal{P}_{p-1}(D)^3$ denote the Nédélec space of degree p [**37**], and define the trace map $\mathrm{tr}^D_{\mathrm{curl}} : H(\mathrm{curl}, D) \to H^{-1/2}(\mathrm{div}, \partial D)$ such that $\mathrm{tr}^D_{\mathrm{curl}}(\vec{w}) = \hat{\nu} \times \vec{w}|_{\partial D}$ for $\vec{w} \in H(\mathrm{curl}, D)$. The numerical solution is sought in the discrete subspace $Y_h \times X_{0,h} \times \hat{X}_h \subset Y \times X_0 \times \hat{X}$, where the component spaces are defined as

$$(91a)\qquad Y_h = \{\vec{v}_h \in Y \ : \ \vec{v}_h|_K \in \mathcal{N}_{p+3}(K) \text{ for all } K \in \Omega_h\},$$

$$(91b)\qquad X_{0,h} = \{\vec{w}_h \in X_0 \ : \ \vec{w}_h|_K \in \mathcal{P}_p(K)^3 \text{ for all } K \in \Omega_h\},$$

$$(91c)\qquad \hat{X}_h = \{\hat{\nu} \times \hat{w}_h \in \hat{X} \ : \ \hat{\nu} \times \hat{w}_h|_{\partial K} \in \mathrm{tr}^K_{\mathrm{curl}} \mathcal{P}_{p+1}(K)^3\}.$$

We thus obtain the discretization of (90): Find $(e_h, \vec{E}^{\mathrm{s}}_h, \hat{\nu} \times \hat{M}_h) \in Y_h \times X_{0,h} \times \hat{X}_h$ such that

$$(92a)\qquad (\vec{e}_h, \vec{v}_h)_Y + b((\vec{E}^{\mathrm{s}}_h, \hat{\nu} \times \hat{M}_h), \vec{v}_h) = l(\vec{v}_h),$$

$$(92b)\qquad \overline{b((\vec{F}_h, \hat{\nu} \times \hat{W}_h), \vec{e}_h)} + c((\vec{E}^{\mathrm{s}}_h, \hat{\nu} \times \hat{M}_h), (\vec{F}_h, \hat{\nu} \times \hat{W}_h)) = 0,$$

for all $\vec{v}_h \in Y_h$ and $(\vec{F}_h, \hat{\nu} \times \hat{W}_h) \in X_{0,h} \times \hat{X}_h$.

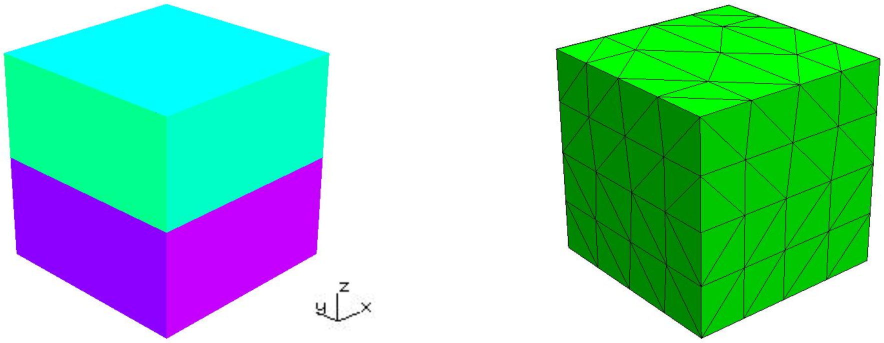
5.3. Instability for small wavenumbers

If the wavenumber k is set to zero in the Maxwell system (85), then the system no longer has a unique solution. Indeed, if $\vec{E}^{\mathrm{s}}$ is a solution, then $\vec{E}^{\mathrm{s}} + \vec{\nabla} f$ will also be a solution for any periodic function f. The DPG method presented in Section 5.2 took no special consideration of this and, consequently, it is not stable for small wavenumbers. After demonstrating the instability with an illustrative example, we propose an alternative DPG method that attempts to adapt the approach of [**16**] to our problem. The approach introduces a Lagrange multiplier to weakly enforce an additional constraint via additional terms in the bilinear form involving the gradient of the Lagrange multiplier. The approach is also similar to the Kikuchi method [**32, 5**]. In our adaptation, the additional constraint is derived from both Gauss's law and an analogue of Gauss's law for surface currents.

5.3.1. An illustrative example. We demonstrate the instability with an example problem that has the same boundary conditions as the nanogap, but which has greatly simplified features in the domain interior. Consider the cube $\Omega = [0,1]\times[0,1]\times[-0.5,0.5]$ with two subdomains forming two layers. The first layer, $\Omega \cap \{z <= 0\}$ is composed of material 1 with wavenumber k_1, and the second layer, $\Omega \cap \{z > 0\}$ is composed of material 2 with wavenumber k_2. Thus, Ω models an infinite domain of two layers, as shown in Figure 5.2.

If the incident field given is a plane wave traveling in the $+z$-direction given by

$$\vec{E}^{\mathrm{inc}} = \begin{bmatrix} e^{\hat{i}k_1 z} \\ 0 \\ 0 \end{bmatrix},$$



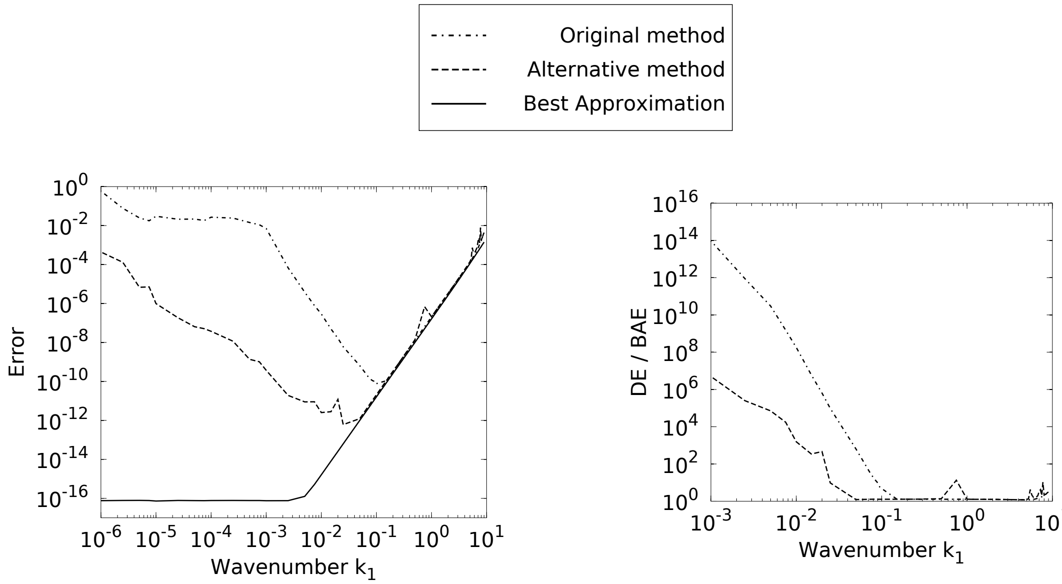
(A) Two-layer geometry. (B) Mesh of two-layer geometry.

FIGURE 5.2. Geometry of a two-layer problem with known exact solution.

then the exact solution to (85) is the scattered wave

$$\vec{E}^{\mathrm{s}} = \begin{cases} \begin{bmatrix} \Gamma e^{-\hat{\imath} k_1 z} \\ 0 \\ 0 \end{bmatrix}, & \text{on layer 1,} \\ \\ \begin{bmatrix} (1+\Gamma) e^{\hat{\imath} k_2 z} - e^{\hat{\imath} k_1 z} \\ 0 \\ 0 \end{bmatrix}, & \text{on layer 2.} \end{cases} \tag{93}$$

To illustrate the instability of the DPG method for small wavenumbers, we computed the order $p = 3$ DPG solution $\vec{E}_h^{\mathrm{s}}$ for a range of k_1, from $k_1 = 10^{-6}$ to $k_1 = 9$, with k_2 determined by $k_2 = 1.1k_1$. Figure 5.3a shows that the error $\|\vec{E}_h^{\mathrm{s}} - \vec{E}^{\mathrm{s}}\|$ of our original method (i.e., the method defined in Section 5.2) is minimal for $k \approx 0.1$, with a minimal value of about 10^{-10}. The error is significantly larger for small wavenumbers– worse than 0.01 for $k \leq 0.00075$. For the wavenumbers $k \approx 0.01$, which is approximately what is used for the nanogap problem, the error is about 3×10^{-7}, which is not terrible, but not great, either.



(A) Dashed and dash-dot curves are DEs of the respective methods. The solid curve is the BAE.

(B) Ratios DE/BAE.

FIGURE 5.3. Comparison of the discretization errors (DE) $\|\vec{E}^{\mathrm{s}}_h - \vec{E}^{\mathrm{s}}\|$ of the original method (92) and the alternative method (98), for a range of wavenumbers k_1, with k_2 determined by $k_2 = 1.1k_1$. The best approximation error (BAE) $\|\mathrm{proj}_{X_{0,h}}(\vec{E}^{\mathrm{s}}) - \vec{E}^{\mathrm{s}}\|$ is also included for reference.

We improve the stability of the method by introducing additional constraints and enforcing them via a Lagrange multiplier. This alternative method, described in the next section, does improve the stability to some degree, as shown in Figure 5.3a.

5.3.2. An alternative DPG method. Our alternative DPG method is based on a variational formulation that introduces a Lagrange multiplier to enforce an additional constraint derived from a form of Gauss's law for surface currents. We first derive the variational formulation from the following equations for $\vec{E}^{\mathrm{s}}$:

(94a)

$$\vec{\nabla}\times(\vec{\nabla}\times\vec{E}^{\mathrm{s}}) - k^2\vec{E}^{\mathrm{s}} = k_0^2(n^2-1)\vec{E}^{\mathrm{inc}}, \quad \text{in } \Omega,$$

(94b) $\vec{\nabla}\cdot(n^2(\vec{E}^{\mathrm{s}}+\vec{E}^{\mathrm{inc}})) = 0,$ in Ω, (Gauss's law)

(94c)

$$\hat{\nu}\times(\vec{\nabla}\times\vec{E}^{\mathrm{s}}) + \hat{\imath}k_0\vec{E}^{\mathrm{s}}_{\mathrm{t}} = \vec{0}, \quad \text{on } \Gamma_z, \qquad \text{(Radiation b.c.)}$$

(94d)

$$-k_0^2\vec{E}^{\mathrm{s}}\cdot\hat{\nu} + \hat{\imath}k_0\,\vec{\nabla}_{\mathrm{t}}\cdot\vec{E}^{\mathrm{s}}_{\mathrm{t}} = 0, \quad \text{on } \Gamma_z. \qquad \text{(Surface current constraint)}$$

Here, $n = k/k_0$ denotes the refractive index, $\vec{E}^{\mathrm{s}}_{\mathrm{t}} = \hat{\nu}\times(\vec{E}^{\mathrm{s}}\times\hat{\nu})$ the tangential component of $\vec{E}^{\mathrm{s}}$, and $\vec{\nabla}_{\mathrm{t}}\cdot\vec{v}$ the surface divergence of a surface vector $\vec{v}$. To state the variational formulation, we will also use the surface gradient, denoted $\vec{\nabla}_{\mathrm{t}}q$ for a surface function q. The Lagrange multiplier will lie in the periodic space

$$X_1 = \{q\in H^1(\Omega) : q(\vec{x}) = q(\vec{x}+50\hat{a}_x), \text{ on } \Gamma_x, \text{ and } q(\vec{x}) = q(\vec{x}+50\hat{a}_y), \text{ on } \Gamma_y\}.$$

Multiplying Maxwell's equation (94a) by a test function $\vec{v}\in Y$, integrating by parts, and using the radiation boundary condition (94c), we obtain

$$a(\vec{E}^{\mathrm{s}},\vec{v}) = k_0^2 g(\vec{v}), \qquad \forall\vec{v}\in Y,$$

with

$$a(\vec{E}^{\mathrm{s}},\vec{v}) = (\vec{\nabla}\times\vec{E}^{\mathrm{s}},\vec{\nabla}\times\vec{v}) - \hat{\imath}k_0\langle\vec{E}^{\mathrm{s}}_{\mathrm{t}},\vec{v}\rangle - k_0^2(n^2\vec{E}^{\mathrm{s}},\vec{v}),$$

and

$$g(\vec{v}) = ((n^2-1)\vec{E}^{\mathrm{inc}},\vec{v}).$$

Next, multiplying Gauss's law (94b) by a test function $q \in X_1$, integrating by parts, then using the surface Gauss law (94d) and, finally, integrating the surface integral by parts, we have

$$\overline{m(q, \vec{E}^{\mathrm{s}})} = g(\vec{\nabla} q),$$

with

$$m(q, \vec{E}^{\mathrm{s}}) = -(n^2 \vec{\nabla} q, \vec{E}^{\mathrm{s}}) - \frac{\hat{\imath}}{k_0} \langle \vec{\nabla}_{\mathrm{t}} q, \vec{E}^{\mathrm{s}}_{\mathrm{t}} \rangle.$$

Introducing a Lagrange multiplier $r \in X_1$, the variational formulation is: Find $(\vec{E}^{\mathrm{s}}, r) \in X_0 \times X_1$ such that

$$a(\vec{E}^{\mathrm{s}}, \vec{v}) + m(r, \vec{v}) = k_0^2 g(\vec{v}), \qquad \forall \vec{v} \in Y, \tag{95a}$$

$$\frac{1}{k_0} \overline{m(q, \vec{E}^{\mathrm{s}})} = \frac{1}{k_0} g(\vec{\nabla} q), \qquad \forall q \in X_1. \tag{95b}$$

We now use this variational formulation to guide the formulation of a new DPG method. It will be expressed as a mixed Galerkin method similar to (90), with the radiation boundary condition imposed in the same way using $c(\cdot, \cdot)$. Thus, we do not need to use the boundary term of $a(\cdot, \cdot)$ to impose the radiation boundary condition as above. We use $b(\cdot, \cdot)$, as defined in equation (87), instead, so (95) is replaced with

$$b((\vec{E}^{\mathrm{s}}, \hat{\nu} \times \hat{M}), \vec{v}) + m(r, \vec{v}) = k_0^2 g(\vec{v}), \qquad \forall \vec{v} \in Y, \tag{96a}$$

$$\frac{1}{k_0} \overline{m(q, \vec{E}^{\mathrm{s}})} = \frac{1}{k_0} g(\vec{\nabla} q), \qquad \forall q \in X_1. \tag{96b}$$

Collecting (96) into one equation,

$$\tilde{b}((\vec{E}^{\mathrm{s}}, \hat{\nu} \times \hat{M}, r), (\vec{v}, q)) = \tilde{l}(\vec{v}, q),$$

with

$$\tilde{b}((\vec{E}^{\mathrm{s}}, \hat{\nu} \times \hat{M}, r), (\vec{v}, q)) = b((\vec{E}^{\mathrm{s}}, \hat{\nu} \times \hat{M}), \vec{v}) + m(r, \vec{v}) + \frac{1}{k_0} \overline{m(q, \vec{E}^{\mathrm{s}})},$$

and

$$\tilde{l}(\vec{v},q) = k_0^2 g(\vec{v}) + \frac{1}{k_0} g(\vec{\nabla} q),$$

we can now state the continuous form of the new DPG method: Find $(\vec{e},\vec{E}^{\mathrm{s}},\hat{\nu}\times\hat{M},r) \in Y \times X_0 \times \hat{X} \times X_1$ such that

$$(\vec{e},\vec{v})_Y + \tilde{b}((\vec{E}^{\mathrm{s}},\hat{\nu}\times\hat{M},r),(\vec{v},q)) = \tilde{l}(\vec{v},q), \tag{97a}$$

$$\overline{\tilde{b}((\vec{F},\hat{\nu}\times\hat{W},q),(\vec{e},r))} + c((\vec{E}^{\mathrm{s}},\hat{\nu}\times\hat{M}),(\vec{F},\hat{\nu}\times\hat{W})) = 0, \tag{97b}$$

for all $\vec{v} \in Y$ and all $(\vec{F},\hat{\nu}\times\hat{W},q) \in X_0 \times \hat{X} \times X_1$. Defining the discrete space

$$X_{1,h} = \{q_h \in X_1 \ : \ q_h|_K \in \mathcal{P}_{p+1}(K) \text{ for all } K \in \Omega_h\},$$

the discretization of (97) is: Find $(\vec{e}_h,\vec{E}_h^{\mathrm{s}},\hat{\nu}\times\hat{M}_h,r_h) \in Y_h \times X_{0,h} \times \hat{X}_h \times X_{1,h}$ such that

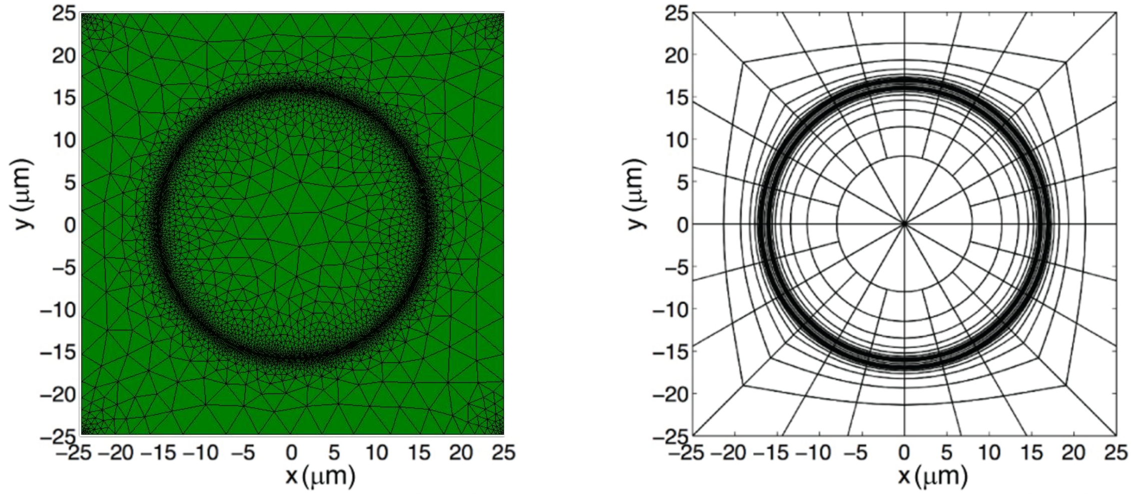
$$(\vec{e}_h,\vec{v}_h)_Y + \tilde{b}((\vec{E}_h^{\mathrm{s}},\hat{\nu}\times\hat{M}_h,r_h),(\vec{v}_h,q_h)) = \tilde{l}(\vec{v}_h,q_h), \tag{98a}$$

$$\overline{\tilde{b}((\vec{F}_h,\hat{\nu}\times\hat{W}_h,q_h),(\vec{e}_h,r_h))} + c((\vec{E}_h^{\mathrm{s}},\hat{\nu}\times\hat{M}_h),(\vec{F}_h,\hat{\nu}\times\hat{W}_h)) = 0, \tag{98b}$$

for all $\vec{v}_h \in Y_h$ and all $(\vec{F}_h,\hat{\nu}\times\hat{W}_h,q_h) \in X_{0,h} \times \hat{X}_h \times X_{1,h}$.

5.4. Implementation using NGSolve with the DPG shared library add-on

The NGSolve software library provides many efficient routines for FEM modeling. The DPG shared library add-on extends the functionality of NGSolve to accommodate the DPG framework. We have made several new contributions to the DPG shared library in order to solve the nanogap problem, including new DPG integrators to compute the left-hand side of (92), and periodic $H(\mathrm{curl},\Omega)$ spaces that are not included in NGSolve (version 6.1). We describe these contributions in more detail below.



(A) A mesh generated by Netgen.

(B) Mesh from [**39**].

FIGURE 5.4. Top-down views of mesh cross-sections near $z = 0.1$.

The first step towards modeling the nanogap problem was to write a geometry file for the nanogap and create a mesh using the Netgen mesh generator [**41**]. The geometry file serves a few purposes. First and foremost, it defines the nonintersecting top-level objects (Au, glass, etc.) via constructive solid geometry (CSG). It also identifies periodic boundaries, and marks the non-periodic boundary components so that the radiation boundary condition may later be imposed. Last but not least, the geometry file specifies that the mesh should have very small elements within and in the immediate vicinity above and below the nanogap, where we expect the solution to vary dramatically on a small scale.

Mesh generation presented a few difficulties. First, the algorithm fails to produce a mesh if the geometric features are extremely small. Even after adjusting the "mesh granularity" setting available in the Netgen GUI, we were not able to produce a mesh when the thickness of the Al_2O_3 layer was $1 - 10$nm, as it is in [**39**]. Fortunately, we can justify that, for the purposes of testing the DPG method, it is not a problem to use a thicker Al_2O_3 layer. It is explained in [**39**] that, while the resonance frequency is sensitive to the thickness of the Al_2O_3 layer for thin layers, it becomes less sensitive when the thickness of the Al_2O_3 layer exceeds the nanogap gap size w. Any thicknesses greater than the gap size will result in a resonance frequency that is practically the same as the resonance frequency resulting from a thickness equal to the gap size. Thus, we use an Al_2O_3 layer of 300 nm.

A second issue with the mesh generation step was the large number of elements. Meshes generated by Netgen typically had ~ 10^5 elements. A top-down view of a cross-section of one of our meshes is shown in Figure 5.4a. For comparison, the paper [**39**] created a mesh that was better tailored to the nanogap problem and which had only 22644 elements. Their mesh is shown in Figure 5.4b. Additional elements imply additional degrees of freedom, a larger linear system and, thus, more intensive computational time and memory requirements. The memory constraint limited how high we could set the order p for the discrete spaces (91).

Once a mesh has been created, the next step is to read it into NGSolve and specify everything needed to set up and solve the DPG system, including the finite element spaces, the bilinear form integrators that determine the matrix, the linear form integrators that determine the right-hand side, and the solution method to be used– either an iterative method or a direct solver– along with a preconditioner. All this is done within a single file with the extension `.pde`. We load the DPG shared library into the `nanogap.pde` file to access the periodic spaces and DPG integrators.

Assembling the matrix of equation (92) requires the computation of integrals of the following forms for each element K:

(1) $\int_K \vec{u}_h \cdot \vec{v}_h$

(2) $\int_K (\vec{\nabla}\times \vec{u}_h) \cdot (\vec{\nabla}\times \vec{v}_h)$

(3) $\int_K (\vec{\nabla}\times \vec{G}_h) \cdot (\vec{\nabla}\times \vec{v}_h)$

(4) $-\int_K k^2 \vec{G}_h \cdot \vec{v}_h$

$$
\begin{array}{ll}
(5) & \hat{\imath}\int_{\partial K}(\hat{\nu}\times\hat{V}_h)\cdot\vec{v}_h = \hat{\imath}\int_{\partial K}\hat{V}_h\cdot(\vec{v}_h\times\hat{\nu}) \\
(6) & -\int_{\partial K\cap\Gamma_z}|k|^2(\vec{G}_h\times\hat{\nu})\cdot(\vec{F}_h\times\hat{\nu}) \\
(7) & -\int_{\partial K\cap\Gamma_z}(\hat{V}_h\times\hat{\nu})\cdot(\hat{W}_h\times\hat{\nu}) \\
(8) & \int_{\partial K\cap\Gamma_z}k\vec{G}_h\cdot(\hat{W}_h\times\hat{\nu})
\end{array}
$$

where

$$
\begin{aligned}
&\vec{u}_h,\vec{v}_h \in Y_h,\\
&\vec{G}_h,\vec{F}_h \in X_{0,h},\ \text{and}\\
&\hat{V}_h,\hat{W}_h \in \hat{X}_h
\end{aligned}
$$

are basis functions in their respective spaces. Here, integrals (1) and (2) contribute to $(\vec{e}_h,\vec{v}_h)_Y$, integrals (3)–(5) contribute to $b((\vec{E}^{\mathrm{s}}_h,\hat{\nu}\times\hat{M}_h),\vec{v})$, and integrals (6)–(8) contribute to $c((\vec{E}^{\mathrm{s}}_h,\hat{\nu}\times\hat{M}_h),(\vec{F}_h,\hat{\nu}\times\hat{W}_h))$. All of the Petrov-Galerkin integrators (corresponding to integrals (3), (4), (5), and (8)) are new contributions to the DPG shared library [**26**]. We shall discuss one integrator, the C++ class CurlCurlPG, as an example. First, the definition of the class:

hcurlintegrators.cpp

```
 #include <fem.hpp>
 #include "dpgintegrators.hpp"
 using namespace ngsolve;
 .
 .
 .

////////////////////////////////////////////////////////////
// Integrate a(x) * curl U . curl V, where U and V are in
// Hcurl spaces, and a(x) is a complex or real coefficient

template<int D> class CurlCurlPG : public DPGintegrator {

  shared_ptr<CoefficientFunction> coeff_a;

  template<class SCAL>
  void T_CalcElementMatrix (const FiniteElement & base_fel,
              const ElementTransformation & eltrans,
              FlatMatrix<SCAL> elmat,
              LocalHeap & lh)  const ;
public:

  CurlCurlPG(const Array<shared_ptr<CoefficientFunction>> & coeffs)
    : DPGintegrator(coeffs), coeff_a(coeffs[2])  {
    cout << "Using DPG integrator " << Name() << " with components "
  << GetInd1()+1 << " and " << GetInd2()+1 << endl ;
  }

  virtual bool IsSymmetric() const { return !coeff_a->IsComplex() ; }

  virtual string Name () const { return "CurlCurlPG"; }

  virtual bool BoundaryForm () const { return false; }

  void CalcElementMatrix (const FiniteElement & base_fel,
            const ElementTransformation & eltrans,
            FlatMatrix<double> elmat,
            LocalHeap & lh) const {
    T_CalcElementMatrix<double>(base_fel,eltrans,elmat,lh);
  }

  void CalcElementMatrix (const FiniteElement & base_fel,
            const ElementTransformation & eltrans,
            FlatMatrix<Complex> elmat,
            LocalHeap & lh) const {
   T_CalcElementMatrix<Complex>(base_fel,eltrans,elmat, lh);
  }
};
.
.
.
```

The computation of the integral is done by the member function `CalcElementMatrix`, which calls a template function with a parameter type to distinguish between real and complex versions of the integrator. Let us now look at the template function.

hcurlintegrators.cpp

```
.
.
.

template<int D> template <class SCAL>
void CurlCurlPG<D>::T_CalcElementMatrix (const FiniteElement & base_fel,
                        const ElementTransformation & eltrans,
                        FlatMatrix<SCAL> elmat,
                        LocalHeap & lh) const {

  const CompoundFiniteElement &  cfel  // product space
    =  dynamic_cast<const CompoundFiniteElement&> (base_fel);

  const HCurlFiniteElement<D> & fel_u =  // U space
    dynamic_cast<const HCurlFiniteElement<D>&> (cfel[GetInd1()]);
  const HCurlFiniteElement<D> & fel_v =  // V space
    dynamic_cast<const HCurlFiniteElement<D>&> (cfel[GetInd2()]);

  elmat = SCAL(0.0);

  // Degrees of freedom
  IntRange ru = cfel.GetRange(GetInd1());
  IntRange rv = cfel.GetRange(GetInd2());
  int ndofu = ru.Size();
  int ndofv = rv.Size();

  FlatMatrixFixWidth<D> curl_um(ndofu,lh); // to store curl(U-basis)
  FlatMatrixFixWidth<D> curl_vm(ndofv,lh); // to store curl(V-basis)

  ELEMENT_TYPE eltype                  // get the type of element:
    = fel_u.ElementType();             // ET_TET in 3d.

  const IntegrationRule &              // Note: p = fel_u.Order()-1
    ir = SelectIntegrationRule(eltype, fel_u.Order()+fel_v.Order()-2);

  FlatMatrix<SCAL> submat(ndofv,ndofu,lh);
  submat = SCAL(0.0);

  for(int k=0; k<ir.GetNIP(); k++) {

    MappedIntegrationPoint<D,D> mip (ir[k],eltrans);
    // set curl(U-basis) and curl(V-basis) at mapped integration points
    fel_u.CalcMappedCurlShape( mip, curl_um );
    fel_v.CalcMappedCurlShape( mip, curl_vm );
    // evaluate coefficient
    SCAL fac = coeff_a -> T_Evaluate<SCAL>(mip);
    fac *= mip.GetWeight() ;

    //             [ndofv x D] * [D x ndofu]
    submat +=  fac * curl_vm  * Trans(curl_um) ;
  }

  elmat.Rows(rv).Cols(ru) += submat;

  if (GetInd1() != GetInd2())
    elmat.Rows(ru).Cols(rv) += Conj(Trans(submat));
}
.
.
.
```

We see that, for the element associated with ElementTransformation eltrans, the T_CalcElementMatrix function template adds the contribution of the integrator to the element matrix elmat by adding submat. The computation of submat is done in a loop over integration points. For each integration point, the integrand is evaluated which, for this example, involves the curl of the basis functions (i.e., shape functions) for both spaces. The curls are easily computed thanks to the built-in NGSolve function CalcMappedCurlShape, which sets curl_um and curl_vm. Other integrators differ mainly within the loop over integration points. The boundary integrators are also treated somewhat differently.

To understand what was involved in the implementation of the discrete periodic $H(\mathrm{curl}, \Omega)$ spaces, let us first look at the definition of our PeriodicHCurlSpace class from [**26**], which is derived from NGSolve's HCurlHigherOrderFESpace class.

periodichcurl.cpp

```
#include <comp.hpp>
using namespace ngcomp;
```

```
class PeriodicHCurlSpace : public HCurlHighOrderFESpace
{
private:

  Array<int> vertmapx;
  Array<int> vertmapy;
  Array<int> dofmapx;
  Array<int> dofmapy;

  // In 2D: if segment i is copied to segment j, then idx = j.
  // In 3D: index is the number of the "identify periodic " statement
  //        in geo file.

  int idx;
  int idy;

public:

  PeriodicHCurlSpace (shared_ptr<MeshAccess> ama, const Flags & flags);
  virtual ~PeriodicHCurlSpace ();
  virtual string GetClassName () const
    {
      return "PeriodicHCurlSpace";
    }

  virtual void Update (LocalHeap & lh);

  virtual void GetDofNrs (int elnr, Array<int> & dnums) const;
  virtual void GetSDofNrs (int selnr, Array<int> & dnums) const;
  virtual FiniteElement & GetFE (int enr, LocalHeap & lh) const;
  virtual FiniteElement & GetSFE (int enr, LocalHeap & lh) const;
  virtual FiniteElement & GetFE (ElementId ei, Allocator & alloc) const;
};
.
.
.
```

The `Update` function of a finite element space is called by NGSolve to create or update its table of degrees of freedom (known as the dof table) for the space. For a *periodic* space, we must ensure that the degrees of freedom that are geometrically associated with the periodic boundaries are correctly accounted for in the dof table.

For example, the x periodicity in the nanogap problem identifies $\partial\Omega \cap \{x = -25\}$ and $\partial\Omega \cap \{x = 25\}$. We visualize these sets as two distinct surfaces in $\mathbb{R}^3$, with every vertex in $\partial\Omega \cap \{x = -25\}$ duplicated in $\partial\Omega \cap \{x = 25\}$. Hence, the mesh generated by Netgen has two different indices for each vertex, edge, and face within these surfaces. However, these geometric objects should not have distinct degrees of freedom. To address this, the members `vertmapx` and `vertmapy` of the `PeriodicHCurlSpace` class are mappings that associate "duplicate" vertices to a single vertex index. Similarly, the members `dofmapx` and `dofmapy` associate "duplicate" degrees of freedoms to a single index in the dof table. We now take a closer look at how this is done in the `Update` function of the `PeriodicHCurlSpace` class.

periodichcurl.cpp

```
.
.
.

void PeriodicHCurlSpace :: Update (LocalHeap & lh)
{
  dofmapx.SetSize(0); // Sentinels: dofmaps are not yet set
  dofmapy.SetSize(0);

  HCurlHighOrderFESpace::Update(lh);

  // set dof maps to identity
  dofmapx.SetSize (GetNDof());
  for (int i = 0; i < dofmapx.Size(); i++)
    dofmapx[i] = i;

  dofmapy.SetSize (GetNDof());
  for (int i = 0; i < dofmapy.Size(); i++)
    dofmapy[i] = i;

  // vertex-pair to edge hashtable
  HashTable<INT<2>, int> vp2e(ma->GetNEdges());

  for (int enr = 0; enr < ma->GetNEdges(); enr++)
  {
    int v1, v2;
    ma->GetEdgePNums (enr, v1, v2);
    if (v1 > v2) Swap (v1, v2);
    vp2e[INT<2>(v1,v2)] = enr;
  }

  // vertex-triple-or-quartet to face hashtable
  // (triangular faces get a dummy vertex number equal to -1)
  HashTable<INT<4>, int> v2f(ma->GetNFaces());

  Array<int> pnums;
  for (int fnr = 0; fnr < ma->GetNFaces(); fnr++)
  {
    ma->GetFacePNums (fnr, pnums);
    INT<4> i4;
    if (pnums.Size() == 3)
      i4 = {-1, pnums[0], pnums[1], pnums[2]};
    if (pnums.Size() == 4)
      i4 = {pnums[0], pnums[1], pnums[2], pnums[3]};
    i4.Sort();
    v2f[i4] = fnr;
  }

  // idx

  // vertex slave -> master array
  vertmapx.SetSize(ma->GetNV());
  for (int i = 0; i < vertmapx.Size(); i++)
    vertmapx[i] = i;

  Array<INT<2> > periodic_verts;
  ma->GetPeriodicVertices(idx, periodic_verts);

  for (auto pair : periodic_verts)
    vertmapx[pair[1]] = pair[0];

  // find periodic edges (using vertex-pair to edge hashtable)
  for (int enr = 0; enr < ma->GetNEdges(); enr++)
  {
    int v1, v2;
    ma->GetEdgePNums (enr, v1, v2);
    // number of master-vertices
    int mv1 = vertmapx[v1];   //
    int mv2 = vertmapx[v2];
    if (v1 != mv1 && v2 != mv2) // edge shall be mapped
    {
      if (mv1 > mv2) Swap (mv1, mv2);
      int menr = vp2e[INT<2>(mv1,mv2)];  // the master edge-nr

      dofmapx[enr] = menr;

      IntRange edofs = GetEdgeDofs (enr);   // dofs on slave edge
      IntRange medofs = GetEdgeDofs (menr); // dofs on master edge
      for (int i = 0; i < edofs.Size(); i++)
        dofmapx[edofs[i]] = medofs[i];
    }
  }

  // find periodic faces (using vertex-triple to face hashtable)
  for (int fnr = 0; fnr < ma->GetNFaces(); fnr++)
  {
    ma->GetFacePNums (fnr, pnums);
    INT<4> i4;

    if (pnums.Size() == 3)
    {
      i4 = {-1, vertmapx[pnums[0]], vertmapx[pnums[1]],
            vertmapx[pnums[2]]};
      if (i4[1] != pnums[0] && i4[2] != pnums[1] && i4[3] != pnums[2])
      {
        i4.Sort();
        int mfnr = v2f[i4];
        IntRange fdofs = GetFaceDofs (fnr);
        IntRange mfdofs = GetFaceDofs (mfnr);
        for (int i = 0; i < fdofs.Size(); i++)
          dofmapx[fdofs[i]] = mfdofs[i];
      }
    }

    if (pnums.Size() == 4)
    {
      i4 = {vertmapx[pnums[0]], vertmapx[pnums[1]], vertmapx[pnums[2]],
            vertmapx[pnums[3]]};
      if (i4[0] != pnums[0] && i4[1] != pnums[1] && i4[2] != pnums[2]
          && i4[3] != pnums[3])
      {
        i4.Sort();
        int mfnr = v2f[i4];

        IntRange fdofs = GetFaceDofs (fnr);
        IntRange mfdofs = GetFaceDofs (mfnr);
        for (int i = 0; i < fdofs.Size(); i++)
          dofmapx[fdofs[i]] = mfdofs[i];
      }
    }
  }

  for (int i = 0; i < dofmapx.Size(); i++)
    if (dofmapx[i] != i)
      ctofdof[i] = UNUSED_DOF;

  // idy (similar to idx above)
  .
  .
  .

}
.
.
.
```

The function begins with a call to the base class's `Update` function and preliminarily sets `dofmapx` and `dofmapy` as identity mappings. Next, two hash tables are created that will facilitate resetting `dofmapx` and `dofmapy` to the actual mappings. Given two vertices connected by an edge, the `vp2e` hash table returns the edge number. Similarly, given three (resp., four) vertices of a triangular (resp, rectangular) face, the `v2f` hash table returns the face number.

The function then accounts for each of the periodic direction in turn. The x periodicity is accounted for first, resulting in the correct `vertmapx` and `dofmapx`. The `vertmapx` array is straightforward to set, because ordered pairs of periodic vertices were made accessible by Netgen when the mesh was generated. These periodic vertex pairs are retrieved using the `MeshAccess` object `ma`. Their index 0 components are designated as the "master" vertices and their index 1 components are "slaves". Thus, any subsequent occurrence of a slave vertex number can be replaced by its master vertex number using `vertmapx`.

Next, dofmapx is set. Periodic edges are accounted for within a loop over all edges. For each edge, its two vertices are compared to their mappings under vertmapx. If both vertices are found to be slave vertices, then the edge is a slave edge and all of its degrees of freedom must match those of its master edge. Similarly, slave periodic faces are found within a loop over all faces, and their associated degrees of freedom are mapped to those of the master face. With dofmapx set correctly, all subsequent occurrences of slave dof numbers can be replaced by master dof numbers. The final step in accounting for x periodicity is to get rid of all of the slave degrees of freedom from the ctofdof array.

The y periodicity is treated in practically the same way, so the setting of vertmapy and dofmapy is omitted from the code snippet above.

For an example of how dofmapx and dofmapy are used, we look at the GetDofNrs function.

periodichcurl.cpp

```
.
.
.

void PeriodicHCurlSpace::GetDofNrs (int elnr, Array<int> & dnums) const
{
  HCurlHighOrderFESpace::GetDofNrs(elnr,dnums);

  if (dofmapx.Size())
    for(int i=0; i<dnums.Size(); i++)
      dnums[i] = dofmapy[dofmapx[dnums[i]]];
}

.
.
.
```

We see that the base class supplies its degrees of freedom in dnums, then these are mapped to the true degrees of freedom via dofmapy composed with dofmapx.

5.5. Simulation results

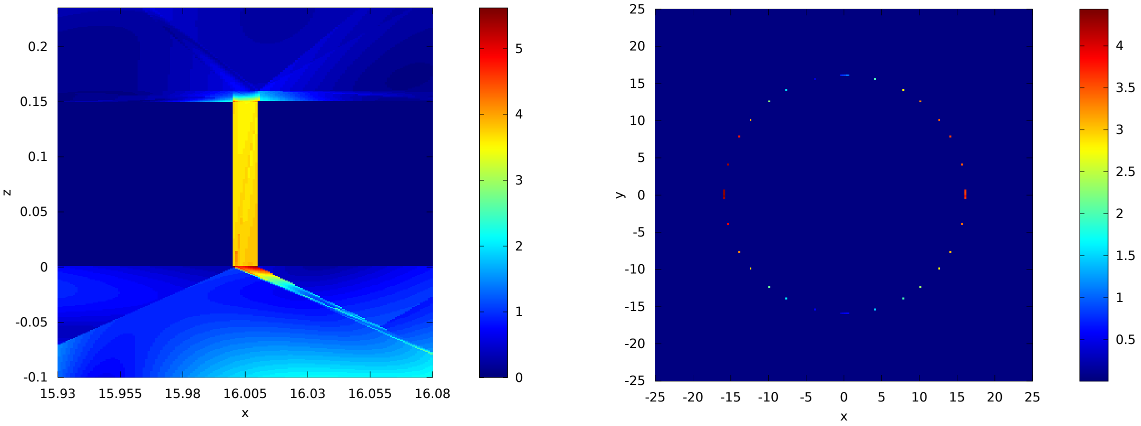
We report the results of a simulation using the DPG method of Section 5.2 with the order of the discrete spaces, defined in (91), set to $p = 2$. We fix the nanogap width at $w = 10$ nm, and the incident wave frequency at $f = 0.625$, which results in wavenumber

$$\begin{aligned} k_0 &= 0.01310 \\ k_{\text{glass}} &= 0.01830 \\ k_{\text{Al}_2\text{O}_3} &= 0.02004 \\ k_{\text{Au}} &= 4.389\hat{\imath} \end{aligned}$$

and which, according to [**39**], produces a resonance when the thickness of the Al_2O_3 layer is 10 nm. As we discussed in the previous section, thicker Al_2O_3 layers should also result in a resonance. We used the conjugate gradient method with a local preconditioner to solve the DPG system.

Results from the simulation are shown in Figure 5.5. Figure 5.5a shows a cross-sectional view of the nanogap at $y = 0$. The Au region is clearly visible, as the field there is nearly zero. The field is strongest within the nanogap, as well as directly above and below it. This is consistent with what we expect. However, the streaks below the nanogap are evidence that the mesh should be finer there.

Figure 5.5b shows a top-down view of the nanogap at $z = 0.1$ µm. While we do see some field enhancement, the lack of smooth transition again indicates that perhaps we need a finer mesh.



(A) Cross-sectional view at $y = 0$.

(B) Top-down view at $z = 0.1$.

FIGURE 5.5. Two views of $|E|$ simulated by the DPG method, for an incident wave of unit amplitude. Length units are µm.

6

Conclusions

We presented and analyzed the DPG$_\varepsilon$ method for the Helmholtz equation. The case $\varepsilon = 1$ was analyzed previously in [**14**]. The numerical results in [**14**] showed that in a comparison of the ratio of L^2 norms of the discretization error to the best approximation error, the DPG method had superior properties. The pollution errors reported in [**14**] for a higher order DPG method were so small that its growth could not be determined conclusively there. In this dissertation, by performing a dispersion analysis on the DPG method for the lowest possible order, we found that the method has pollution errors that asymptotically grow with k at the same rate as other comparable methods.

In addition, we found both dispersive and dissipative type of errors in the lowest order DPG method. The dissipative errors manifest in computed solutions as artificial damping of wave amplitudes (e.g., as illustrated in Figure 4.1).

Our results show that the DPG solutions have higher accuracy than an L^2-based least-squares method with a stencil of identical size. However, the errors in the (lowest order) DPG method did not compare favorably with a standard (higher order) finite element method having a stencil of the same size. Whether this disadvantage can be offset by the other advantages of the DPG methods (such as the regularizing effect of ε, and the fact that it yields Hermitian positive definite linear systems and good gradient approximations) remains to be investigated.

We provided the first theoretical justification for considering the ε-modified DPG method. If the test space were exactly computed, then Theorem 2.2.1 shows that the errors in numerical fluxes and traces will improve as $\varepsilon \to 0$. However, if the test space is inexactly computed using the enrichment degree r, then the numerical results from the dispersion analysis showed that errors continually decreased as ε was decreased only for odd r. A full theoretical explanation of such discrete effects and the limiting behavior when ε is 0 deserves further study.

For the HDG method, we found that here are values of stabilization parameters τ that will cause the HDG method to fail in time-harmonic electromagnetic and acoustic simulations using complex wavenumbers (equation (49) et seq.). If the wavenumber k is complex, then choosing τ so that $\mathrm{Re}(\tau)\,\mathrm{Im}(k) \le 0$ guarantees that the HDG method is uniquely solvable (see Theorem 3.2.2). Even when the wave problem is not well-posed (such as at a resonance), the HDG method remains uniquely solvable when $\mathrm{Im}(k) = 0$ and $\mathrm{Re}(\tau) \neq 0$. However, in such cases, we found the discrete stability to be tenuous. (See Figure 3.2 and accompanying discussion.) For real wavenumbers k, we found that the HDG method introduces small amounts of artificial dissipation (see equation (69)). In 1D, the optimal values of τ that asymptotically minimize both dissipative and dispersive errors in the discrete wavenumber are $\tau = \pm\hat{\imath}$ (see equation (70)). In 2D, for real wavenumbers k, the best values of τ are dependent on the propagation angle. Overall, values of τ that asymptotically minimize the error in the discrete wavenumber over all angles is $\tau = \pm\hat{\imath}\sqrt{3}/2$ (per equation (78)).

References

[1] M. Ainsworth, *Discrete dispersion relation for hp-version finite element approximation at high wave number*, SIAM Journal on Numerical Analysis, 42 (2004), pp. 553–575.

[2] D. N. Arnold and F. Brezzi, *Mixed and nonconforming finite element methods: implementation, postprocessing and error estimates*, RAIRO Modél. Math. Anal. Numér., 19 (1985), pp. 7–32.

[3] I. Babuška and S. Sauter, *Is the pollution effect of the FEM avoidable for the Helmholtz equation considering high wave numbers?*, SIAM Review, 42 (2000), pp. 451–484, http://dx.doi.org/10.1137/S0036142994269186.

[4] P. Bochev and M. Gunzburger, *Least-Squares Finite Element Methods*, Applied Mathematical Sciences, Springer, 2009.

[5] D. Boffi, M. Fortin, and F. Brezzi, *Mixed finite element methods and applications*, Springer series in computational mathematics, Springer, Berlin, Heidelberg, 2013.

[6] T. Bui-Thanh, L. Demkowicz, and O. Ghattas, *A unified discontinuous Petrov-Galerkin method and its analysis for Friedrichs' systems*, SIAM J. Numer. Anal., 51 (2013), pp. 1933–1958.

[7] Z. Cai, R. Lazarov, T. A. Manteuffel, and S. McCormick, *First-order system least squares for second-order partial differential equations: Part I*, SIAM J. Numer. Anal., 31 (1994), pp. 1785–1799.

[8] C. Carstensen, L. Demkowicz, and J. Gopalakrishnan, *Breaking spaces and forms for the DPG method and applications including maxwell equations*, submitted, (2016).

[9] B. Cockburn and J. Gopalakrishnan, *A characterization of hybridized mixed methods for the Dirichlet problem*, SIAM J. Numer. Anal., 42 (2004), pp. 283–301.

[10] B. Cockburn, J. Gopalakrishnan, and R. Lazarov, *Unified hybridization of discontinuous Galerkin, mixed, and continuous Galerkin methods for second order elliptic problems*, SIAM Journal on Numerical Analysis, 47 (2009), pp. 1319–1365, http://dx.doi.org/10.1137/070706616.

[11] J. Cui and W. Zhang, *An analysis of HDG methods for the Helmholtz equation*, IMA Journal of Numerical Analysis, 34 (2014), pp. 279–295, http://imajna.oxfordjournals.org/content/34/1/279.full.pdf+html.

[12] L. Demkowicz and J. Gopalakrishnan, *A class of discontinuous Petrov-Galerkin methods. Part I: The transport equation*, Comput. Methods Appl. Mech. Engrg., 199 (2010), pp. 1558–1572.

[13] 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.

[14] L. Demkowicz, J. Gopalakrishnan, I. Muga, and J. Zitelli, *Wavenumber explicit analysis of a DPG method for the multidimensional Helmholtz equation*, Computer Methods in Applied Mechanics and Engineering, 213 (2012), pp. 126–138.

[15] L. Demkowicz, J. Gopalakrishnan, and A. Niemi, *A class of discontinuous Petrov-Galerkin methods. Part III: adaptivity*, Applied numerical mathematics, 62 (2012), pp. 396–427.

[16] L. Demkowicz and L. Vardapetyan, *Modeling of electromagnetic absorption/scattering problems using hp-adaptive finite elements*, Computer Methods in Applied Mechanics and Engineering, 152 (1998), pp. 103 – 124. Containing papers presented at the Symposium on Advances in Computational Mechanics.

[17] L. F. Demkowicz and J. Gopalakrishnan, *Recent Developments in Discontinuous Galerkin Finite Element Methods for Partial Differential Equations: 2012 John H Barrett Memorial Lectures*, Springer International Publishing, Cham, 2014, ch. An Overview of the Discontinuous Petrov Galerkin Method, pp. 149–180.

[18] A. Deraemaeker, I. Babuška, and P. Bouillard, *Dispersion and pollution of the FEM solution for the Helmholtz equation in one, two and three dimensions*, International journal for numerical methods in engineering, 46 (1999), pp. 471–499.

[19] T. W. Ebbesen, H. J. Lezec, H. F. Ghaemi, T. Thio, and P. A. Wolff, *Extraordinary optical transmission through sub-wavelength hole arrays*, Nature, 391 (1998), pp. 667–669.

[20] G. J. Fix and M. D. Gunzburger, *On numerical methods for acoustic problems*, Computers & Mathematics with Applications, 6 (1980), pp. 265–278.

[21] G. Giorgiani, S. Fernández-Méndez, and A. Huerta, *High-order continuous and discontinuous Galerkin methods for wave problems*, Int. J. Numer. Methods Fluids, 73 (2013), pp. 883–903.

[22] J. Gopalakrishnan, *Five lectures on DPG methods*, ArXiv e-prints, (2013), 1306.0557.

[23] J. Gopalakrishnan, S. Lanteri, N. Olivares, and R. Perrussel, *Stabilization in relation to wavenumber in HDG methods*, Advanced Modeling and Simulation in Engineering Sciences, 2 (2015), p. 13.

[24] J. Gopalakrishnan, I. Muga, and N. Olivares, *Dispersive and dissipative errors in the DPG method with scaled norms for Helmholtz equation*, SIAM Journal on Scientific Computing, 36 (2014), pp. A20–A39.

[25] J. Gopalakrishnan and W. Qiu, *An analysis of the practical DPG method*, Mathematics of Computation, 83 (2014), pp. 537–552.

[26] J. Gopalakrishnan, J. Schöberl, L. Kogler, and N. Olivares, *DPG shared library add-on to NGSolve*. `https://github.com/jayggg/DPG`.

[27] R. Griesmaier and P. Monk, *Error analysis for a hybridizable discontinuous Galerkin method for the Helmholtz equation*, Journal of Scientific Computing, 49 (2011), pp. 291–310.

[28] F. Ihlenburg, *Finite element analysis of acoustic scattering*, vol. 132, Springer Science & Business Media, 1998.

[29] F. Ihlenburg and I. Babuška, *Dispersion analysis and error estimation of galerkin finite element methods for the helmholtz equation*, International journal for numerical methods in engineering, 38 (1995), pp. 3745–3774.

[30] F. Ihlenburg and I. Babuška, *Finite element solution of the Helmholtz equation with high wave number Part I: The h-version of the FEM*, Computers & Mathematics with Applications, 30 (1995), pp. 9 – 37.

[31] F. Ihlenburg and I. Babuška, *Finite element solution of the Helmholtz equation with high wave number Part II: The h-p version of the FEM*, SIAM Journal on Numerical Analysis, 34 (1997), pp. 315–358, http://dx.doi.org/10.1137/S0036142994272337.

[32] F. Kikuchi, *On a discrete compactness property for the Nédélec finite elements*, J. Fac. Sci. Univ. Tokyo Sect. IA Math., 36 (1989), pp. 479–490.

[33] L. Li, S. Lanteri, and R. Perrussel, *Numerical investigation of a high order hybridizable discontinuous Galerkin method for 2d time-harmonic Maxwell's*

equations, COMPEL-The international journal for computation and mathematics in electrical and electronic engineering, 32 (2013), pp. 1112–1138.

[34] ——, *A hybridizable discontinuous Galerkin method combined to a Schwarz algorithm for the solution of 3d time-harmonic Maxwells equation*, Journal of Computational Physics, 256 (2014), pp. 563–581.

[35] S. Maier, *Plasmonics: Fundamentals and Applications*, Springer, 2007.

[36] J. M. Melenk, *On generalized finite element methods*, PhD thesis, The University of Maryland, 1995.

[37] J. C. Nédélec, *Mixed finite elements in* $\mathbb{R}^3$, Numerische Mathematik, 35 (1980), pp. 315–341.

[38] N. C. Nguyen, J. Peraire, and B. Cockburn, *Hybridizable discontinuous Galerkin methods for the time-harmonic Maxwells equations*, Journal of Computational Physics, 230 (2011), pp. 7151–7175.

[39] H.-R. Park, X. Chen, N.-C. Nguyen, J. Peraire, and S.-H. Oh, *Nanogap-enhanced terahertz sensing of 1 nm thick (/106) dielectric films*, ACS Photonics, 2 (2015), pp. 417–424.

[40] P.-A. Raviart and J. M. Thomas, *A mixed finite element method for 2nd order elliptic problems*, in Mathematical aspects of finite element methods (Proc. Conf., Consiglio Naz. delle Ricerche (C.N.R.), Rome, 1975), Springer, Berlin, 1977, pp. 292–315. Lecture Notes in Math., Vol. 606.

[41] J. Schöberl, *Netgen.* `https://sourceforge.net/projects/netgen-mesher/`.

[42] ——, *NGSolve.* `https://sourceforge.net/projects/ngsolve/`.

[43] G. Strang and G. J. Fix, *An analysis of the finite element method*, Prentice-Hall series in automatic computation, Prentice-Hall, Englewood Cliffs, N.J., 1973.

[44] L. L. Thompson and P. M. Pinsky, *Complex wavenumber Fourier analysis of the p-version finite element method*, Computational Mechanics, 13 (1994), pp. 255–275.

[45] J. Zitelli, I. Muga, L. Demkowicz, J. Gopalakrishnan, D. Pardo, and V. Calo, *A class of discontinuous petrovgalerkin methods. part iv: The optimal test norm and time-harmonic wave propagation in 1d*, Journal of Computational Physics, 230 (2011), pp. 2406 – 2432.