An Adaptive Anisotropic hp-Refinement Algorithm for the 2D Maxwell Eigenvalue Problem

An Adaptive Anisotropic hp-Refinement Algorithm for the 2D Maxwell Eigenvalue Problem

Jeremiah Corrado, Jake J. Harmon, *Graduate Student Member, IEEE*, and Branislav M. Notaroš, *Fellow, IEEE*

***Abstract*—We present a novel adaptive mesh refinement algorithm for efficiently solving a continuous Galerkin formulation of the 2D Maxwell Eigenvalue Problem over quadrilateral discretizations. The algorithm harnesses a Refinement-by-Superposition framework with support for anisotropic hp-adaptivity. Following a brief summary of the underlying methodology, we develop a novel approach to directing and deploying intelligent fully anisotropic hp-refinements. Numerical examples targeting the Maxwell Eigenvalue problem verify the ability to achieve exponential rates of convergence on a challenging benchmark and demonstrate that including anisotropic refinement yields a significant enhancement of computational efficiency. The success of this algorithm, along with the simplicity of the underlying Refinement-by-Superposition approach, creates a promising path for implementing highly efficient yet lightweight finite element method (FEM) codes for a variety of applications in computational electromagnetics (CEM) and elsewhere.**

***Index Terms*—computational electromagnetics, continuous Galerkin, finite element method, higher order methods, adaptive refinement, dual-weighted-residual, *hp*-refinement, anisotropic h-refinement, anisotropic p-refinement, multi-level, refinement-by-superposition**

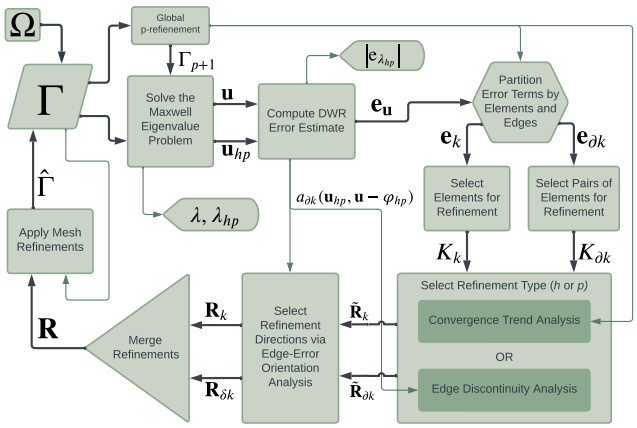


Fig. 1: Refinement Algorithm Overview. Execution of the algorithm begins with the Domain (Ω) located in the top left corner. Bold arrows show the primary flow of data, while light arrows provide some additional detail.

I. Introduction

ADAPTIVE refinement algorithms are a crucial component of any finite element method (FEM) implementation. Efficient acquisition of accurate results necessitates the employment of highly specialized FEM discretizations. The arrangement of elements and basis functions must adhere to the dynamics of interest and the intricacies of the physical model; however, simulations generally begin with a discretization that is a minimally complex realization of the geometry of interest (i.e., the initial discretization is yielded directly from the 3D modeling software with no concern for its fitness to the simulation). For example, an accurate simulation of the radar-cross-section of a jet requires an accurate map of the current density on the aircraft's surface, which in turn generally requires a fine discretization of any sharp edges along the surface (many elements must be concentrated around sharp edges). There is no guarantee that the 3D modeling software will produce such a discretization, and thus, adaptive refinement is necessary to progressively transform the geometrically driven mesh into a numerically optimal mesh. This work describes an adaptive algorithm that uses a refinement-by-superposition (RBS) implementation of hp-refinement to progressively generate an accurate FEM solution from a granular mesh. We use a 2D FEM problem to develop and test the methodology; however, a 3D generalization is trivial.

In many cases, it is possible to achieve highly accurate results in FEM using higher-order elements (or p-refined elements) alone; however, we also include h-refinements in this work to better address challenging solution behaviors such as discontinuities, singularities, and multi-scale behavior. In these cases, the exponential convergence rates typically afforded by p-refinement are disrupted, and we are left with algebraic rates of convergence. The intelligent application of h-refinement around the challenging behavior can restore fast convergence rates by sequestering singular behavior to a small region of the mesh; or, in the case of multi-scale behavior, we can adapt the scale of the mesh such that the size of the behavior is on the order of the element size. Specifically, in the domain of Computational Electromagnetics (CEM), p-refinements are most effective when the element scale is on the order of a wavelength, and thus in the presence of much smaller-scale (or higher frequency) variations, we benefit from employing a combination of h- and p-refinement.

The RBS approach to hp-refinement is a powerful alternative to the more traditional Refinement-by-Replacement (RBR) implementations of h-adaptivity for quadrilateral or hexahedral elements. Like RBR, it has been shown to produce exponential rates of convergence on challenging benchmark problems for the Maxwell Eigenvalue Problem [1]. Here, singular and sharp eigenfunctions were simulated efficiently using a mixture of h-

Jeremiah Corrado, Jake J. Harmon, and Branislav M. Notaroš are with the department of Electrical and Computer Engineering, Colorado State University, Fort Collins, CO 80523-1373 USA (e-mail: jcorrado@rams.colostate.edu, jake.harmon@ieee.org, branislav.notaros@colostate.edu).

refinement around the problematic (sharp or singular) solution behavior and p-refinement elsewhere in the mesh. Similar results have been achieved outside of CEM on 2D and 3D problems [2].

Although the end result does not differ from RBR in terms of performance, the RBS approach achieves exponential convergence using a much lighter weight software implementation. [3], [4]. The most challenging component of RBR–continuity enforcement over hanging-nodes–is avoided entirety by adopting a slightly more complex mesh data-structure which allows continuity to be enforced by construction. Using this approach, we implement an H(curl) continuity condition using only a few hundred lines of code. The simplicity of the RBS implementation, allows engineers to adopt and modify FEM codes more quickly and intuitively. It also yields smaller binaries that are faster and easier to install.

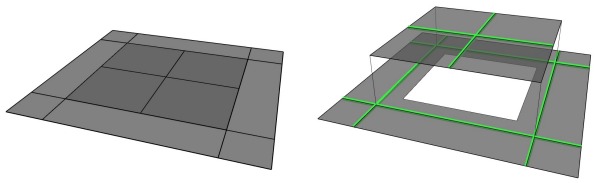
Because RBS renders hanging-nodes trivial, it also supports a relatively straightforward implementation of anisotropic h-refinement, which is significantly more challenging to develop under the constraints of RBR. Anisotropic hp-refinements have been shown to yield improved efficiency on the benchmark problem discussed in this work [5], as well as in other domains [6].

The remainder of this work is organized as follows. Section II provides a brief summary of the RBS approach to h-refinement for 2D FEM. A more detailed explanation of the underlying methodology and the anisotropic refinement implementation can be found in [1] and [5] respectively. Section III gives a detailed breakdown of the adaptive refinement methodology for the Maxwell Eigenvalue Problem in 2D FEM as shown in Fig. 1. This includes subsections for each procedure in the algorithm and the requisite mathematical foundations. Finally, Section IV gives a numerical analysis of the methodology using the aforementioned benchmark problem with singular solutions. Here, we investigate the usefulness of the methodology as a whole, and provide a comparison with an isotropic version of the same algorithm.

II. The Refinement-by-Superposition Methodology

The RBS approach to FEM constitutes a generalization of the typical Refinement-by-Replacement approach; however, it has significant effects on the enforcement of continuity conditions. With an RBR implementation, an h-refinement introduces hanging nodes around the border of the refined element (4 hanging nodes in 2D and 18 in 3D), requiring specialized algorithms to enforce continuity between the new elements and neighboring elements. Alternatively, an RBS h-refinement leaves the parent element in the discretization to handle continuity enforcement and introduces new elements "above" it, making hanging-nodes inconsequential.

This difference is depicted in Fig. 2, where the RBR refinement on the left shows the introduction of four hanging nodes, while the RBS refinement on the right shows the new elements superimposed over the base element. Here, active edges are highlighted in green to show that the element on the base layer is enforcing continuity with its neighbors, while the edges along the border of the new layer are inactive. We also show that the face of the refined element is inactive, while the faces of the child elements are active. As such, the new elements have inherited all basis functions from their parent, except those necessary to maintain continuity with the neighboring elements.



(a) RBR (b) RBS

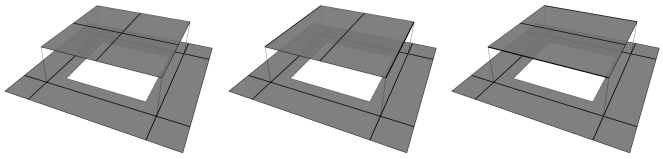
Fig. 2: Comparison of hanging nodes in RBS and RBR for 2D FEM. Extension into the vertical axis is not physical, it is merely a conceptual tool to represent superposition.

With this in mind, we note that the RBS mesh data structure is composed of a list of element trees, where the root elements are those from the initial discretization, and h-refinements append new elements to the leaves of the trees. Edges are also organized in tree structures and maintain lists of adjacent elements on both sides. This arrangement is distinct from an RBR mesh implementation where we simply have a list of elements and a list of half-edges (who may connect to one other half-edge, or multiple other half-edges due to h-refinements).

After executing refinements, a simple procedure follows to define a basis set over our domain such that continuity is enforced among neighboring elements and overlapping layers maintain linear independence. We give a summary here; however, a full specification can be found in [1], [5]:

1) Iterate over each element and define basis functions according to the local expansion orders
2) Associate basis functions with the relevant geometric components based on the desired continuity conditions:
   - **H(curl)**: The non-zero tangential components
   - **H(div)**: The non-zero normal components

   We separate basis functions into three types, based on the value of the function along the edges associated with its vectorial direction:
   - **Element**: Zero along all relevant edges
   - **Edge**: Non-zero along one relevant edge
   - **Node**: Non-zero along two adjacent edges
3) Add the element-type basis functions on all the leaf elements to the system
4) Iterate over each edge tree:
   - Find the most h-refined edges, who have at least one element on each side (if there are multiple elements on a side, choose the most h-refined element)
   - For each of the selected edges, match the edge-type basis functions on the selected pairs of elements and add them to the system
5) Iterate over each node:



(a) T-Type (*isotropic*) (b) U-Type (c) V-Type

Fig. 3: h-Refinement Variants

- Find the most h-refined groups of four elements (such that each pair of neighboring elements shares an edge)
- Match the node-type basis functions on the four elements and add them to the system

This algorithm is significantly more lightweight than an RBR equivalent and is fully compatible with any hierarchical basis compatible with RBR [7], [8]. It carries the additional benefit of imposing no limit on mesh irregularity, meaning that an edge can support any number of hanging nodes. Few RBR implementations of FEM support discretizations with mesh irregularity above one hanging-node per edge. This is not a severe limitation in practice; however, it removes the need for the mesh to support transition elements between regions of dense h-refinement and sparse h-refinement. Notably, stark transitions with many hanging nodes along an edge will reduce the sparsity of the system matrices and make parallelization more challenging; however, some work has been done to overcome these difficulties [9].

Additionally, RBS imposes no constraints on the shape or orientation of the elements on the superimposed layers (so long as continuity is enforced among them). Thus anisotropic h-refinement is straightforward to implement as it requires no changes to the underlying methodology. For 2D FEM, we utilize an implementation with three h-refinement variants: one isotropic and two anisotropic, as shown in Fig. 3. Using a combination of these anisotropic h-refinements and anisotropic p-refinements (which are trivial to implement), one can achieve significant efficiency improvements on sharp or singular problems [5]. In other words, anisotropic hp-refinement facilitates a more surgical introduction of new Degrees of Freedom (DoFs) over elements that have inaccuracies in only one direction or region.

III. Adaptive Refinement Strategy

We now develop an adaptive refinement strategy to leverage the anisotropic hp-refinements afforded by the RBS methodology from the previous section. This algorithm focuses specifically on 2D problems; however, the principles described here immediately generalize to other applications, including 3D FEM. In particular, we target the Maxwell Eigenvalue problem for its solution behavior and availability of benchmark solutions.

The Maxwell Eigenvalue problem is formulated as follows, where we constrain the possible solutions to TE propagation modes by asserting that the solution to the electric field is purely transversal [8]:

Find an eigenpair $U = \{\mathbf{u}_{hp},\, \lambda_{hp}\} \in B_{hp} \times \mathbb{R}_{>0}$ such that:

$$a(\mathbf{u}_{hp},\, \boldsymbol{\phi}_{hp}) = \lambda_{hp} m(\mathbf{u}_{hp},\, \boldsymbol{\phi}_{hp}) \quad \forall \boldsymbol{\phi}_{hp} \in B_{hp} \tag{1}$$

for $B_{hp} \subset H_0(\text{curl}; \Omega)$, $\Omega \subset \mathbb{R}^2$, where:

$$m(\mathbf{u}_{hp},\, \boldsymbol{\phi}_{hp}) = \langle \mathbf{u}_{hp},\, \boldsymbol{\phi}_{hp} \rangle \tag{2}$$

$$a(\mathbf{u}_{hp},\, \boldsymbol{\phi}_{hp}) = \langle \nabla_t \times \mathbf{u}_{hp},\, \nabla_t \times \boldsymbol{\phi}_{hp} \rangle \tag{3}$$

With this problem in mind, we will discuss the details of the refinement algorithm summarized in Fig. 1. Each of the vital steps will be outlined in detail in the following sections. It is important to note that the main loop in the diagram (from the initial discretization Γ, to our refined discretization $\hat{\Gamma}$) shows the process for executing a single refinement iteration. This process is designed to be repeated many times, and various stop conditions can be used to terminate it. For example, a predefined number of iterations can execute, or refinement can continue until a desired accuracy is achieved.

A. Dual Weighted Residual Error Estimation

A refinement iteration begins with the computation of a solution and an accompanying error estimate generated using the Dual Weighted Residual (DWR) procedure. We will give a high-level description of how DWR applies to this problem; however more detail is found in [10]. This procedure is the driving force behind the refinement algorithm, as we must have some spatially-localized notion of error contributions in order to choose which elements to refine. Without such information, one must employ indicators (as opposed to estimators) that may inadequately relate the properties of the discretization to the accuracy of goal computables.

For dictating refinements, the error,

$$e_{\lambda_{hp}} = \lambda - \lambda_{hp}, \tag{4}$$

may be expressed in the following dual weighted residual (DWR) form [10]:

$$\begin{aligned} e_{\lambda_{hp}}(1 - \tfrac{1}{2}M) = a(\mathbf{u}_{hp}, \mathbf{u} - \varphi_{hp}) - \\ \lambda_{hp} m(\mathbf{u}_{hp}, \mathbf{u} - \varphi_{hp}) \quad \forall \varphi_{hp} \in B_{hp} \end{aligned} \tag{5}$$

Here, an overall estimate of the error in the eigenvalue accuracy is given in terms of an approximate solution to the primal problem $\mathbf{u}_{hp}$ and an exact solution to the dual problem $\mathbf{u}$. Because this exact solution is not available, a higher-order solution takes its place as an approximation. This computation gives an accurate estimate of the error. We compute the higher-order solution by applying a global isotropic p-refinement of magnitude 1 to the entire mesh and re-solving the eigenproblem.

The validity of the above equality relies on a pre-normalization of the primal and dual solutions in terms of the L2 norm:

$$\langle \mathbf{u}, \mathbf{u} \rangle = 1 \quad \text{and} \quad \langle \mathbf{u}_{hp}, \mathbf{u}_{hp} \rangle = 1 \tag{6}$$

To extract the maximum amount of useful information from the error contribution estimates, φ_{hp} is chosen to be $\boldsymbol{\Pi}_{hp}^{curl}\mathbf{u}$, which is the curl conforming projection of the dual solution onto the primal solution space. The subtraction of this

information from the dual solution leaves only higher-order information in the error estimate. As such, terms that would cancel in the global assembly of the error contributions (due to Galerkin orthogonality) are excluded preemptively.

Following [10], we extract more spatially granular data from the DWR procedure by separating the stiffness-matrix integral, shown below:

$$a(\mathbf{u},\boldsymbol{\phi}) = \langle \nabla_t \times \mathbf{u}, \nabla_t \times \boldsymbol{\phi} \rangle = \int_{\Omega_K} \nabla_t \times \mathbf{u} \cdot \nabla_t \times \boldsymbol{\phi} \quad d\Omega_K \quad (7)$$

into its by-parts form. In this expanded form, the surface term is computed separately from the edge terms, yielding direct access to an error estimate associated with the solution on each edge:

$$a(\mathbf{u},\boldsymbol{\phi}) = \int_{\Omega_K} (\nabla \times \nabla_t \times \mathbf{u}) \cdot \boldsymbol{\phi} \quad d\Omega_K - \oint_{\partial\Omega_K} (\nabla_t \times \mathbf{u}) \times \boldsymbol{\phi} \cdot \hat{n} \quad dS_K \quad (8)$$

These individual edge-terms are stored to be used later for discontinuity analysis and refinement direction selection. The full-valued solution of $a(\mathbf{u},\boldsymbol{\phi})$ is used for the purpose of evaluating Equations 5 and/or 9.

Initially we accumulate error contributions in terms of dual DoFs. The following equation shows the interactions of the ith dual DoF with all primal DoFs; however, in practice, only the overlapping integrals must be computed. Additionally, we note that the higher-order portions of the dual solution constitute the only non-zero error terms, as the others are eliminated by the curl conforming projection operator.

$$\mathbf{e}_{\mathbf{u}i} = \sum_{\boldsymbol{\nu}_{hp} \in \mathbf{u}_{hp}} a(\mathbf{u}_{hp}, (\mathbf{u} - \boldsymbol{\Pi}_{hp}^{curl}\mathbf{u})_i) - \lambda_{hp} m(\mathbf{u}_{hp}, (\mathbf{u} - \boldsymbol{\Pi}_{hp}^{curl}\mathbf{u})_i) \quad (9)$$

With this vector of error coefficients, we have an estimate of how local approximation error hampers the quality of the (global) eigenvalue quantity. As such, it is appropriate to combine the coefficients of nearby DoFs to arrive at a list of elements whose refinement would yield the most significant improvement in accuracy.

B. Error Coefficient Partitioning and Thresholding

Here, we choose to accumulate the error coefficients into two separate lists. One is composed of element DoFs and the other of edge DoFs (because we are focused on TE modes only, there are no DoFs associated with nodes). Specifically, for each element in the mesh, the error terms associated with its local Element-Type DoFs are summed up to generate an error coefficient for the portion of the solution that lies strictly on the element's surface:

$$\mathbf{e}_k = \left| \sum_{i \in \mathbf{u}(\Omega_k)} \mathbf{e}_{\mathbf{u}i} \right| \quad (10)$$

Then, for each edge in the mesh, the error terms associated with its Edge-Type DoFs are summed up to produce an error coefficient for the solution which spans the two neighboring elements. [1]

$$\mathbf{e}_{\partial k} = \left| \sum_{i \in \mathbf{u}(\Omega_{\partial k})} \mathbf{e}_{\mathbf{u}i} \right| \quad (11)$$

We now have two vectors of error coefficients, depicted by the output of the hexagonal block in Fig. 1. This separation is made to facilitate more effective h-refinement decisions. In experimental analyses, singular or sharp behavior often induced relatively large error terms associated with nearby Edge-Type DoFs, but did not necessarily do the same for Element-Type DoFs. As such, generating an error coefficient for all DoFs (Element-Type and Edge-Type) over an element can cause large edge-error terms to be ignored, leading to a general under h-refinement of the mesh.

Two groups of elements are then selected for refinement by choosing the top contributors from both lists. Specifically, any element with an error coefficient greater than the mean element coefficient is marked for refinement:

$$\mathbf{K}_k = \{k : \mathbf{e}_k > \overline{\mathbf{e}_k}\} \quad (12)$$

Any edge with a coefficient greater than the mean edge coefficient marks both of its active elements for refinement:

$$\mathbf{K}_{\partial k} = \left\{k^+, k^- : \mathbf{e}_{\partial k} > \overline{\mathbf{e}_{\partial k}}\right\} \quad (13)$$

Now, with these lists of elements, we must choose between h- and p-refinement and make refinement-direction selections. The following sections discuss the heuristics used to make these decisions.

C. Refinement Type Selection

The primary hp-decision heuristic is based on the theoretical difference in convergence rates associated with h- and p-refinements. This strategy is underpinned by the inequality shown in the following equation [11]–[13]:

$$|e_k| \leq C h_i^{min(\xi-1,p_i)} p_i^{-(\xi-1)} ||\mathbf{u}||_{H^\xi(\Omega)} \quad (14)$$

where e_k is the error associated with a given element k, C is a solution specific constant, ξ represents the local regularity of the solution, h_k is the diameter of the element, and p_k is the polynomial expansion order on the element. If the regularity is large enough, the error magnitude is dominated by an exponential relationship with the expansion order (as shown in equation 15). When the regularity is too small–or the solution is non-smooth–the error magnitude is dominated by an algebraic relationship with the expansion order (equation 16).

$$|e_k| \leq C \frac{h_k^{p_k}}{p_k^{\xi-1}} ||\mathbf{u}||_{H^\xi(\Omega)} \quad (15)$$

$$|e_k| \leq C \left(\frac{h_k}{p_k}\right)^{\xi-1} ||\mathbf{u}||_{H^\xi(\Omega)} \quad (16)$$

[1]These coefficients, $\mathbf{e}_{\partial k}$, are not to be confused with the error terms computed using integration-by-parts (Equation 8). Those terms only focus on the error estimate along the inside of the edge, while these are associated with the edge-type basis functions which are defined over the entirety of the two neighboring elements.

Thus, in smooth regions of a solution, one can expect the local error to shrink exponentially with respect to p_i, indicating that p-refinement would be appropriate to efficiently move towards a more accurate solution. Alternatively, if the solution is non-smooth, the accuracy will converge algebraically, indicating that h-refinement must be employed to sequester the non-smooth behavior before p-refinements can be employed profitably.

This fact offers a simple heuristic to identify whether h- or p-refinement would better address a large error contribution associated with an element or edge. As such, throughout the refinement process, each element maintains a record of its estimated error contributions and the local expansions order associated with those contributions, as does each edge. For the elements, these are denoted as e_{k_i} (the estimated error on the k^{th} element for the i^{th} refinement iteration) and p_{k_i} (the expansion order on the k^{th} element for the i^{th} refinement iteration) respectively. Edges also maintain a record of local error contributions denoted as $e_{\partial k_i}$. They use the mean of the expansion orders on the two neighboring elements, denoted as $p_{\partial k_i}$. The minimum of the two expansion orders could also be used due to the inactivity of the higher order edge-type basis functions on the element with a larger expansion order.

We now want to select between h- and p-refinement for all elements in $\mathbf{K}_k$. If at least three data points are available (two historical, and one from the current refinement iteration), we execute the following procedure:

1) Separately fit the (p_{k_i}, e_{k_i}) samples to an exponential function and an algebraic function
2) Compute the sum of the squared residuals for both functional fits ($\hat{e}_{(exp)}$ and $\hat{e}_{(alg)}$):

$$\hat{\beta_{exp}} = \sum_i (|e_i| - \hat{e}_{(exp)}(p_i))^2 \tag{17}$$

$$\hat{\beta_{alg}} = \sum_i (|e_i| - \hat{e}_{(alg)}(p_i))^2 \tag{18}$$

3) If the exponential residual sum β_{exp} is smaller than the algebraic residual sum β_{alg} (meaning that the convergence behavior is better described by an exponential relationship with p), then the element is marked for p-refinement, otherwise it is marked for h-refinement

The same procedure is executed for each of the edges in $\mathbf{K}_{\partial k}$ using the local convergence data $(p_{\partial k_i}, e_{\partial k_i})$. Again, this requires at least three data points.

In order to ensure that the functional fits are relevant to the current state of the mesh, each stack of historical convergence data is limited to 5 entries. This way, if the refinement process begins in the pre-asymptotic region, the initially erroneous convergence information will be forgotten during later iterations. Additionally, after an h-refinement, the most recent data point is inherited by the child elements, giving them some indication of the local convergence behavior without over-influencing future refinement decisions.

During the first few iterations of adaptive refinement, elements will not be able to complete convergence-trend analysis as they will not have a sufficient number of error estimates to form functional fits. This problem also arises when new elements are added to the mesh via h-refinement and have not yet accumulated sufficient local convergence data. As such, we need a secondary decision procedure to make hp-refinement decisions when convergence-trend analysis cannot be completed.

Our alternate heuristic analyzes discontinuities in the solution error among neighboring elements. Here, we analyze the relevant edges by computing the magnitude of the difference between the edge error terms on both adjacent elements (these are the edge-error terms generated via integration-by-parts in the DWR procedure). The following equation gives the magnitude difference:

$$\delta_{edge} = \left| \left| \int_{\partial\Omega_{k^+}} (\nabla_t \times \mathbf{u}_{hp}) \times (\mathbf{u} - \varphi_{hp}) \cdot \hat{n} \quad dS_{k^+} \right| - \left| \int_{\partial\Omega_{k^-}} (\nabla_t \times \mathbf{u}_{hp}) \times (\mathbf{u} - \varphi_{hp}) \cdot \hat{n} \quad dS_{k^-} \right| \right| \tag{19}$$

where $\partial\Omega_{k^+}$ and $\partial\Omega_{k^-}$ designate the bounds for the integrals along opposing sides of the same edge.

A small value of δ_{edge} indicates that the discretization is well balanced. In other words, because both neighboring elements "agree" about the accuracy of the solution along their shared edge, we can assume that there is no small-scale or discontinuous behavior that the current discretization fails to capture. As such, p-refinement will be employed on the elements associated with the edge in question. Alternatively, when neighboring elements "disagree" about the solution accuracy on a given edge, we can assume that there is some discontinuity or small-scale behavior near the edge. As such, an h-refinement is likely to be more profitable for improving solution accuracy.

The δ_{edge} terms are computed for all elements in $\mathbf{K}_k$ and $\mathbf{K}_{\partial k}$ that did not have a sufficient number of data points for convergence-trend analysis. For element-refinement decisions, the mean of the four local δ_{edge} vales is used. If this value is large, according to some threshold, we select h-refinement, otherwise we select p-refinement. For the edge-refinement decisions, we only consider the discontinuity along the edge in question. Again, this value is thresholded to select between h- and p-refinement.

We now have two partially-defined lists of refinements, $\tilde{\mathbf{R}}_k$ and $\tilde{\mathbf{R}}_{\partial k}$ generated from our two lists of elements: $\mathbf{K}_k$ and $\mathbf{K}_{\partial k}$ respectively. Under an isotropic formulation of the same algorithm, we would be ready to apply these refinements to the mesh; however, due to the availability of anisotropic refinements, we need to make direction selections to optimally address the orientation and location of the solution error.

D. Refinement Direction Selection

For each h-refinement, we would like to decide between the three available refinement types: T-, U-, and V-Type (shown in Fig. 3), as well as some more advanced composites of these types. For p-refinements, we would like to decide between an isotropic refinement of degree one (increment the expansion order by 1 in $both$ directions) and the degree-one anisotropic p-refinements in $either$ direction (increment the expansion order by 1 in only one direction).

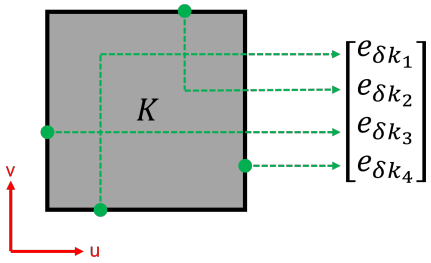


Fig. 4: Physical arrangement of edge-error terms as used in Equations 21 and 22.

This procedure analyzes the edge-error terms computed during the DWR procedure via integration-by-parts. We define a vector of edge-error coefficients for each element k as follows:

$$\begin{bmatrix} e_{\delta k_1} \\ e_{\delta k_2} \\ e_{\delta k_3} \\ e_{\delta k_4} \end{bmatrix} = \begin{bmatrix} \int_{\partial\Omega_{k_1}} (\nabla_t \times \mathbf{u}_{hp}) \times (\mathbf{u} - \varphi_{hp}) \cdot \hat{n}_1 & dS_{k_1} \\ \int_{\partial\Omega_{k_2}} (\nabla_t \times \mathbf{u}_{hp}) \times (\mathbf{u} - \varphi_{hp}) \cdot \hat{n}_2 & dS_{k_2} \\ \int_{\partial\Omega_{k_3}} (\nabla_t \times \mathbf{u}_{hp}) \times (\mathbf{u} - \varphi_{hp}) \cdot \hat{n}_3 & dS_{k_3} \\ \int_{\partial\Omega_{k_4}} (\nabla_t \times \mathbf{u}_{hp}) \times (\mathbf{u} - \varphi_{hp}) \cdot \hat{n}_4 & dS_{k_4} \end{bmatrix} \tag{20}$$

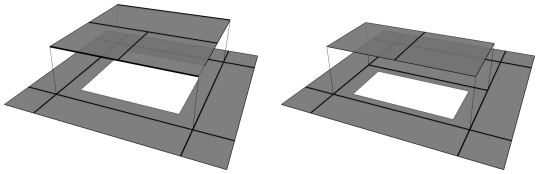
where the subscripts on k represent the four edge indices. The orientated of the indices is given in Fig. 4.

The edge-error vectors from the elements associated with each of the refinements in $\tilde{\mathbf{R}}_k$ and $\tilde{\mathbf{R}}_{\partial k}$ are multiplied by a directionality matrix which generates a weighting vector for each refinement direction. The matrices for p- and h-refinement are given in equations 21 and 22 respectively. Each row is a magnitude-4 kernel that aims to produce a large coefficient whenever the error vector points in a particular direction. The refinements take whichever direction is associated with the largest coefficient in the weighting vector $\mathbf{W}$.

In both matrices, the isotropic refinement variants have multiple associated kernels. The first, and most obvious, has a large magnitude when the edge-error coefficients are nearly equal across all edges. The other isotropic kernels produce large coefficients when two adjacent edges have large error magnitudes. In these cases, isotropic refinement is still preferable as there is no clear separation in direction that could be addressed more efficiently with an anisotropic refinement. The next two kernels identify cases where two opposing edges have significantly larger error coefficients than the other two, indicating that anisotropic refinements are best suited to improve accuracy without introducing superfluous DoFs.

$$\mathbf{W} = \begin{bmatrix} W_{(1,1)} \\ W_{(1,1)} \\ W_{(1,1)} \\ W_{(1,1)} \\ W_{(1,1)} \\ W_{(1,0)} \\ W_{(0,1)} \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 0 & 0 & 2 \\ 2 & 0 & 2 & 0 \\ 0 & 2 & 0 & 2 \\ 0 & 2 & 2 & 0 \\ 2 & 2 & 0 & 0 \\ 0 & 0 & 2 & 2 \end{bmatrix} \times \begin{bmatrix} e_{\delta k_1} \\ e_{\delta k_2} \\ e_{\delta k_3} \\ e_{\delta k_4} \end{bmatrix} \tag{21}$$

The h-refinement matrix has four additional refinement kernels associated with composite h-refinement types. These aim to efficiently address scenarios where most of the error is concentrated around a single edge. During experimental analysis, these refinement types emerged naturally over multiple iterations; thus, their explicit inclusion can yield the same result in fewer iterations.



(a) z := u-level (b) z := v-level

Fig. 5: Example of a composite h-refinement type. The central element has been refined using the "$V_{\downarrow}$" h-refinement that specifically targets its bottom edge. The same refinement is plotted twice for different z-axis definitions.

$$\mathbf{W} = \begin{bmatrix} W_T \\ W_T \\ W_T \\ W_T \\ W_T \\ W_U \\ W_V \\ W_{V_\downarrow} \\ W_{V_\uparrow} \\ W_{U_\leftarrow} \\ W_{U_\rightarrow} \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 0 & 0 & 2 \\ 2 & 0 & 2 & 0 \\ 0 & 2 & 0 & 2 \\ 0 & 2 & 2 & 0 \\ 2 & 2 & 0 & 0 \\ 0 & 0 & 2 & 2 \\ 2.5 & 0 & 0.75 & 0.75 \\ 0 & 2.5 & 0.75 & 0.75 \\ 0.75 & 0.75 & 2.5 & 0 \\ 0.75 & 0.75 & 0 & 2.5 \end{bmatrix} \times \begin{bmatrix} e_{\delta k_1} \\ e_{\delta k_2} \\ e_{\delta k_3} \\ e_{\delta k_4} \end{bmatrix} \tag{22}$$

The composite refinement of an element successively applies two anisotropic h-refinements in one step. For example, the refinement type "$V_{\downarrow}$" is invoked when the bottom edge has an overpowering error coefficient. This will apply a V-Type refinement to the element, then a U-Type refinement to the resultant child element along the bottom edge. This way, the scale of the bottom edge has been reduced, but the scale of the top edge is left unchanged, leaving its continuity with neighboring elements intact. This composite refinement is depicted in Figure 5.

Applying direction selection to both lists of partial refinements yields two lists of fully-defined refinements which we denote as $\mathbf{R}_k$ and $\mathbf{R}_{\partial k}$. These must now be combined and applied to the mesh.

E. Refinement Combination and Application

Elements can be marked for refinement up to five times: once by its surface DoF error term, and once for each of its edge DoF error terms. As such, we must be able to compute a sum of multiple refinements aimed at the same element. An extension of this process follows naturally for 3D FEM where faces must be accounted for.

For mixed scenarios (h- and p-refinements on the same element), we simply execute the p-refinement before the h-refinements in order to ensure that the resultant child elements have the increased expansion order that was intended for the

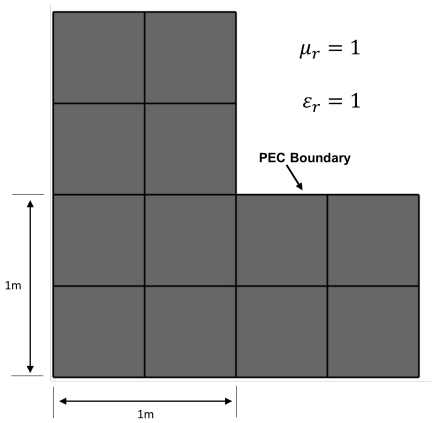
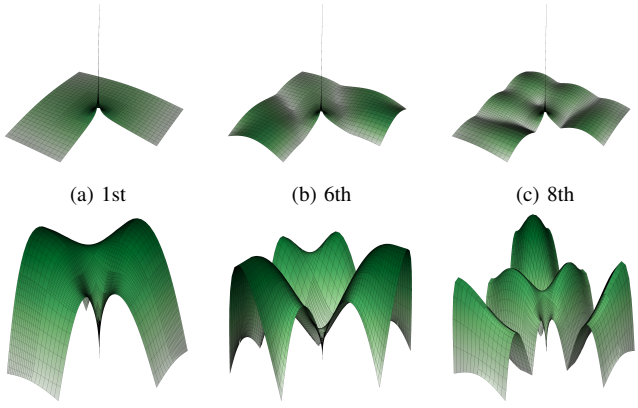


Fig. 6: Re-Entrant Corner Benchmark Mesh



(a) 1st (b) 6th (c) 8th

(d) 2nd (e) 5th (f) 9th

Fig. 7: Eigenfunctions of interest for the re-entrant corner waveguide benchmark problem. The Electric Field Magnitudes are shown for the singular (1st, 6th, 8th) and Sharp (2nd, 5th, 9th) functions. All elements are plotted with an 8x8 grid of points which is shown superimposed on the fields

parent element. Executing the refinements in the opposite order could cause p-refinements to be ignored entirety as neighboring h-refinements could remove the need for edge-type DoFs on the parent element.

When multiple h-refinements are defined over the same element, they add constructively. For example, a U-Type and V-Type refinement will add to create a T-Type refinement. P-refinements add in the obvious fashion; however we limit the magnitude to 2 in each direction to prevent over refinement.

The combined list of refinements is denoted as **R** in Fig. 1. These are applied to the mesh creating a new discretization for the subsequent iteration.

IV. Numerical Results

We use the re-entrant corner waveguide mesh shown in Fig. 6 to assess the effectiveness of the refinement algorithm described in the previous section. We target this model problem as hp-refinement is necessary to efficiently solve most of its eigenpairs and due to the availability of high-accuracy benchmark eigenvalue computations [14]. We evaluate convergence behavior for six of the waveguide's eigenfunctions, three of which are singular about the re-entrant corner and possess varying degrees of complexity elsewhere in the mesh. The other three are sharp about the re-entrant corner and present higher-energy variation elsewhere in the mesh. The electric field magnitudes for these eigenfunctions are shown in Fig. 7. We also investigate the value of anisotropic hp-adaptivity by executing the same experiment with and without anisotropic refinements. For the isotropic version of the algorithm, the same exact DWR and hp-decision procedures are used, but the direction selection algorithm always defaults to the isotropic option.

Figure 8 shows the solution convergence behavior for both classes of eigenfunction, and both versions of the refinement algorithm. Results are plotted on a cube-root–log scale where exponential convergence rates are indicated by straight lines. Each data point is associated with one refinement iteration and shows the relative error of the eigenvalue with respect to the number of degrees of freedom used in that iteration.

Exponential convergence is achieved by the isotropic and anisotropic versions of the adaptive algorithm. Figure 8 also indicates that anisotropic refinements can provide faster convergence rates and significant improvements in efficiency when the algorithm is applied to singular eigenfunctions. For all three singular cases, and for the 2nd eigenpair, there is a significant reduction in the number of DoFs needed to achieve a given accuracy. By the final iteration, the difference is quite substantial. This is a significant improvement from the *a priori* refinement strategy used in [5], where efficiency gains became slightly reduced over the course of refinement.

The same separation is not achieved for the 5th and 9th eigenpairs, indicating that further tuning of the algorithm is likely needed for cases where sharp and smooth behavior exist in close vicinity. Regardless of the slight difference in isotropic and anisotorpic results, both algorithms achieve exponential convergence on these difficult eigenfunctions, clearly exhibiting the general viability of the methodology for other challenging problems in CEM.

Figures 9 and 10 depict the final refinement states of the re-entrant corner mesh for the singular eigenpairs. The isotropic mesh arrangements are reminiscent of the *a priori* strategy used in [5], while the anisotropic strategies show less uniform refinement decisions and tend to introduce more h-refinement farther from the corner. It is also clear that these algorithms rely heavily on anisotropic p-refinements, which present significant NDoF savings. For example, in the anisotropic mesh for the first eignepair, some elements have an expansion order of 10 in one direction and 4 in the other. An equivalent isotropic refinement would require the element to support 6 additional degrees of polynomials in the other direction, which are not necessarily beneficial to solution accuracy.

V. Conclusion

By leveraging a Refinement-by-Superposition approach to FEM discretizations, we developed a powerful adaptive refinement strategy that efficiently conducts anisotropic hp-refinements to solve complex problems in computational

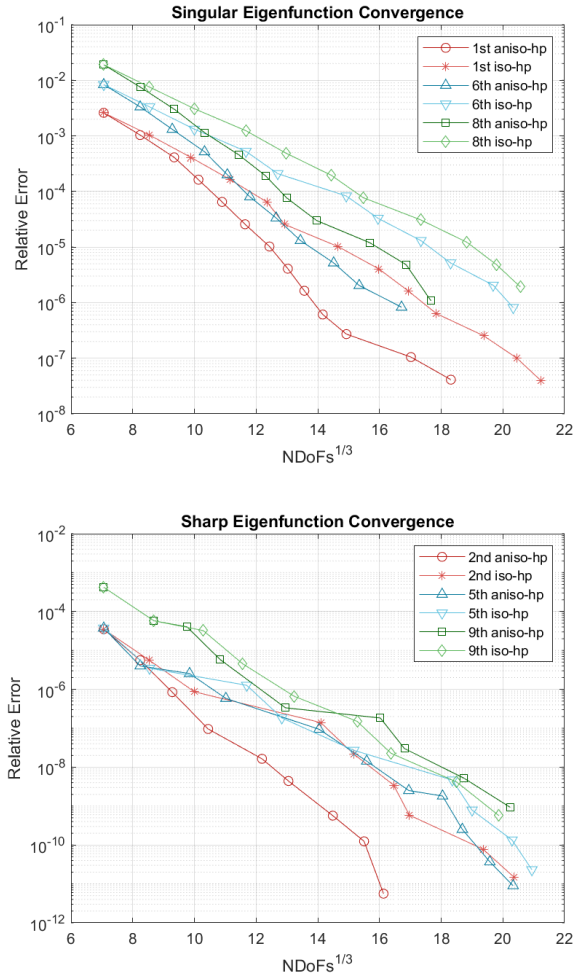
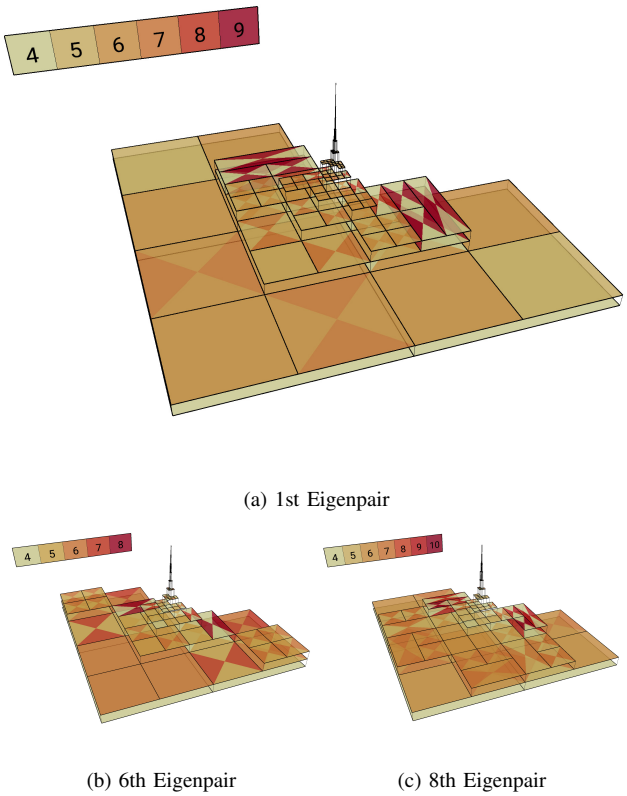


Fig. 8: Comparison of convergence rates for the isotropic and anisotropic adaptive refinement algorithms. Behavior for the Singular and Sharp eigenfunctions are shown on a cubed-root–log scale



(a) 1st Eigenpair

(b) 6th Eigenpair

(c) 8th Eigenpair

Fig. 9: Final Mesh states for the Singular Eigenpairs after *anisotropic* refinement. The keys in the top left show the expansion orders

electromagnetics and other areas of applied mathematics. This approach is composed of a DWR error estimator, two hp-decision heuristics, and a direction selection heuristic (which is necessary to guide anisotropy). Without any pre-defined knowledge of how to address a re-entrant corner–which typically introduces challenging multi-scale solution behavior–the algorithm converged exponentially on accurate solutions for all six eigenfunctions of a benchmark Maxwell Eigenvalue problem. Additionally, we noted cases where the anisotropic version of the algorithm significantly outperformed the isotropic version in terms of efficiency.

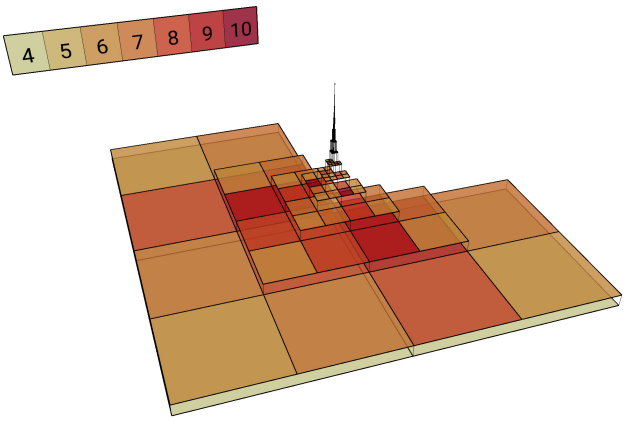
Specifically, comparing the efficiencies of the final iteration of the anisotropic and isotoropic simulations, we achieved improvements of 1.5x, 1.77x, and 2.8x on the 1st, 6th and 8th eigenvalues respectively (where efficiency is defined as the ratio of the achieved accuracy to the number of degrees of freedom used). These results are an excellent indication of the algorithms success, as they imply reduced memory consumption (or improved accuracy) in practical applications that require h-refinement around singularities. This is especially encouraging for shared memory applications (ex: running simulations on a PC), where fast simulations require that the entire system matrices fit in RAM.

For the 5th and 9th eigenvalues, efficiency gains were approximately 1x over the course of refinement, meaning that anisotropy was neither a help nor hindrance towards the simulation efficiency. Thus, we conclude that there is further room for improvement in the algorithm's implementation. It is likely that a more careful tuning of the somewhat arbitrary thresholds and directionality kernels used in the algorithm would result in a better use of the anisotropic refinements on the sharp eigenfunctions.

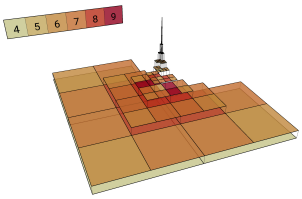
For the adaptive solution of complicated partial differential equations, the algorithm described in this work is a valuable demonstration of an adaptive refinement strategy compatible with RBS and anisotropic hp-refinements. It's success serves as further indication of the usefulness of RBS and anisotropy for addressing challenging problems in CEM. These results will likely generalize to other problem domains that require enforcement of continuity conditions over quadrilateral/hexahedral elements.

References

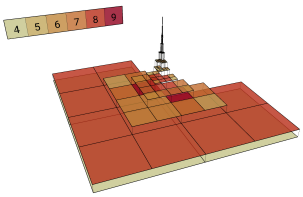
[1] J. Harmon, J. Corrado, and B. Notaros, "A refinement-by-superposition hp-method for h(curl)- and h(div)-conforming discretizations," *TechRxiv*, 6 2021.



(a) 1st Eigenpair



(b) 6th Eigenpair



(c) 8th Eigenpair

Fig. 10: Final Mesh states for the Singular Eigenpairs after*isotropic* refinement. The keys in the top left show the expansion orders

[2] N. Zander, T. Bog, S. Kollmannsberger, D. Schillinger, and E. Rank, "Multi-level hp-adaptivity: High-order mesh adaptivity without the difficulties of constraining hanging nodes," *Computational Mechanics*, vol. 55, no. 3, p. 499–517, 2015.

[3] J. Corrado, J. Harmon, B. Notaros, and M. M. Ilic, "Fem_2d: A rust package for 2d finite element method computations with extensive support for hp-refinement," 2 2022. [Online]. Available: https://www.techrxiv.org/articles/preprint/FEM_2D_A_Rust_Package_for_2D_Finite_Element_Method_Computations_with_Extensive_Support_for_hp-refinement/19166339

[4] J. Corrado, "Fem 2d," 2022. [Online]. Available: https://github.com/jeremiah-corrado/fem_2d

[5] J. Corrado, J. Harmon, and B. Notaros, "A Refinement-by-Superposition Approach to Fully Anisotropic hp-Refinement for Improved Efficiency in CEM," 10 2021. [Online]. Available: https://www.techrxiv.org/articles/preprint/A_Refinement-by-Superposition_Approach_to_Fully_Anisotropic_hp-Refinement_for_Improved_Efficiency_in_CEM/16695163

[6] N. Zander, H. Bériot, C. Hoff, P. Kodl, and L. Demkowicz, "Anisotropic multi-level hp-refinement for quadrilateral and triangular meshes," *Finite Elements in Analysis and Design*, vol. 203, p. 103700, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0168874X21001748

[7] M. M. Kostić and B. M. Kolundžija, "Maximally orthogonalized higher order bases over generalized wires, quadrilaterals, and hexahedra," *IEEE Transactions on Antennas and Propagation*, vol. 61, no. 6, pp. 3135–3148, 2013.

[8] M. M. Ilic, A. Z. Ilic, and B. M. Notaros, "Efficient large-domain 2-D FEM solution of arbitrary waveguides using p-refinement on generalized quadrilaterals," *IEEE Transactions on Microwave Theory and Techniques*, vol. 53, no. 4, pp. 1377–1383, 2005.

[9] J. N. Jomo, N. Zander, M. Elhaddad, A. Özcan, S. Kollmannsberger, R.-P. Mundani, and E. Rank, "Parallelization of the multi-level hp-adaptive finite cell method," *Computers & Mathematics with Applications*, vol. 74, no. 1, pp. 126–142, 2017, 5th European Seminar on Computing ESCO 2016. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0898122117300147

[10] J. Harmon and B. Notaros, "Adaptive hp-refinement for 2-d maxwell eigenvalue problems: Method and benchmarks," *IEEE Transactions on Antennas and Propagation*, pp. 1–1, 01 2022.

[11] W. Gui and I. Babuška, "The h, p and h-p versions of the finite element method in 1 dimension, part i." *Numerische Mathematik*, vol. 49, no. 6, pp. 577–612, Nov 1986.

[12] W. Gui and I. Babuška, "The h, p and h-p versions of the finite element method in 1 dimension, part ii." *Numerische Mathematik*, vol. 49, no. 6, pp. 613–657, Nov 1986.

[13] W. Gui and I. Babuška, "The h, p and h-p versions of the finite element method in 1 dimension, part iii." *Numerische Mathematik*, vol. 49, no. 6, pp. 659–683, Nov 1986.

[14] M. Dauge, "Benchmark computations for maxwell equations for the approximation of highly singular solutions," Aug 2004. [Online]. Available: https://perso.univ-rennes1.fr/monique.dauge/benchmax.html