The Dissertation Committee for Stefan Klaus Wilhelm Henneking certifies that this is the approved version of the following dissertation:

A Scalable *hp*-Adaptive Finite Element Software with Applications in Fiber Optics

Committee:

Leszek F. Demkowicz, Supervisor

Björn Engquist

Omar Ghattas

Jay Gopalakrishnan

Jacob Grosek

Karen E. Willcox

A Scalable *hp*-Adaptive Finite Element Software with Applications in Fiber Optics

by

Stefan Klaus Wilhelm Henneking

Dissertation

Presented to the Faculty of the Graduate School
of the University of Texas at Austin
in Partial Fulfillment
of the Requirements
for the Degree of

Doctor of Philosophy

The University of Texas at Austin
May 2021

Acknowledgements

I have many co-workers, mentors, friends, and family members to thank for their support and guidance throughout my academic journey. Without their help and dedication, I could not have completed this dissertation.

Foremost, I am grateful to my advisor and mentor, Dr. Leszek F. Demkowicz. As a teacher, he patiently explained to me the involved finite element theory he himself mastered through many decades of devoted studies. His encouragement to look for a deeper understanding of the mathematics that underpin our everyday work has shaped my thinking fundamentally. His profound appreciation for the history and people whose work we build upon has been a valuable lesson to me. As a mentor, he was able to offer the best guidance I could have hoped for at every moment of my doctoral studies. I will take his mentorship as an example that I hope to live up to in my own life.

I would like to thank Dr. Björn Engquist, Dr. Omar Ghattas, Dr. Jay Gopalakrishnan, Dr. Jacob Grosek, and Dr. Karen E. Willcox for serving on my dissertation committee. In particular, I thank Dr. Jay Gopalakrishnan and his collaborators at Portland State University—Dow Drake, Tathagata Goswami, and Benjamin Parker—for their work and collaboration on fiber amplifier simulations. I am deeply grateful to Dr. Jacob Grosek, as well as his colleagues at the Laser Division of the Air Force Research Laboratory in Albuquerque, NM, without whose expertise and commitment our research on modeling fiber laser amplifiers would not at all have been feasible.

This work was partially supported by the Air Force Office of Scientific Research

(AFOSR) grants no. FA9550-17-1-0090, no. 18RDCOR018, and no. FA9550-19-1-0237. I would like to extend my sincere thanks to Dr. Fariba Fahroo and Dr. Jean-Luc Cambier who have served as the program officers of the AFOSR Computational Mathematics program and whose continued support has been essential for this project. I would also like to thank Dr. Nathan K. Crane for the support I received to conduct software development work under Sandia National Laboratories grant 2040897-V2.

I would like to acknowledge the work of previous students in Dr. Demkowicz's group, especially Dr. Paolo Gatto, Dr. Kyungjoo Kim, Dr. Federico Fuentes and Dr. Brendan Keith for their contributions to *hp*3D. I thank Dr. Nathan V. Roberts for his expert advice on coding parallel finite element methods and for mentoring me as an intern at Sandia National Laboratories. I am very thankful to Dr. Socratis Petrides for teaching me much about solving high-frequency wave problems, patiently explaining our 3D finite element code, and spending countless hours on helpful discussions. I am thankful to Dr. Sriram Nagaraj who pioneered the first version of the DPG fiber amplifier model that I was able to build my own work on. And I thank Dr. Jaime Mora and Jacob Badger for contributing vitally important sum factorization implementations for the DPG Maxwell formulations.

I am grateful to the faculty, staff, and students of the Oden Institute for creating an amazing collaborative and collegial space. I would like to especially thank Stephanie Rodriguez, Lauren Constant, and Lisa Gentry, for going above and beyond to support me as a student throughout my doctoral studies. I also want to thank the Oden Institute for awarding me financial support as part of the National Initiative for Modeling and Simulation (NIMS) Graduate Research Fellowship Program.

At last, I wish to thank my dear family and friends who have given me unwavering support during all of these years: my mother and father, who taught me the courage to follow my ambitions; my sister, who I can always rely on; and Shadhi, who has been at my side as a supportive and loving partner. Thank you all so very much.

A Scalable *hp*-Adaptive Finite Element Software with Applications in Fiber Optics

by

Stefan Klaus Wilhelm Henneking, Ph.D.
The University of Texas at Austin, 2021

Supervisor: Leszek F. Demkowicz

In this dissertation, we present a scalable parallel version of *hp*3D—a finite element (FE) software for analysis and discretization of complex three-dimensional multiphysics applications. The developed software supports hybrid MPI/OpenMP parallelism for large-scale computation on modern manycore architectures. The focus of the effort lies on the development and optimization of the parallel software infrastructure underlying all distributed computation. We discuss the challenges of designing efficient data structures for isotropic and anisotropic *hp*-adaptive meshes with tetrahedral, hexahedral, prismatic, and pyramidal elements supporting discretization of the exact sequence energy spaces. While the code supports standard Galerkin methods, special emphasis is given to systems arising from discretization with the discontinuous Petrov–Galerkin (DPG) method. The method guarantees discrete stability by employing locally optimal test functions, and it has a built-in error indicator which we exploit to guide mesh adaptivity. In addition to interfacing with third-party packages for various tasks, we have developed our own tools including a parallel nested dissection solver suitable for scalable FE computation of waveguide geometries. We present weak-scaling results with up to 24 576 CPU cores and numerical simulations with more than one billion degrees of freedom.

The new software capabilities enable solution of challenging wave propagation problems with important applications in acoustics, elastodynamics, and electromagnetics. These applications are difficult to solve in the high-frequency regime because the FE discretization suffers from significant numerical pollution errors that increase with the wavenumber. It is critical to control these errors to obtain a stable and accurate method. We study the pollution effect for waveguide problems with more than 8 000 wavelengths in the context of robust DPG FE discretizations for the time-harmonic Maxwell equations. We also discuss adaptive refinement strategies for multi-mode fiber waveguides where the propagating transverse modes must be resolved sufficiently. Our study shows the applicability of the DPG error indicator to this class of problems.

Finally, we present both modeling and computational advancements to a unique three-dimensional DPG FE model for the simulation of laser amplification in a fiber amplifier. Fiber laser amplifiers are of interest in communication technology, medical applications, military defense capabilities, and various other fields. Silica fiber amplifiers can achieve high-power operation with great efficiency. At high optical intensities, multi-mode amplifiers suffer from undesired thermal coupling effects which pose a major obstacle in power-scaling of such devices. Our nonlinear 3D vectorial model is based on the time-harmonic Maxwell equations, and it incorporates both amplification via an active dopant and thermal effects via coupling with the heat equation. The model supports co-, counter-, and bi-directional pumping configurations, as well as inhomogeneous and anisotropic material properties. The high-fidelity simulation comes at the cost of a high-order FE discretization with many degrees of freedom per wavelength. To make the computation more feasible, we have developed a novel longitudinal model rescaling, using artificial material parameters with the goal of preserving certain quantities of interest. Numerical tests demonstrate the applicability and utility of this scaled model in the simulation of an ytterbium-doped, step-index fiber amplifier that experiences laser amplification and heating.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation

***hp* finite elements.** *hp*-adaptive methods are considered to be among the most powerful but difficult to implement finite element (FE) technologies. By orchestrating an optimal distribution of element size h and polynomial order p, one can achieve exponential rates of convergence for irregular solutions. Combined with the preasymptotic stability delivered by the discontinuous Petrov–Galerkin (DPG) method [32], the methodology offers a unique tool for fast and reliable discretization of challenging modeling problems. The finite element software *hp*3D, developed by Demkowicz et al. [37, 57, 83], supports many advanced FE technologies, including exact-sequence conforming discretizations, fully anisotropic *hp*-refinements, and hybrid meshes with elements of "all shapes" (tetrahedra, hexahedra, prisms, pyramids). Because of that, *hp*3D is applicable to a great variety of complex multiphysics models. However, these models often require high-order discretizations, complex geometries, and coupled multiphysics variables, yielding very large problems that typically exceed the memory and computational capacities of a single compute node. While parallel FE libraries are widely available, few support *hp* meshes. For that reason, a distributed-memory, scalable parallel implementation of *hp*3D is of great interest to advancing the state-of-the-art computational capabilities in *hp*-adaptive FE computation.

Fiber laser amplifiers. Optical fiber amplifiers play a critical role in our world. For example, optic communication technology relies on fiber amplifiers to transfer data over long distance [107], and astronomers use fiber lasers as highly coherent light sources to calibrate telescopes [126]. Silica fiber amplifiers have emerged as excellent candidates for achieving high power outputs with great efficiency while providing superior beam quality. The research in, fabrication of, and applications for optical fibers have greatly benefited from the growth and ubiquity of the telecommunications industry, leading to more reliable, lower cost, and higher power fiber laser systems [75, 66]. Nonetheless, the efforts of power-scaling beam combinable fiber amplifiers have encountered numerous roadblocks in the form of nonlinear effects [75]. In continuous wave operation, stimulated Brillouin scattering (SBS) [84] and stimulated Raman scattering (SRS) [117] impose limits on the achievable power. One effective mitigation strategy for these nonlinear effects is to increase the core size while simultaneously decreasing the fiber length. Unfortunately, these large-mode-area fibers permit additional guided higher-order modes, which ultimately can reduce the output beam quality if any significant amount of power is transmitted in these modes. This happens above a certain power threshold, at which point the fundamental mode begins to exchange energy with the higher-order modes. This effect is called the transverse mode instability (TMI) [44]. The origins of these mode instabilities are understood to be tied to thermal effects and the interference patterns between the guided modes of the amplifier [76].

Experimental investigations of nonlinear effects in fiber amplifiers are slow, costly, and provide limited data. Indeed, the fabrication of rare-earth doped fiber amplifiers is very expensive, and manufacturing techniques limit the options and time-to-delivery of custom configurations. Additionally, experimental data are limited with regard to accuracy and placement of sensors. In fact, measurements are largely confined to observing the output of the fiber amplifier. Therefore, the need for modeling and simulation of fiber amplifiers is evident.

1.2 Objectives

The objectives of this dissertation are:

- The development of a mathematical model and DPG finite element discretization for analyzing optical fiber amplifiers.
- The parallelization and optimization of the *hp*3D finite element software for supporting large-scale numerical simulations of complex multiphysics applications.

1.3 Background

Following a brief introduction to the Maxwell equations, this section provides an overview of common approaches to fiber amplifier modeling, some of the challenges related to solving high-frequency wave propagation problems, and applications of the discontinuous Petrov–Galerkin method.

1.3.1 Maxwell's equations

The propagation of optical fields in fibers is governed by Maxwell's equations. Stated in differential form, the Maxwell equations are,

$$\nabla \cdot \boldsymbol{\mathcal{D}} = \rho_f, \tag{1.1}$$

$$\nabla \cdot \boldsymbol{\mathcal{B}} = 0, \tag{1.2}$$

$$\nabla \times \boldsymbol{\mathcal{E}} = -\frac{\partial \boldsymbol{\mathcal{B}}}{\partial t}, \tag{1.3}$$

$$\nabla \times \boldsymbol{\mathcal{H}} = \boldsymbol{\mathcal{J}}_f + \frac{\partial \boldsymbol{\mathcal{D}}}{\partial t}, \tag{1.4}$$

with the definitions given in Table 1.1 (see Table A.1 for an overview of SI units).

$\boldsymbol{\mathcal{J}}_f$ and ρ_f represent the sources for the electromagnetic field. $\boldsymbol{\mathcal{D}}$ is also referred to

Symbol	Description	Unit
$\boldsymbol{\mathcal{E}}$	Electric field vector	V/m
$\boldsymbol{\mathcal{H}}$	Magnetic field vector	A/m
$\boldsymbol{\mathcal{D}}$	Electric flux density vector	C/m
$\boldsymbol{\mathcal{B}}$	Magnetic flux density vector	$\mathrm{Wb/m^2}$
$\boldsymbol{\mathcal{J}}_f$	(Free) Current density vector	$\mathrm{A/m^2}$
ρ_f	(Free) Charge density	$\mathrm{C/m^3}$

Table 1.1: Maxwell's equations: symbols

as the electric displacement vector, and $\partial\boldsymbol{\mathcal{D}}/\partial t$ is the displacement current $\boldsymbol{\mathcal{J}}_d$. Maxwell's equations imply the continuity equation for charge and current:

$$\nabla \cdot \boldsymbol{\mathcal{J}}_f + \frac{\partial \rho_f}{\partial t} = 0. \tag{1.5}$$

Remark: The electric polarization $\boldsymbol{\mathcal{P}}$, defined in (1.6), produces a bound charge $\rho_b = -\nabla\cdot\boldsymbol{\mathcal{P}}$; a change in the electric polarization causes a polarization current $\boldsymbol{\mathcal{J}}_p = \partial\boldsymbol{\mathcal{P}}/\partial t$. The magnetic polarization $\boldsymbol{\mathcal{M}}$, defined in (1.7), produces a bound current $\boldsymbol{\mathcal{J}}_b = \nabla \times \boldsymbol{\mathcal{M}}$. The Maxwell equations imply that the total current density $\boldsymbol{J} = \boldsymbol{\mathcal{J}}_f + \boldsymbol{\mathcal{J}}_p + \boldsymbol{\mathcal{J}}_b$ and charge density $\rho = \rho_f + \rho_b$ satisfy the continuity equation.

Optical fibers are made of silica glass. In the absence of free charges in a *dielectric* medium (electrical conductivity $\sigma = 0$) such as silica glass, $\boldsymbol{\mathcal{J}}_f = \sigma\boldsymbol{\mathcal{E}} = \mathbf{0}$ (by Ohm's law) and $\rho_f = 0$. The flux densities $\boldsymbol{\mathcal{D}}$ and $\boldsymbol{\mathcal{B}}$ are a response to the electromagnetic fields propagating inside the medium. The following constitutive equations [64, 74] relate the flux densities to the field vectors:

$$\boldsymbol{\mathcal{D}} = \varepsilon_0\boldsymbol{\mathcal{E}} + \boldsymbol{\mathcal{P}}, \tag{1.6}$$

$$\boldsymbol{\mathcal{H}} = \frac{1}{\mu_0}\boldsymbol{\mathcal{B}} - \boldsymbol{\mathcal{M}}, \tag{1.7}$$

where $\boldsymbol{\mathcal{P}}$ and $\boldsymbol{\mathcal{M}}$ are the induced electric and magnetic polarization, respectively, ε_0 is the electric permittivity in vacuum, μ_0 is the magnetic permeability in vacuum, and $c = 1/\sqrt{\varepsilon_0\mu_0}$

is the speed of light (cf. Table A.2). $\boldsymbol{\mathcal{P}}$ and $\boldsymbol{\mathcal{M}}$ are generally nonlinear functions of the electric and magnetic field, obtained by constitutive modeling. In a *linear* medium,

$$\boldsymbol{\mathcal{D}} = \varepsilon \boldsymbol{\mathcal{E}}, \tag{1.8}$$

$$\boldsymbol{\mathcal{B}} = \boldsymbol{\mu} \boldsymbol{\mathcal{H}}. \tag{1.9}$$

In *isotropic* materials, ε and μ are scalar values; in *anisotropic* materials, $\boldsymbol{\varepsilon}$ and $\boldsymbol{\mu}$ are second-rank tensors. In a *homogeneous* medium, ε and μ are uniform. Assuming the dielectric medium is linear, isotropic, and homogeneous, we can rewrite the Maxwell system as:

$$\nabla \cdot \boldsymbol{\mathcal{E}} = 0, \tag{1.10}$$

$$\nabla \cdot \boldsymbol{\mathcal{H}} = 0, \tag{1.11}$$

$$\nabla \times \boldsymbol{\mathcal{E}} = -\mu \frac{\partial \boldsymbol{\mathcal{H}}}{\partial t}, \tag{1.12}$$

$$\nabla \times \boldsymbol{\mathcal{H}} = \varepsilon \frac{\partial \boldsymbol{\mathcal{E}}}{\partial t}. \tag{1.13}$$

Time-harmonic Maxwell equations in a linear medium. The time-varying electric and magnetic fields are $\boldsymbol{\mathcal{E}}$ and $\boldsymbol{\mathcal{H}}$; let the corresponding time-harmonic fields be denoted by $\boldsymbol{E}$ and $\boldsymbol{H}$, respectively. Then, the time-harmonic version of the Maxwell equations is derived using the following ansatz:

$$\boldsymbol{\mathcal{E}}(x, y, z, t) = \mathfrak{Re}\left\{\boldsymbol{E}(x, y, z) e^{i\omega t}\right\}, \tag{1.14}$$

$$\boldsymbol{\mathcal{H}}(x, y, z, t) = \mathfrak{Re}\left\{\boldsymbol{H}(x, y, z) e^{i\omega t}\right\}, \tag{1.15}$$

where $i = \sqrt{-1}$, ω is the angular frequency of the (monochromatic) light, and $\mathfrak{Re}\{\cdot\}$ refers to the real part of the complex-valued vector field.

The time-harmonic linear Maxwell equations are then given by:

$$\nabla \times \boldsymbol{E} = -i\omega\mu\boldsymbol{H}, \tag{1.16}$$

$$\nabla \times \boldsymbol{H} = i\omega\varepsilon\boldsymbol{E}. \tag{1.17}$$

Remark: It is important to note that we chose the ansatz $\exp(+iwt)$ *while others may choose the ansatz* $\exp(-iwt)$*. Either choice is suitable, but it is important to keep the particular ansatz in mind for correctness of the sign as we proceed.*

1.3.2 Modeling of fiber amplifiers

The state of the art in numerical simulation of fiber amplifiers consists of beam propagation methods (BPMs), coupled mode theory (CMT) approaches, and a variety of other models that make certain approximations and assumptions to achieve simplified, but efficient simulations. Some models couple to a time-dependent heat equation [125, 96], others solve the thermal problem in the frequency domain [67, 116]. Generally, these models are derived from the time-harmonic Maxwell equations and make certain assumptions to arrive at simpler models that are easier to discretize and compute. BPMs postulate that the propagating fields are guided along the longitudinal fiber direction with some propagation constant (wavenumber); they proceed to solve the corresponding field envelope by stepping along the fiber in the wave propagation direction, treating the problem as an initial value problem. Both 3D vectorial BPMs [110] and, more commonly, scalar BPMs [59, 124] have been proposed. These BPMs tend to work better in frequency domain problems where each transverse guided mode is given its own unique frequency and wavenumber, leading to a coupled system with a different partial differential equation (PDE) for each guided mode. Otherwise, one is left with a single PDE with a given wavenumber that must capture all of the guided modes simultaneously, straining the limits of the slowly varying envelope approximation.

Scalar models additionally assume that the propagating fields are strictly polarized in one of the transverse directions, and that the fiber is polarization maintaining, thereby eliminating two of the vector components from the equations. Moreover, the field envelopes may be assumed to be slowly varying in the direction of propagation, reducing the model to a 2D BPM. A further simplification is made by the CMT approach that decomposes the electric field into a discrete set of propagating guided modes of the fiber, which are explicitly connected to one another via coupling coefficients [96, 65].

These models, posed as an initial value problem, are most amenable to forward propagating light. While they can be made to handle bi-directional light propagation, doing so tends to be very computationally intensive. For this reason, Brillouin scattering in fibers is most often modeled separately in codes written specifically for that phenomenon.

1.3.3 Numerical solution of high-frequency wave problems

The pollution effect. Discretizing the time-harmonic Maxwell equations for an optical fiber domain poses the difficulty of capturing a high-frequency wave over many wavelengths. It is well-known that an accurate numerical solution for wave problems with high frequency is difficult to obtain [9, 89, 48]. Finite element discretizations are usually dependent on satisfying the *Nyquist stability criterion*, implying that all propagating wave frequencies must be "resolved" to a certain extent in order to have a stable discretization. Typically, this requires a FE mesh where the element size h is on the order of one wavelength or smaller. However, even if the Nyquist criterion is satisfied, the error still grows with increasing wavenumber; this is called the *numerical pollution effect* [9]. Some FE methods such as the discontinuous Petrov–Galerkin method can circumvent the stability problem and deliver a robust discretization for any wavenumber [127, 39, 102, 104], but they do not eliminate the pollution error in multiple dimensions. Numerical pollution can manifest itself in different forms: commonly, we observe a diffusive effect causing wave attenuation and/or a dispersive effect resulting in

a phase shift. It is critical to control the pollution error for obtaining accurate results. The pollution effect in wave propagation has been studied extensively for the Bubnov–Galerkin FE method, as well as discontinuous Galerkin methods, least-squares methods, and various other approaches (see [72, 73, 9, 40, 3, 89, 39] and references therein).

Solution schemes. Overcoming the pollution error in high-frequency applications is computationally expensive as it typically requires fine meshes and high-order discretizations. Additionally, acoustic or electromagnetic wave propagation problems in the time-harmonic setting lead to PDEs with an indefinite Helmholtz or Maxwell operator, respectively. Therefore, the resulting linear systems are both very large and difficult to solve. In particular, many advanced solver techniques are not directly applicable to these problems: direct solvers are often prohibitively expensive both in memory and computational complexity while most state-of-the-art iterative solvers are not reliably convergent for these systems [48]. In the last decade, however, notable advances were made toward more robust iterative solution schemes [56], e.g., the sweeping preconditioner [45, 46, 123], source transfer methods [22, 85], the L-sweeps method [121], shifted Laplacian techniques [55], multigrid methods [120] including a DPG-based adaptive multigrid scheme [103, 104], and others. Still, these techniques are not always scalable in parallel computation, sometimes are limited to certain types of domains or source functions, or may have other limitations. Thus, developing fast and robust solution schemes for wave propagation problems remains an active research field.

1.3.4 Discontinuous Petrov–Galerkin methods

The discontinuous Petrov–Galerkin (DPG) FE method of Demkowicz and Gopalakrishnan [30, 29, 33, 127, 32] offers guaranteed discrete stability for well-posed variational formulations. The stability comes from using so-called *optimal test functions* that are computed *locally* (i.e., per element) on-the-fly. To localize the determination of optimal test functions, *broken* (discontinuous) *test spaces* are used. These broken test spaces give rise to additional

interface terms on the element boundaries. In other words, the DPG method is a hybrid method with a group variable consisting of the solution defined on elements and additional *traces* defined on the mesh skeleton. The additional unknowns (traces) are discretized by using traces of standard element shape functions [32]. The DPG method can be classified as a minimum residual method with the residual measured in an (approximate) dual norm; it therefore always delivers a positive definite Hermitian matrix. It can also be formulated as a mixed method with an extra unknown—the *Riesz representation of the residual*—that serves as a built-in a-posteriori error indicator. The method is therefore *adaptivity ready* without further work on a-posteriori error estimation. The guaranteed stability of the DPG method comes at the computational cost of first locally determining optimal test functions and then solving a linear system with additional (trace) degrees of freedom. The method is thus more expensive than the weak Galerkin formulation. However, this increased computational complexity can be justified for challenging problems where standard techniques may not yield robustly stable discretizations. Its stability properties make the DPG method particularly applicable to solving high-frequency wave propagation problems [103, 104] or singular perturbation problems [71, 51, 111], where preasymptotic stability is essential for guiding efficient *hp*-adaptivity.

The DPG method can be applied to any well-posed variational formulation [20]. It has been used to solve applications in convection-diffusion [34, 18], elasticity [99, 79, 49], fluid dynamics [21, 108, 80, 106], plate bending [53, 54], acoustic wave propagation [127, 39, 60, 19], Maxwell's equations [20, 98, 102], and others (see also [15] and references therein). Additionally, DPG methods have been used for space-time discretizations [36, 63, 47], time-marching schemes [52, 109, 95, 94], polygonal and polyhedral elements [7, 10], and in the Banach space setting [93, 86]. DPG-based preconditioners have been developed in [62, 87, 13, 104].

1.4 Contributions

Fiber amplifier model. The first accomplishment of this dissertation is the development of a 3D vectorial Maxwell model for the simulation of laser gain in optical fiber amplifiers. The model is based on a DPG finite element discretization for the ultraweak variational formulation of the time-harmonic Maxwell equations (see Chapter 2). This model is unique in several aspects. Rather than treating the problem as an initial value problem, the model states a boundary value problem. In this approach, we make as few assumptions as possible with the aim to provide a tool for high-fidelity numerical simulations. The 3D model builds on the work of Nagaraj et al. [98], who have used a Maxwell model to simulate passive Raman amplification in a fiber. Building on that framework, we added the ability to model the more common active gain amplification through a rare-earth, lanthanide metal dopant in the fiber core region [70]. Like Raman gain, active gain causes our Maxwell system to become nonlinear. Additionally, we augmented our simulation with a thermal model that analyzes the interplay between heat deposition, the induced thermal perturbations to the refractive index of the medium, and the response of the propagating optical fields to this perturbed medium. To the best of our knowledge, this is the only fiber model that is computed using a 3D finite element discretization. This fiber amplifier model is discussed in Chapter 5.

Solving the time-harmonic Maxwell model numerically poses a challenge because it is inherently a high-frequency wave propagation problem. The FE discretization is therefore suffering from numerical pollution errors. We have investigated the pollution effect for DPG finite element discretizations of the time-harmonic Maxwell equations in a numerical study for waveguide problems with more than 8 000 wavelengths and high order of approximation [69]. We corroborate theoretical results by Melenk and Sauter [89, 90] and provide guidance on how to best discretize wave problems with high frequency. The results and our approach to mitigating these errors are discussed in Chapter 4; there, we also study how the DPG method can be effectively used to perform isotropic or anisotropic mesh adaptivity to resolve higher-

order modes in fiber simulations. The DPG methodology offers a framework for setting up automatic *hp*-adaptivity, which may be pivotal to capturing the onset of nonlinear effects in a fiber amplifier.

Finite element software. The second key contribution of this dissertation is the development of a parallel MPI/OpenMP version of the *hp*3D finite element code [37, 57, 83], detailed in Chapter 3. This effort extends the capabilities of the FE software from shared-memory, single-node computation to distributed-memory computation across many compute nodes. The underlying distributed data structures and parallel routines fully support all of the advanced features of *hp*3D, particularly the *hp*-adaptive refinement capabilities. The parallel code also exploits shared-memory parallelization via OpenMP threading and performs efficiently in hybrid MPI/OpenMP computation. Additionally, interfaces to well-established third-party libraries were created for various tasks: linear solve (MUMPS [4], PETSc [12]), mesh partitioning and load balancing (ParMETIS [78], PT-Scotch [23], Zoltan [41]), I/O and visualization (HDF5 [122], ParaView [2]), and others. Besides the parallelization, several other optimizations contributed to increasing the simulation capabilities of *hp*3D. This included the development of a more efficient computation for conforming traces in DPG discretizations and the implementation of a static condensation module for accelerating the element assembly procedure.

For solving the fiber amplifier model at large scale, we developed a custom parallel nested dissection solver that exploits the structure of the distributed linear system arising from discretizing the fiber domain. This solver is discussed in Chapter 3.

In Chapter 6, we study the performance and scalability of the parallel code for the fiber amplifier simulation, using the Stampede2 supercomputer at the Texas Advanced Computing Center (TACC). We also conduct large-scale numerical experiments for various configurations of the fiber amplifier model. We were able to scale the size of the full 3D fiber model simulation from computing less than 100 wavelengths to solving circa 10 000 wavelengths.

Chapter 2

Discontinuous Petrov–Galerkin (DPG) Methodology

The DPG finite element method can be introduced from various points of view [32]. We first introduce it as a Petrov–Galerkin method with optimal test functions that directly addresses the issue of discrete stability in an abstract variational setting. Two other perspectives are then discussed: DPG as a minimum residual method, and DPG as a mixed method. Each point of view provides new insight into the methodology.

2.1 The ideal DPG method

Consider an abstract variational problem of the form,

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

where $\mathcal{U}$ (trial space) and $\mathcal{V}$ (test space) are Hilbert spaces equipped with norms $\|\cdot\|_{\mathcal{U}}$ and $\|\cdot\|_{\mathcal{V}}$ induced by the inner products $(\cdot,\cdot)_{\mathcal{U}}$ and $(\cdot,\cdot)_{\mathcal{V}}$, respectively; $b(\cdot,\cdot)$ is a continuous bilinear (sesquilinear) form on $\mathcal{U}\times\mathcal{V}$ (with continuity constant M),

$$|b(u,v)| \leq M \|u\|_{\mathcal{U}} \|v\|_{\mathcal{V}}, \tag{2.2}$$

that satisfies the continuous inf–sup condition (with inf–sup constant γ),

$$\inf_{\|u\|_{\mathcal{U}}=1} \sup_{\|v\|_{\mathcal{V}}=1} |b(u,v)| =: \gamma > 0; \tag{2.3}$$

and the continuous linear (antilinear) form $l(\cdot)$ satisfies the compatibility condition,

$$l(v) = 0 \ \forall v \in \mathcal{V}_0, \quad \text{where } \mathcal{V}_0 := \{v \in \mathcal{V} : \ b(u,v) = 0 \ \forall u \in \mathcal{U}\}. \tag{2.4}$$

Let $\mathcal{U}'$ and $\mathcal{V}'$ denote the space of continuous linear (antilinear) functionals on $\mathcal{U}$ and $\mathcal{V}$, respectively. By the Babuška–Nečas theorem, the variational problem (2.1) is well-posed, i.e., there exists a unique solution $u \in \mathcal{U}$ that depends continuously upon the data,

$$\|u\|_{\mathcal{U}} \leq \frac{1}{\gamma} \|l\|_{\mathcal{V}'}. \tag{2.5}$$

Consider finite-dimensional subspaces $\mathcal{U}_h \subset \mathcal{U}$ and $\mathcal{V}_h \subset \mathcal{V}$, where $\dim(\mathcal{U}_h) = \dim(\mathcal{V}_h)$, and the corresponding discrete abstract variational problem,

$$\begin{cases} u_h \in \mathcal{U}_h \subset \mathcal{U}, \\ b(u_h, v_h) = l(v_h), \ v_h \in \mathcal{V}_h \subset \mathcal{V}. \end{cases} \tag{2.6}$$

If the discrete inf–sup condition is satisfied, i.e.,

$$\inf_{\|u_h\|_{\mathcal{U}}=1} \sup_{\|v_h\|_{\mathcal{V}}=1} |b(u_h,v_h)| =: \gamma_h > 0, \tag{2.7}$$

then the discrete problem (2.6) is well-posed, and by Babuška's theorem [8, 27],

$$\|u - u_h\|_{\mathcal{U}} \leq \frac{M}{\gamma_h} \inf_{\omega_h \in \mathcal{U}_h} \|u - \omega_h\|_{\mathcal{U}}, \tag{2.8}$$

where u is the exact solution of (2.1), M/γ_h is the stability constant, and $\inf_{\omega_h \in \mathcal{U}_h} \|u - \omega_h\|_{\mathcal{U}}$

is the best approximation error measured in the trial norm $\|\cdot\|_{\mathcal{U}}$. The continuous inf–sup condition (2.3) does *not* in general imply the discrete inf–sup condition (2.7). A natural question arises: is it possible to find a test space $\mathcal{V}_h$ such that (2.3)⇒(2.7)?

2.1.1 A Petrov–Galerkin method with optimal test functions

In the DPG method, the issue of discrete stability is solved by finding a unique test space, called the *optimal test space* $\mathcal{V}^{\text{opt}}$. Given any trial space $\mathcal{U}_h \subset \mathcal{U}$, we define its optimal test space by

$$\mathcal{V}^{\text{opt}} := T(\mathcal{U}_h), \tag{2.9}$$

where the *trial-to-test operator* $T : \mathcal{U} \to \mathcal{V}$ is defined by

$$(Tu_h, v)_{\mathcal{V}} = b(u_h, v) \quad \forall u_h \in \mathcal{U}_h, v \in \mathcal{V}. \tag{2.10}$$

For any $u_h \in \mathcal{U}_h$, equation (2.10) uniquely defines Tu_h by the Riesz representation theorem. Let $\mathcal{B} : \mathcal{U} \to \mathcal{V}'$ denote the linear operator induced by the form $b(\cdot, \cdot)$,

$$\langle \mathcal{B}u, v\rangle_{\mathcal{V}'\times\mathcal{V}} = b(u, v), \ v \in \mathcal{V}, \tag{2.11}$$

where $\langle \cdot, \cdot \rangle_{\mathcal{V}'\times\mathcal{V}}$ denotes the duality pairing on $\mathcal{V}' \times \mathcal{V}$. Then, $T = \mathcal{R}_{\mathcal{V}}^{-1}\mathcal{B}$, where $\mathcal{R}_{\mathcal{V}} : \mathcal{V} \to \mathcal{V}'$ is the Riesz map. In other words, for every trial function u_h, the trial-to-test operator defines a unique optimal test function, $v_{u_h} = \mathcal{R}_{\mathcal{V}}^{-1}\mathcal{B}u_h$. The optimal test functions realize the supremum in the inf–sup condition. Indeed,

$$\sup_{0\neq v\in\mathcal{V}} \frac{|b(u_h, v)|}{\|v\|_{\mathcal{V}}} = \sup_{0\neq v\in\mathcal{V}} \frac{|(Tu_h, v)_{\mathcal{V}}|}{\|v\|_{\mathcal{V}}} = \|Tu_h\|_{\mathcal{V}} = \frac{|b(u_h, v_{u_h})|}{\|v_{u_h}\|_{\mathcal{V}}}. \tag{2.12}$$

Therefore, $\gamma_h \geq \gamma$, i.e., discrete stability is guaranteed by construction.

2.1.2 A minimum residual method

An equivalent formulation arises from minimizing the residual in the dual test norm,

$$u_h = \underset{w_h \in \mathcal{U}_h}{\arg\min} \|l - \mathcal{B}w_h\|^2_{\mathcal{V}'} = \underset{w_h \in \mathcal{U}_h}{\arg\min} \|\mathcal{R}_{\mathcal{V}}^{-1}(l - \mathcal{B}w_h)\|^2_{\mathcal{V}}, \tag{2.13}$$

where we used the fact that the Riesz map is an isometric isomorphism. u_h is the solution to (2.6) with optimal test functions if and only if u_h minimizes the residual in (2.13). Taking the Gâteaux derivative, we obtain a minimum residual formulation:

$$\begin{cases} u_h \in \mathcal{U}_h, \\ (\mathcal{R}_{\mathcal{V}}^{-1}(l - \mathcal{B}u_h), \mathcal{R}_{\mathcal{V}}^{-1}\mathcal{B}w_h)_{\mathcal{V}} = 0, \quad w_h \in \mathcal{U}_h. \end{cases} \tag{2.14}$$

Furthermore, we define the energy norm $\|\cdot\|_E$ on the trial space $\mathcal{U}$ by

$$\|u\|_E := \|\mathcal{B}u\|_{\mathcal{V}'} = \|\mathcal{R}_{\mathcal{V}}^{-1}\mathcal{B}u\|_{\mathcal{V}}. \tag{2.15}$$

By construction, the stability constant is equal to 1 in the energy norm; hence,

$$\|u - u_h\|_E \leq \inf_{w_h \in \mathcal{U}_h} \|u - w_h\|_E, \tag{2.16}$$

so u_h is the best approximation in the energy norm.

2.1.3 A mixed method

Let ψ be defined as the Riesz representation of the residual,

$$\psi := \mathcal{R}_{\mathcal{V}}^{-1}(l - \mathcal{B}u_h). \tag{2.17}$$

When u_h minimizes the residual (2.13), then

$$(\psi, \mathcal{R}_{\mathcal{V}}^{-1}\mathcal{B}w_h)_{\mathcal{V}} = 0, \quad w_h \in \mathcal{U}_h. \tag{2.18}$$

We arrive at a mixed Galerkin formulation:

$$\begin{cases} u_h \in \mathcal{U}_h, \ \psi \in \mathcal{V}, & \\ (\psi, v)_{\mathcal{V}} + b(u_h, v) = l(v), & v \in \mathcal{V}, \\ b(w_h, \psi) = 0, & w_h \in \mathcal{U}_h. \end{cases} \tag{2.19}$$

The error measured in the energy norm can be computed explicitly,

$$\|u - u_h\|_E = \|\mathcal{B}(u - u_h)\|_{\mathcal{V}'} = \|l - \mathcal{B}u_h\|_{\mathcal{V}'} = \|\mathcal{R}_{\mathcal{V}}^{-1}(l - \mathcal{B}u_h)\|_{\mathcal{V}} = \|\psi\|_{\mathcal{V}}, \tag{2.20}$$

hence $\|\psi\|_{\mathcal{V}}$ offers a built-in a-posteriori error indicator. Finally, note that the choice of the test norm $\|\cdot\|_{\mathcal{V}}$ is critical, as it dictates the norm in which the method converges.

Optimal test norm. The ideal DPG method delivers the orthogonal projection in the energy norm, i.e.,

$$\|u - u_h\|_E = \inf_{w_h \in \mathcal{U}_h} \|u - w_h\|_E. \tag{2.21}$$

Then, given a trial norm $\|\cdot\|_{\mathcal{U}}$, we can ask how to choose the test norm $\|\cdot\|_{\mathcal{V}}$, so that

$$\|u - u_h\|_{\mathcal{U}} \lesssim \|u - u_h\|_E = \|\psi\|_{\mathcal{V}}. \tag{2.22}$$

Consider the linear operator $\mathcal{B}' : \mathcal{V} \to \mathcal{U}'$, induced by the bilinear form $b(\cdot, \cdot)$,

$$\mathcal{B}' : \mathcal{V} \ni v \to \langle \mathcal{B}'v, \cdot \rangle_{\mathcal{U}' \times \mathcal{U}} = \overline{b(\cdot, v)} \in \mathcal{U}', \tag{2.23}$$

where $\overline{b(\cdot, v)}$ denotes the complex conjugate of $b(\cdot, v)$. Suppose that $\mathcal{B}$ is injective; then,

$$\|v\|_{\mathcal{V}} := \sup_{u \in \mathcal{U}} \frac{|b(u,v)|}{\|u\|_{\mathcal{U}}} = \|\mathcal{B}'v\|_{\mathcal{U}'}, \tag{2.24}$$

defines a norm, and

$$\|u - u_h\|_E = \|u - u_h\|_{\mathcal{U}}. \tag{2.25}$$

Use of this test norm within the ideal DPG method delivers the projection in the trial norm $\|\cdot\|_{\mathcal{U}}$; it is therefore called the *optimal test norm* [32].

Among the various variational formulations that can be discretized with the DPG method, the ultraweak formulation is special because the optimal test norm can be derived explicitly. We will show this for the case of the ultraweak Maxwell formulation in Section 2.5.1.

2.2 Breaking the test space

In the discussion so far, we have neglected computational aspects of the DPG method. One question that arises immediately in the context of practicality is the cost of the inversion of the global Riesz map $\mathcal{R}_{\mathcal{V}}$. Let $\Omega \subset \mathbb{R}^3$ denote a bounded domain (open and connected set) with Lipschitz boundary $\Gamma \equiv \partial\Omega$, and let Ω_h denote a suitable finite element triangulation of Ω with mesh skeleton Γ_h. By "breaking" the test space, i.e., employing a larger discontinuous test space, $\mathcal{V}(\Omega_h) \supset \mathcal{V}(\Omega)$, the inversion of the Riesz map on Ω is localized and can be done independently *element-wise*. The element-local computational costs are still significant, but they can be parallelized efficiently. By enlarging the test space, new interface terms arise on the mesh skeleton with interface unknowns $\hat{u}$. The resulting variational problem is

$$\begin{cases} u \in \mathcal{U}, \hat{u} \in \hat{\mathcal{U}}, \\ b(u,v) + \langle \hat{u}, v \rangle_{\Gamma_h} = l(v), \ v \in \mathcal{V}(\Omega_h), \end{cases} \tag{2.26}$$

where $\langle \cdot, \cdot \rangle_{\Gamma_h}$ denotes an appropriate duality pairing on the mesh skeleton. The new interface unknowns may be interpreted as Lagrange multipliers that enforce conformity of test functions [32, 27]. The stability of the formulation with broken test spaces is inherited from the continuous problem. In particular, the broken formulation (2.26) is well-posed with a mesh-independent stability constant of the same order as the stability constant for the continuous problem [20].

2.3 The practical DPG method

Until now, the trial-to-test operator T has only been defined in the infinite-dimensional setting (2.10). To compute optimal test functions in practice, the inversion of the global Riesz map must be approximated on a *truncated* finite-dimensional test space $\mathcal{V}^r \subset \mathcal{V}$, where $\dim(\mathcal{V}^r) \gg \dim(\mathcal{U}_h)$; $\mathcal{V}^r$ is also called the *enriched test space* [31, 61]. With the Riesz map defined on this truncated test space, $\mathcal{R}_{\mathcal{V}^r} : \mathcal{V}^r \to (\mathcal{V}^r)'$, the approximate trial-to-test operator $T^r : \mathcal{U}_h \to \mathcal{V}^r$ is defined by:

$$T^r := \mathcal{R}_{\mathcal{V}^r}^{-1} \iota^T \mathcal{B}, \tag{2.27}$$

where $\iota : \mathcal{V}^r \to \mathcal{V}$ is the inclusion map. Consequently, the practical DPG method with optimal test functions solves

$$\begin{cases} u_h \in \mathcal{U}_h \subset \mathcal{U}, \\ b(u_h, v_h) = l(v_h), \ v_h \in \mathcal{V}^{r,\mathrm{opt}} = T^r \mathcal{U}_h, \end{cases} \tag{2.28}$$

with the additional interface term from (2.26) when breaking the test space $\mathcal{V}^r$.

The enriched test space in practice. In choosing the enriched test space, the goal is to approximate the semi-continuous problem (2.19), i.e., the residual ψ, as well as possible. The test space is typically enlarged by increasing the discretization order p of the trial space

uniformly to $p + \Delta p$, where $\Delta p \in \{1, 2, 3, \ldots\}$. Of course, the element-wise computational and memory complexity increases with larger Δp. Some alternatives for enriching the test space are discussed in [111]. Another approach is the so-called *double-adaptivity paradigm* where the test space is enlarged by local *hp*-adaptivity in a separate test mesh [25, 38]. The numerical results reported in this dissertation were obtained with uniform polynomial enrichment $\Delta p = 1$.

Fortin operator. Because the optimal test functions are approximated, some stability loss is inevitable. Several papers have addressed the issue of controlling and quantifying the stability loss in the DPG method [61, 20, 97]. The main tool for analyzing the stability of the practical DPG method is the Fortin operator. We state the main result of [61]: Assuming the existence of a linear and continuous operator $\Pi : \mathcal{V} \to \mathcal{V}^r$ such that for all $v \in \mathcal{V}$,

$$b(w_h, \Pi v - v) = 0, \ w_h \in \mathcal{U}_h, \tag{2.29}$$

then the variational problem (2.28) is well-posed and the following stability estimate holds:

$$\|u - u_h\|_{\mathcal{U}} \leq \frac{MC_\Pi}{\gamma_h} \inf_{\omega_h \in \mathcal{U}_h} \|u - \omega_h\|_{\mathcal{U}}, \tag{2.30}$$

where C_Π is the continuity constant of the Fortin operator Π.

2.4 Energy spaces of the exact sequence

The DPG method can be applied to any well-posed variational formulation. A conforming discretization of DPG formulations requires the conforming discretization of the *energy spaces of the exact sequence*. The standard energy spaces on a bounded domain $\Omega \subset \mathbb{R}^3$ with

Lipschitz boundary $\Gamma \equiv \partial\Omega$ are defined as follows:

$$\begin{aligned} L^2(\Omega) &:= \{y : \Omega \to \mathbb{F} : \|y\| < \infty\}, \\ H^1(\Omega) &:= \{w : \Omega \to \mathbb{F} : w \in L^2(\Omega), \nabla w \in (L^2(\Omega))^3\}, \\ H(\mathrm{curl}, \Omega) &:= \{\boldsymbol{q} : \Omega \to \mathbb{F}^3 : \boldsymbol{q} \in (L^2(\Omega))^3, \nabla \times \boldsymbol{q} \in (L^2(\Omega))^3\}, \\ H(\mathrm{div}, \Omega) &:= \{\boldsymbol{v} : \Omega \to \mathbb{F}^3 : \boldsymbol{v} \in (L^2(\Omega))^3, \nabla \cdot \boldsymbol{v} \in L^2(\Omega)\}, \end{aligned} \tag{2.31}$$

where $\|\cdot\|$ is the $L^2(\Omega)$ norm induced by the $L^2(\Omega)$ inner product $(\cdot, \cdot)$, $\mathbb{F} = \mathbb{R}$ (or $\mathbb{C}$), and the differential operators (grad, curl, div) are understood in the distributional sense. Assuming that Ω is homeomorphic with an open ball, the energy spaces form an exact sequence [27]:

$$H^1(\Omega) \xrightarrow{\nabla} H(\mathrm{curl}, \Omega) \xrightarrow{\nabla\times} H(\mathrm{div}, \Omega) \xrightarrow{\nabla\cdot} L^2(\Omega). \tag{2.32}$$

In the definition of the broken energy spaces, we follow the exposition in [20]. Suppose Ω is partitioned into a set Ω_h of open disjoint elements $\{K\}_{K\in\Omega_h}$ with Lipschitz element boundaries $\{\partial K\}_{K\in\Omega_h}$. The broken energy spaces defined on the finite element mesh Ω_h are:

$$\begin{aligned} L^2(\Omega_h) &:= \{y \in L^2(\Omega) : y|_K \in L^2(K)\ \forall K \in \Omega_h\} = L^2(\Omega), \\ H^1(\Omega_h) &:= \{w \in L^2(\Omega) : w|_K \in H^1(K)\ \forall K \in \Omega_h\} \supset H^1(\Omega), \\ H(\mathrm{curl}, \Omega_h) &:= \{\boldsymbol{q} \in (L^2(\Omega))^3 : \boldsymbol{q}|_K \in H(\mathrm{curl}, K)\ \forall K \in \Omega_h\} \supset H(\mathrm{curl}, \Omega), \\ H(\mathrm{div}, \Omega_h) &:= \{\boldsymbol{v} \in (L^2(\Omega))^3 : \boldsymbol{v}|_K \in H(\mathrm{div}, K)\ \forall K \in \Omega_h\} \supset H(\mathrm{div}, \Omega). \end{aligned} \tag{2.33}$$

Additionally, energy spaces are needed for the trace unknowns that arise from breaking the test space. These spaces are defined on the mesh skeleton Γ_h. On each element $K \in \Omega_h$,

the following (continuous and surjective) trace operators are defined:

$$\begin{aligned}
H^1(K) \ni w &\longmapsto \gamma^K w := w|_{\partial K} \equiv \hat{w} \in H^{1/2}(\partial K), \\
H(\mathrm{curl}, K) \ni \boldsymbol{q} &\longmapsto \gamma^K_{t,\perp} \boldsymbol{q} := (\hat{\boldsymbol{n}} \times \boldsymbol{q}) \times \hat{\boldsymbol{n}}|_{\partial K} \equiv \hat{\boldsymbol{q}}_{\boldsymbol{t},\perp} \in H^{-1/2}(\mathrm{curl}, \partial K), \\
H(\mathrm{curl}, K) \ni \boldsymbol{q} &\longmapsto \gamma^K_{t,\dashv} \boldsymbol{q} := \hat{\boldsymbol{n}} \times \boldsymbol{q}|_{\partial K} \equiv \hat{\boldsymbol{q}}_{\boldsymbol{t},\dashv} \in H^{-1/2}(\mathrm{div}, \partial K), \\
H(\mathrm{div}, K) \ni \boldsymbol{v} &\longmapsto \gamma^K_n \boldsymbol{v} := \boldsymbol{v}|_{\partial K} \cdot \hat{\boldsymbol{n}} \equiv \hat{v}_n \in H^{-1/2}(\partial K),
\end{aligned} \tag{2.34}$$

where $\hat{\boldsymbol{n}}$ is the outward unit normal on ∂K. Corresponding to the definition of the trace operators, we will refer to the unknowns in these trace spaces as continuous traces, tangential traces, rotated tangential traces, and normal traces, respectively.

The global trace operators in the broken spaces can now be defined element-wise:

$$\begin{aligned}
H^1(\Omega_h) \ni w &\longmapsto \gamma w := \prod_{K \in \Omega_h} \gamma^K(w|_K) \in \prod_{K \in \Omega_h} H^{1/2}(\partial K), \\
H(\mathrm{curl}, \Omega_h) \ni \boldsymbol{q} &\longmapsto \gamma_{t,\perp} \boldsymbol{q} := \prod_{K \in \Omega_h} \gamma^K_{t,\perp}(\boldsymbol{q}|_K) \in \prod_{K \in \Omega_h} H^{-1/2}(\mathrm{curl}, \partial K), \\
H(\mathrm{curl}, \Omega_h) \ni \boldsymbol{q} &\longmapsto \gamma_{t,\dashv} \boldsymbol{q} := \prod_{K \in \Omega_h} \gamma^K_{t,\dashv}(\boldsymbol{q}|_K) \in \prod_{K \in \Omega_h} H^{-1/2}(\mathrm{div}, \partial K), \\
H(\mathrm{div}, \Omega_h) \ni \boldsymbol{v} &\longmapsto \gamma_n \boldsymbol{v} := \prod_{K \in \Omega_h} \gamma^K_n(\boldsymbol{v}|_K) \in \prod_{K \in \Omega_h} H^{-1/2}(\partial K).
\end{aligned} \tag{2.35}$$

The trace spaces on the mesh skeleton are defined as element-wise traces of globally conforming functions, hence they are single-valued on element interfaces:

$$\begin{aligned}
H^{1/2}(\Gamma_h) &:= \gamma H^1(\Omega), \\
H^{-1/2}(\mathrm{curl}, \Gamma_h) &:= \gamma_{t,\perp} H(\mathrm{curl}, \Omega), \\
H^{-1/2}(\mathrm{div}, \Gamma_h) &:= \gamma_{t,\dashv} H(\mathrm{curl}, \Omega), \\
H^{-1/2}(\Gamma_h) &:= \gamma_n H(\mathrm{div}, \Omega).
\end{aligned} \tag{2.36}$$

These trace spaces are equipped with *minimum energy extension* norms:

$$\begin{aligned}
\|\hat{w}\|_{H^{1/2}(\Gamma_h)} &:= \inf_{\substack{w\in H^1(\Omega)\\ \gamma w=\hat{w}}} \|w\|_{H^1(\Omega)} \ , \\
\|\hat{\boldsymbol{q}}_{\boldsymbol{t},\perp}\|_{H^{-1/2}(\mathrm{curl},\Gamma_h)} &:= \inf_{\substack{\boldsymbol{q}\in H(\mathrm{curl},\Omega)\\ \gamma_{t,\perp}\boldsymbol{q}=\hat{\boldsymbol{q}}_{\boldsymbol{t},\perp}}} \|\boldsymbol{q}\|_{H(\mathrm{curl},\Omega)} \ , \\
\|\hat{\boldsymbol{q}}_{\boldsymbol{t},\dashv}\|_{H^{-1/2}(\mathrm{div},\Gamma_h)} &:= \inf_{\substack{\boldsymbol{q}\in H(\mathrm{curl},\Omega)\\ \gamma_{t,\dashv}\boldsymbol{q}=\hat{\boldsymbol{q}}_{\boldsymbol{t},\dashv}}} \|\boldsymbol{q}\|_{H(\mathrm{curl},\Omega)} \ , \\
\|\hat{v}_n\|_{H^{-1/2}(\Gamma_h)} &:= \inf_{\substack{\boldsymbol{v}\in H(\mathrm{div},\Omega)\\ \gamma_n\boldsymbol{v}=\hat{v}_n}} \|\boldsymbol{v}\|_{H(\mathrm{div},\Omega)} \ .
\end{aligned} \tag{2.37}$$

We can now define the mesh skeleton term $\langle\cdot,\cdot\rangle_{\Gamma_h}$ in the broken formulation (2.26). Suppose that trace unknown $\hat{\boldsymbol{E}} \in H^{-1/2}(\mathrm{curl},\Gamma_h)$ and broken test function $\boldsymbol{F} \in H(\mathrm{curl},\Omega_h)$. Then,

$$\langle \boldsymbol{n}\times\hat{\boldsymbol{E}}, \boldsymbol{F}\rangle_{\Gamma_h} := \sum_{K\in\Omega_h} \langle \hat{\boldsymbol{n}}\times\hat{\boldsymbol{E}}|_{\partial K}, \gamma_{t,\perp}^K(\boldsymbol{F}|_K)\rangle_{\partial K}, \tag{2.38}$$

where $\langle\cdot,\cdot\rangle_{\partial K}$ denotes the $H^{-1/2}(\mathrm{div},\partial K)\times H^{-1/2}(\mathrm{curl},\partial K)$ duality pairing on element boundary ∂K (see [20, Lem. 2.2]).[1]

Polynomial spaces. In the FE computation, we use polynomial subspaces of the energy spaces that satisfy the exact sequence property at the discrete level. That is, the polynomial spaces,

$$W^p \subset H^1(\Omega),\ Q^p \subset H(\mathrm{curl},\Omega),\ V^p \subset H(\mathrm{div},\Omega),\ Y^p \subset L^2(\Omega), \tag{2.39}$$

form an exact sequence analogous to (2.32). The polynomial spaces used in this dissertation are defined for standard elements of all shapes in [50]. The spaces correspond to Nédélec's sequence of the first type; for each element, W^p contains complete polynomials of order p, while Q^p, V^p, and Y^p contain complete polynomials of order $p-1$. We refer to the polynomial order p as the order of approximation.

[1] In practice, the arguments of the boundary operator $\langle\cdot,\cdot\rangle_{\partial K}$ are piecewise smooth functions so that the operator becomes a surface integral over ∂K.

2.5 The DPG method for Maxwell's equations

In this section, the DPG methodology is applied to the time-harmonic Maxwell equations. We introduce the ultraweak variational formulation that is used throughout this dissertation. For this, we consider the linear equations (1.16)–(1.17). The set of nonlinear equations used for modeling active gain and thermal coupling in fiber amplifiers is discussed in Chapter 5; however, most of the following discussion on variational formulations translates directly to the nonlinear problem. Error and residual convergence results for the DPG method are shown with different orders of approximation for a low-frequency manufactured solution. In the context of multimode fiber waveguides, we discuss the need for absorbing boundary layers in Section 2.5.2.

2.5.1 Ultraweak variational formulation

Because DPG is a Petrov–Galerkin method, it can solve any well-posed linear variational problem, with or without symmetric functional setting. Various variational formulations can be derived for the time-harmonic Maxwell equations (1.16)–(1.17). More precisely, we can formulate six different variational formulations, each one stated in a different energy setting. In [20], all six formulations are presented, and it is shown that well-posedness of one formulation implies the well-posedness of the other formulations. However, in the context of high-frequency wave propagation problems, the ultraweak DPG formulation has proven superior in the sense that it has better approximability properties than the other DPG formulations [127, 98, 102]. Moreover, the ultraweak formulation stands out because it is the only formulation for which the optimal test norm can be derived explicitly. In the following, we derive the ultraweak variational formulation in the globally conforming and in the broken setting, discuss the choice of the test norm, and show convergence results for a simple manufactured problem.

Let $\Omega \subset \mathbb{R}^3$ be bounded with Lipschitz boundary $\Gamma \equiv \partial\Omega$. Consider the time-harmonic

Maxwell problem in a linear, isotropic, homogeneous, and dielectric medium,

$$\begin{cases} \nabla \times \boldsymbol{E} + i\omega\mu \boldsymbol{H} = \boldsymbol{0} & \text{in } \Omega, \\ \nabla \times \boldsymbol{H} - i\omega\varepsilon \boldsymbol{E} = \boldsymbol{0} & \text{in } \Omega, \\ \boldsymbol{n} \times \boldsymbol{E} = \boldsymbol{n} \times \boldsymbol{E}_0 & \text{on } \Gamma_1, \\ \boldsymbol{n} \times \boldsymbol{H} = \boldsymbol{n} \times \boldsymbol{H}_0 & \text{on } \Gamma_2, \end{cases} \tag{2.40}$$

where $\Gamma_1 \cap \Gamma_2 = \emptyset,\ \bar{\Gamma}_1 \cup \bar{\Gamma}_2 = \Gamma$, with sufficiently regular boundary data.

The variational form can be stated as a second-order problem or as a first-order system. Depending on how the equations are relaxed (integrated by parts), six different variational formulations can be derived. All formulations are simultaneously well- or ill-posed [20]. The ultraweak formulation is the weak form of the first-order system with the most relaxed functional setting for the trial functions; it is obtained by testing (2.40) with test functions $(\boldsymbol{F}, \boldsymbol{G})$, integrating over Ω, and relaxing both equations:

$$\begin{cases} \boldsymbol{E}, \boldsymbol{H} \in (L^2(\Omega))^3, \\ (\boldsymbol{E}, \nabla \times \boldsymbol{F}) + (i\omega\mu \boldsymbol{H}, \boldsymbol{F}) = -\langle \boldsymbol{n} \times \boldsymbol{E}_0, \boldsymbol{F} \rangle_{\Gamma_1}, \quad \boldsymbol{F} \in \mathcal{V}_2, \\ (\boldsymbol{H}, \nabla \times \boldsymbol{G}) - (i\omega\varepsilon \boldsymbol{E}, \boldsymbol{G}) = -\langle \boldsymbol{n} \times \boldsymbol{H}_0, \boldsymbol{G} \rangle_{\Gamma_2}, \quad \boldsymbol{G} \in \mathcal{V}_1, \end{cases} \tag{2.41}$$

where $\mathcal{V}_i := \{\boldsymbol{q} \in H(\mathrm{curl}, \Omega) : \boldsymbol{n} \times \boldsymbol{q} = \boldsymbol{0} \text{ on } \Gamma_i\},\ i \in \{1, 2\}$. Notice that the test functions were chosen to satisfy homogeneous boundary conditions such that the boundary terms with unknown tangential traces vanish from the formulation.

Remark: Similar to (2.38), the boundary term $\langle \cdot, \cdot \rangle_\Gamma$ is defined as the $H^{-1/2}(\mathrm{div}, \Gamma) \times H^{-1/2}(\mathrm{curl}, \Gamma)$ duality pairing on boundary Γ. The prescribed boundary data is assumed to be in the corresponding energy space. However, on subset $\Gamma_i \subset \Gamma$ of the boundary, $H^{-1/2}(\mathrm{div}, \Gamma_i)$ and $H^{-1/2}(\mathrm{curl}, \Gamma_i)$ are not dual to each other. To make sense of the boundary term $\langle \cdot, \cdot \rangle_{\Gamma_i}$, additional regularity is assumed in this case. For example, $\boldsymbol{F} \in \{\boldsymbol{q} \in \mathcal{V}_2 : \boldsymbol{q}|_{\Gamma_1} \in L^2(\Gamma_1)\}$ and $\boldsymbol{G} \in \{\boldsymbol{q} \in \mathcal{V}_1 : \boldsymbol{q}|_{\Gamma_2} \in L^2(\Gamma_2)\}$. See [27] for further discussion of this issue.

Broken ultraweak formulation. Because of the relaxed setting in the ultraweak formulation, the trial unknowns—electric field $\boldsymbol{E}$ and magnetic field $\boldsymbol{H}$—have no global continuity requirements. Conforming discrete trial spaces can therefore be constructed locally (i.e., per element). However, when breaking test spaces, the additional trace unknowns do require global conformity over the mesh skeleton. The discontinuous (broken) test spaces are discretized by locally $H(\mathrm{curl})$-conforming test functions. Corresponding to the abstract formulation (2.26), we obtain the broken ultraweak Maxwell formulation:

$$\begin{cases} \boldsymbol{E}, \boldsymbol{H} \in (L^2(\Omega))^3, \ \hat{\boldsymbol{E}} \in \hat{\mathcal{U}}_1, \ \hat{\boldsymbol{H}} \in \hat{\mathcal{U}}_2, \\ (\boldsymbol{E}, \nabla_h \times \boldsymbol{F}) + \langle \boldsymbol{n} \times \hat{\boldsymbol{E}}, \boldsymbol{F} \rangle_{\Gamma_h} + (i\omega\mu \boldsymbol{H}, \boldsymbol{F}) = 0, \quad \boldsymbol{F} \in H(\mathrm{curl}, \Omega_h), \\ (\boldsymbol{H}, \nabla_h \times \boldsymbol{G}) + \langle \boldsymbol{n} \times \hat{\boldsymbol{H}}, \boldsymbol{G} \rangle_{\Gamma_h} - (i\omega\varepsilon \boldsymbol{E}, \boldsymbol{G}) = 0, \quad \boldsymbol{G} \in H(\mathrm{curl}, \Omega_h), \end{cases} \tag{2.42}$$

where

$$\begin{aligned} \hat{\mathcal{U}}_1 &:= \big\{ \hat{\boldsymbol{q}} \in H^{-1/2}(\mathrm{curl}, \Gamma_h) : \boldsymbol{n} \times \hat{\boldsymbol{q}} = \boldsymbol{n} \times \boldsymbol{E}_0 \text{ on } \Gamma_1 \big\}, \\ \hat{\mathcal{U}}_2 &:= \big\{ \hat{\boldsymbol{q}} \in H^{-1/2}(\mathrm{curl}, \Gamma_h) : \boldsymbol{n} \times \hat{\boldsymbol{q}} = \boldsymbol{n} \times \boldsymbol{H}_0 \text{ on } \Gamma_2 \big\}, \end{aligned} \tag{2.43}$$

and h denotes element-wise operations. The trace space $H^{-1/2}(\mathrm{curl}, \Gamma_h)$ and the duality pairing $\langle \cdot, \cdot \rangle_{\Gamma_h}$ on the mesh skeleton were defined in (2.36) and (2.38), respectively. The previous assumption for test functions to vanish on part of the boundary is no longer necessary; the corresponding terms can be computed as part of the duality pairing on the mesh skeleton involving the new unknowns—electric field trace $\hat{\boldsymbol{E}}$ and magnetic field trace $\hat{\boldsymbol{H}}$. In the DPG finite element computation, these unknowns are approximated as the traces of $H(\mathrm{curl})$-conforming elements.

Adjoint operator. The ultraweak formulation can also be stated more succinctly using the adjoint operator. First, we define the following group variables:

$$\mathfrak{u} := (\boldsymbol{E}, \boldsymbol{H}), \quad \mathfrak{v} := (\boldsymbol{F}, \boldsymbol{G}). \tag{2.44}$$

The Maxwell operator from (2.40) can be written as:

$$A\mathfrak{u} = (\nabla \times \boldsymbol{E} + i\omega\mu\boldsymbol{H},\ \nabla \times \boldsymbol{H} - i\omega\varepsilon\boldsymbol{E})\,, \tag{2.45}$$

where, using $\mathfrak{L}^2(\Omega) := (L^2(\Omega))^3 \times (L^2(\Omega))^3$, the domain of A is given by:

$$D(A) := \left\{ \mathfrak{u} \in \mathfrak{L}^2(\Omega) : A\mathfrak{u} \in \mathfrak{L}^2(\Omega),\ \begin{array}{ll} \boldsymbol{n} \times \boldsymbol{E} = \boldsymbol{n} \times \boldsymbol{E}_0 & \text{on } \Gamma_1, \\ \boldsymbol{n} \times \boldsymbol{H} = \boldsymbol{n} \times \boldsymbol{H}_0 & \text{on } \Gamma_2. \end{array} \right\}. \tag{2.46}$$

Then, the sesquilinear form corresponding to the ultraweak formulation (2.41) is

$$b(\mathfrak{u}, \mathfrak{v}) = (\mathfrak{u}, A^*\mathfrak{v}), \tag{2.47}$$

with the formal adjoint operator defined by:

$$A^*\mathfrak{v} = (\nabla \times \boldsymbol{F} + i\omega\varepsilon\boldsymbol{G},\ \nabla \times \boldsymbol{G} - i\omega\mu\boldsymbol{F})\,, \tag{2.48}$$

with domain $D(A^*) := \{\mathfrak{v} \in \mathfrak{L}^2(\Omega) : A^*\mathfrak{v} \in \mathfrak{L}^2(\Omega),\ \boldsymbol{n} \times \boldsymbol{F} = \boldsymbol{0} \text{ on } \Gamma_2,\ \boldsymbol{n} \times \boldsymbol{G} = \boldsymbol{0} \text{ on } \Gamma_1\}$.

In summary, we can state the ultraweak formulation as:

$$\begin{cases} \mathfrak{u} \in \mathfrak{L}^2(\Omega), \\ (\mathfrak{u}, A^*\mathfrak{v}) = l(\mathfrak{v}), \quad \mathfrak{v} \in D(A^*), \end{cases} \tag{2.49}$$

where $l(\mathfrak{v}) = -\langle \boldsymbol{n} \times \boldsymbol{E}_0, \boldsymbol{F} \rangle_{\Gamma_1} - \langle \boldsymbol{n} \times \boldsymbol{H}_0, \boldsymbol{G} \rangle_{\Gamma_2}$.

Test norm. As mentioned before, the choice of the test norm $\|\cdot\|_{\mathcal{V}}$ in the DPG method is important because it determines the norm in which the method converges. In the ideal DPG method with unbroken test spaces, the optimal test norm (2.24) can be derived explicitly for the ultraweak variational formulation [32]. In this case, the optimal test norm is the adjoint

norm: $\|\mathfrak{v}\|_{\mathcal{V}} = \|A^*\mathfrak{v}\|$. With this particular test norm,

$$\|\mathfrak{u}\| = \sup_{\mathfrak{v}\in\mathcal{V}} \frac{|b(\mathfrak{u},\mathfrak{v})|}{\|\mathfrak{v}\|_{\mathcal{V}}} = \|\mathfrak{u}\|_E. \tag{2.50}$$

Consequently, the method delivers the L^2 projection.

In the broken setting, however, this test norm can not be employed because it is not localizable. It can be augmented with an additional term to obtain a (quasi-optimal) test norm: $\|\mathfrak{v}\|^2_{\mathcal{V}(\Omega_h)} = \|A_h^*\mathfrak{v}\|^2 + \alpha\|\mathfrak{v}\|^2$, where the notation A_h^* indicates element-wise operations. We refer to this norm as the adjoint graph norm (with scaling parameter $\alpha \in \mathcal{O}(1)$). The optimal test norm and the adjoint graph norm are equivalent norms and, crucially, the equivalence constants are independent of frequency ω (assuming the bounded domain Ω is star-shaped with smooth boundary or convex) [88, 39, 20]. We revisit this point in the context of studying numerical pollution in Section 4.2. If not otherwise mentioned, we use $\alpha = 1$ in all numerical experiments.

In the same way that elimination of the boundary conditions in $D(A^*)$ (i.e., testing on the whole boundary) leads to the introduction of additional unknowns on the boundary Γ, the breaking of the test space leads to the introduction of additional unknowns on the entire mesh skeleton Γ_h. For the broken formulation, we define $\hat{\mathfrak{u}} = (\hat{\boldsymbol{E}}, \hat{\boldsymbol{H}})$ and the corresponding interface terms:

$$\langle \hat{\mathfrak{u}}, \mathfrak{v}\rangle_{\Gamma_h} := \langle \boldsymbol{n}\times\hat{\boldsymbol{E}}, \boldsymbol{F}\rangle_{\Gamma_h} + \langle \boldsymbol{n}\times\hat{\boldsymbol{H}}, \boldsymbol{G}\rangle_{\Gamma_h}. \tag{2.51}$$

The energy space for the trace $\hat{\mathfrak{u}}$ is defined element-wise:

$$\hat{\mathcal{U}} := \left\{\hat{\mathfrak{u}} : \exists \mathfrak{u} \in D(A) : \mathrm{tr}_{\Gamma_h}\mathfrak{u} = \hat{\mathfrak{u}}\right\}, \tag{2.52}$$

where the (tangential) trace operator tr_{Γ_h} is defined similar to (2.35). The trace space is

equipped with a minimum energy extension norm:

$$\|\hat{\mathfrak{u}}\|_{\hat{\mathcal{U}}} := \inf_{\substack{\mathfrak{u}\in D(A) \\ \mathrm{tr}_{\Gamma_h}\mathfrak{u}=\hat{\mathfrak{u}}}} \|\mathfrak{u}\|_Q, \tag{2.53}$$

where $\|\mathfrak{u}\|_Q^2 := \|A\mathfrak{u}\|^2 + \|\mathfrak{u}\|^2$. The broken ultraweak variational problem (2.42) can then be written as:

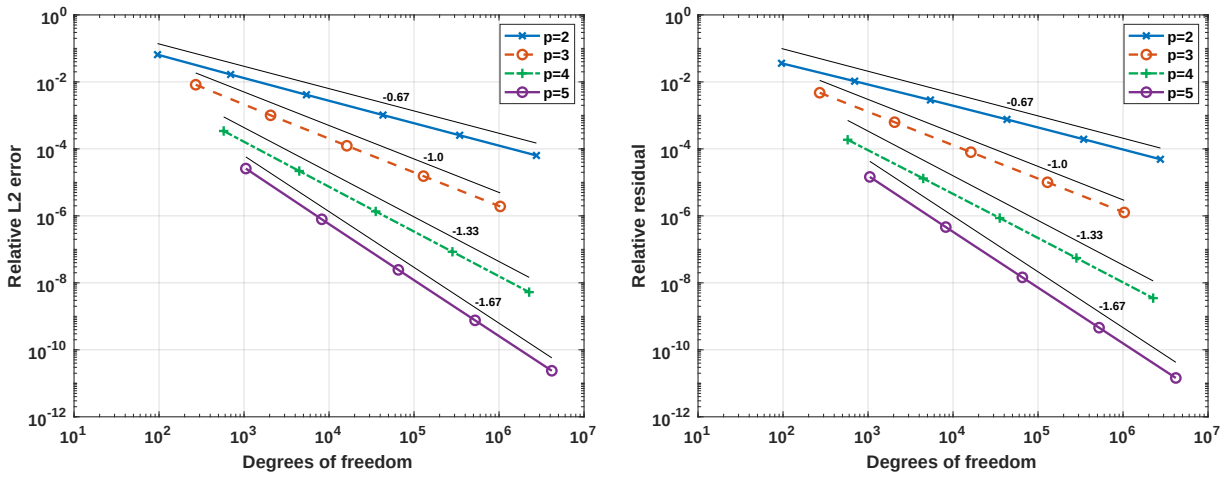
$$\begin{cases} \mathfrak{u} \in \mathfrak{L}^2(\Omega), \hat{\mathfrak{u}} \in \hat{\mathcal{U}}, \\ (\mathfrak{u}, A_h^*\mathfrak{v}) + \langle \hat{\mathfrak{u}}, \mathfrak{v}\rangle_{\Gamma_h} = l(\mathfrak{v}), \quad \mathfrak{v} \in \mathcal{V}(\Omega_h), \end{cases} \tag{2.54}$$

where $\mathcal{V}(\Omega_h) := \{\mathfrak{v} \in \mathfrak{L}^2(\Omega) : A_h^*\mathfrak{v} \in \mathfrak{L}^2(\Omega)\}$, and the load $l(\mathfrak{v}) = 0$ in this case.

Remark: Both the functional setting and the norm used for the traces derive from the choice of the norm for the broken test space. For a more thorough discussion of this point, we refer to [20, 27].

Convergence rates. To conclude this section, we show convergence results for the broken ultraweak Maxwell formulation (2.42) for a manufactured solution on a uniformly refined cube. Let $\Omega = (0,1)^3$, and let $(E_x(x,y,z), 0, 0)^T = (\sin(x)\sin(y)\sin(z), 0, 0)^T$ be the exact solution of the electric field. The boundary conditions, derived from the exact solution, are imposed on the tangential trace of the electric field, i.e., $\Gamma_1 = \Gamma$ in (2.42).

Figure 2.1 shows the convergence rates for the relative error of the electric and magnetic fields, measured in L^2, and for the relative residual, computed with (2.20) and measured in the (quasi-optimal) test norm ($\alpha = 1$). The rates were obtained by uniform h-refinements, using fixed order of approximation, $p \in \{2, 3, 4, 5\}$; consequently, according to the discrete exact sequence spaces (2.39), the fields $(\boldsymbol{E}, \boldsymbol{H})$ are approximated with linear, quadratic, cubic, and quartic polynomials, respectively. The obtained convergence rates are thus optimal for both the error and residual.



(a) Error (b) Residual

Figure 2.1: Error and residual convergence rates for a manufactured low-frequency solution on a uniformly refined cube, computed with the ultraweak DPG Maxwell formulation. We observe optimal convergence rates for any order of approximation.

2.5.2 Perfectly matched layer

The numerical simulation of waveguide problems such as the propagation of light in an optical fiber is performed in a truncated computational domain of finite length. At the end of the domain, usually where the light exits the optical fiber, it is important to avoid unrealistic reflections of the electromagnetic wave that may be caused by imposing (artificial) boundary conditions. Absorbing boundary conditions are designed to eliminate or at least mitigate this issue. In the context of multimode fiber simulations (Chapter 4) or the nonlinear gain problem (Chapter 5), it is not sufficient to impose impedance boundary conditions through an impedance relation between the electric and magnetic field since an exact impedance relation only works for a single propagating mode. In these numerical simulations, we employ a *perfectly matched layer* (PML) to absorb the propagating wave at the fiber end. PMLs, first introduced in [14], are a popular tool in the finite element simulation of wave propagation phenomena in unbounded domains (see [24, 17, 91, 6] and references therein); in recent work, stretched coordinate PMLs have been extended to the DPG methodology [6, 98, 102]. Details of our PML implementation are given in Appendix B.3.

Chapter 3

*hp*3D Finite Element Software

*hp*3D is a three-dimensional finite element software developed by Demkowicz et al. [26, 37]. The current (fifth) version of the code is implemented in Fortran 90 and supports the solution of coupled problems and discretization of the H^1-, $H(\text{curl})$-, $H(\text{div})$-, and L^2-conforming finite elements necessary for complex multiphysics models. Moreover, *hp*3D supports the use of hybrid meshes using elements of "all shapes": tetrahedra, hexahedra, prisms, and pyramids. The use of orientation-embedded shape functions [50] and projection-based interpolation [28] enables efficient computation of hanging (constrained) nodes and anisotropic *hp*-refinements. For a fundamental discussion of the underlying finite element code design in *hp*3D, we refer to [37, 57, 83, 50].

*The main contribution of this work to the hp*3D *code is the development of a scalable MPI/OpenMP parallel version for distributed hp-adaptive finite element computation of complex multiphysics problems. In particular, we are interested in large-scale simulations of the fiber amplifier model introduced in Chapter 5 to study nonlinear gain in high-power fiber laser amplifiers.*

This work's contributions to *hp*3D also include the implementation of DPG interface variables (trace unknowns) and efficient static condensation for local element matrices, both of which are discussed in this chapter (see Sections 3.4.2 and 3.4.3, respectively). In the context of DPG methods, recent work showed the importance of sum factorization for the

integration of tensor product shape functions to accelerate element computation [92, 11]. This fast integration technique is briefly described and its computational performance shown in the ultraweak Maxwell setting in Section 3.4.1.

We begin with a description of the parallel mesh distribution, where each MPI process is assigned a distinct subdomain and stores the corresponding degrees of freedom (DOFs). This approach builds on the lean data structures of the sequential *hp*3D code, discussed in Section 3.1. To perform static partitioning and dynamic repartitioning, Zoltan [41] has been integrated with *hp*3D, see Section 3.2.2. Load balancing is essential to maintain an even computational workload and memory usage between compute nodes, particularly for successive *hp*-adaptive mesh refinements. Some numerical examples for this are discussed in the context of multimode propagation in linear waveguides in Section 4.3. OpenMP support is enabled in *hp*3D to exploit intra-node shared memory parallelism, which provides significant speedup for single-node computation on modern manycore architectures [102]. Besides parallelizing computation of the finite element stiffness and load assembly, *hp*3D interfaces with parallel sparse direct solvers (e.g., MUMPS [4]) and iterative solver packages (e.g., PETSc [12]). We have implemented a custom parallel nested dissection solver, as well as a custom repartitioner for load balancing purposes, to achieve excellent scaling for certain applications (e.g., waveguide problems). We discuss its implementation in Section 3.3 and present scaling results in Section 6.1. Recent work on DPG multigrid solvers [103, 102] provides a starting point for the development of a parallel iterative solver with geometric multigrid preconditioning in a future project. Additionally, a host of MPI related infrastructure has been implemented in *hp*3D as part of this work, including parallel I/O and visualization.

3.1 Data structures and *hp*-adaptivity

Although the concept of *hp*-refinements is relatively intuitive, its realization in finite element codes is challenging. In particular, *h*-refinements require flexible mesh data structures and

advanced processing routines to enforce conformity of basis functions on irregular meshes. The idea of automatic *hp*-adaptivity where both element size h and polynomial order p are determined based on a-posteriori error estimation goes back to a series of papers published by Demkowicz et al. in the late eighties [35, 100, 105]. The need for such capability was driven by the aim for exponential convergence, i.e., an exponential decrease in the error in terms of the number of degrees of freedom, for difficult problems with singularities or boundary layers. One principle idea that is anchored in the *hp*3D data structure design to support these refinements is the "growing" of trees for all topological entities in the mesh—edges, faces, and element interiors—instead of growing element trees only.

The backbone of the *hp*3D code are two data structures: `ELEMS` and `NODES`. The array `ELEMS` stores information on the elements in the initial mesh, and it is statically allocated in the beginning of the program run. Each initial mesh element (object) consists of attributes typical for a classical FE code for unstructured meshes: element nodes, neighbors, etc. The initial element mesh is built from a regular mesh (containing no hanging nodes) that may range from a few elements to several thousand. This is done via a Geometric Modeling Package (GMP), described in [57], that supports the use of both isoparametric and exact geometry elements; interfaces to established unstructured mesh generators (e.g., CUBIT [16], NETGEN [112]) are also provided.

The other essential data structure is the `NODES` array, consisting of abstract *nodes*: vertices, edges, faces, and element interiors. We refer to these node types as *vertex node, edge node, face node,* and *middle node,* respectively. The `NODES` array keeps information on node attributes such as the type of the node, polynomial order, nodal tree structure (father and sons), etc., as well as the degrees of freedom. It also stores information about the type of physics variables that are supported on each node and the corresponding boundary or interface conditions. The p-refinement of an element translates into updating the information for its nodes (new DOFs), while the h-refinement translates into "breaking" nodes. For example, an edge breaks into two edge- and one vertex-son. New nodes, i.e., entries in

the NODES array, are created, and the corresponding information on nodal trees is updated. Anisotropic refinements can therefore result in various different ways to break a node.[1] We emphasize that the NODES data structure represents a tree but is stored as an array. That is, the nodal tree can be reconstructed on-the-fly but instead of storing pointers, we use unique node IDs that correspond to the location of a node in the NODES array. This serves primarily for fast lookup of nodal information. Newly created nodes are always appended at the end of the array. Figure 3.1 illustrates how nodal trees are grown for a simple 2D example.

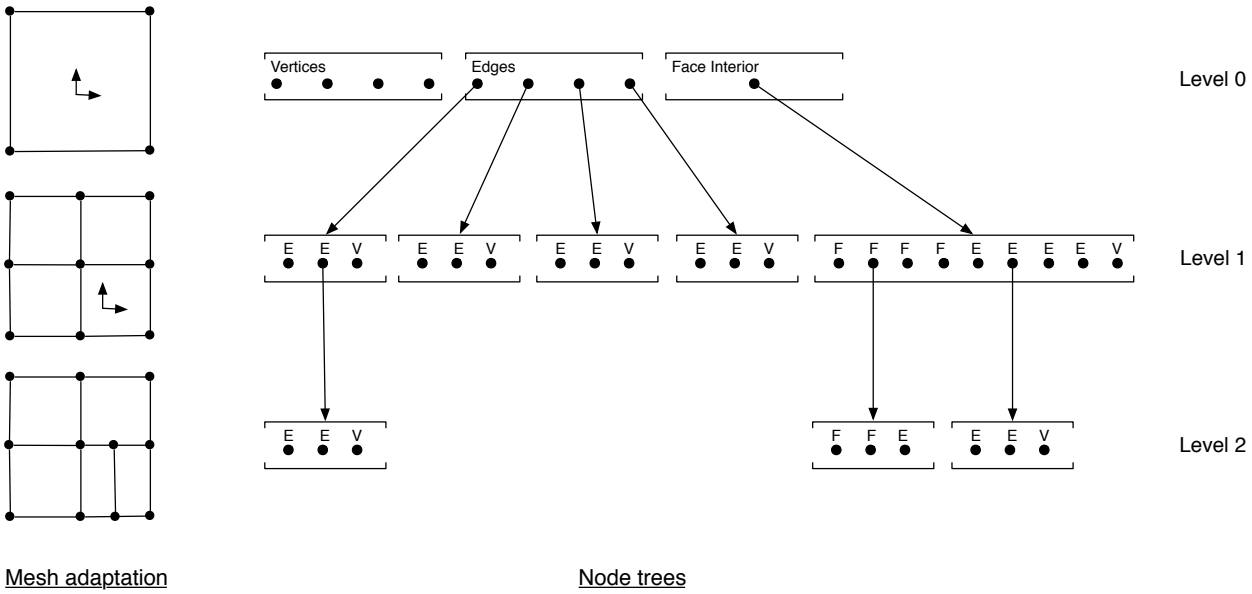


Figure 3.1: Nodal trees in *hp*3D [83, Fig. 3.6]. The NODES data structure supports "growing" trees for all topological entities in the mesh: edges, faces, and element interiors.

3.2 MPI/OpenMP parallelization

The *hp*3D finite element software is primarily designed for solving complex multiphysics applications that require mesh adaptivity, including anisotropic adaptivity (e.g., to resolve boundary layers) and locally adaptive order of approximation (p-adaptivity), discretization of physics variables in different energy spaces (possibly supported in only a part of the domain),

[1] *hp*3D ensures that the new mesh is *1-irregular:* that is, parent nodes of constrained (hanging) nodes must be unconstrained [37]. The process of enforcing this rule is called "closing" the mesh and may lead to additional refinements but makes the implementation of constrained approximation more efficient.

or hybrid meshes with elements of different shapes. If none of these advanced finite element features are required for an application, for example to compute the classical variational Poisson problem with a Galerkin discretization in a low-order uniform mesh (or a-priori defined mesh), then simpler codes can be used that will most likely perform faster. In recent years, DPG methods have become a focus for applications computed with *hp*3D because they naturally fit into the *hp*-adaptive framework. In this setting, computations usually start with a coarse mesh and guide automatic *hp*-adaptivity with a reliable a-posteriori error indicator to successively build a hierarchy of adaptively refined meshes. Using mixed isotropic and anisotropic *h*-adaptivity in hybrid meshes requires additional code complexity (e.g., to maintain 1-irregular meshes). The *hp*3D code hides this complexity from the user; the same approach was taken for the MPI/OpenMP parallelization that mostly concerns the data structures hidden from the user application. The use of MPI or OpenMP is optional; each one can be switched on independently. The changes needed to accommodate the parallel environment in the user application are minimal. In particular, all of the data structures used in the *hp*3D library support both distributed as well as single-node computation. This is a distinction from other FE libraries that define specific parallel datatypes that require the user to adapt the application code to the computing environment.

Compute architectures. The parallel code is designed to run efficiently on small machines (e.g., a single workstation or laptop), as well as large-scale computing facilities (e.g., Stampede2 at TACC). Due to the quickly changing high-performance computing landscape, the meaning of "large-scale" computation has evolved constantly; nowadays, large-scale computation could be understood in terms of a large number of degrees of freedom (e.g., several billion DOFs), big storage requirements (e.g., hundreds of Terabytes of main memory), many compute cores (e.g., thousands of CPU cores, or millions of GPU cores), or various other measures. Additionally, supercomputing platforms have become more heterogeneous and traditional performance measures (e.g., FLOPS) are only one metric of interest. In *hp*3D,

we have targeted CPU-based manycore architectures which require less specialized code than GPU-based systems. The Intel Xeon "Skylake" (SKX) compute nodes of TACC's Stampede2 cluster are state-of-the-art manycore chips that were used for development, tests and productive runs with *hp*3D. Each compute node consists of a two-socket NUMA architecture.[2] Per socket, 24 cores are available, and each compute node has a total of 192 GB main memory. In a hybrid MPI/OpenMP setting, one or a few MPI processes per compute node are typically used so that shared-memory parallelism is exploited via OpenMP threading. The number of OpenMP threads per MPI process is selected in such a way that the total number of threads per compute node equals the total number of available compute cores (e.g., 48 on TACC's Stampede2 SKX nodes). The parallel *hp*3D code was used to conduct numerical experiments with up to 512 compute nodes (24 576 cores) and two billion DOFs. Scaling results are discussed in Section 6.1.

3.2.1 Mesh distribution

Key to the parallel code efficiency is an efficient design of the parallel data structures, i.e., one that supports a distributed finite element mesh, partitioned into subdomains, that requires only a minimum amount of communication between MPI processes. We are leveraging the existing data structures for that purpose. Mesh distribution in the *hp*-adaptive setting requires dynamic load balancing; therefore, a certain amount of communication is unavoidable. In this section, we describe in detail our approach to implementing the distributed data structures.

Partitioning. The mesh partitioning follows a hybrid MPI/OpenMP parallelization approach where mesh data is distributed to MPI processes and each MPI process uses OpenMP threading to parallelize its computational workload. We distinguish between two stages of

[2]Non-uniform memory access (NUMA) architectures provide local memory for each processor which can be accessed faster than the non-local memory. In other words, memory access times depend on the memory location.

the mesh distribution process: the logical partitioning that determines the distribution of the mesh, and the actual distribution of the data structures that represent the distributed mesh. Logically, the finite element mesh is partitioned into N (non-overlapping) subdomains, where N is the total number of MPI processes. Each subdomain is "owned" by one distinct MPI process, and the subdomain ID is equal to the corresponding MPI rank, numbered from 0 to $N-1$. Neighboring subdomains interface through faces, edges, and vertices at the subdomain boundary. The logical mesh partitioning into subdomains is accomplished by setting a subdomain flag, called `subd`, for all (unbroken) middle nodes of the (active) mesh. Middle nodes correspond to the element interiors and can therefore be identified uniquely with an element. Within each subdomain, MPI processes use OpenMP threading to compute tasks in parallel. Usually, within the subdomain, one thread is performing the workload for one distinct element at a time. Due to p-adaptivity and other factors that contribute to varying element workload, dynamic thread scheduling typically performs best.

Data distribution. The initial mesh elements (depending on the specific application and geometry), stored in `ELEMS`, typically account for a negligible portion of the overall storage requirements. The bulk of the main memory is occupied by the dynamically growing `NODES` array and by the internal data structures of the direct or iterative linear solver. Therefore, each MPI process initially receives one copy of the `ELEMS` array that it retains throughout the program run. This is done only once at the beginning and may thus be considered part of the startup cost of the parallel *hp*3D solver. The redundancy in storage requirements is compensated for by the communication-avoiding access to the `ELEMS` array throughout the rest of the program run. We considered two different approaches for distributing the `NODES` data structure. We will argue why we favor one over the other.

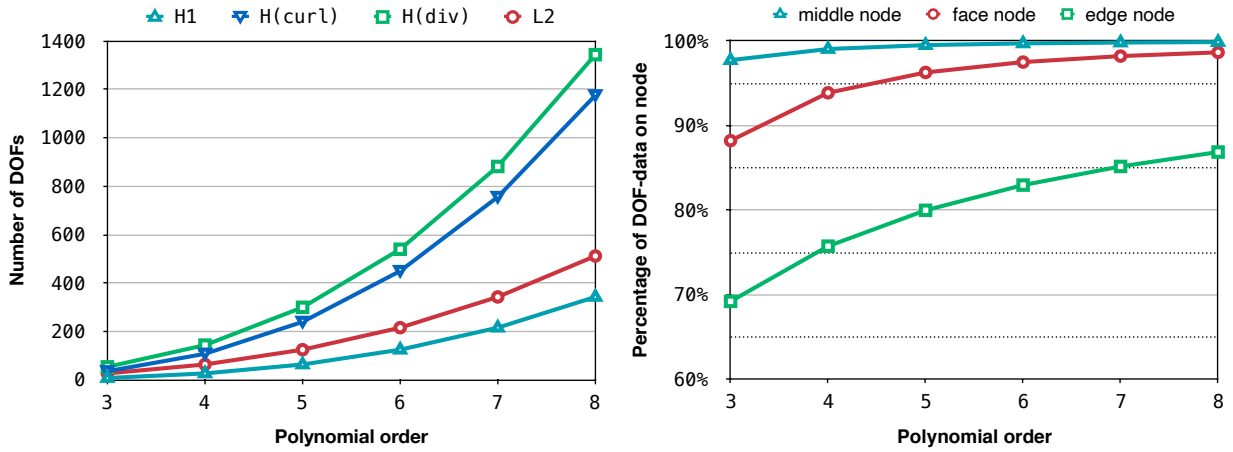
First approach: distributing nodes. The first approach immediately suggests itself: since we are looking to decompose the domain primarily based on geometrical information

and workload (dynamic load balancing), we may use the nodes' attributes and spatial coordinates to distribute the `NODES` array accordingly. Every node stores geometry data which can easily be accessed. This approach avoids storing redundant `NODES` data except where absolutely necessary (at subdomain interfaces). The issue with this approach lies in the fact that there is no data locality with respect to nodal connectivities in the `NODES` array due to adaptive mesh refinements. As a mesh refinement is executed, new nodal sons of an edge, face, or element interior are appended at the end of the `NODES` array (cf. Figure 3.1). Continuous reordering of the `NODES` data structure to guarantee data locality would be very costly with regard to memory operations; additionally, as the location of each node in the `NODES` array is also used as its unique ID for fast access, nodal connectivity information would have to be rewritten in any reordering. Therefore, forcing data locality is not a feasible (scalable) option in the current setting. If data locality is not possible, then the `NODES` array would have to be distributed in small chunks, and connecting location of a node to its ID would involve complex calculations and additional information about offset and size of each chunk to be kept by each MPI process. Every mesh refinement and repartitioning would require this information to be updated accordingly.

Second approach: distributing degrees of freedom. Given these limitations, we have implemented an alternative decomposition of the `NODES` array. We believe that storing a certain amount of redundant data reduces the complexity and provides better scalability by avoiding communication between MPI processes. The idea is to distinguish node attributes (type, order, father and sons, ...) from local data (DOFs). The node attributes are essential in coordinating across MPI processes (e.g., in mesh refinement) but the majority of the data associated with a node are typically DOFs.

To illustrate that, we need to consider a specific application because the size of the local data on a node depends on many factors: type of the node, polynomial order of the shape functions, number of physics variables and their respective number of components (e.g,

vector-valued or scalar-valued), whether they are complex- or real-valued, and the energy space $(H^1, H(\text{curl}), H(\text{div}), L^2)$. On the other hand, the amount of data allocated for node attributes is fixed. Through successive optimizations, the non-DOF data has been reduced to only 64 bytes per node.



(a) Interior solution DOFs per variable (b) DOF data per node in ultraweak Maxwell

Figure 3.2: Degrees of freedom for a hexahedral element, depending on the polynomial order. For high-order finite elements, DOFs account for the majority of data in the `NODES` array.

To make this notion more precise, consider the ultraweak DPG Maxwell formulation (see (2.42)), where the degrees of freedom describe a time-harmonic electromagnetic field. That is, DOFs define complex-valued vector fields with L^2 components and $H(\text{curl})$ interface variables for both the electric and magnetic field. This adds up to a total of six L^2 variables (interior solution DOFs) and two $H(\text{curl})$ variables (interface solution DOFs). The middle node of a fifth-order hexahedral element therefore stores 750 interior solution DOFs (cf. Figure 3.2a). Using complex double precision, its solution DOFs occupy a total of 12 kB. Additionally, geometry DOFs must be stored for each node (three H^1 variables). With $p = 5$, DOF data occupies about 99.5 percent of middle node storage (cf. Figure 3.2b).

Based on these observations, we let each MPI process keep *global information about node attributes* while storing only *local DOFs associated with its subdomain*. To provide an estimate for the scalability of this approach, we consider the ultraweak Maxwell problem.

Scalability of the approach. Each of TACC's Stampede2 SKX compute nodes has two sockets and a total of 192 GB main memory. Assuming one MPI process per socket—a typical hybrid MPI/OpenMP configuration—96 GB memory are available per subdomain. Further assuming that it is reasonable to allocate at most ten percent of local memory with global data structures, the storage for the `NODES` array should not surpass 10 GB.

To give a storage estimate for the ultraweak Maxwell problem, we consider a uniform fifth-order hexahedral mesh. In this mesh, the (average) number of nodes per hexa element is eight (one vertex node, three edge nodes, three face nodes, one middle node).[3] Since each node allocates 64 bytes for its attributes, a hexa element needs 512 bytes on average. Thus, as shown in Table 3.1, nodal attributes for a total of 10 000 000 hexahedral elements can be stored with ca. 5 GB memory. For the ultraweak Maxwell problem, each hexahedral element has (on average) 1 020 solution DOFs: 750 L^2 DOFs and 270 $H(\mathrm{curl})$ trace DOFs. Therefore, the total number of solution DOFs for a uniform mesh with 10 000 000 hexa elements is approximately 10 200 000 000 (cf. Table 3.1).

Mesh data		Number of solution DOFs	
Hexahedral elements	`NODES` (MB)	L^2 fields	$H(\mathrm{curl})$ traces
1 000	0.5	750 000	270 000
1 000 000	512	750 000 000	270 000 000
10 000 000	5120	7 500 000 000	2 700 000 000

Table 3.1: Nodal data and ultraweak Maxwell solution DOFs in a uniformly refined fifth-order hexahedral mesh. To store 10 000 000 hexahedral elements, global information in the `NODES` array accounts for ca. 5 GB main memory. On this mesh, the total number of solution degrees of freedom is 10 200 000 000.

The above storage estimate constitutes a practical limit for the scalability of the finite element code. However, at this point we believe solving systems with up to several billion

[3] A hexahedral element consists of 27 nodes (8 vertices, 12 edges, 6 faces, and 1 element interior), but most of these nodes are shared with neighboring elements. Considering hexa elements in the domain interior (away from the boundary) of a uniformly refined mesh, there are precisely 3 faces, 3 edges, 1 vertex, and 1 element interior, per hexa element.

degrees of freedom is providing sufficient scalability to justify the described approach.

Refinements. Parallel mesh refinement in the *hp*-adaptive setting poses a challenging task: mesh reconciliation. The main ideas behind mesh reconciliation in the context of *hp*3D are described in [37]. First, to make decisions on *h*- and *p*-refinements, automatic *hp*-adaptivity requires global communication to exchange element-local error indicators between processes. Then, assuming each MPI process computes the mesh refinements only within its own subdomain and updates its `NODES` array accordingly, different MPI processes would obtain a different version of the `NODES` array (i.e., mesh), hence the need for mesh reconciliation. This would require additional global communication and a complex reconciliation procedure [37] that is counter to our approach of storing global node attributes for simplicity and efficiency.

Storing the entire `NODES` array (except DOFs) on every compute node opens up a different possibility to perform the mesh refinements. After the error indicators have been collected by every MPI process, each one could perform the (adaptive) mesh refinement on the `NODES` array globally. This poses the scalability question again. The mesh refinement can be seen as a two-step procedure: first, it is decided if and how elements are to be refined, and for every *h*-refinement new nodal sons are created—including allocating space for DOFs; the second step involves updating the respective geometry DOFs. While the first step is essential for the structure of the `NODES` array and building consistent nodal trees, the second one involves only local computations (except for hanging nodes on the subdomain interface). Moreover, the first step is computationally cheap while the second one may be expensive, especially for isoparametric elements with higher *p*. Therefore, the first step of the mesh refinement is performed globally on each processor (though DOFs are only allocated for nodal sons within the subdomain) but the second step is parallelized based on the mesh distribution. This approach incurs some amount of redundant work but greatly simplifies the refinement procedure by avoiding complex mesh reconciliation and the associated global communication.

3.2.2 Dynamic load balancing

High-order finite element methods are typically computationally intensive and are thus good candidates for distributed large-scale computations which are often memory-bandwidth bound otherwise. In the context of adaptive mesh refinements, dynamic load balancing is essential to provide a balanced workload. This is even more important in a hybrid MPI/OpenMP implementation for manycore architectures, as one idle MPI process may cause many compute cores to be idle. A sufficient workload must be maintained within each subdomain so that OpenMP can be exploited.

In the first stage of load balancing, *hp*3D interfaces with Zoltan to determine a mesh partition. Zoltan is a dynamic load balancing library maintained by Sandia National Laboratories [41]. The library supports many load balancing algorithms that can be selected by changing a single parameter in the call to Zoltan's load balancing routine. The geometric partitioners include algorithms based on space filling curves as well as recursive bisection; the latter category is applicable to waveguide problems, particularly when used with nested dissection solvers. The geometric partitioners require spatial coordinates of each element and a weight representing the respective computational workload, both of which are available or can be computed based on the data in the `NODES` array. Zoltan also provides graph-based partitioners where elements serve as the graph's vertices, and faces between elements serve as edges in the graph. The corresponding weights are simply the number of interior DOFs and local interface DOFs, respectively. In Section 4.3.2, different load balancing strategies for waveguide problems are discussed in more detail.

The second step of the load balancing is data migration. Once the decision about optimal load balance has been made, i.e., new subdomains have been defined, the mesh has to be redistributed. In our approach, this means both logical redistribution of the mesh, as well as allocating DOFs within the subdomain and deallocating them otherwise. It may also entail exchanging DOFs between MPI processes when element ownership has changed and

the current solution DOFs are required for the next solution process (e.g., in time-stepping methods). Geometry DOFs are always recomputed.

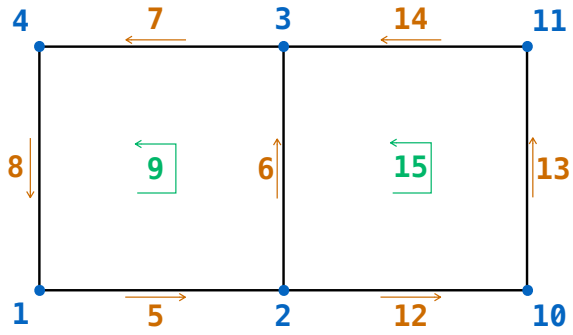
3.2.3 Assembly

In this section, we discuss the implementation of the distributed finite element matrix and load assembly. In particular, we show how local-to-global DOF maps are computed in parallel based on the mesh distribution. This process serves to assemble a distributed sparse global matrix, usually in COO-format, which can either be directly used by a linear solver (direct or iterative) or converted to another sparse matrix format if necessary. Matrix-free methods have not yet been implemented in the distributed code.

While the mesh distribution is based on partitioning elements, the assembly routine works with *degrees of freedom that are associated with nodes*. This is an important distinction, because global DOF maps can be directly computed using the `NODES` array rather than first building global element-wise information. In essence, an MPI process can avoid constructing local node lists and corresponding constraints for elements outside of its subdomain that share nodes with elements within. In both the sequential and distributed *hp*3D code, the assembly process is done in two steps: 1) gathering of assembly information, and 2) distributed stiffness and load assembly, including element integration. The sequential assembly that is performed when computing without MPI parallelism executes a subset of the distributed assembly tasks corresponding to the assembly within a single subdomain. While most of the computational workload is in the conveniently parallel element integration, it has proved important to optimize the first step (gathering assembly information) for parallel scalability.

Gathering of assembly information. To describe the assembly algorithm, we are breaking down the implementation of this part into four distinct steps (A)–(D), each illustrated by means of a simple 2D example. Consider a mesh (or a part of the mesh) with two quadri-

lateral elements ("quads"), as shown in Figure 3.3, each belonging to one distinct subdomain (or MPI process). Each quad consists of four vertex nodes, four edge nodes, and one face (interior) node. One edge node and two vertex nodes are shared between the two elements, i.e., these nodes are at the subdomain interface. The displayed nodal information (`node`, `type`, and `subd`) is accessible to both MPI processes through the `NODES` array. The `subd` values correspond to the partitioning (i.e., the left element is owned by the processor with rank 0, and the right element is owned by the processor with rank 1). Suppose we are interested in gathering assembly information for the discretization of one H^1 variable with third-order polynomial spaces (e.g., classical variational Poisson formulation with Galerkin discretization). Then, there are one, two, and four solution DOFs per vertex, edge, and face node, respectively.




```
node = [ 1, 2, 3, 4, 5, 6, 7, 8, 9,10,11,12,13,14,15]
type = [ v, v, v, v, e, e, e, e, f, v, v, e, e, e, f]
subd = [ -, -, -, -, -, -, -, -, 0, -, -, -, -, -, 1]
```

Figure 3.3: Distributed assembly illustrated by a simple 2D example. Some nodal information is accessible across all processors.

A. Node ownership: To compute local-to-global DOF maps in parallel, we introduce the concept of *node ownership*. Recall that during the partitioning of the mesh, element ownership was defined through the assignment of an element (resp. middle node) to a specific subdomain. Unlike middle nodes that are always in the interior of a subdomain, vertex,

edge, and face nodes cannot be identified with a distinct subdomain when they are at the subdomain interface. Each of these node types could be shared with elements of multiple other subdomains. In order to compute unique global indices for DOFs associated with such nodes, we determine one subdomain as the distinct owner of the node in each case. First, every MPI process marks nodes that are needed by elements within their subdomain with their own rank; then, through a global reduction with the "min" operator, every node is uniquely assigned to an owner—the MPI process with the lowest rank which had previously marked the node. This owner will now be responsible for determining the correct mapping for DOFs associated with the node.

In our simple 2D example, middle nodes are face (interior) nodes. Face node 9 is owned by `[0]` (the MPI process with rank 0), and face node 15 is owned by `[1]` (the MPI process with rank 1). Both MPI processes create a node ownership array, called `NOD_OWN`, where they mark the nodes that are connected to their respective element. Other entries are initialized with N, the total number of processes (here, $N = 2$). As illustrated by Figure 3.4, both processes, `[0]` and `[1]`, have marked the nodes 2, 3, and 6, which are at the subdomain boundary. Through a reduction operation, `[0]` is determined as the owner of these nodes.

```
[0]  NOD_OWN = [ 0, 0, 0, 0, 0, 0, 0, 0, 0, N, N, N, N, N, N]
[1]  NOD_OWN = [ N, 1, 1, N, N, 1, N, N, N, 1, 1, 1, 1, 1, 1]
-> allreduce(min): NOD_OWN
     NOD_OWN = [ 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1]
```

Figure 3.4: Distributed assembly. Processes communicate to determine *node ownership*.

B. Local offsets: Once node ownership has been determined, MPI processes compute local node offsets for all of their owned nodes. The offsets are used to map the DOFs of a node to the global matrices and vectors. This step of the process corresponds to the gathering of assembly information in the sequential code, whereas steps (A), (C), and (D) are additional communication needed in the distributed version. Figure 3.5 shows the accumulated information in the 2D example: node DOFs (`NOD_DOF`), offsets (`NOD_OFF`), and a

total DOF count per subdomain (SUBD_DOF).

```
[0]  NOD_DOF = [ 1, 1, 1, 1, 2, 2, 2, 2, 4, 0, 0, 0, 0, 0, 0]
[0]  NOD_OFF = [ 0, 1, 2, 3, 4, 6, 8,10,12, 0, 0, 0, 0, 0, 0]
[0] SUBD_DOF = [16, 0]
[1]  NOD_DOF = [ 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 2, 2, 2, 4]
[1]  NOD_OFF = [ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 2, 4, 6, 8]
[1] SUBD_DOF = [ 0,12]
```

Figure 3.5: Distributed assembly. Processes determine local node offsets per subdomain.

C. Global offsets: To compute global offsets, MPI processes only need to exchange the number of DOFs for owned nodes per subdomain (in SUBD_DOF) and calculate their respective subdomain offset via a prefix sum (in SUBD_OFF); consequently, the global ordering of DOFs is determined first by subdomain, then by elements within the subdomain. This a-priori defined ordering is advantageous for data locality and matches the expected partitioning of sparse matrices into successive blocks of rows typically used by parallel linear solvers (e.g., PETSc). At the end of this step, processes have computed global DOF offsets (in NOD_OFF) for all nodes owned by their subdomain, as illustrated by Figure 3.6.

```
-> allreduce(max): NOD_DOF, SUBD_DOF
     NOD_DOF = [ 1, 1, 1, 1, 2, 2, 2, 2, 4, 1, 1, 2, 2, 2, 4]
    SUBD_DOF = [16, 12]
    SUBD_OFF = [ 0, 16]
[0]  NOD_OFF = [ 0, 1, 2, 3, 4, 6, 8,10,12, 0, 0, 0, 0, 0, 0]
[1]  NOD_OFF = [ 0, 0, 0, 0, 0, 0, 0, 0, 0,16,17,18,20,22,24]
```

Figure 3.6: Distributed assembly. Processes communicate to determine global offsets.

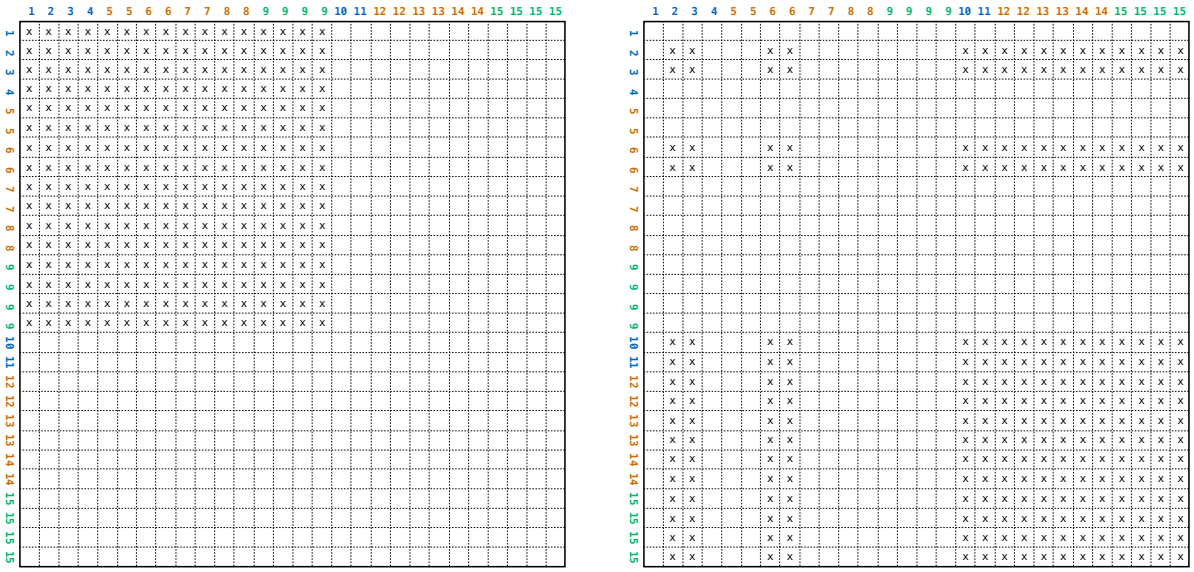
D. Exchanging information: In the last step, processes must exchange offset information to receive DOF offsets for nodes at their subdomain boundary that are owned by another process. Figure 3.7 illustrates the final DOF mapping with global offsets for every node.

```
-> allreduce(max): NOD_OFF
     NOD_OFF = [ 0, 1, 2, 3, 4, 6, 8,10,12,16,17,18,20,22,24]
```

Figure 3.7: Distributed assembly. Processes communicate offsets for owned nodes.

All in all, the assembly information is gathered with the help of a few global reductions over integer arrays indicating node ownership and offsets. Apart from these reduction operations, the computation is local to subdomains and thus can be easily parallelized.

Element integration, and distributed stiffness and load assembly. Since offsets have previously been computed, the elements in each subdomain can be processed independently and in parallel by OpenMP threads. From the assembly perspective, it is at this point not relevant whether the element matrices are computed for a DPG formulation or a standard Galerkin discretization. The user sets flags to indicate the properties of the matrix (e.g., symmetry, definiteness). Dirichlet boundary conditions and constraints from hanging nodes are incorporated during the construction of the element matrices (these features are hidden from the user application). In this part of the code, the distributed *hp*3D assembly works in the same way as the sequential code; in other words, every MPI process independently works on assembling their part of the global stiffness matrix and load vector that corresponds to the subdomain they own. The final stiffness matrix is stored in a distributed COO-format where values with duplicate indices are understood to be summed (this occurs for nodal interactions at subdomain interfaces). Many parallel linear solvers can work directly with this distributed matrix format. Figure 3.8 illustrates how the non-zero values of the global stiffness matrix are distributed in the 2D example.



(a) Subdomain `[0]` stiffness matrix

(b) Subdomain `[1]` stiffness matrix

Figure 3.8: Distributed assembly. The stiffness matrix is distributed according to the partitioning into subdomains. Non-zero values (marked by `x`) are stored in COO-format.

Remark: The PETSc solver interface is slightly more complicated than the described assembly procedure in this section, because the stiffness matrix must be distributed by rows in such a way that all interactions of a degree of freedom (a particular row) with any other degree of freedom (a particular column) is assembled by the owner of the row. Therefore, some interactions must be communicated when degrees of freedom are associated with nodes at the subdomain boundary.

3.3 Nested dissection solver

Nested dissection solvers are a class of direct methods for solving sparse linear systems. The method was originally proposed by Alan George [58] for solving the linear system arising from finite element discretizations. In essence, nested dissection is a method for finding a specific ordering of unknowns in Gaussian elimination that strives to reduce the fill-in from factorization. The algorithm is based on a divide-and-conquer strategy that makes it suitable for parallelization, because subproblems can be computed independently. Nested dissection

can be explained from various perspectives; in our discussion, we will illustrate the method by means of a simple example that is relevant to solving waveguide applications at large scale. For a general discussion of nested dissection algorithms, we refer to [82] and references therein.

Overview. The implementation of a parallel nested dissection solver in *hp*3D was motivated by the optical waveguide application (Chapter 5), but it extends to other types of problems with similarly partitioned domains. The optical fiber has a cylinder shape where the cylinder length (e.g., 10 000 wavelengths $\approx 7\,000$ μm) is substantially larger than its radius (≈ 200 μm). If subdomains are defined through cuts normal to the (longitudinal) fiber axis (e.g., by recursive bisection), interfaces between subdomains are cross-sections of the cylinder. A nested dissection solver is a natural choice to tackle this problem, because the size of each subdomain interface problem is relatively small due to the cylindrical shape of the domain. Furthermore, the subdomain interface does not increase in size (geometrically) when the fiber length is increased to simulate more wavelengths.

Initially, the interior DOFs of each subdomain are eliminated with static condensation; unlike the static condensation inside elements (cf. Section 3.4.3), this involves a *sparse* linear solve in each subdomain. In this first step, communication between MPI processes is not needed. The remaining interface problem is coupled between adjacent subdomains, as shown in Figure 3.9 for eight subdomains. This implies that the matrix representing the global interface problem is sparse and its non-zero entries are confined along a diagonal band. The matrix structure is illustrated for eight subdomains (i.e., seven interfaces) in Figure 3.10; each of the dense 4×4 blocks corresponds to the interaction of interface DOFs with each other (blocks on the diagonal) or with neighboring interface DOFs (blocks on the off-diagonal). The size of each dense block depends on the number of unknowns per interface. The total size of the coupled interface problem is therefore proportional to the number of interfaces between subdomains.

Algorithm. In a recursive (nested) process, coupled interface problems are separated into subproblems that are solved independently. Essentially, this process is equivalent to statically condensing subdomain interior DOFs onto the interfaces but with each separator defining a new interface. In a parallel implementation, communication between MPI processes that own neighboring subdomains is required. The large coupled interface problem is recursively subdivided into smaller ones that can efficiently be solved by a single processor (or some small number of processors) with a direct solver. In the recursive backtracking, the solution of the interface problem is propagated (broadcasted) to the corresponding subdomains (i.e., the MPI processes who own them) and the local interface solutions can be extracted independently by every processor. In the last step, backward substitution is executed in each subdomain to compute the subdomain interior DOFs from the interface solution.

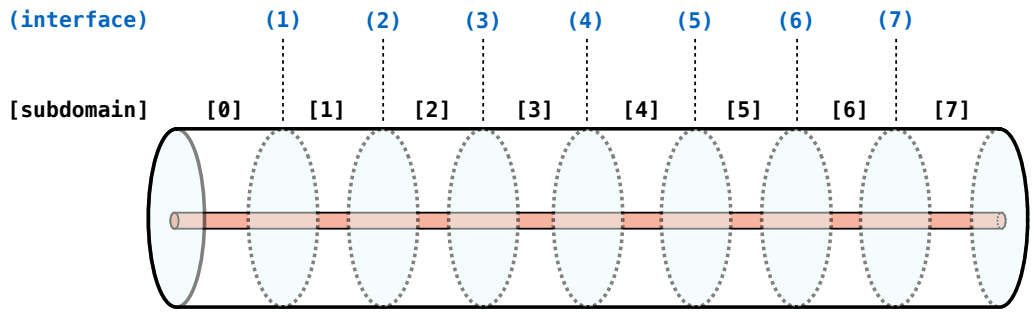


Figure 3.9: Coupled interface problem for the optical waveguide with eight subdomains. After MPI processes have independently eliminated interior subdomain DOFs, the remaining interface problem `(1-7)` is coupled between neighboring subdomains `[0-7]`.

It is perhaps most descriptive to illustrate the recursive structure of the algorithm with a binary separator tree. Nested dissection can be viewed as a recursive divide-and-conquer algorithm on an undirected graph; it uses graph separators (sets of vertices) whose removal divides the graph approximately in half (for binary trees). In the optical waveguide, graph separators correspond to the set of degrees of freedom on a particular fiber cross-section. For example, the first separator can be chosen as the middle interface; Figure 3.11 shows

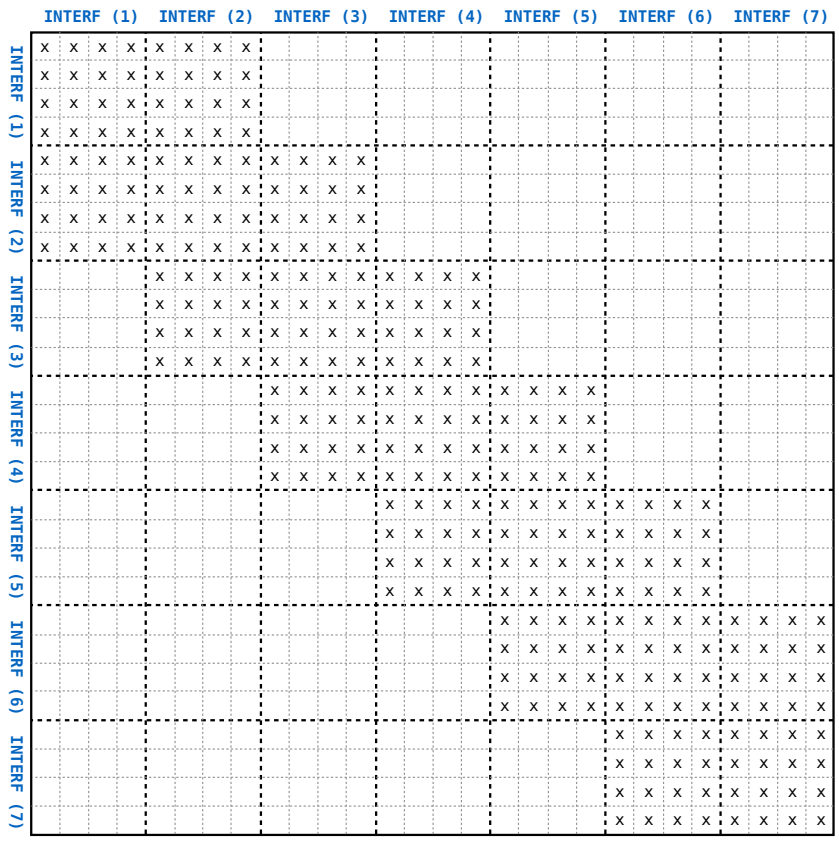


Figure 3.10: Structure of the (Hermitian positive definite) stiffness matrix for the coupled interface problem with eight subdomains. The matrix is sparse and its non-zero entries are confined along a diagonal band. Each dense block indicates the interaction of an interface with itself or with one of its neighboring interfaces.

how the interactions of the corresponding degrees of freedom separate the non-zero matrix entries. Once the separator has been defined, the non-zero matrix entries are divided into two uncoupled subproblems representing the interfaces on the left and on the right of the separator, as depicted in Figure 3.12. Each subproblem is computed by a subset of the processors: `[0-3]` and `[4-7]` independently work on the subproblems defined by interfaces `(1-3)` and `(5-7)`, respectively. Note that this split naturally follows from the partitioning of the domain, because the processors will work on a part of the domain where their respective subdomain lies. Once these subproblems are solved (i.e., statically condensed onto the separator interface), the remaining separator problem `(4)` can be computed. The separator solution is used to retrieve the final solution to the subproblems. For large problem instances,

the method is applied recursively. For example, if the subproblems (1-3) and (5-7) are too large to be statically condensed directly, then a new separator can be defined for each one. In this example, choosing the middle interface of each subproblem as a separator defines four independent subproblems (see Figure 3.13). Each subproblem consists of one interface only and is solved by the two MPI processes that own the attached subdomains. At this point, the binary separator tree consists of three stages: first, all eight processes participate to solve the uncoupled subproblems (1,3,5,7) (leafs of the tree); then, four processes compute the two uncoupled separator problems (2,6); at last, two processes solve the original separator problem (4) (root of the tree).

The remainder of this section is dedicated to the implementation of the nested dissection solver in *hp*3D. A performance analysis and numerical experiments for the fiber application are shown in Section 6.1.

Implementation. There are two distinctions between the previous description of the algorithm and the *hp*3D implementation of the solver: 1) the separator tree is not necessarily a binary tree; the current implementation supports binary trees, quadtrees, octrees, or any combination of trees of size $2^n, n = 1, 2, 3, \ldots$; and 2) the implemented solver traverses the tree structure bottom-up, i.e., it is eliminating the smallest size subproblems first (corresponding to the leafs of the separator tree), then recursively calling itself on the remaining (and possibly still large) coupled interface problem, thereby progressing one level up in the tree. With every additional level, fewer MPI processes are involved in the solution process.

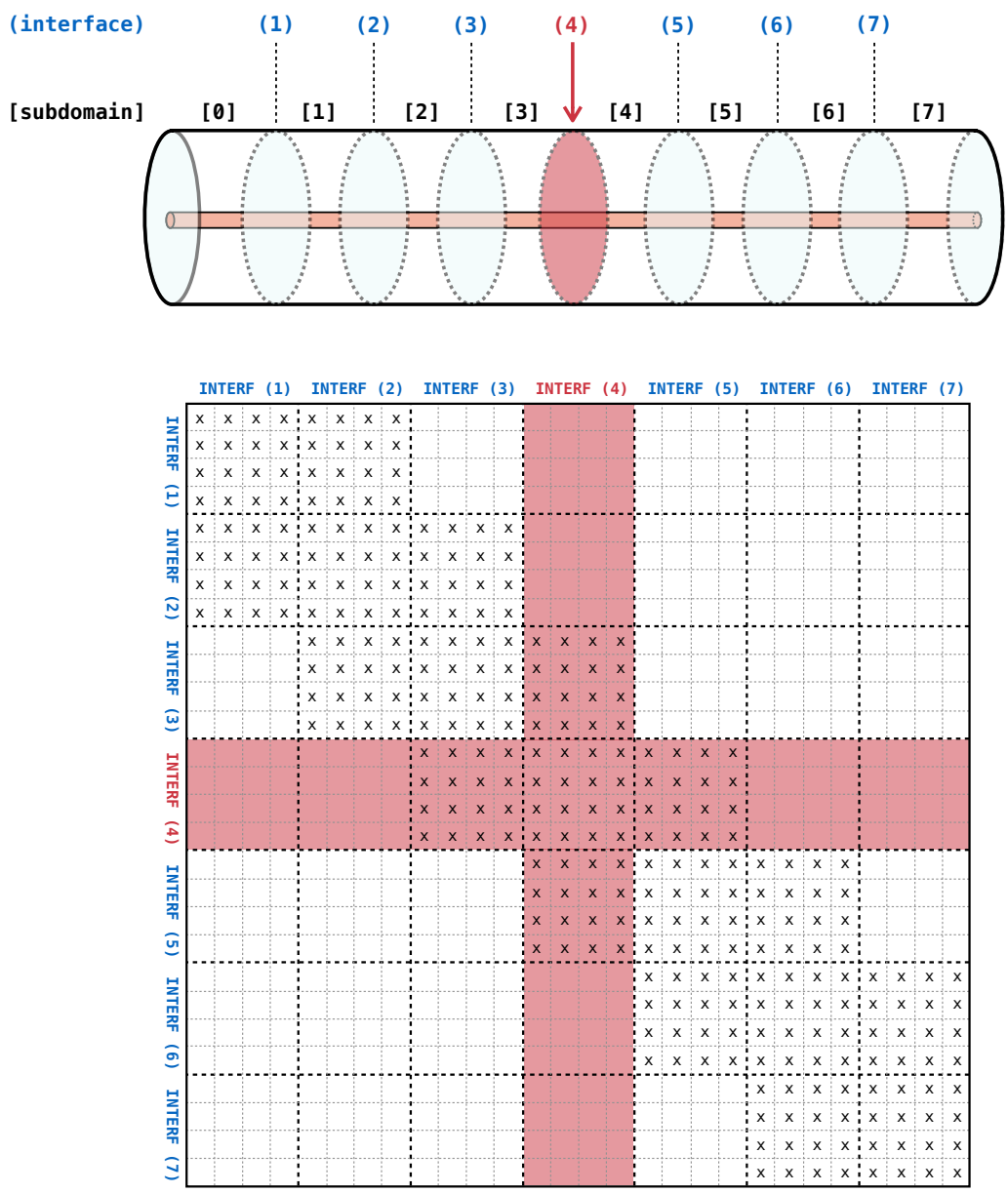


Figure 3.11: Coupled interface problem. Choosing a separator corresponds to selecting the set of degrees of freedom from a particular fiber cross-section.

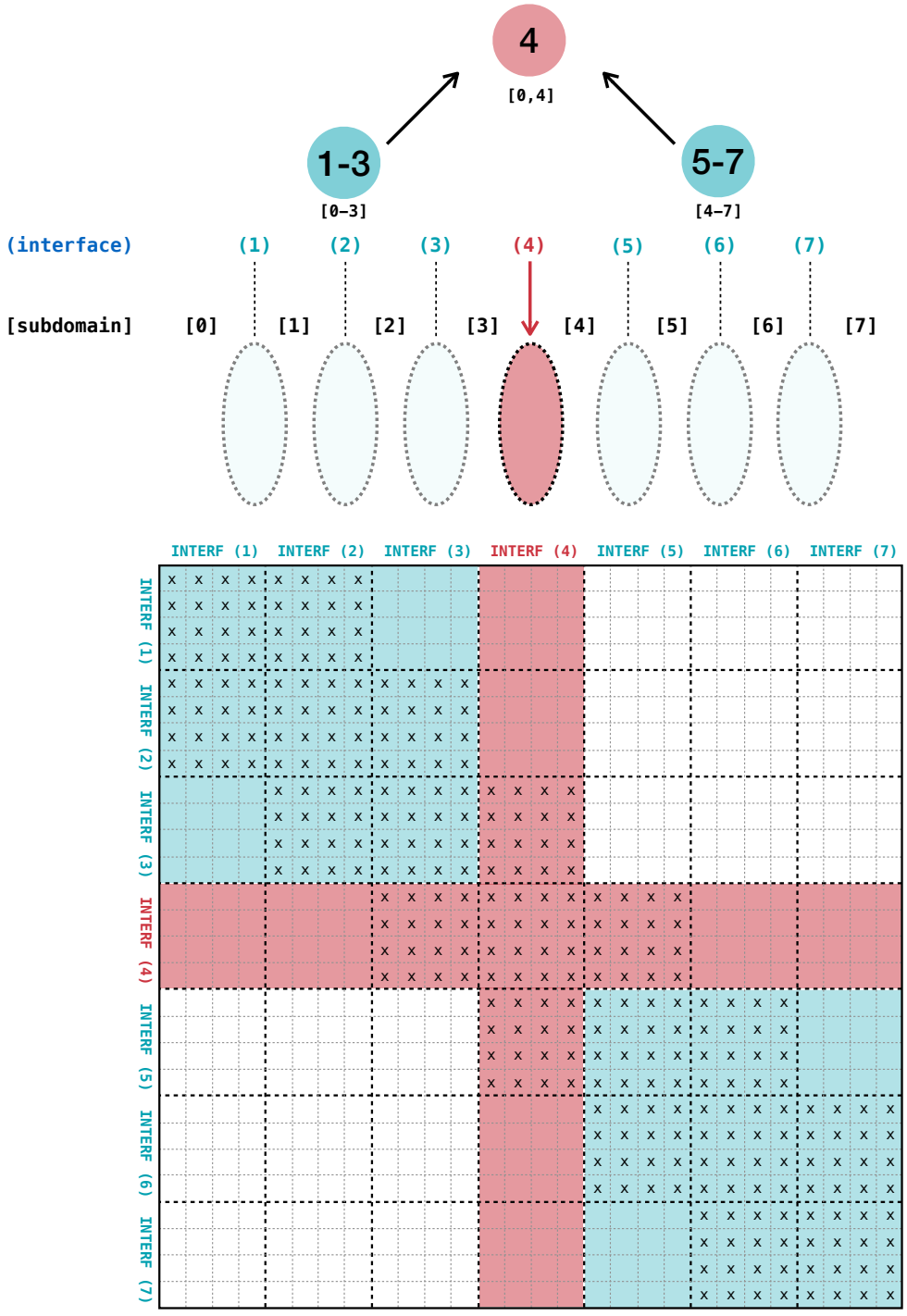


Figure 3.12: Coupled interface problem. The separator divides the global problem into two subproblems that can be solved independently. This is equivalent to statically condensing the left interfaces (1-3) and right interfaces (5-7) onto the separator interface (4).

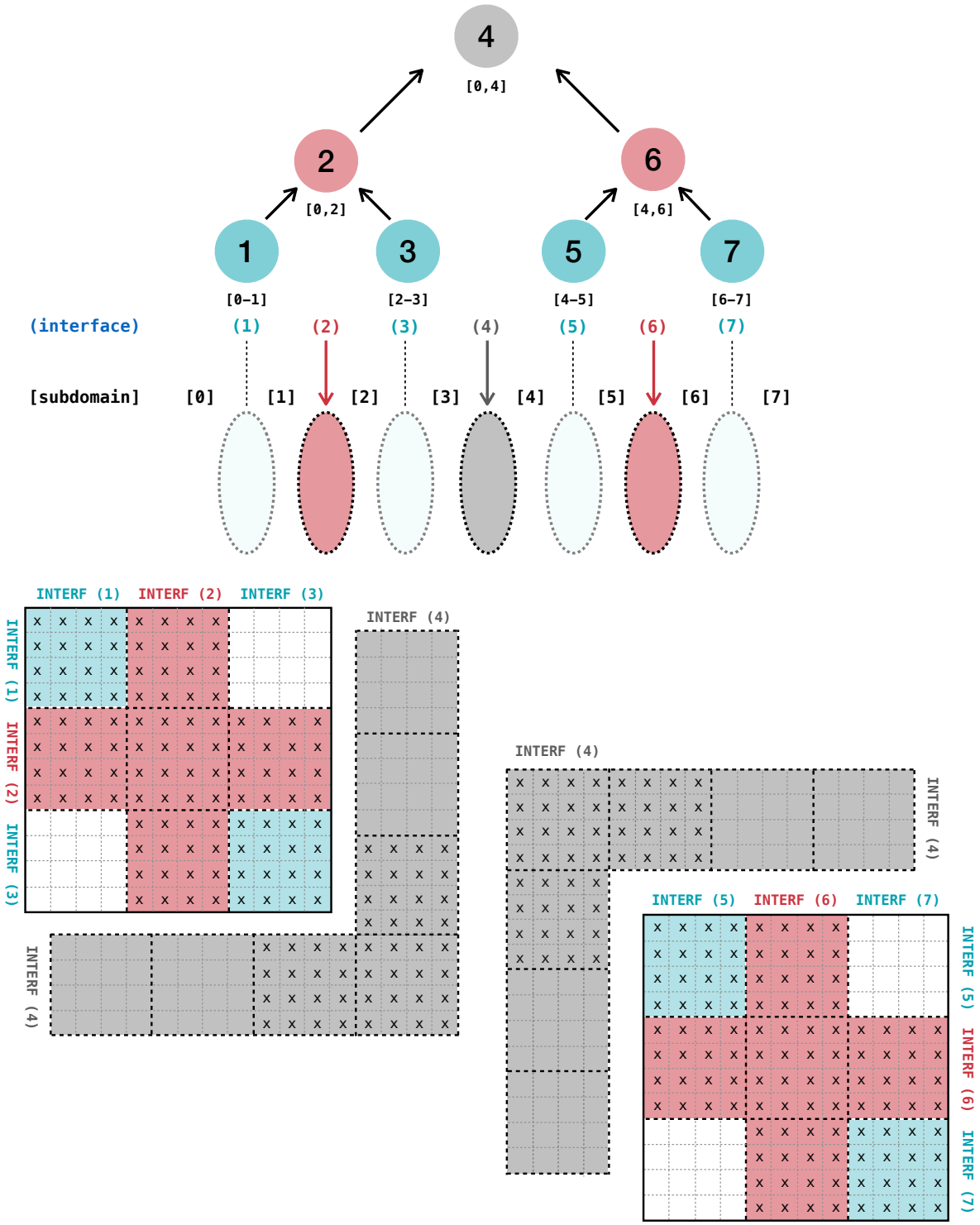


Figure 3.13: Coupled interface problem. Recursive application of the algorithm produces a (binary) separator tree, dividing coupled interface problems into smaller subproblems that can be solved efficiently by a small number of processors.

Preliminary step. Condensing the subdomain interior. As previously mentioned, the first level of the nested dissection solver is the static condensation of a subdomain interior onto its interfaces. The assembly of this subdomain problem only involves one communication step between MPI processes to identify whether a node is shared with other processes (i.e., it is on the subdomain interface) or whether it is in their subdomain interior. This step is similar to computing node ownership in the distributed assembly (cf. Section 3.2.3). Since eliminating subdomain interior DOFs does not involve communication between MPI processes, it can be done independently per subdomain with a sparse direct solver. In *hp*3D, the sequential MUMPS solver is used for this. To be precise, there are N independent MUMPS instances (one per subdomain) and each MPI communicator consists of a single MPI process that owns a particular subdomain. Once subdomain interior DOFs are eliminated, the remaining coupled interface problem is assembled in a distributed MUMPS instance (i.e., the banded sparse matrix, as in Figure 3.10, is distributed across all processors). This instance is the input for the recursive nested dissection routine. We break the routine into five steps.

Step 1. Communicators for the subproblems. To simplify the description of the algorithm, we assume that the number of processes in the input instance, called `mPROCS`, is a power of two; that is, $\texttt{mPROCS} = 2^l$, where $l \in \{1, 2, 3, \ldots\}$. Every process has a unique rank in this input instance: $\texttt{mRANK} \in \{0, 1, \ldots, \texttt{mPROCS} - 1\}$. Next, we define the size of the base case for the recursion in terms of the number of processes (subdomains) involved: `mSUB_PROCS`. Each subproblem involves a parallel direct solve of this size, and we assume that $\texttt{mSUB\_PROCS} = 2^k$, where $k \in \{1, 2, 3, \ldots\}$. A subproblem consists of $2^k - 1$ interfaces. The choice of the subproblem size determines the structure of the separator tree: $\texttt{mSUB\_PROCS} = 2$ results in a binary tree, $\texttt{mSUB\_PROCS} = 4$ in a quadtree, and so on. A subproblem has to be at least of size $\texttt{mSUB\_PROCS} = 2$ in which case it involves two (neighboring) MPI processes solving a single interface problem. If the size of the input problem is small enough ($l \leq k$), then the instance will be solved directly without further recursion. Other-

wise, the input problem is separated into $2^l/2^k = 2^{l-k}$ uncoupled subproblems by choosing $2^{l-k} - 1$ separators. To compute subproblems independently, the MPI communicator of the input instance is split into subcommunicators (groups) of size `mSUB_PROCS`. Every process calculates its group number, `mSUB_COMM` $\in \{0, 1, \ldots, 2^{l-k} - 1\}$, and subcommunicator rank, `mSUB_RANK` $\in \{0, 1, \ldots, 2^k - 1\}$, from their input rank:

```
mSUB_COMM = mRANK / mSUB_PROCS;
mSUB_RANK = MOD(mRANK,mSUB_PROCS).
```

Step 2. Communicator for the separator problem. One more subcommunicator is defined to compute the remaining separator problem with $2^{l-k} - 1$ interfaces. Since assembling this distributed separator problem requires information (Schur complements) from the subproblems, each group has one representative (MPI process) that participates in computing the separator problem. That group representative is responsible for communicating information from the static condensation of their subproblem to the separator problem, as well as broadcasting the solution from the separator problem to the other group members so they can retrieve their respective subdomain solutions. In each group, the process with `mSUB_RANK` $= 0$ is dedicated as the representative. The number of processes participating in the separator problem therefore is `mINT_PROCS` $= 2^{l-k}$. Every group representative obtains their rank in the new subcommunicator, called `mINT_RANK`, from their group number, `mSUB_COMM`. The defined variables are summarized below, and an example for an input instance with 32 processes and subproblems with four processes is shown in Table 3.2.

`mPROCS`: Number of processes in the input instance;
`mRANK`: Rank of each process in the input instance;
`mSUB_PROCS`: Number of processes in each subproblem;
`mSUB_RANK`: Rank of each process in the subproblem;
`mINT_PROCS`: Number of processes in the (separator) interface problem;
`mINT_RANK`: Rank of each process in the (separator) interface problem.

mRANK	mSUB_RANK	mINT_RANK
0	0	0
1	1	-
2	2	-
3	3	-
4	0	1
5	1	-
6	2	-
7	3	-
8	0	2
9	1	-
10	2	-
11	3	-
12	0	3
13	1	-
14	2	-
15	3	-
⋮	⋮	⋮
28	0	7
29	1	-
30	2	-
31	3	-

Table 3.2: Nested dissection communicators: an example with 32 subdomains (`mPROCS` = 32), split into eight subproblems (`mINT_PROCS` = 8) of four subdomains each (`mSUB_PROCS` = 4).

Step 3: Assembling the subproblems. Once all subcommunicators are defined, the corresponding subproblems can be extracted from the input instance. Because of the particular structure of the partitioning that is assumed a-priori, this subproblem assembly does not require any communication between MPI processes. Consider a coupled interface problem with one separator:

$$\begin{bmatrix} A_{ll} & A_{li} & - \\ A_{il} & A_{ii} & A_{ir} \\ - & A_{ri} & A_{rr} \end{bmatrix} \begin{bmatrix} x_l \\ x_i \\ x_r \end{bmatrix} = \begin{bmatrix} b_l \\ b_i \\ b_r \end{bmatrix}, \tag{3.1}$$

where the indices $\{l, i, r\}$ refer to contributions from degrees of freedom "left" of the interface, on the interface, and "right" of the interface, respectively. The processes in the two subcommunicators ("left" and "right") assemble and solve the two subproblems independently:

Left subproblem:

$$\begin{aligned} \texttt{LHS} &= [A_{ll}], \\ \texttt{RHS} &= [b_l \mid A_{li}], \\ \texttt{Solve} &\Rightarrow [A_{ll}^{-1} b_l \mid A_{ll}^{-1} A_{li}]; \end{aligned}$$

Right subproblem:

$$\begin{aligned} \texttt{LHS} &= [A_{rr}], \\ \texttt{RHS} &= [b_r \mid A_{ri}], \\ \texttt{Solve} &\Rightarrow [A_{rr}^{-1} b_r \mid A_{rr}^{-1} A_{ri}]. \end{aligned}$$

Step 4: Assembling the separator problem. In the previous step, the right-hand side was assembled with the load $\{b_l, b_r\}$ and with interface interactions $\{A_{li}, A_{ri}\}$ to compute Schur complement factors. These are now used by the group representatives to statically condense the left and right subproblems onto the separator:

Left subproblem:

$$\begin{aligned} A_l^{\text{Schur}} &= A_{il} A_{ll}^{-1} A_{li}, \\ b_l^{\text{Schur}} &= A_{il} A_{ll}^{-1} b_l; \end{aligned}$$

Right subproblem:

$$\begin{aligned} A_r^{\text{Schur}} &= A_{ir} A_{rr}^{-1} A_{ri}, \\ b_r^{\text{Schur}} &= A_{ir} A_{rr}^{-1} b_r. \end{aligned}$$

Separator (interface) problem:

$$\begin{aligned} \texttt{LHS} &= [A_{ii} - A_l^{\text{Schur}} - A_r^{\text{Schur}}], \\ \texttt{RHS} &= [b_i - b_l^{\text{Schur}} - b_r^{\text{Schur}}], \\ \texttt{Solve} &\Rightarrow x_i. \end{aligned}$$

In this particular example, the separator problem consists of only one interface and can be solved directly. Generally, this problem instance ("solving x_i") is computed by recursively applying nested dissection to it. In other words, the assembled problem as shown above serves as the input to the next level of the nested dissection solver and steps (1)–(5) are repeated for that instance.

Step 5: Retrieving the solution to subproblems. Once the separator problem has been solved directly or through recursive application of nested dissection, the group representatives compute the solutions to the subproblems and broadcast them to the other group members:

Left subproblem:

$$x_l = A_{ll}^{-1} b_l - A_{ll}^{-1} A_{li} x_i;$$

Right subproblem:

$$x_r = A_{rr}^{-1} b_r - A_{rr}^{-1} A_{ri} x_i.$$

Final step: Retrieving the solution in the subdomain interior. In the same way that the separator solution of step (4) serves to retrieve the subproblem solution in step (5), the solution to the coupled interface problem solved in steps (1)–(5) serves to retrieve the solution to subdomain interior DOFs that were eliminated in the preliminary step. This final step is local to each subdomain, hence it does not involve communication between MPI processes.

3.4 Coding DPG

DPG methods have been used in *hp*3D to compute variational formulations for a variety of multiphysics applications (see Section 1.3.4). Therefore, optimizing the computational performance of DPG implementations has been a focus of the *hp*3D development in recent years. In this section, some of these optimizations that the author has contributed to are presented:

- Computing optimal test functions in the enriched test space comes at a computational cost in the element integration, including construction of the Gram matrix. In Section 3.4.1, we discuss sum factorization techniques for the DPG linear system that can accelerate numerical integration by more than one order of magnitude.

- Trace unknowns that arise from breaking the test space in the DPG method can be discretized conformingly by using shape functions for standard elements of the exact sequence, evaluated on the element boundary. In Section 3.4.2, we show how trace variables have been incorporated in the *hp*3D code.

- Static condensation is a technique widely used in finite element coding, but we would like to emphasize the importance of it in the context of ultraweak variational formulations. In Section 3.4.3, we describe *hp*3D's static condensation module which is hidden from both the user application and the constrained approximation routines.

Before these points are discussed in more detail, we briefly recap how the DPG linear system is constructed.

The DPG linear system. In the description of the DPG linear system, we follow the concise exposition of [32, 102]. We would like to point out that other DPG variants, e.g., discrete least-squares FE [81], lead to a different linear system. The implementation of the DPG method can be explained from the perspective of the mixed problem (2.19). Consider the corresponding broken formulation:

$$\begin{cases} u_h \in \mathcal{U}_h,\ \hat{u}_h \in \hat{\mathcal{U}}_h,\ \psi \in \mathcal{V}^r(\Omega_h), & \\ (\psi, v)_{\mathcal{V}} + b_h(u_h, v) + \langle \hat{u}_h, v \rangle_{\Gamma_h} = l(v), & v \in \mathcal{V}^r(\Omega_h), \\ b_h(w_h, \psi) = 0, & w_h \in \mathcal{U}_h, \\ \langle \hat{w}_h, \psi \rangle_{\Gamma_h} = 0, & \hat{w}_h \in \hat{\mathcal{U}}_h. \end{cases} \tag{3.2}$$

We reduce (3.2) to a matrix equation. Let $\mathfrak{U}_h = \{\mathfrak{u}_i\}_{i=1}^{N}$, $\hat{\mathfrak{U}}_h = \{\hat{\mathfrak{u}}_i\}_{i=1}^{\hat{N}}$, and $\mathfrak{V}^r = \{\mathfrak{v}_i\}_{i=1}^{M}$ (where $M > N + \hat{N}$) denote bases for the discrete trial space $\mathcal{U}_h \times \hat{\mathcal{U}}_h$ and the enriched test space V^r, respectively. We define the stiffness matrix for the modified bilinear form, the

Gram matrix, and the load vector as follows:

$$\mathrm{B}_{ij} = b(\mathfrak{u}_j, \mathfrak{v}_i), \quad \hat{\mathrm{B}}_{ij} = \langle \hat{\mathfrak{u}}_j, \mathfrak{v}_i \rangle, \quad \mathrm{G}_{ij} = (\mathfrak{v}_j, \mathfrak{v}_i)_V, \quad \mathrm{l}_i = l(\mathfrak{v}_i). \tag{3.3}$$

In matrix form, the problem can now be formulated in the following way:

Find the set of coefficients (over field $\mathbb{F} = \mathbb{R}$ (or $\mathbb{C}$))

$$\mathrm{w} = [\mathrm{w}_i]_{i=1}^{N} \in \mathbb{F}^N, \quad \hat{\mathrm{w}} \in [\hat{\mathrm{w}}_i]_{i=1}^{\hat{N}} \in \mathbb{F}^{\hat{N}}, \text{ and } \mathrm{q} = [\mathrm{q}_i]_{i=1}^{M} \in \mathbb{F}^M \tag{3.4}$$

such that

$$\mathrm{u_h} = \sum_{i=1}^{N} \mathrm{w}_i \mathfrak{u}_i, \quad \hat{\mathrm{u}}_\mathrm{h} = \sum_{i=1}^{\hat{N}} \hat{\mathrm{w}}_i \hat{\mathfrak{u}}_i, \text{ and } \Psi = \sum_{i=1}^{M} \mathrm{q}_i \mathfrak{v}_i \tag{3.5}$$

satisfy

$$\begin{bmatrix} \mathrm{G} & \mathrm{B} & \hat{\mathrm{B}} \\ \mathrm{B}^* & 0 & 0 \\ \hat{\mathrm{B}}^* & 0 & 0 \end{bmatrix} \begin{bmatrix} \Psi \\ \mathrm{u_h} \\ \hat{\mathrm{u}}_\mathrm{h} \end{bmatrix} = \begin{bmatrix} \mathrm{l} \\ 0 \\ 0 \end{bmatrix}. \tag{3.6}$$

The residual Ψ can be eliminated by static condensation so that we obtain a linear system for the approximate solution $[\mathrm{u_h}\ \hat{\mathrm{u}}_\mathrm{h}]$:

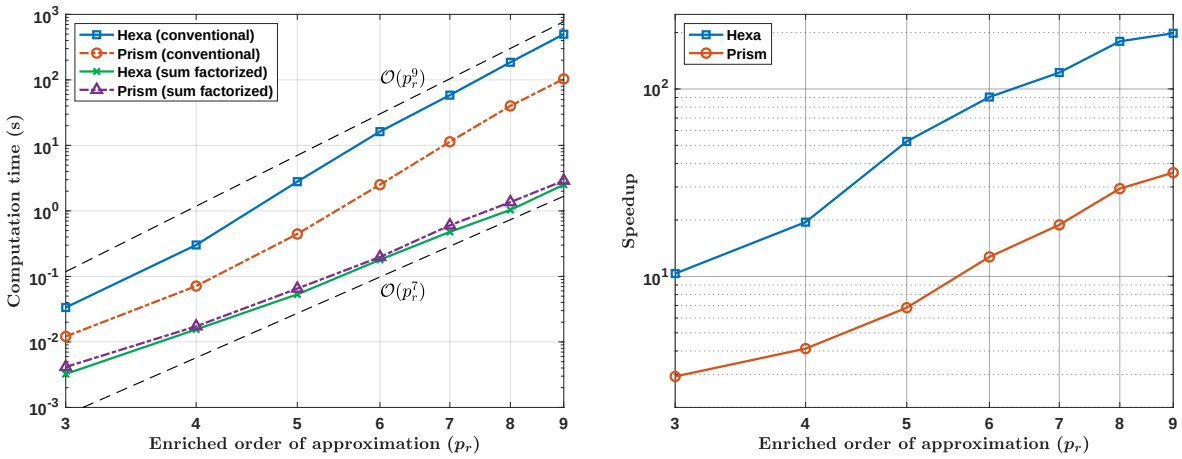
$$\begin{bmatrix} \mathrm{B}^*\mathrm{G}^{-1}\mathrm{B} & \mathrm{B}^*\mathrm{G}^{-1}\hat{\mathrm{B}} \\ \hat{\mathrm{B}}^*\mathrm{G}^{-1}\mathrm{B} & \hat{\mathrm{B}}^*\mathrm{G}^{-1}\hat{\mathrm{B}} \end{bmatrix} \begin{bmatrix} \mathrm{u_h} \\ \hat{\mathrm{u}}_\mathrm{h} \end{bmatrix} = \begin{bmatrix} \mathrm{B}^*\mathrm{G}^{-1}\mathrm{l} \\ \hat{\mathrm{B}}^*\mathrm{G}^{-1}\mathrm{l} \end{bmatrix}. \tag{3.7}$$

Note that the positive definite Gram matrix is block-diagonal because the test space is discontinuous. Therefore, Ψ can be removed from the system element-wise. Later, the residual may be post-processed if needed (e.g., for adaptivity). Once the matrices and vectors in (3.7) are set up for each element, we can proceed with the distributed assembly as described in Section 3.2.3. The remainder of this section describes techniques on how to efficiently implement the construction of the linear systems (3.6)–(3.7).

3.4.1 Sum factorization

One common critique of the DPG method is the computational complexity due to determining optimal test functions. While the computation of optimal test functions is element-local and can be easily parallelized, it can indeed become a computational bottleneck when using a high-order discretization. The main reason for this is the computational cost of standard numerical integration (Gaussian quadrature) techniques. For example, integration for hexahedral and prismatic elements usually requires $\mathcal{O}(p^9)$ operations, where p is the polynomial order. The optimal test functions are computed by using an enriched test space that has a larger dimension than the trial space. This is typically accomplished by using a higher order of discretization for the test space (see Section 2.3). This "enriched order" is defined by $p_r := p + \Delta p$, where p is the discretization order of the trial space; we use $\Delta p = 1$ uniformly. Therefore, the Gram matrix G that derives from the test inner product (cf. (3.3)) can be particularly costly to compute. It is essential to optimize the numerical integration to achieve an efficient method. Mora and Demkowicz [92] and Badger et al. [11] have shown that the computational complexity of the element integration for DPG linear systems can be reduced from $\mathcal{O}(p_r^9)$ to $\mathcal{O}(p_r^7)$ for hexahedral and prismatic elements, respectively. This is accomplished by employing sum factorization techniques that exploit the tensor-product structure of the element shape functions for these element types (full tensorization of the hexahedron and partial tensorization of the prism). For algorithms and further discussion of sum factorization techniques, see [92, 11] and references therein.

The implementations for fast numerical integration in *hp*3D were created by the authors of [92, 11]. Figure 3.14 illustrates the importance of sum factorization for computing the matrices $\mathrm{G}, \mathrm{B}, \mathrm{l}$ (defined in (3.3)) in the ultraweak DPG Maxwell formulation. In this formulation, the Gram matrix is computed using the test inner product that induces the adjoint graph norm (Section 2.5.1). As depicted in Figure 3.14a, compute times are reduced dramatically with sum factorization. For both element types, hexahedra and prisms, the



(a) Runtime: conventional vs. sum-factorized

(b) Sum factorization speedup

Figure 3.14: Sum factorization for the element-local computation of the matrices $\mathrm{G}, \mathrm{B}, \mathrm{l}$ in the ultraweak DPG Maxwell formulation: (a) the asymptotic complexity is reduced from $\mathcal{O}(p_r^9)$ to $\mathcal{O}(p_r^7)$; (b) for high-order elements, the computation is accelerated by more than one order of magnitude. Runtimes collected on Stampede2 SKX compute node with multi-threading enabled (one thread per element). Algorithms and implementations from [92, 11].

observed asymptotic complexity is reduced from $\mathcal{O}(p_r^9)$ to $\mathcal{O}(p_r^7)$. High-order-element computation is significantly accelerated in particular: we observe up to 35x speedup for prisms and 200x speedup for hexahedra (see Figure 3.14b).

3.4.2 Trace unknowns

In this section, we outline how the DPG trace unknowns are implemented in *hp*3D.

Recall that depending on the energy space setting of the variational formulation (Section 2.4), the broken DPG formulation requires trace unknowns that are *element-wise traces of globally conforming functions* in H^1, $H(\mathrm{div})$, or $H(\mathrm{curl})$ (cf. (2.36)). In other words, on element interfaces, traces of H^1 must be continuous, traces of $H(\mathrm{div})$ must be continuous in normal direction, and traces of $H(\mathrm{curl})$ must be tangentially continuous. In a three-dimensional FE code that supports discretizations of these energy spaces, i.e., discretizations with H^1-, $H(\mathrm{div})$-, and $H(\mathrm{curl})$-conforming elements,[4] the simplest way to discretize

[4]In *hp*3D, conforming discretizations of vector-valued $H(\mathrm{div})$- and $H(\mathrm{curl})$-functions are realized with Raviart–Thomas and Nédélec elements, respectively. See [50] for details.

the DPG trace unknowns is by using restrictions of these elements to the element boundary. This way, much of the code infrastructure that is in place for standard Galerkin discretizations can be reused for the traces (e.g., implementation of element shape functions). While there is no notion of a trace for L^2-functions, L^2-conforming elements are needed for the discretization of ultraweak formulations.

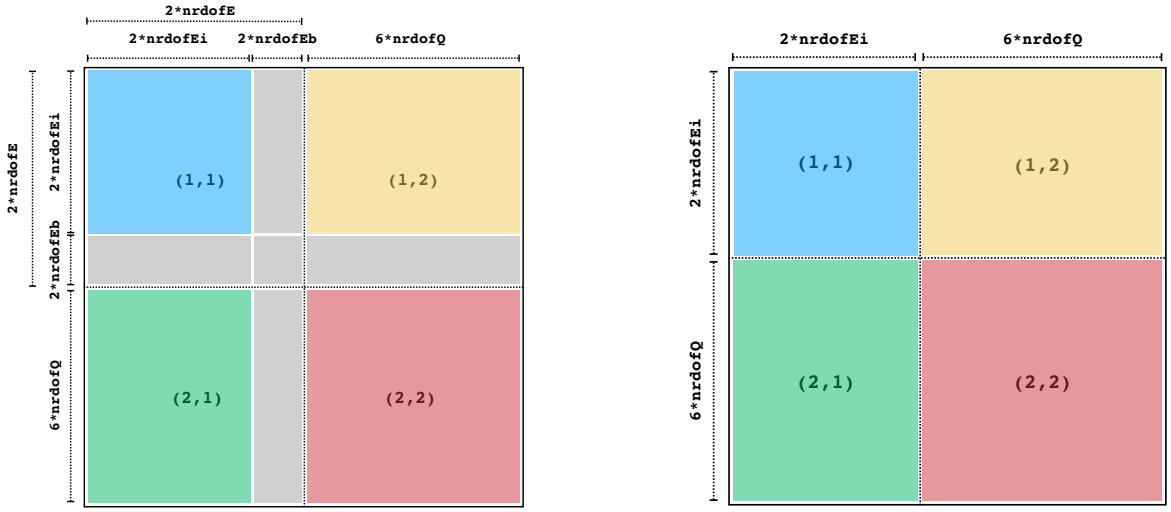
Parameter	Description	DOFs
`nrdofE`	$H(\text{curl})$ variable	$3p(p+1)^2$
`nrdofEi`	$H(\text{curl})$ trace	$12p^2$
`nrdofEb`	$H(\text{curl})$ bubble	$3p(p-1)^2$
`nrdofQ`	L^2 component	p^3

Table 3.3: Trial DOFs in a hexahedral element for the ultraweak DPG Maxwell formulation, depending on the polynomial order p.

Consider the ultraweak DPG Maxwell formulation (2.42) discretized with a hexahedral FE mesh. To compute the electric field $\boldsymbol{E}$ and magnetic field $\boldsymbol{H}$, we set up two vector-valued L^2-variables with three components each. Additionally, we require two vector-valued $H(\text{curl})$-variables for the traces $\hat{\boldsymbol{E}}$ and $\hat{\boldsymbol{H}}$. The number of trial DOFs for each component in a hexahedral element of order p is given in Table 3.3. Consequently, there are $2 * \texttt{nrdofEi} + 6 * \texttt{nrdofQ}$ trial DOFs for a hexahedron in the ultraweak DPG Maxwell problem.

If standard $H(\text{curl})$-conforming elements are used for the discretization of the traces, then the corresponding element-interior (bubble) DOFs must be deleted from the linear system. In *hp*3D, a simple but effective procedure is implemented where each H^1-, $H(\text{div})$-, and $H(\text{curl})$-variable has a flag that specifies whether it is a standard variable defined on the entire domain or a trace variable defined on the mesh skeleton. If a variable is defined as a trace, then its bubble DOFs are not allocated in the local element matrix for assembly. Since the *constrained approximation* routines[5] do not modify bubble DOFs, the usual assembly routines can operate directly on this reduced local element matrix.

[5]For elements with constrained (hanging) nodes, the constrained approximation routines are used to construct the *modified element* that expresses constrained element DOFs as linear combinations of unconstrained (parent) DOFs. Element-interior (bubble) DOFs cannot be constrained DOFs. See [37] for details.



(a) With standard $H(\text{curl})$ variable

(b) With $H(\text{curl})$ trace variable

Figure 3.15: Eliminating bubble DOFs for DPG trace unknowns. The local element matrices for the DPG ultraweak Maxwell formulation avoid storing element-interior DOFs for the traces of $H(\text{curl})$-conforming elements.

Figure 3.15 illustrates the element-local matrix entries for the ultraweak DPG Maxwell problem, discretized with or without specifying an interface (trace) variable in *hp*3D. The parameters used in Figure 3.15 relate to the quantities shown in Table 3.3. As depicted in Figure 3.15, the submatrix `(2,2)` does not change because it only concerns the L^2 degrees of freedom. However, any interactions with the element trace unknown's interior DOFs (which are meaningless) are omitted from the matrix. This reduces both storage requirements as well as computation time since subsequent modification to eliminate these contributions is no longer needed.

3.4.3 Static condensation

Static condensation plays an important role in computing DPG finite element discretizations. Firstly, the local residual DOFs are usually eliminated (statically condensed) in the construction of the DPG linear system (3.7) as previously described. Then, static condensation is used to locally eliminate bubble DOFs associated with the middle node of an element.

In ultraweak formulations, this includes all L^2-trial DOFs. The corresponding bubble DOF interactions form independent blocks (submatrices) within the element stiffness matrix in the sense that they have no global conformity requirements. These DOFs can therefore be eliminated from the system element-wise. The resulting global problem is a significantly smaller linear system that involves only the interface unknowns (i.e., DOFs associated with shape functions whose support extends to more than one element). Particularly in high-order discretizations, the number of eliminated interior DOFs can be significant, leading to a drastic reduction of the computational cost in the linear solve. For example, the ultraweak DPG Maxwell problem has $6p^3$ bubble L^2-DOFs and $24p^2$ trace $H(\text{curl})$-DOFs locally per hexahedral element (cf. Table 3.3). Therefore, the size of each local element matrix is reduced by a factor of $6p^3/(6p^3 + 24p^2) = p/(p + 4)$. In the global system, the factor is even larger since interface DOFs are shared among elements.

In *hp*3D, we developed a static condensation module that is integrated with the assembly procedure and removes bubble DOFs from the system automatically. A significant computational advancement came from eliminating these bubble DOFs directly from the local element matrix *before* creating the modified element matrix via constrained approximation routines. The static condensation of interior DOFs for each variable can be enabled or disabled by the user. The user may also control whether Schur complement factors are stored or later recomputed on-the-fly when the element-interior DOFs are retrieved by back-substitution.

Remark: The statically condensed system has the same size for all DPG formulations of a particular problem. However, the conditioning of the global linear system is not the same for every formulation [102].

Chapter 4

DPG Method for Linear Optical Waveguides

In this chapter, we introduce the governing equations of linear waveguide theory and the closed-form solutions (*eigenmodes*) to the corresponding eigenproblem in step-index fibers. We discuss the pollution effect in long waveguides and conduct numerical waveguide simulations with DPG discretizations of the ultraweak Maxwell system. We also study mesh adaptivity for multi-mode waveguide propagation in weakly-guiding step-index fibers, leading into the computation of the nonlinear fiber model in the next chapter. In the context of large-scale distributed computation, we discuss the benefits of dynamic load balancing and show numerical results.[1]

4.1 Linear waveguide theory

A step-index fiber is an optical waveguide. The propagation of the (confined) optical field in the fiber can be described as a *guided wave* along the longitudinal fiber axis. The theory of linear waveguide problems is well established [64, 74, 1]. In this section, we derive the guided wave equations in a linear medium. The solutions (eigenmodes) to the corresponding eigenvalue problem are the *modes of the waveguide*. Any propagating (guided) field in the waveguide is a linear combination of these (guided) modes.

[1]The contents of this chapter are partially taken from a previous publication: S. Henneking and L. Demkowicz. "A numerical study of the pollution error and DPG adaptivity for long waveguide simulations". In: *Comput. Math. Appl.* (2020).
The author contributed to both numerical implementation of the method and analysis of the results.

4.1.1 Guided modes

Recall the time-harmonic Maxwell equations in a linear, isotropic and homogeneous medium (cf. (1.16)–(1.17)):

$$\nabla \times \boldsymbol{E} = -i\omega\mu\boldsymbol{H}, \tag{4.1}$$

$$\nabla \times \boldsymbol{H} = i\omega\varepsilon\boldsymbol{E}. \tag{4.2}$$

In the absence of free charges in a dielectric medium, this time-harmonic Maxwell system reduces to vectorial Helmholtz equations:[2]

$$\Delta\boldsymbol{E} + \mu\varepsilon\omega^2\boldsymbol{E} = \boldsymbol{0}, \tag{4.3}$$

$$\Delta\boldsymbol{H} + \mu\varepsilon\omega^2\boldsymbol{H} = \boldsymbol{0}. \tag{4.4}$$

Suppose that the center of the waveguide is aligned with the z-axis, so that (x, y) are the transverse directions. At the radial boundary (i.e., the boundary of the transverse domain), we assume perfect electrical conductor (PEC) boundary conditions; that is, the tangential electrical field vanishes on the boundary. Assuming a guided wave propagating along the z-direction, the fields take the form:

$$\boldsymbol{E}(x, y, z) = \boldsymbol{E}(x, y)e^{\pm ikz}, \tag{4.5}$$

$$\boldsymbol{H}(x, y, z) = \boldsymbol{H}(x, y)e^{\pm ikz}, \tag{4.6}$$

where $\beta = \pm ik$ is called the *propagation constant* and k is the *wavenumber*. We consider waves that are traveling in the forward $(+z)$ direction, hence we assume the ansatz $e^{iwt-ikz}$. Note that, in principle, backward traveling waves are possible as well, and they are usually discussed in the context of resonant cavity problems. Using the ansatz (4.5) and (4.6) for the

[2] a) apply the curl operator to (4.1); b) substitute $\nabla \times H$ via (4.2); c) use vector identity $\nabla \times (\nabla \times E) = \nabla(\nabla \cdot E) - \Delta E$; d) apply Gauss's Law in the absence of free charges: $\nabla \cdot E = 0$. Analagous for (4.2).

Helmholtz equations (4.3) and (4.4), respectively, we obtain (transverse) Helmholtz equations for the field envelopes,

$$\left[\Delta_t + (\mu\varepsilon\omega^2 - k^2)\right] \left\{ \begin{array}{c} \boldsymbol{E}(x,y) \\ \boldsymbol{H}(x,y) \end{array} \right\} = \mathbf{0}, \tag{4.7}$$

where Δ_t is the transverse part of the Laplacian operator. Analysis of the corresponding eigenvalue problem, $(\Delta_t + \gamma^2)\Psi = 0$, where $\gamma^2 \equiv \mu\varepsilon\omega^2 - k^2$, with appropriate boundary conditions, yields a spectrum of positive eigenvalues γ_λ^2, and eigenmodes $\Psi_\lambda, \lambda = 1, 2, \ldots$; *guided modes* are those for which the corresponding wavenumber k_λ is real-valued; otherwise, the mode is decaying and called *evanescent*.

Modes in a weakly-guiding step-index fiber. A circular step-index fiber consists of a fiber core of radius r_1 and a surrounding cladding of radius r_2. The material refractive index of the core (n_1) is larger than the refractive index of the cladding (n_2), i.e.,

$$n(r) = \begin{cases} n_1 \quad\quad\quad , & r \le r_1, \\ n_2 < n_1 \ , & r_1 < r < r_2. \end{cases} \tag{4.8}$$

where $r = \sqrt{x^2 + y^2}$. We equivalently refer to these quantities as $r_{\text{core}}, r_{\text{clad}}, n_{\text{core}}$, and n_{clad}, respectively. The step-index fiber is called *weakly-guiding* if $(n_1 - n_2)/n_1 \ll 1$. Under the weakly-guiding condition, the wave equation may be posed for the transverse electric field components:

$$\text{Core:} \quad \left[\Delta_t + \left(\frac{\omega^2}{c^2} n_1^2 - k^2\right)\right] \left\{ \begin{array}{c} E_x \\ E_y \end{array} \right\} = 0, \tag{4.9}$$

$$\text{Cladding:} \quad \left[\Delta_t - \left(k^2 - \frac{\omega^2}{c^2} n_2^2\right)\right] \left\{ \begin{array}{c} E_x \\ E_y \end{array} \right\} = 0. \tag{4.10}$$

The corresponding eigenvalue problem with appropriate boundary and core-cladding interface conditions yields transverse *core-guided* modes ψ_λ that satisfy

$$\frac{\omega}{c}n_2 < |k_\lambda| < \frac{\omega}{c}n_1. \tag{4.11}$$

These modes have two possible linear polarizations in the transverse directions: $\hat{\boldsymbol{e}}_{\boldsymbol{x}}$ and $\hat{\boldsymbol{e}}_{\boldsymbol{y}}$. They are therefore called *LP modes*. In cylindrical coordinates, they must satisfy the following characteristic equation involving l-th order Bessel functions J_l and modified Bessel functions K_l (cf. [74, Eqn. (8.128)]):

$$\frac{(\gamma r_1)J_l'(\gamma r_1)}{J_l(\gamma r_1)} = \frac{(\beta r_1)K_l'(\beta r_1)}{K_l(\beta r_1)}, \quad l = 0, 1, 2, \ldots, \tag{4.12}$$

as well as

$$(r_1\gamma)^2 + (r_1\beta)^2 = r_1^2\frac{\omega^2}{c^2}(n_1^2 - n_2^2) \equiv V^2, \tag{4.13}$$

where

$$\gamma^2 = \left(\frac{\omega^2}{c^2}n_1^2 - k^2\right), \quad \beta^2 = \left(k^2 - \frac{\omega^2}{c^2}n_2^2\right). \tag{4.14}$$

V is called the *normalized frequency* or V-number, and NA $:= \sqrt{n_1^2 - n_2^2}$ is the fiber core numerical aperture.

Given any suitable fiber parameters n_1, n_2, r_1, r_2, and frequency ω, for every $l = 0, 1, 2, \ldots$, there are infinitely many solutions γ, β that satisfy (4.12) but only finitely many of these may satisfy (4.13) as well. These are denoted γ_{lp}, β_{lp}, $p = 1, 2, \ldots, N$. We find that only for $l = 0$, there exists a solution for any $V > 0$. This *fundamental mode* is the LP_{01} mode, and it does not have a *cutoff frequency* (i.e., a frequency V_c below which the mode cannot propagate). All other LP modes do have a cutoff frequency and can only propagate if the V-number is larger than their cutoff frequency. These cutoff frequencies can be calculated for each mode and are shown in Table 4.1 for the lowest-order LP modes (cf. roots of Bessel functions—see Tables A.9 and A.10).

Guided mode	LP_{01}	LP_{11}	LP_{21}, LP_{02}	LP_{31}	LP_{12}	$\cdots$
Cutoff frequency V_c	-	2.405	3.832	5.136	5.520	$\cdots$

Table 4.1: Cutoff frequencies of lowest-order LP modes in a weakly-guiding step-index fiber. The fundamental mode (LP_{01}) has no cutoff and can propagate at any frequency.

Denoting $\Psi_0 \equiv AJ_l(\gamma a) = CK_l(\beta a)$ for some constants A, C, we can write the modes as:

$$\Psi(r,\theta) = \Psi_0 \cos(l\theta) \begin{cases} J_l(\gamma r)/J_l(\gamma a), & r \le r_{\text{core}}, \\ K_l(\beta r)/K_l(\beta a), & r_{\text{core}} < r \le r_{\text{clad}}. \end{cases} \tag{4.15}$$

Since the $\cos(l\theta) \equiv 1$ for $l = 0$, the LP_{0p} modes are radially symmetric. For example, Figure 4.1 shows the transverse profile of the fundamental mode (LP_{01}) in a weakly-guiding step-index fiber.

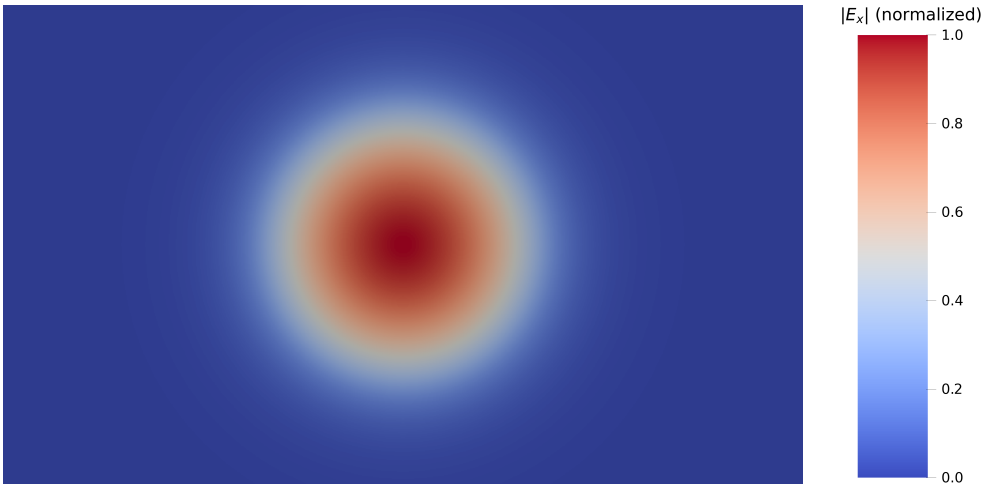


Figure 4.1: Transverse guided mode profile (magnitude of the electric field). The fundamental mode (LP_{01}) in a weakly-guiding step-index fiber is radially symmetric.

4.1.2 Optical power

The optical power in a waveguide is one of the essential quantities of interest. In particular, when studying (nonlinear) gain amplification in a fiber laser (Chapter 5), we model the transfer of optical power from one field to another and want to measure the power in each field

at any point (z) along the length of the fiber. The average power flow of the electromagnetic field is given by the real part of the time-averaged Poynting vector:

$$\boldsymbol{S} := \boldsymbol{E} \times \boldsymbol{H}^*, \tag{4.16}$$

where $\boldsymbol{H}^*$ denotes the complex conjugate of $\boldsymbol{H}$. For a fixed position $0 < \bar{z} < L$, let $\Omega_t(\bar{z}) := \{(x, y, \bar{z}) : x^2 + y^2 < r_{\text{clad}}^2\}$ denote the transverse domain of the fiber, and let $\hat{\boldsymbol{n}}$ be the normal vector on Ω_t (i.e., $\hat{\boldsymbol{n}} = \hat{\boldsymbol{e}}_{\boldsymbol{z}}$). We assume that the power flow in the waveguide is in the forward direction (guided wave assumption), and thus the power is calculated by

$$P(\bar{z}) := \left| \int_{\Omega_t(\bar{z})} \hat{\boldsymbol{n}} \cdot \mathfrak{Re}\{\boldsymbol{S}\} dxdy \right|. \tag{4.17}$$

Recall that the broken ultraweak Maxwell formulation has trace unknowns $\hat{\boldsymbol{E}}$ and $\hat{\boldsymbol{H}}$ that are defined on the mesh skeleton Γ_h. To be precise, these trace unknowns are defined in the energy space $H^{-1/2}(\text{curl}, \Gamma_h)$, thus for every element $K \in \Omega_h$, the duality pairing

$$\langle \hat{\boldsymbol{n}} \times \hat{\boldsymbol{E}}, \hat{\boldsymbol{H}} \rangle_{H^{-1/2}(\text{div}, \partial K) \times H^{-1/2}(\text{curl}, \partial K)} \tag{4.18}$$

is well-defined. Since we assume that the power flow is in the forward direction, contributions from faces that are not orthogonal to the fiber axis ($\hat{\boldsymbol{e}}_{\boldsymbol{z}}$) vanish and the computation of the power can be interpreted as a duality pairing on fiber cross-sections:

$$P(\bar{z}) = \left| \int_{\Omega_t(\bar{z})} \hat{\boldsymbol{n}} \cdot \mathfrak{Re}\{\hat{\boldsymbol{E}} \times \hat{\boldsymbol{H}}^*\} dxdy \right| = \left| \int_{\Omega_t(\bar{z})} \mathfrak{Re}\{(\hat{\boldsymbol{n}} \times \hat{\boldsymbol{E}}) \cdot \hat{\boldsymbol{H}}^*\} dxdy \right|. \tag{4.19}$$

Therefore, the power flow along the fiber is computed directly from the solution to the Maxwell problem without additional post-processing.

Confinement ratio. Another quantity of interest in our simulations is the *confinement ratio*, denoted by Γ. The confinement ratio is defined as the fraction of the energy of the propagating wave that is confined to the core of the fiber. In general, a guided mode is more tightly confined to the core if the normalized frequency (V-number) is much larger than the mode's cutoff frequency (i.e., $V \gg V_c$). As V approaches V_c, the cladding fields become more significant. For an LP_{lp} mode that is polarized along x, the confinement ratio is given by:

$$\Gamma_{lp} := \frac{P_{lp}^{\text{core}}}{P_{lp}^{\text{total}}} = \frac{\int_0^{r_1} |E_x|^2 r \; dr}{\int_0^{r_2} |E_x|^2 r \; dr}. \tag{4.20}$$

The mode confinement ratio plays an important role for discretizing the transverse domain. This point is investigated in the multi-mode fiber adaptivity study in Section 4.3. Additionally, the confinement ratio of the propagating field indicates how the power is distributed in the fiber cross-section (more power outside the fiber core indicates that higher-order modes are carrying more energy). Consider the transverse domain of the fiber core: $\Omega_t^{\text{core}}(\bar{z}) := \{(x, y, \bar{z}) : x^2 + y^2 < r_{\text{core}}^2\}$. Then, using the computation of optical power in the waveguide, we can calculate the confinement ratio of the propagating field at a fixed position $0 < \bar{z} < L$ by

$$\Gamma(\bar{z}) = \left| \int_{\Omega_t^{\text{core}}(\bar{z})} \mathfrak{Re}\{(\hat{\boldsymbol{n}} \times \hat{\boldsymbol{E}}) \cdot \hat{\boldsymbol{H}}^*\} dxdy \right| / \left| \int_{\Omega_t(\bar{z})} \mathfrak{Re}\{(\hat{\boldsymbol{n}} \times \hat{\boldsymbol{E}}) \cdot \hat{\boldsymbol{H}}^*\} dxdy \right|. \tag{4.21}$$

4.1.3 Mode projection

Given a time-harmonic electromagnetic field in a weakly-guiding circularly symmetric step-index fiber, we want to determine how much of the power is in each of the guided modes. This section shows how the a-priori knowledge of the fiber modes can be utilitized to compute the power of each propagating mode. Suppose we have a (multi-mode) fiber operated at a normalized frequency of $V \approx 4.43$. From Table 4.1, we know that the fiber supports four guided modes: $\{\mathrm{LP}_{01}, \mathrm{LP}_{11}, \mathrm{LP}_{21}, \mathrm{LP}_{02}\}$ (with one additional rotation each for the asymmet-

ric modes LP_{11} and LP_{21}). As usual, the fiber axis is assumed to be aligned with the z-axis, the length of the fiber is L, and for a fixed position $0 < \bar{z} < L$, we denote the transverse domain of the fiber by $\Omega_t(\bar{z})$, and the respective core and cladding transverse domains by $\Omega_t^{\text{core}}(\bar{z})$ and $\Omega_t^{\text{clad}}(\bar{z})$. The eigenmode solutions to the transverse Helmholtz problem (4.9)–(4.10) form an L^2-orthogonal basis on the transverse domain $\Omega_t(\bar{z})$ for any $0 < \bar{z} < L$. Therefore, any (guided) propagating field $\boldsymbol{E}(x, y, \bar{z})$ is approximately a linear combination of the (guided) modes of the fiber. Approximately, because the Helmholtz eigenvalue problem is derived with certain simplifying assumptions that are not exactly true for the real step-index fiber. In other words,

$$\boldsymbol{E}(x, y, \bar{z}) \approx \sum_{i=1}^{N} \left[\boldsymbol{A}_m \cdot \boldsymbol{\varphi}_m(x, y) e^{-ik_m \bar{z}} \right] \ , \tag{4.22}$$

where N is the number of guided modes, $\boldsymbol{A}_m$ the amplitude (vector) of the m-th mode, $\boldsymbol{\varphi}_m$ the (normalized) eigenmode, and k_m the propagation constant of the mode. For a guided mode, k_m is a real number. In the multi-mode fiber with $V \approx 4.43$, $N = 4$. By orthogonality of the eigenmodes,

$$(\boldsymbol{\varphi}_m, \boldsymbol{\varphi}_n) = \int_{\Omega_t} \boldsymbol{\varphi}_m \cdot \boldsymbol{\varphi}_n^* \, dxdy = \begin{cases} 0, & m \neq n \ , \\ 1, & m = n \ . \end{cases} \tag{4.23}$$

Theorem 4.1.1. (Hilbert space projection) [5, Thm. 3.14] *Suppose $\mathcal{H}$ is a (complex) Hilbert Space equipped with the inner product $(\cdot, \cdot)$, and $\{\boldsymbol{u}_1, \ldots, \boldsymbol{u}_N\} \subset \mathcal{H}$ is an orthonormal set. Let $\boldsymbol{x} \in \mathcal{H}$. Then, the orthogonal projection of $\boldsymbol{x}$ onto $\mathcal{M} = span\{\boldsymbol{u}_1, \ldots, \boldsymbol{u}_N\}$ is given by:*

$$\mathcal{P}_{\mathcal{M}} \boldsymbol{x} = \sum_{i=1}^{N} (\boldsymbol{x}, \boldsymbol{u}_i) \boldsymbol{u}_i \ . \tag{4.24}$$

By the Hilbert space projection theorem, the orthogonal projection of the transverse

field $\boldsymbol{E}(x, y, \bar{z})$ onto mode $\boldsymbol{\varphi}_m$ is

$$\left[\int_{\Omega_t(\bar{z})} \boldsymbol{E}(x, y, \bar{z}) \cdot \boldsymbol{\varphi}_m^* \, dxdy\right] \boldsymbol{\varphi}_m(x, y) \equiv \alpha_m \boldsymbol{\varphi}_m \ , \tag{4.25}$$

where α_m is the (complex) projection coefficient.

Recall that we can calculate the power flow of the electromagnetic field through the transverse domain $\Omega_t(\bar{z})$ by (4.19). Suppose we want to calculate how much of the power resides in the m-th mode at $0 < \bar{z} < L$. Let $P_m(\bar{z})$ denote the power of the orthogonal projection of the field onto the m-th mode. Then,

$$\begin{aligned} P_m(\bar{z}) &= \left| \int_{\Omega_t(\bar{z})} \hat{\boldsymbol{n}} \cdot \mathfrak{Re}\left\{ \alpha_m \boldsymbol{\varphi}_m(x, y, \bar{z}) \times \left[-\frac{1}{i\omega\mu} \nabla \times (\alpha_m \boldsymbol{\varphi}_m) \right]^* (x, y, \bar{z}) \right\} dxdy \right| \\ &= |\alpha_m|^2 \left| \int_{\Omega_t(\bar{z})} \hat{\boldsymbol{n}} \cdot \mathfrak{Re}\left\{ \boldsymbol{\varphi}_m(x, y, \bar{z}) \times \frac{1}{i\omega\mu} \nabla \times \boldsymbol{\varphi}_m^*(x, y, \bar{z}) \right\} dxdy \right| \ , \end{aligned} \tag{4.26}$$

where we used Faraday's law, $\nabla \times \boldsymbol{E} = -i\omega\mu \boldsymbol{H}$. Based on (4.26), the power that resides in each of the guided modes for a given propagating field can be computed.

Numerical experiments and verification. For convenience, we restrict ourselves to injecting x-polarized electric fields only; consequently, only the x-polarized electric fields of the guided modes are considered. We denote the (unnormalized) guided modes by

$$\boldsymbol{\psi}_m = A_m \boldsymbol{\varphi}_m, \quad m = \{1, 2, 3, 4\} \equiv \{\mathrm{LP}_{01}, \mathrm{LP}_{11}, \mathrm{LP}_{21}, \mathrm{LP}_{02}\} \ . \tag{4.27}$$

Remark: LP_{lp} modes with $l > 0$ have two possible rotations in each polarization state (see Appendix B.2). We show results for one of the rotations of the LP_{11} and LP_{21} modes.

L^2 norm of the unnormalized eigenmodes. For the (unnormalized) LP modes, defined in (4.15), we can directly calculate $\|\boldsymbol{\psi}_{lp}\|$ by

$$\|\boldsymbol{\psi}_{lp}\| = \left[\int_0^{2\pi} \cos^2(l\theta)d\theta \left(\frac{\int_0^{r_1} J_l^2(\gamma_{lp}r)rdr}{J_l^2(\gamma_{lp}r_1)} + \frac{\int_{r_1}^{r_2} K_l^2(\beta_{lp}r)rdr}{K_l^2(\beta_{lp}r_1)}\right)\right]^{1/2}. \tag{4.28}$$

We obtain: $\|\boldsymbol{\psi}_1\| \approx 5.74447$, $\|\boldsymbol{\psi}_2\| \approx 2.63182$, $\|\boldsymbol{\psi}_3\| \approx 2.09447$, and $\|\boldsymbol{\psi}_4\| \approx 3.15927$. Numerically, these norms are computed by integrating over the element boundaries $\{\Omega_t^f\}_{f=1}^F$ that discretize the transverse domain Ω_t, i.e.,

$$\|\boldsymbol{\psi}_m\| = \left[\int_{\Omega_t} \boldsymbol{\psi}_m \cdot \boldsymbol{\psi}_m^*\right]^{1/2} = \left[\sum_{f=1}^{F} \int_{\Omega_t^f} \boldsymbol{\psi}_m \cdot \boldsymbol{\psi}_m^*\right]^{1/2}. \tag{4.29}$$

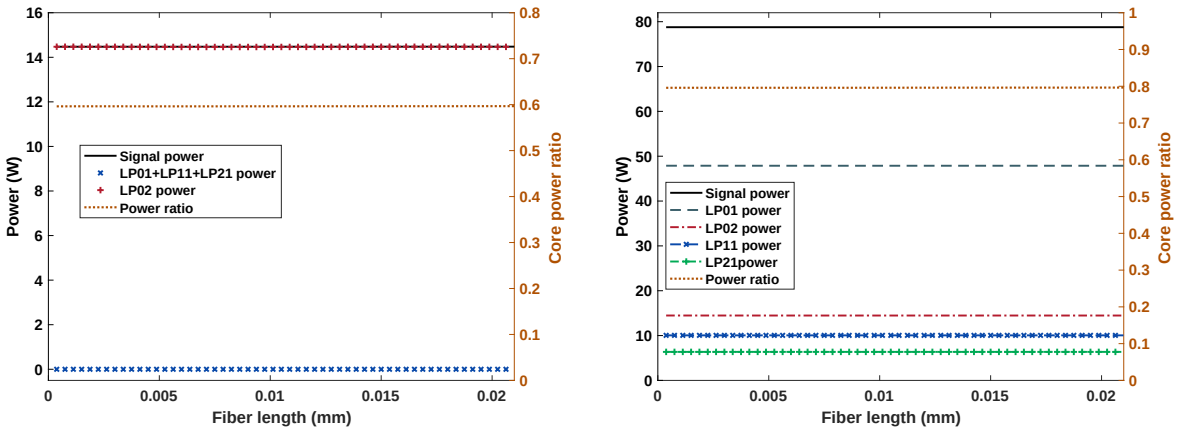
The numerically calculated values for the discretized modes (on the initial geometry) are: $\|\tilde{\boldsymbol{\psi}}_1\| \approx 5.74447$, $\|\tilde{\boldsymbol{\psi}}_2\| \approx 2.63180$, $\|\tilde{\boldsymbol{\psi}}_3\| \approx 2.09443$, and $\|\tilde{\boldsymbol{\psi}}_4\| \approx 3.15926$. As the mesh is further refined in the radial direction, the numerical values approach the analytically computed values for the exact modes.

Linear combination of guided modes. If exactly one of the guided modes is injected into the fiber, say $\boldsymbol{\psi}_m$, then the coefficients take values $\alpha_m = 1, \alpha_n = 0, n \neq m$. In this case, the power $P(\bar{z})$ of the field should be equal to the power $P_m(\bar{z})$ of the orthogonal projection onto $\boldsymbol{\psi}_m$, while the powers $P_n(\bar{z}), n \neq m$, should equal zero. The numerical solutions for the discretized modes are shown in Table 4.2.

Injected mode	Projection coefficient			
	$\tilde{\alpha}_1$	$\tilde{\alpha}_2$	$\tilde{\alpha}_3$	$\tilde{\alpha}_4$
$\boldsymbol{\psi}_1$	1.00	10^{-7}	10^{-5}	10^{-5}
$\boldsymbol{\psi}_2$	10^{-10}	1.00	10^{-10}	10^{-10}
$\boldsymbol{\psi}_3$	10^{-6}	10^{-8}	1.00	10^{-5}
$\boldsymbol{\psi}_4$	10^{-5}	10^{-7}	10^{-5}	1.00

Table 4.2: Projection coefficients of the numerical solution for each guided mode. The obtained values are accurate for all tested modes.

Figure 4.2 shows the numerical solution for the power of the projected field for each guided mode in two scenarios: in Figure 4.2a, the fiber was excited with the LP_{02} mode, and in Figure 4.2b, the fiber was excited with a combination of all four guided LP modes. In the case of a single mode, the entire energy (referred to as the power of the signal in the plots) projects onto the LP_{02} mode, as expected from Table 4.2. In the case of the linear combination of injected modes, the sum of the power in the projections is equal the total power (i.e., the projection coefficients sum up to 1). The confinement ratio Γ is also shown in both plots ("core power ratio"): in the case of the single LP_{02} mode, the confinement ratio is $\Gamma_{02} = 59.58\%$, while for the multi-mode propagation, the ratio is a weighted sum (weighted with the squared projection coefficients) of the ratios $\Gamma_{lp},\ (l,p) \in \{(0,1),(1,1),(2,1),(0,2)\}$. In both cases, the numerical results are accurate.



(a) Injecting the LP_{02} mode (b) Injecting a combination of guided modes

Figure 4.2: Computation of the optical power projected onto modes for a guided field in a fiber of 32 wavelengths. The numerical results for the projected power are accurate when the fiber is excited with a linear combination of guided modes.

Exciting the fiber with a symmetric field. Next, the fiber is excited with a field that is not composed of guided modes only. Consider the following (electric) field as the fiber

input:

$$\boldsymbol{E}_{\tilde{r}} = (E_x, 0, 0)_{\tilde{r}} = e^{-i\omega z} \begin{cases} 1 & , \ r \leq \tilde{r}, \\ e^{-(r-\tilde{r})^2} & , \ r > \tilde{r}, \end{cases}$$

where $r_{\text{core}} < \tilde{r} < r_{\text{clad}}$. Because the field $\boldsymbol{E}_{\tilde{r}}$ is circularly symmetric, we expect most of the energy to be projected onto the circularly symmetric modes LP_{01}, LP_{02}. Indeed, the analytical calculations show, for $\tilde{r} \approx r_{\text{core}}/0.9$,

$$\{\alpha_{01}, \alpha_{11}, \alpha_{21}, \alpha_{02}\} \approx \{2.52567, 0.00, 0.00, 2.09587\}, \tag{4.30}$$

$$\{P_{01}, P_{11}, P_{21}, P_{02}\}/P \approx \{47.53, 0.00, 0.00, 32.73\}\% \ . \tag{4.31}$$

Therefore, approximately 80.26% of the field's power is captured by the guided modes while the remaining energy should occupy either evanescent (decaying) modes or (guided) cladding modes. The discrete numerical solution yields the following approximations:

$$\{\tilde{\alpha}_{01}, \tilde{\alpha}_{11}, \tilde{\alpha}_{21}, \tilde{\alpha}_{02}\} \approx \{2.52563, 10^{-8}, 10^{-4}, 2.09586\}, \tag{4.32}$$

$$\{\tilde{P}_{01}, \tilde{P}_{11}, \tilde{P}_{21}, \tilde{P}_{02}\}/\tilde{P} \approx \{47.52, 0.00, 0.00, 32.71\}\% \ . \tag{4.33}$$

Figure 4.3 shows the power of the orthogonal projections onto the guided modes. Clearly, the LP_{01} and LP_{02} mode carry most of the total energy, but a significant amount of power ($\approx 20\%$) is not captured by any of the guided modes. If the remaining power resides in cladding modes, it propagates similar to the core-guided modes without loss of energy in the simulation (except for numerical errors).

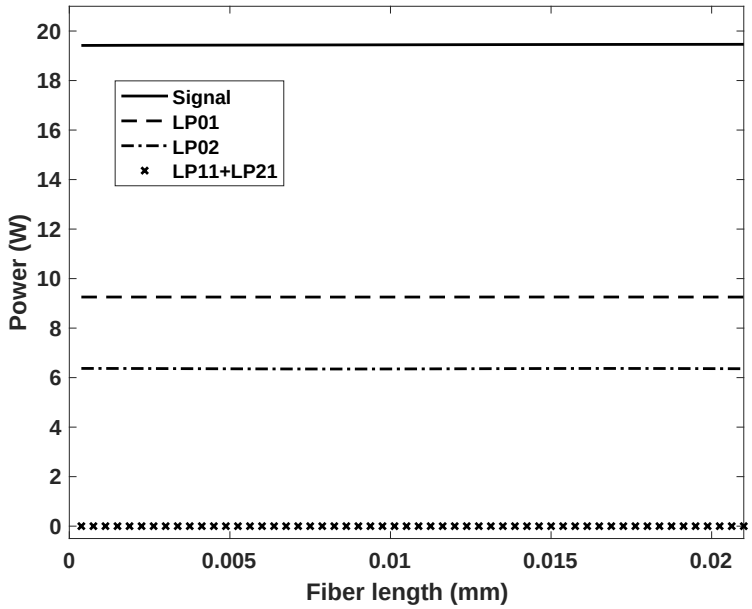


Figure 4.3: Computation of the optical power projected onto modes for an unguided field in a fiber of 32 wavelengths. The numerical results accurately reflect that only some of the energy is captured by guided modes when the fiber is excited with a (symmetric) field that is not composed of guided modes only.

4.2 Numerical pollution in long waveguide simulations

It is well-known that an accurate numerical solution for wave problems with high frequency is difficult to obtain. Unfortunately, the finite element discretization for these problems suffers from significant numerical pollution errors that increase with the wavenumber [9]. It is critical to control these errors to obtain a stable and accurate method. Section 1.3.3 discusses previous work on mitigating the pollution effect and some recent advances in developing robust solution schemes for high-frequency wave problems. However, to the best of our knowledge, numerical studies in three dimensions have mostly been limited to acoustic wave problems with a moderate number of wavelengths. As mentioned in Section 1.3.3, the DPG method can circumvent the stability problem and deliver a robust discretization for any wavenumber [39], but it does not eliminate the numerical pollution error in multiple dimensions. In this section, we discuss how the frequency ω enters the DPG error analysis,

and we report numerical results for the propagation of the fundamental mode in a rectangular waveguide. In particular, we show results for the DPG method applied to the 3D vectorial time-harmonic Maxwell problem in waveguides with more than 8 000 wavelengths, using high order of approximation. Our results corroborate previous analysis for the Galerkin discretization of the Helmholtz and Maxwell operators by Melenk and Sauter [89, 90].

4.2.1 Pollution estimates

The mathematical setting for different variational formulations of the time-harmonic Maxwell equations in context of DPG is analyzed in much detail in [20]. We recap a few points that are relevant for our discussion regarding the ultraweak formulation. Recall from Section 2.5.1 that, defining group variables $\mathfrak{u} = (\boldsymbol{E}, \boldsymbol{H}), \mathfrak{v} = (\boldsymbol{F}, \boldsymbol{G})$, and $\hat{\mathfrak{u}} = (\hat{\boldsymbol{E}}, \hat{\boldsymbol{H}})$, the broken ultraweak Maxwell problem (2.42) can be written as:

$$\begin{cases} \mathfrak{u} \in \mathcal{U}, \hat{\mathfrak{u}} \in \hat{\mathcal{U}}, \\ b(\mathfrak{u}, \mathfrak{v}) + \langle \hat{\mathfrak{u}}, \mathfrak{v} \rangle_{\Gamma_h} = l(\mathfrak{v}), \quad \mathfrak{v} \in \mathcal{V}(\Omega_h), \end{cases} \tag{4.34}$$

where the load $l(\mathfrak{v}) = 0$, $b(\mathfrak{u}, \mathfrak{v}) = (\mathfrak{u}, A_h^* \mathfrak{v})$, and $\langle \hat{\mathfrak{u}}, \mathfrak{v} \rangle_{\Gamma_h}$ is given by (2.51). In the ultraweak formulation with conforming test functions, the optimal test norm is the adjoint norm: $\|\mathfrak{v}\|_{\mathcal{V}} = \|A^* \mathfrak{v}\|$; with this norm, the method delivers the L^2 projection. For the broken formulation (2.54), the optimal test norm must be augmented with an additional term, and we obtain a quasi-optimal test norm—the adjoint graph norm: $\|\mathfrak{v}\|^2_{\mathcal{V}(\Omega_h)} = \|A_h^* \mathfrak{v}\|^2 + \alpha \|\mathfrak{v}\|^2$, with scaling parameter $\alpha \in \mathcal{O}(1)$ (see Section 2.5.1 for additional details). The adjoint graph norm is robustly equivalent with the optimal test norm, i.e., independent of the frequency ω, and the robust stability constant is maintained in the broken formulation [20]. This implies that the approximation error is bounded by the best approximation error (BAE) uniformly in ω [39]:

$$\|\mathfrak{u} - \mathfrak{u}_h\|^2 + \|\hat{\mathfrak{u}} - \hat{\mathfrak{u}}_h\|^2_{\hat{\mathcal{U}}} \leq C \left[\inf_{\mathfrak{w}_h} \|\mathfrak{u} - \mathfrak{w}_h\|^2 + \inf_{\hat{\mathfrak{w}}_h} \|\hat{\mathfrak{u}} - \hat{\mathfrak{w}}_h\|^2_{\hat{\mathcal{U}}} \right], \tag{4.35}$$

where constant C is independent of the mesh and frequency ω, and $\|\cdot\|_{\hat{\mathcal{U}}}$ refers to the minimum energy extension norm defined in (2.53). The estimate (4.35) implies that the L^2 best approximation is pollution free because it is independent of ω. In one dimension, the BAE for the traces is zero, thus the method is in fact pollution free [127, 39, 103]. In multiple dimensions, however, the BAE for the traces is measured in the operator-dependent minimum energy extension norm $\|\cdot\|_{\hat{\mathcal{U}}}$, and this norm does depend on the frequency ω. In other words, the projection in the minimum energy extension norm is *not* pollution free, hence the method exhibits numerical pollution.

The estimate for the standard Galerkin method on the other hand has a stability constant that is not ω-independent, thus the Galerkin discretization is not robustly stable. The DPG method hides the perturbation parameter ω in the best approximation and by doing so yields a *stable discretization for any wavenumber*. This can practically be exploited by starting computation on a coarse mesh where the pollution error is high, and driving *hp*-adaptivity with the DPG error indicator. This approach yields superior meshes for resolving localized waves [102, 104].

A wavenumber explicit analysis for the Helmholtz equation is presented for the DPG method in [39] and for the Galerkin method in [89]. For the Galerkin discretization, Melenk and Sauter show that quasi-optimality is obtained under the conditions that $\omega h/p$ is sufficiently small and p is at least $\mathcal{O}(\log\omega)$, where h is the mesh size and p the polynomial order of approximation [89]. A similar estimate was shown for the Maxwell operator [90]. Based on these estimates, the best approach to counter the pollution error may be an *hp*-strategy that preferably increases the polynomial order p while keeping ωh constant for increasing frequency.

In the next section, we study the pollution error with numerical experiments for many wavelengths and discuss the observations with regard to the suggested *hp*-strategy and its applicability to the DPG method for the time-harmonic Maxwell problem.

4.2.2 Numerical results

The propagation of an electromagnetic field in a waveguide is governed by the Maxwell equations. We assume that the time-harmonic setting is justified and that the waveguide medium is nonmagnetic, dielectric, and no free charges are present. While nonlinear effects and anisotropic, inhomogeneous material properties play important roles in research on fiber optics, for the purpose of this pollution study we assume the waveguide medium is linear, isotropic and homogeneous. We prescribe inhomogeneous Dirichlet boundary conditions (BCs) at the waveguide input, to excite the waveguide, and impedance BCs at the output, to absorb the wave, but impose PEC BCs everywhere else. For this simplified setting, the propagating field in a waveguide can be described as a superposition of guided modes. Consider the following rectangular waveguide domain Ω, in Cartesian coordinates:

$$\Omega = (0.0, 1.0) \times (0.0, 0.5) \times (0, L),$$

where L is the length of the waveguide. The fundamental mode in this waveguide is the transverse electric TE_{10} mode, depicted in Figure 4.4. The fundamental mode is not very oscillatory in the transverse direction. At the waveguide end, we employ an absorbing impedance boundary condition that matches the wave impedance for the fundamental mode. In the rectangular waveguide experiments, the cross-section is modeled with two hexahedral elements, which is justified by the simple transverse mode profile (cf. Figure 4.4).

Uniform polynomial order p. In our first experiment, we analyze the relative field error, measured in the L^2 norm, for the propagating fundamental mode in waveguides of different length L. The smallest waveguide has a length equivalent to one wavelength of the fundamental mode, and the longest one has 8 192 wavelengths. As we increase the length L, we keep the number of elements per wavelength (i.e., degrees of freedom (DOFs) per wavelength) constant. In particular, we choose a discretization with four elements per

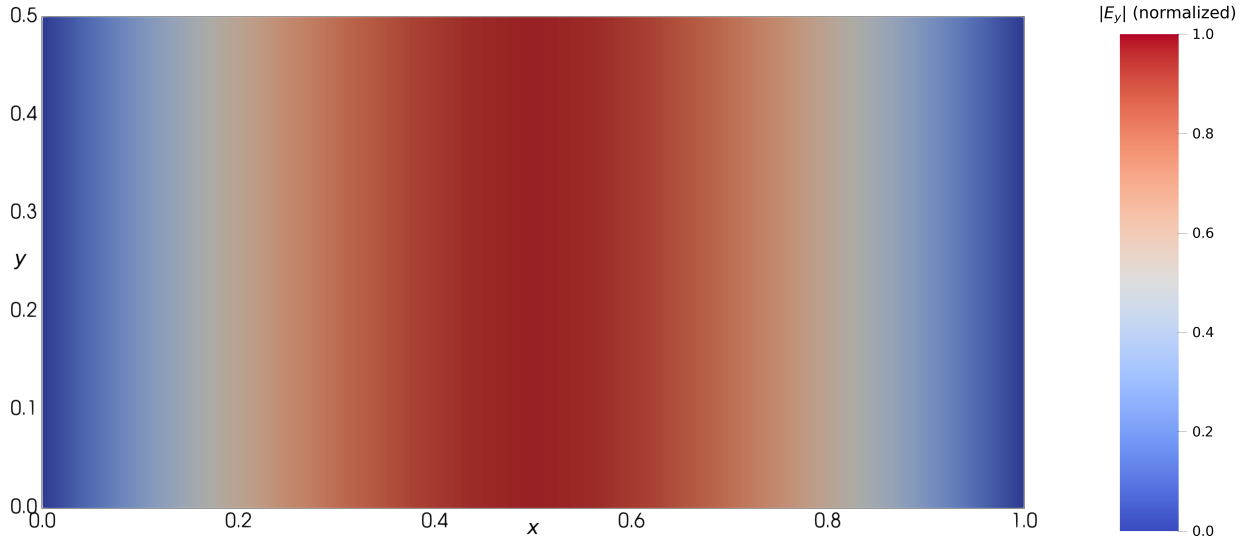


Figure 4.4: TE_{10} transverse electric field in a rectangular waveguide in a plane normal to the z-axis. The simple transverse profile of the fundamental mode justifies a geometry discretization with few elements in the cross-section.

wavelength. Figure 4.5 shows the relative field error for these waveguides for uniform order of approximation, ranging from $p = 4$ to $p = 8$. In all numerical experiments, we are using the enrichment order $\Delta p = 1$ for the test space to approximate optimal test functions.

We make several observations: 1) for a fixed number of wavelengths, higher polynomial order yields significantly smaller (more than one order of magnitude) errors, as expected; 2) for every order of approximation, the field error starts to increase if the waveguide is long enough despite keeping the DOFs per wavelength constant; and 3) for higher p, this pollution effect is "kicking in" at a later point, i.e., more wavelengths can be computed with higher order before the pollution error is measurable. Furthermore, to maintain some desired accuracy, one needs to increase the polynomial order in nearly regular intervals. For example, to achieve 1% accuracy for 4 wavelengths, it is sufficient to use $p = 4$; at 64 wavelengths, $p = 5$ is needed; with $p = 6$, computing up to 1 024 wavelengths is feasible with this error margin; and $p = 7$ would most likely be sufficient for 16 384 wavelengths. At a closer look, these intervals resemble a logarithmic dependency on the polynomial order ($4 * 2^4 = 64,\ 64 * 2^4 = 1\,024,\ 1\,024 * 2^4 = 16\,384$). In other words, these results corroborate

theoretical estimates by Melenk and Sauter predicting that control of the pollution error would require increasing p logarithmically with the wavenumber.

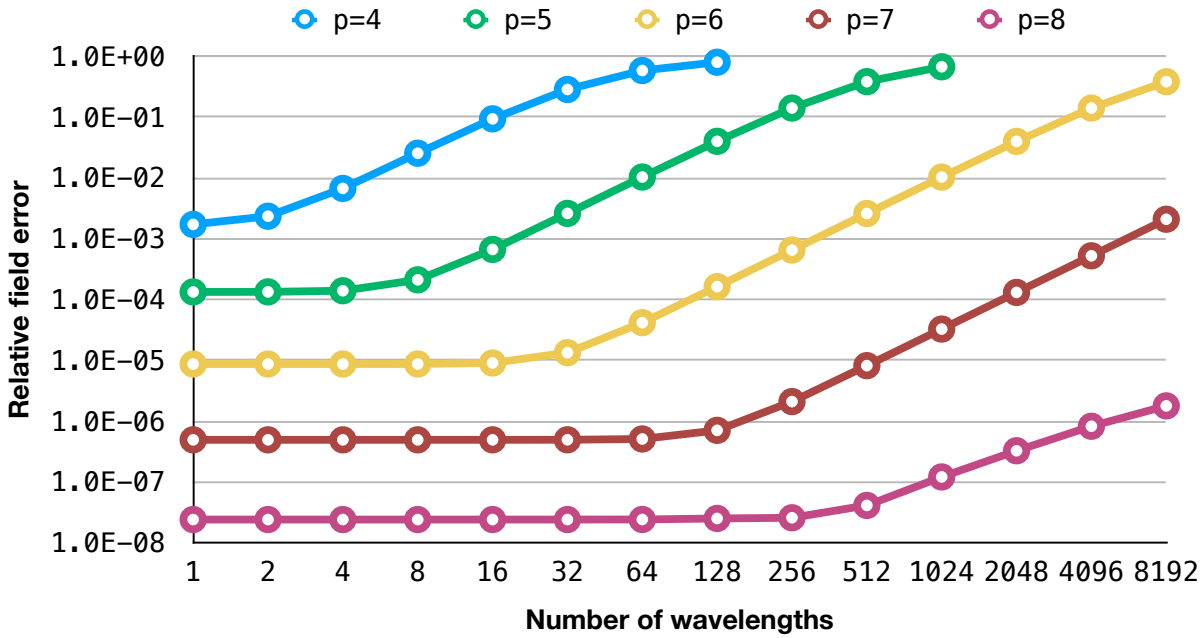


Figure 4.5: Relative field error with uniform order of approximation. As the number of wavelengths is increased, we keep the ratio of elements per wavelength constant. To counter the pollution error, the polynomial order of approximation must be increased logarithmically with the number of wavelengths.

In our numerical experiments, the pollution manifests itself primarily as a diffusive error causing wave attenuation. This is in agreement with previous observations for the DPG method [39]. A practical way of measuring this diffusivity in waveguide applications is to compute the power flux through the cross-section of the waveguide at different points in z. In a linear, dielectric waveguide with PEC boundary conditions, the fundamental mode should ideally be carried without loss of power. Figure 4.6 shows the measured power loss between the waveguide input ($z = 0$) and output ($z = L$) for different polynomial orders. Note that $p = 8$ has less than 0.005% loss of power in all tested waveguides. The pollution error is clearly visible in terms of power loss. We also observe the same logarithmic dependency for increasing polynomial order, illustrated by the near-equidistant parallel character of the lines.

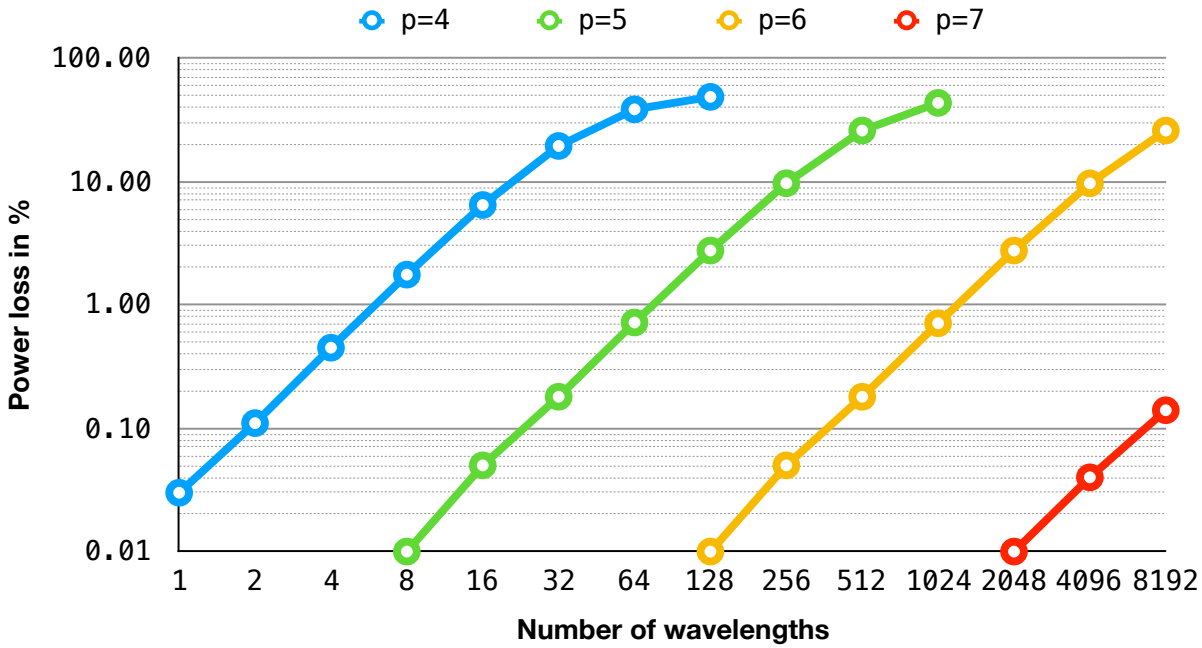


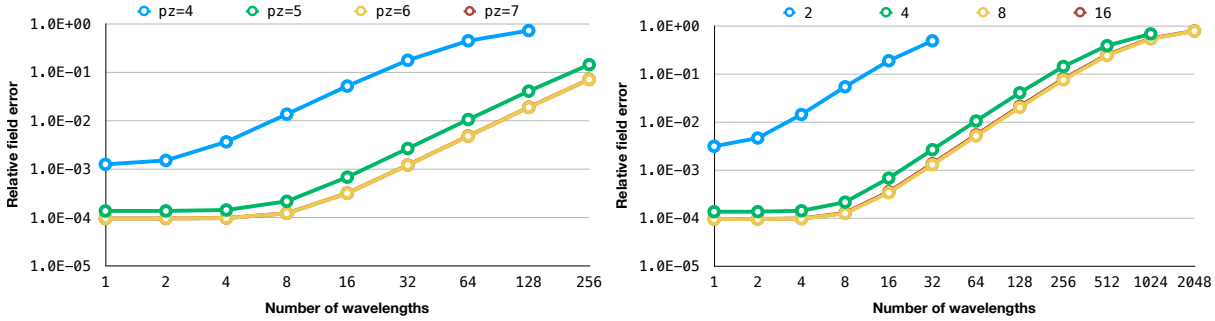
Figure 4.6: Power loss with uniform order of approximation. In the DPG discretization, we observe that the pollution error has primarily a diffusive effect. This implies that the power flux, measured perpendicular to the waveguide cross-section, decreases along the waveguide. The depicted quantity is the loss of power between the input and output for waveguides of different lengths.

Anisotropic polynomial order (p_x, p_y, p_z) and element size h. Moving on to additional experiments, we keep our focus on the same rectangular waveguide but with different potential approaches of dealing with the pollution error. It may be reasonable to assume that since the wave is propagating in one direction (along z), it will be sufficient to increase the order of approximation anisotropically or to increase the number of elements through anisotropic h-refinements in z. Exploring both of these options (cf. Figure 4.7), we find that neither one of these approaches yields satisfactory results. First, in Figure 4.7a, we use fifth-order polynomials in the radial discretization ($p_x = p_y = 5$) of the waveguide and increase the anisotropic order from $p_z = 4$ to $p_z = 7$. While the error decreases initially, it begins stagnating at $p_z = 6$ (note that the $p_z = 7$ error coincides almost exactly with $p_z = 6$). The same observation is made for uniform order $p = 5$ with varying number of elements per wavelength (ranging from 2 elements to 16 elements). Our findings indicate that the pollution error depends on the interplay between the mode resolution (radial discretization) and

the wave resolution in the direction of propagation. In other words, *increasing the number of DOFs anisotropically does not suffice asymptotically to control the pollution error.*

Finally, we measure the loss of power for both anisotropic refinement cases, plotted in Figure 4.8. As expected, we observe the same stagnation in the diffusive pollution, consistent with the errors measured in the previous plot.

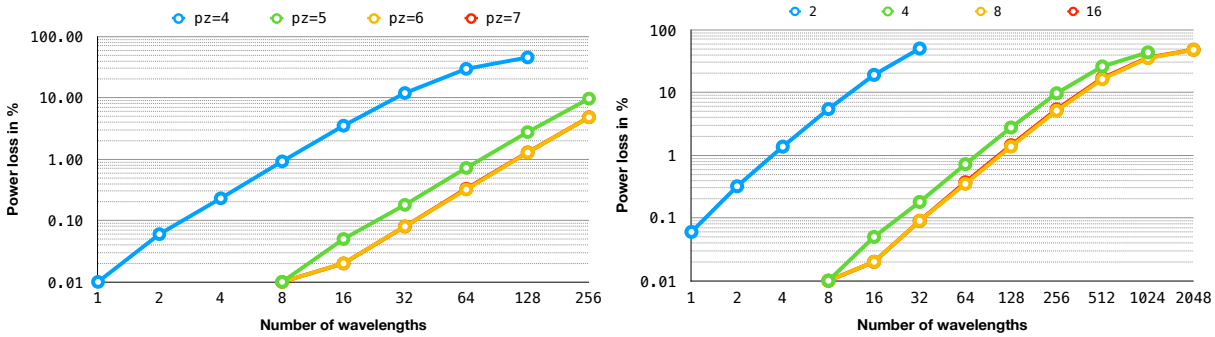
We have conducted these experiments on different waveguides (rectangular waveguides, circular waveguides, and step-index fibers) with various propagating modes, and the observations are all consistent with the observations presented up to here; we therefore omit showing additional numerical results for those cases.



(a) Anisotropic approximation order: $p_x = p_y = 5$, $p_z \in \{4, 5, 6, 7\}$

(b) Anisotropic element size: $h_x = h_y = 0.5$, $h_z \in \lambda/\{2, 4, 8, 16\}$

Figure 4.7: Relative field error with anisotropic refinements. Increasing the number of DOFs anisotropically via h- or p-refinements in the direction of wave propagation does not suffice to control the pollution error in our experiments.



(a) Anisotropic approximation order: $p_x = p_y = 5$, $p_z \in \{4, 5, 6, 7\}$

(b) Anisotropic element size: $h_x = h_y = 0.5$, $h_z \in \lambda/\{2, 4, 8, 16\}$

Figure 4.8: Power loss with anisotropic refinements. The diffusive pollution effect is not countered through anisotropic h- or p-refinements.

4.3 Adaptivity for higher-order modes

In our adaptivity study, we focus on a different aspect of resolving the propagating wave. We have shown that the interplay between resolving the wave along the direction of propagation and resolving the transverse mode profile is important in controlling the pollution error. For that reason, the finite element mesh should be sensitive to different mode profiles and adapt to resolve them appropriately. This is especially important in waveguide applications where significant transfer of power occurs between different guided modes. Our tool for adapting the mesh "on-the-fly" is the DPG residual that serves as an error indicator in the energy norm (2.20). Recall that the DPG method can be reformulated as a minimum residual method (2.13) where the residual is minimized in the dual test norm. Therefore, the Riesz representation of the residual, defined in (2.17), is the error measured in the energy norm. We proceed with numerical experiments in multi-mode step-index fibers and also look at the load imbalance that results from adapting the mesh to different propagating modes.

Multi-mode fiber. We consider a dielectric optical waveguide. More precisely, we assume that the waveguide is a weakly-guiding, large-mode-area (LMA), step-index fiber made of silica glass. See Table A.5 for a description of the model parameters for the fiber. For weakly-guiding fibers, $(n_{\text{core}} - n_{\text{clad}})/n_{\text{core}} \ll 1$, and the guided modes are linearly polarized (LP) modes. The V-number is a "normalized frequency" that determines how many guided modes are supported by the particular fiber. For example, if $V < 2.405$, then the fiber is single-mode. LMA fibers have a relatively large core radius and support multiple modes. The fiber we consider has a normalized frequency of $V \approx 4.43$; thus, it supports four guided modes: $\{\text{LP}_{01}, \text{LP}_{11}, \text{LP}_{21}, \text{LP}_{02}\}$. The fiber axis is assumed to be aligned with the z-axis, and the length of the fiber is L. Figure 4.9 illustrates the guided modes for this particular fiber, showing the magnitude of the electric field in the center of the fiber cross-section. For multi-mode propagation, we employ an absorbing PML at the fiber output (cf. Section 2.5.2).

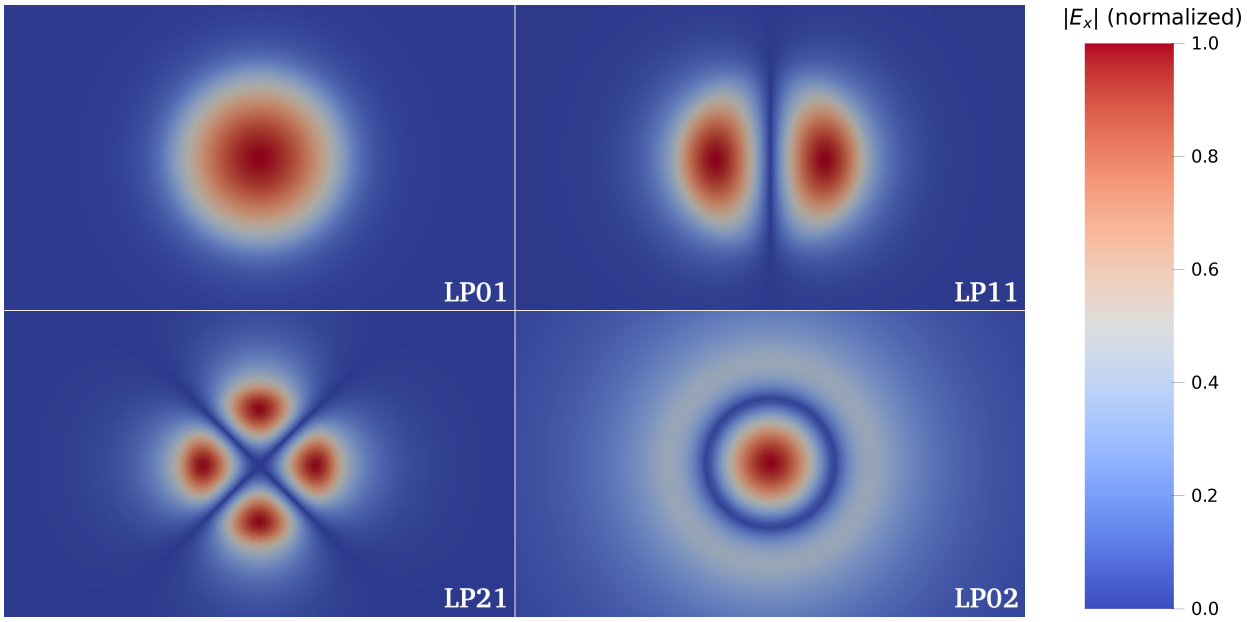


Figure 4.9: Guided modes in LMA fiber (magnitude of the electric field). Higher-order modes are more oscillatory in the transverse direction and carry more energy outside the fiber core. Therefore, they require additional refinements of the fiber cross-section discretization.

For this fiber, the mode confinement (amount of energy confined to the core region) of each mode is (cf. (4.20))

$$\{\Gamma_{01}, \Gamma_{11}, \Gamma_{21}, \Gamma_{02}\} \approx \{96.11, 88.77, 74.79, 59.58\}\%. \tag{4.36}$$

Clearly, the optimal discretization of the fiber cross-section is different for each of these modes. That is, to capture the oscillations of the higher-order modes near the core-cladding interface, a finer discretization is needed than for the fundamental mode LP_{01}. In particular, higher-order modes demand more refinements (or degrees of freedom) outside of the fiber core, when compared to the fundamental mode that is mostly confined to the core region. Therefore, a single geometry cannot be optimal for capturing any propagating mode.

Suppose we are interested in simulating the transverse mode instability (TMI) phenomenon [44] in active gain fiber amplifiers. The TMI is characterized by the chaotic transfer of energy between the fundamental mode and the higher-order modes. One challenge in com-

puting a numerical solution to the corresponding nonlinear Maxwell model (see Chapter 5) is capturing modes accurately when they occur. With mode instabilities, it is not known a-priori which modes will be propagating in which parts of the fiber. Refining the initial geometry globally to better resolve higher-order modes increases the computational complexity dramatically and may render the computation infeasible for large problem instances. Adaptivity, on the other hand, can be used to refine the mesh where it is needed for capturing these modes locally, keeping the overall computational cost significantly lower.

4.3.1 Adaptivity study

In the following experiments, we are aiming to establish the efficacy of adaptivity based on the DPG residual for resolving higher-order modes.

Adaptive refinement strategy. In the broken DPG setting, the residual is computed through element-wise contributions, i.e.,

$$\|\psi\|_{\mathcal{V}}^2 = \sum_{j=1}^{n} \| \psi|_{K_j} \|_{\mathcal{V}(K_j)}^2 \ , \tag{4.37}$$

where $K_j, j = 1, \ldots, n$, denotes the j-th element. After each solve, elements are marked for refinement if they satisfy a certain criterion. We use a strategy for marking elements that is based on Dörfler's marking [42]:

1. Sort the element residuals $\| \psi|_{K_j} \|_{\mathcal{V}(K_j)}^2$ in descending order, i.e.,

$$\| \psi|_{K_1} \|_{\mathcal{V}(K_1)}^2 \geq \| \psi|_{K_2} \|_{\mathcal{V}(K_2)}^2 \geq \cdots \geq \| \psi|_{K_n} \|_{\mathcal{V}(K_n)}^2. \tag{4.38}$$

2. Mark elements $K_j, j = 1, \ldots, J$, where $J \leq n$ is the smallest integer for which the

following is true:

$$\sum_{j=1}^{J} \| \psi|_{K_j} \|^2_{\mathcal{V}(K_j)} \geq \kappa \|\psi\|^2_{\mathcal{V}} \ , \tag{4.39}$$

where $\kappa \in (0, 1)$.

At this point, with some choice of κ, elements have been marked for refinement. However, it is not clear how to optimally refine each marked element when the hp mesh supports anisotropic adaptive refinements in both element size h and polynomial order p. The choice will ultimately be problem-dependent.

Initial mesh. In our experiments, we choose an initial mesh with uniform polynomial order $p = 5$, two elements per wavelength in z-direction (direction of propagation), and a radial (transverse) hybrid discretization using curvilinear hexahedral and prismatic elements.

Figure 4.10 illustrates the initial geometry discretization in the fiber cross-section (not drawn to scale): four prisms are used to model the center of the fiber core, and they are surrounded by three layers of four hexahedra each. We refer to these different layers as "domains" and enumerate them from 1 to 4 moving radially outward from the center of the fiber to the cladding boundary. The choice of the initial discretization was informed by the fiber geometry, the fact that all guided modes decay exponentially in the cladding region, and by conducting numerical tests primarily with the fundamental mode. For a relatively short fiber of 16 wavelengths, this initial geometry captures the fundamental mode very well with regard to several physical quantities of interest (e.g., conservation of power, mode confinement). Higher-order modes are not captured as well, and for fibers with many wavelengths (i.e., several hundred or a few thousand wavelengths) we observe more significant pollution errors in these modes. For example, the errors can be observed in small oscillations of the mode powers along the fiber, diffusive pollution effects, or an unsteady power confinement ratio.

We use DPG to perform multiple adaptive mesh refinements, each based on the re-

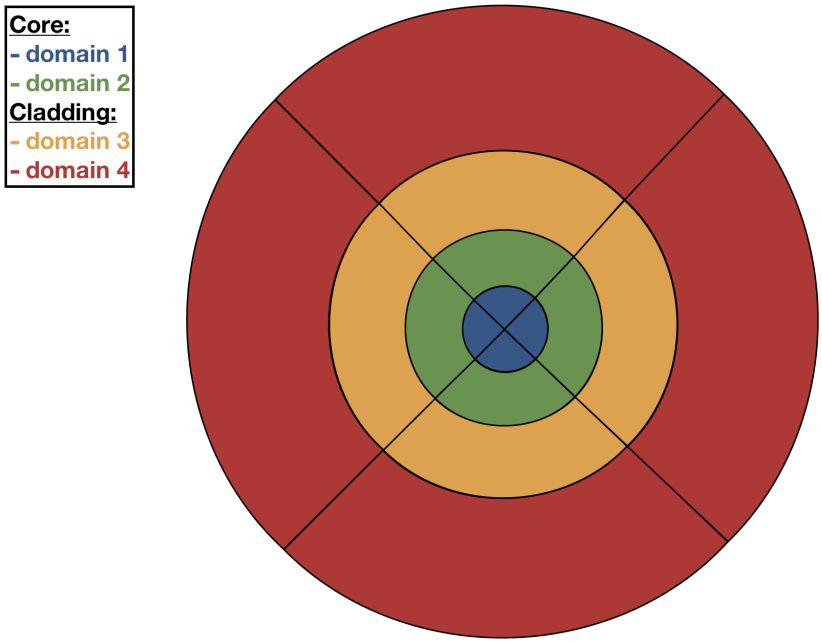


Figure 4.10: Initial geometry discretization in the fiber cross-section (not drawn to scale). This discretization works well for the fundamental mode, but it may not be sufficient to capture higher-order modes.

spective previous solution and residual, to test the residual error indicator for capturing different modes. As a test case, we look at the 16 wavelengths fiber. The goal is to observe the sensitivity of the adaptive refinements toward specific propagating modes. We choose the parameter $\kappa = 0.5$ in (4.39) and proceed with four adaptive refinement steps after the initial solution. Note that enforcing mesh regularity may cause some additional refinements to "close the mesh", i.e., to obtain a mesh that is 1-irregular (cf. Section 3.1).

Isotropic refinements. First, we apply isotropic h-refinements for marked elements. Figure 4.11 shows the domains of refinement in the fiber. Each plot illustrates how the mesh is successively refined for one particular guided mode propagating in the fiber. For the fundamental mode, the error indicator is marking elements for refinement primarily in the fiber core, where most of the energy is located. The first three refinement steps exclusively refine in the outer and inner core region. None of the adaptive refinements for higher-order modes refine inside the inner core region. It is notable how sensitive the error indicator is

to these different modes. In the case of the LP_{02} mode, the code primarily refines in the outer cladding region; this is likely to be the case because the initial cladding geometry discretization is too coarse to capture the exponential decay of the remaining energy in the transverse field.

Anisotropic refinements. Next, we repeat the experiment with anisotropic (radial) h-refinements. Radial refinements are of interest because higher-order modes are only more oscillatory in the transverse field, but they are not more oscillatory in the direction of propagation. In other words, the guided modes have very similar propagation constants (in fact, higher-order modes oscillate slightly slower than the fundamental mode). Therefore, if the numerical pollution is low for the fundamental mode, we may assume that the resolution in the direction of propagation is "good enough" for approximating any higher-order guided modes. Then, radial refinements (in h or p) are the more economical way of capturing these modes. For anisotropic h-adaptive refinements, depicted in Figure 4.12, a similar pattern emerges for the higher-order modes but the picture is quite different for the LP_{01} mode. The fundamental mode repeatedly refines elements in the same domain, because the anisotropic refinements do not decrease the local residuals as well as the isotropic refinements. This indicates that the fundamental mode is already well approximated in the transverse field and the residual demands refinements in z-direction for better accuracy of the numerical solution. This interplay between the resolution in different directions is critical when studying the pollution error for guided modes.

Figure 4.13 shows how the total residual evolves in both scenarios: we observe that the higher-order modes benefit much from anisotropic refinements, making this the preferred choice for improving the numerical solution with fewer degrees of freedom. For the fundamental mode, we find that the residual does not further decrease through anisotropic refinements, indicating the mode is captured quite well by the initial geometry discretization.

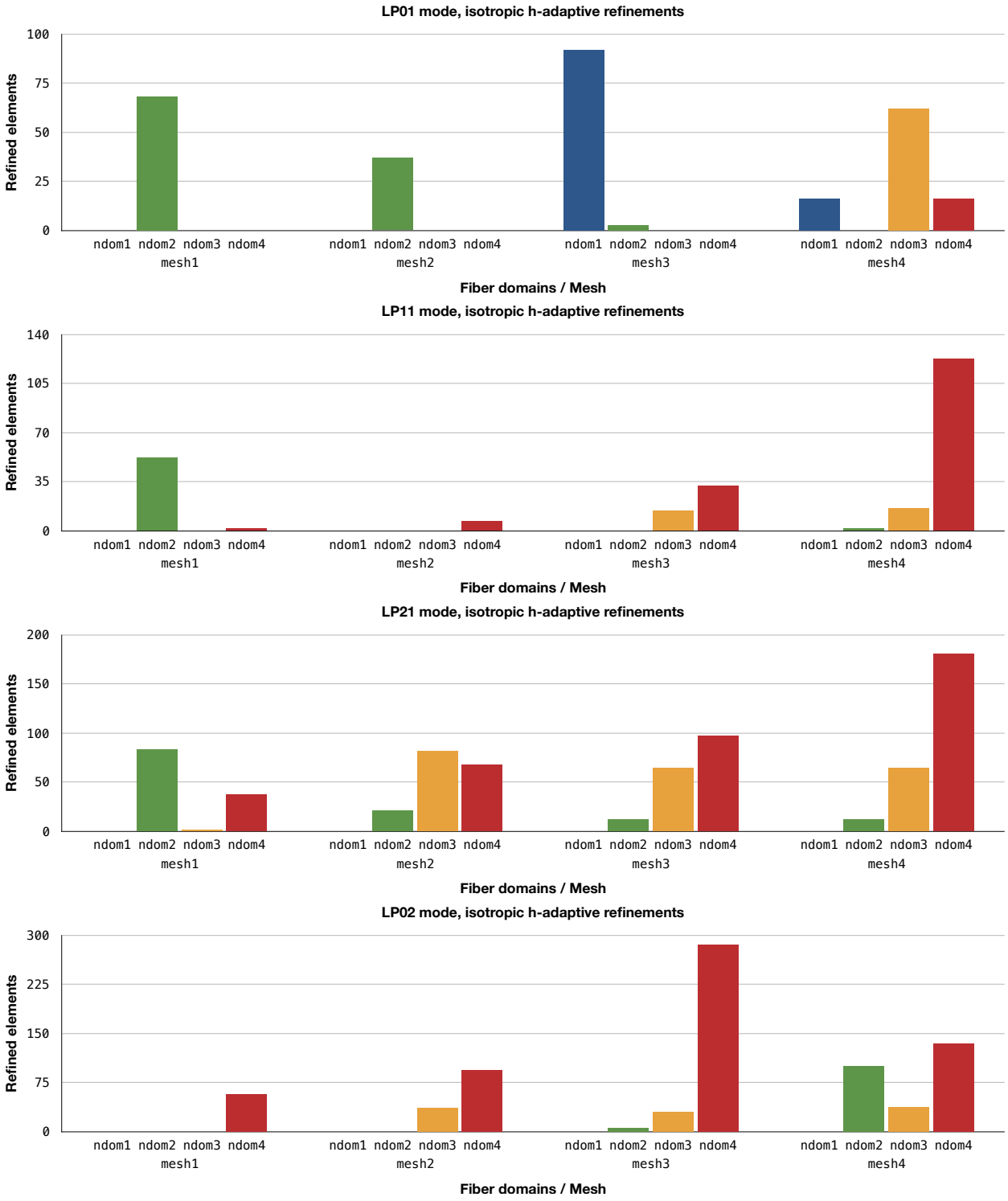


Figure 4.11: Isotropic h-adaptive refinements. Depending on the propagating mode, the DPG error indicator marks elements for refinement in different fiber domains.

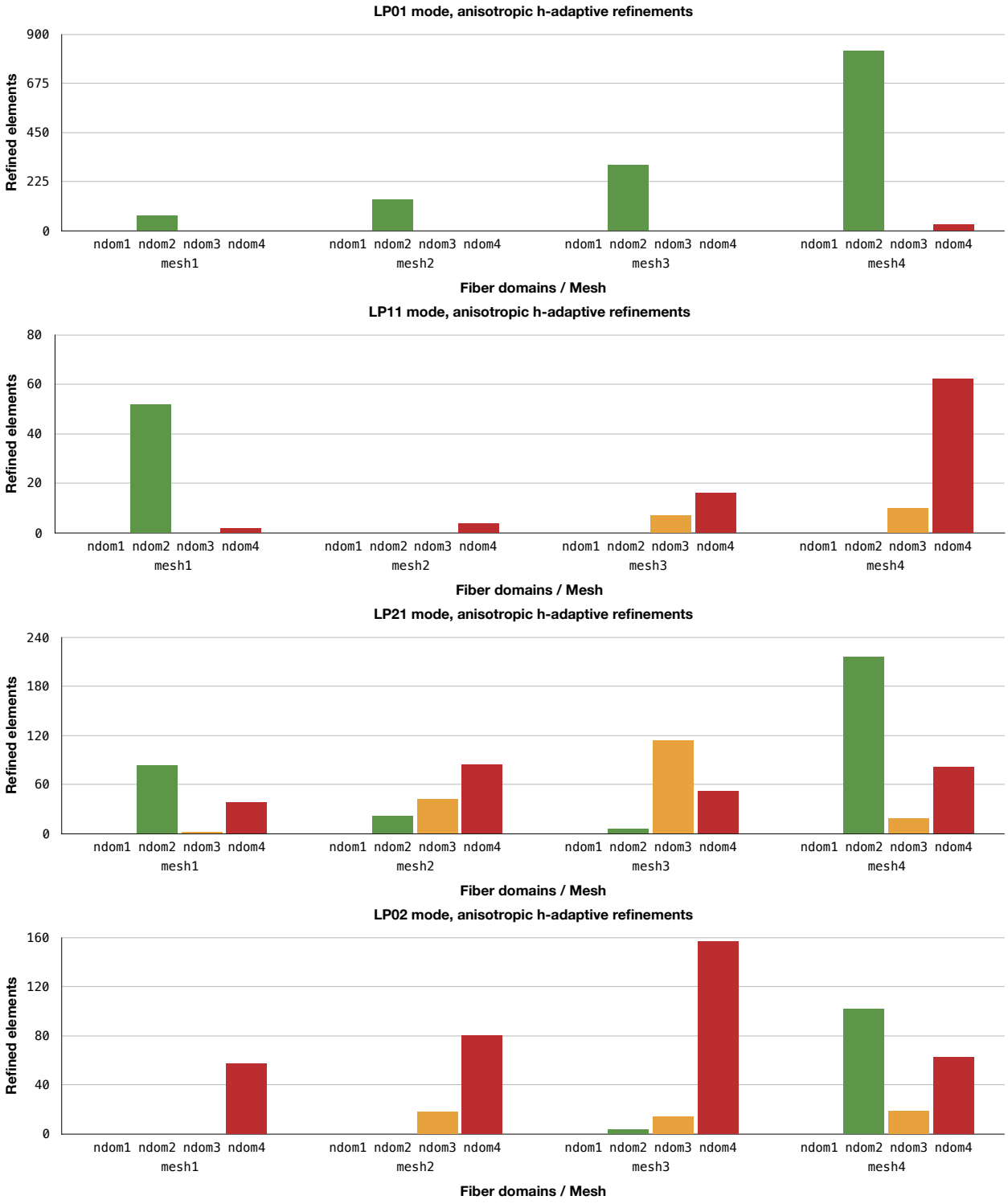
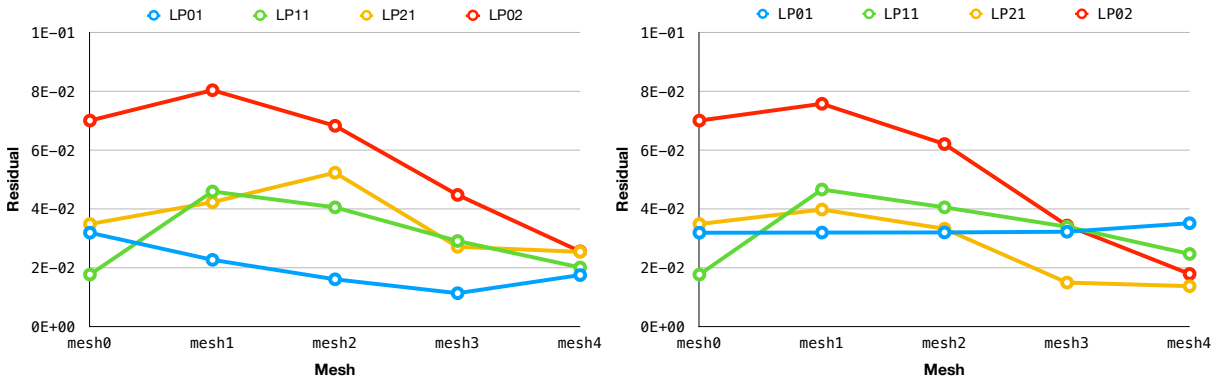


Figure 4.12: Anisotropic h-adaptive refinements in the radial (transverse) direction. For higher-order modes, anisotropic refinements are more computationally efficient to capture the transverse mode profile.



(a) Isotropic h-adaptive refinements

(b) Anisotropic h-adaptive refinements

Figure 4.13: Evolution of the DPG residual in adaptive mesh refinements. With our choice of initial geometry discretization, the anisotropic refinements decrease the residual for higher-order modes but not for the fundamental mode. This illustrates that the interplay of the resolution between the transverse direction and the direction of propagation is critical, and the optimal refinement strategy depends on the propagating modes.

4.3.2 Load balancing

In the parallel computation of the fiber problem, we partition the geometry into subdomains, each owned by one distinct MPI process. The rank of each MPI process is the ID of the subdomain it owns (cf. Section 3.2.3).

Static partitioning. Initially, we partition the fiber directly based on geometric cuts orthogonal to the fiber axis. Figure 4.14 illustrates what the partitioning looks like for four subdomains. This initial static partitioning is a good choice because it keeps the interfaces between subdomains small, making it possible to compute large fibers in parallel with a nested dissection solve and obtain good weak scaling (cf. Section 6.1). As the adaptive mesh refinements proceed, the domain is dynamically repartitioned to retain load balance. A number of different repartitioners are available in third-party software libraries, such as Zoltan [41]. In the fiber domain, graph partitioning that strives for minimum cuts is a good choice because it keeps the subdomain interfaces relatively small. In the broken ultraweak Maxwell formulation, the numbers of element interior DOFs (electromagnetic fields: L^2) are used as weights for graph vertices, and the numbers of trace DOFs on faces (electromagnetic

fluxes: $H(\text{curl})$ trace) serve as weights for the graph's edges. Connectivities from edge degrees of freedom are omitted to provide a sparser graph and accelerate partitioning. ParMETIS or PT-Scotch can be used for approximating the partitioning problem. As an alternative, we use a custom dynamic fiber repartitioner that forces orthogonal cuts through the domain while trying to maximize load balance and minimize data migration, similar to recursive coordinate bisection partitioners. This custom repartitioner can perform orders of magnitude faster than graph partitioning because it relies primarily on geometry information.

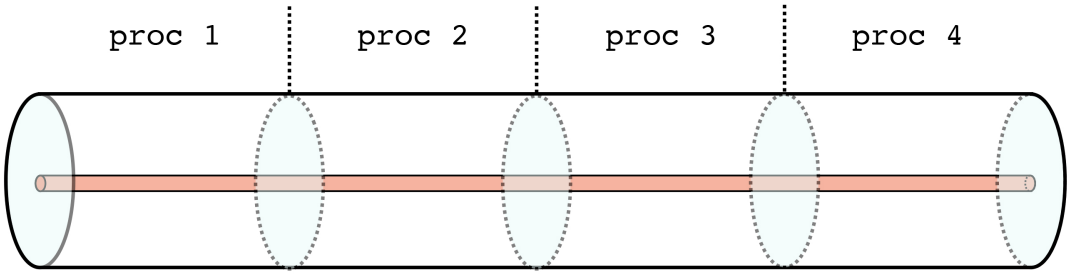


Figure 4.14: Initial (static) load distribution in the step-index fiber. An efficient and balanced distribution is achieved by defining subdomains through orthogonal cuts to the fiber axis.

We study how the workload in different subdomains changes without repartitioning. Figure 4.15 and Figure 4.16 show the workload per MPI rank in a fiber of 16 wavelengths, partitioned into eight subdomains, with h-adaptive isotropic and anisotropic refinements, respectively. Both plots depict results for the higher-order mode LP_{21}. Here, the workload is simply shown as the number of subdomain interior DOFs, i.e., all solution DOFs that are part of a subdomain excluding the trace DOFs on the subdomain interfaces.

Isotropic refinements lead expectedly to higher load imbalance because each isotropic h-refinement increases the local DOFs by another factor of two compared to the anisotropic (radial) h-refinement. Two observations stand out: firstly, the subdomains closer to the fiber input appear to exhibit higher residuals, hence more refinements are observed in that region; secondly, towards the fiber output, many refinements are picked up in the second-to-last subdomain, and almost none in the very last one. The latter observation is an effect from

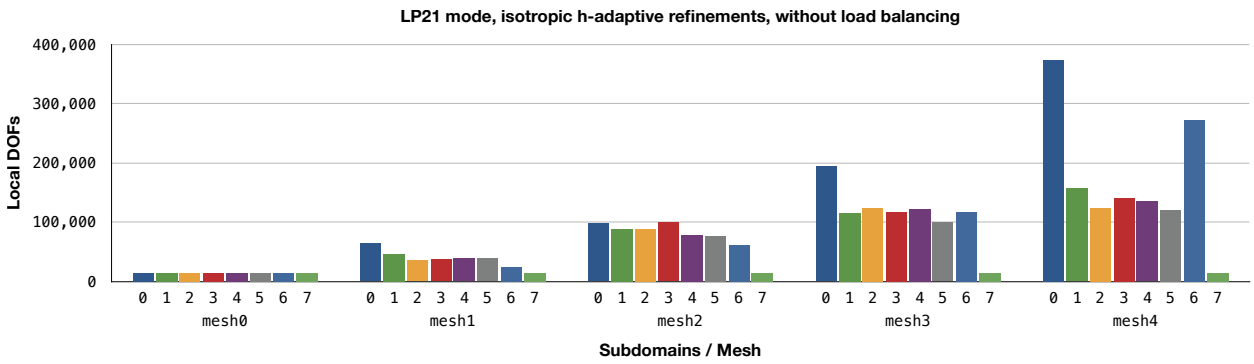


Figure 4.15: Isotropic h-adaptive refinements. The load imbalance increases with every adaptive refinement step. The most refinements are happening at the fiber input (subdomain 0) and within the first few wavelengths of the PML region (subdomain 6).

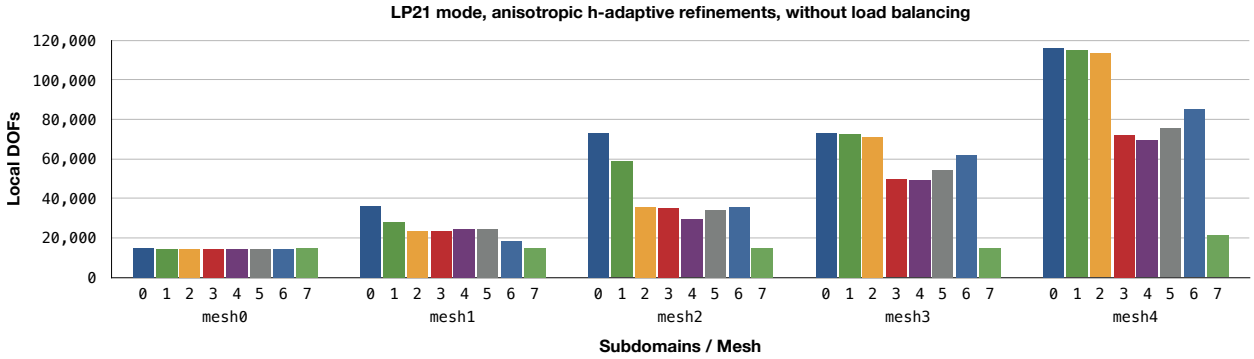


Figure 4.16: Radial (anisotropic) h-adaptive refinements. Load imbalance also occurs with the anisotropic adaptivity but it is less pronounced than in the isotropic case.

the PML region at the fiber end. In this short fiber, the PML is active in the last two subdomains (i.e., the layer encompasses about four wavelengths). When the wave enters the PML region, it exhibits exponential decay due to the coordinate stretching. This initial decay must be captured accurately by the numerical solution. The DPG residual recognizes the need for additional refinements in this region and marks elements in the corresponding subdomain. By the time the wave enters the last subdomain, it has decayed so far that the residual remains fairly small and almost none of the elements are marked in the adaptive procedure.

Dynamic repartitioning. It is evident that dynamic load balancing is necessary for computational efficiency in the simulation of the TMI phenomenon or other applications with

energy transfer between guided modes. By repartitioning the fiber domain, we obtain significant speedup in the total computation time. Figure 4.17 shows the workload per MPI process when the mesh is repartitioned after every isotropic h-adaptive refinement step. Both repartitioners, the graph partitioner based on ParMETIS as well as the custom fiber partitioner, distribute the workload evenly among the processors. The distribution is not exactly even in either case because both also aim to minimize the size of the interface problem that separates the subdomains. That is, the graph partitioner approximates minimum cuts while the custom fiber partitioner keeps all cuts orthogonal to the fiber axis.

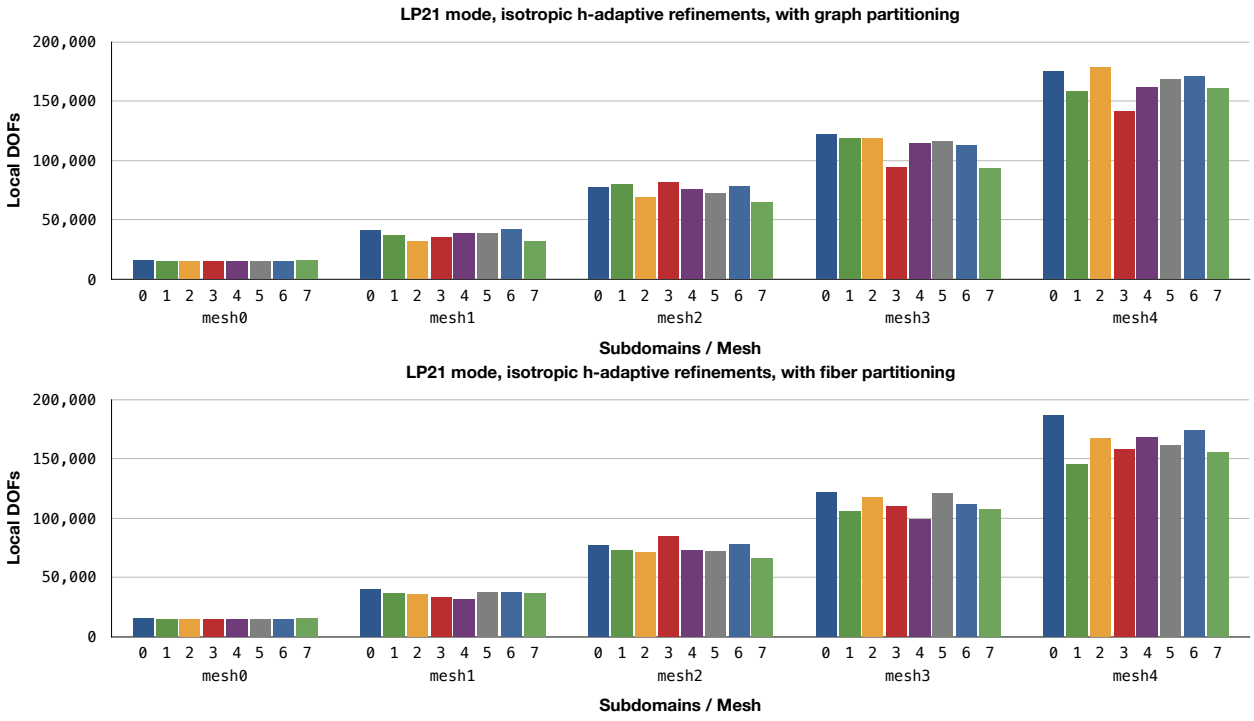
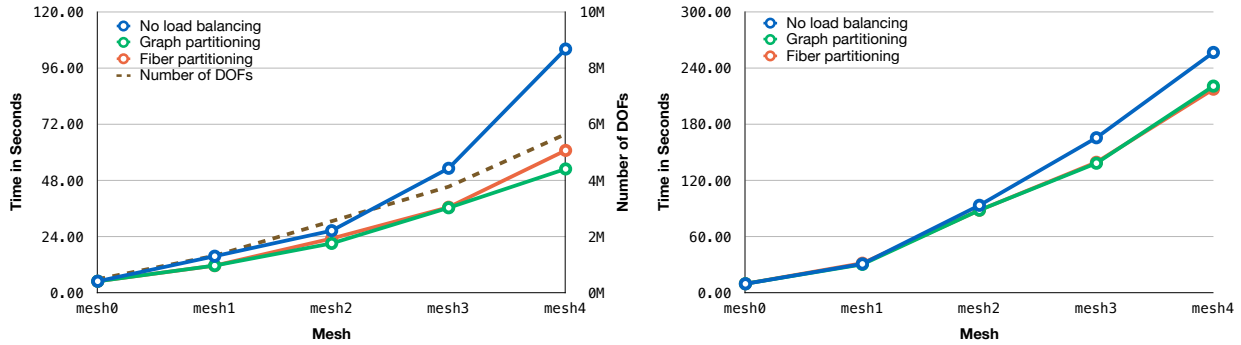


Figure 4.17: Load balancing for isotropic h-adaptive refinements. Both the graph partitioning and the custom fiber partitioning distribute the workload evenly among the processors in every refinement step.

Figure 4.18a displays the computation time for the distributed finite element assembly for each of the isotropic h-adaptive meshes. The assembly time includes the time for element integration and the assembly of every sparse subdomain stiffness matrix and load vector. In DPG methods, the assembly time can account for a substantial part of the entire time to solution because of the computation of optimal test functions in the enriched test space (cf.

Section 3.4.1). On the other hand, the assembly computation is conveniently parallel and exhibits good parallel scaling as long as the workload is balanced. With load balancing, the increase in computation time corresponds to the total increase in the number of DOFs, as expected; in the imbalanced case, where some processes finish early and remain idle until the MPI process with the maximal workload is done, the assembly time increases unproportionally.

The parallel solve time is depicted in Figure 4.18b; the plot shows the total time for analysis, factorization, and linear solve (forward and backward elimination) performed by the distributed MUMPS solver [4]. Both load balancing strategies result in a modest reduction of the solve time and perform equally well. Of course, the time spent on load balancing, i.e., partitioning and data migration, should be taken into account, as well. However, for the problem size considered here, we have found that the computation time for these tasks is almost negligibly small (less than one second per mesh).



(a) Assembly time

(b) Solve time

Figure 4.18: Computation time with isotropic h-adaptive refinements for the guided LP_{21} mode. a) The finite element assembly time (including time for element integration) is proportional to the (maximal) workload per subdomain. With load balancing, the assembly time scales corresponding to the increase in the total number of degrees of freedom. b) The solve time, including the time for analysis, factorization, and linear solve performed by the distributed MUMPS solver, exhibits a modest reduction when load balancing is performed.

Chapter 5

Optical Fiber Amplifier Model

In this chapter, we present our three-dimensional DPG finite element model for the simulation of laser amplification in a fiber amplifier. Our model is based on the time-harmonic Maxwell equations, and it incorporates both amplification via an active dopant and thermal effects via coupling with the heat equation. As a full vectorial finite element simulation, this model distinguishes itself from other fiber amplifier models that are typically posed as an initial value problem and make significantly more approximations. Our model supports co-, counter-, and bi-directional pumping configurations, as well as inhomogeneous and anisotropic material properties. The long-term goal of this modeling effort is to study nonlinear phenomena that prohibit achieving unprecedented power levels in fiber amplifiers, along with validating typical approximations used in lower-fidelity models. The high-fidelity simulation comes at the cost of a high-order finite element discretization with many degrees of freedom per wavelength. This is necessary to counter the effect of numerical pollution due to the high-frequency nature of the wave simulation (cf. Section 4.2). To make the computation more feasible, we have developed a novel longitudinal model rescaling, using artificial material parameters with the goal of preserving certain quantities of interest. Numerical tests demonstrate the applicability and utility of this scaled model in the simulation of an ytterbium-doped, step-index fiber amplifier that experiences laser amplification and heating.

We begin with an overview of a typical fiber amplifier setup in Section 5.1 and state

some of the generally applicable modeling assumptions for silica fibers in Section 5.2. In Section 5.3, we discuss our nonlinear gain amplification model for the time-harmonic Maxwell equations, which involves two weakly-coupled systems. The non-dimensionalization is shown in detail, because the disparate length scales in the fiber pose a modeling and computational challenge. Section 5.4 describes the modeling of the thermal response via coupling with the heat equation. This includes both the heat deposition in the fiber, based on the laser amplification, as well as the induced thermal polarization. The longitudinal scaling of the fiber model is discussed in Section 5.5, where we present new arguments on how to rescale the coupled Maxwell/heat model while preserving several quantities of interest. In particular, we can obtain an accurate heat distribution along the fiber amplifier for an arbitrary scaling factor. Section 5.6 presents numerical results for an ytterbium-doped active gain fiber amplifier with the full vectorial model: first, we introduce the DPG formulation of the coupled Maxwell/heat problem; then, we show the power distribution along the fiber, the amplifier efficiency, and convergence of the DPG residual in the nonlinear solve. The scaling arguments from the preceding section are numerically corroborated by qualitatively and quantitatively comparing the results for fibers of different lengths. Lastly, we show simulation results that illustrate the effect of the heating on the fiber material refractive index. The chapter concludes with a summary of our work and gives a brief outlook on the wide applicability of this unique 3D model.[1]

Remark: In Section 4.1, we established the well-known linear waveguide theory (see also [64, 74] and references therein). This chapter proceeds presuming that the reader is familiar with the relevant concepts (e.g., the modes in a weakly-guiding step-index fiber derived from the linear time-harmonic Maxwell equations).

[1]The contents of this chapter are partially taken from a previous publication: S. Henneking, J. Grosek, and L. Demkowicz. "Model and computational advancements to full vectorial Maxwell model for studying fiber amplifiers". In: *Comput. Math. Appl.* 85 (2021), pp. 30–41.
The author contributed to model development, numerical implementation, and analysis of the results.

5.1 Ytterbium-doped fiber amplifier

Figure 5.1 is a schematic of a final-stage large-mode-area step-index fiber amplifier (not drawn to scale). The typical configuration includes a highly coherent *laser signal*, injected into the fiber core region, and a less coherent *pump field* launched into the inner cladding and core regions simultaneously (as depicted in Figure 5.1), usually by means of the pump combiner that is spliced onto the beginning of the amplifier. However, it is also possible to core-pump the amplifier by launching the pump field into only the fiber core region as is done with the signal. The fact that the core region has a slightly higher refractive index than the inner cladding region, both made primarily of fused silica, allows for the signal to be guided along the fiber length in the core by total internal reflection. Thus, the fiber amplifier is a waveguide. Likewise, the polymer coating, which has a much lower refractive index than the glass, helps guide the pump light (cladding-pumped configuration) in the inner cladding region by total internal reflection. The light naturally falls into a discrete set of guided modes, where the fundamental mode has a Gaussian-like profile (see Figure 4.1), leading to the best beam quality that the fiber waveguide can output. These core-guided modes are derived from the linear waveguide theory for weakly-guiding step-index fibers (see Section 4.1).

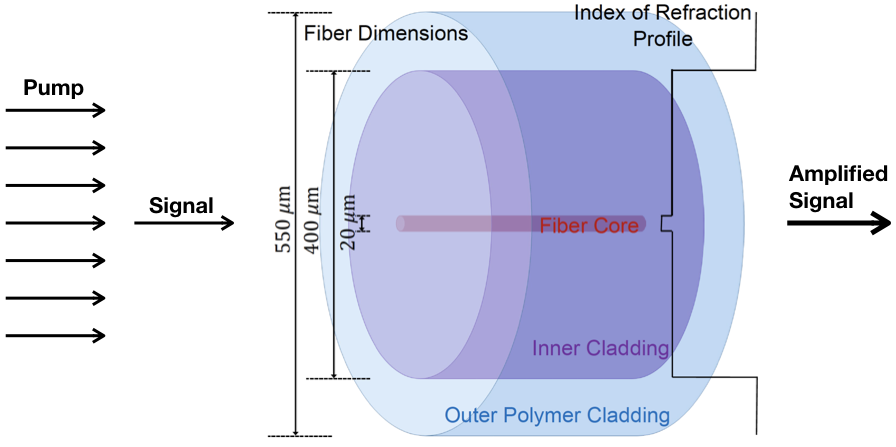


Figure 5.1: Schematic of a weakly-guiding, continuous-wave, double-clad, large-mode-area, step-index fiber amplifier (not drawn to scale) [65].

Active gain fiber amplifiers have doped core regions (e.g., ytterbium dopant). The pump and signal wavelengths are chosen such that the pump field experiences very high absorption by the active dopant, while the signal field is in a regime of high emission probability of the active dopant. Given that sufficiently high pump and signal powers are launched into the amplifier, a large percentage of the pump light can be converted into highly coherent laser signal light by the stimulated emission process. This mechanism is also referred to as *active gain*. Since the pump photons have a higher frequency than the signal photons, some energy is lost in this process, ultimately leading to heat deposition along the fiber via this *quantum defect*. The fiber amplifier is usually 5–20 meters long, chosen so as to absorb a large portion of the launched pump light (e.g., $\sim 95\%$) and/or to limit the onset of detrimental optical nonlinearities (e.g., SBS).

5.2 Maxwell's equations in a fiber amplifier

The propagation of optical fields in fibers is governed by Maxwell's equations. We assume that the free current density and charge density vanish in the silica fiber. And, to a good approximation, optical fibers are nonmagnetic, hence we may neglect induced magnetic polarization in the material [1]. The signal field and the pump field are assumed to each independently satisfy the time-harmonic Maxwell equations, but they are weakly coupled through nonlinear polarization terms. The time-harmonic ansatz is justified by the fact that the signal and pump fields are near-monochromatic, and that the time-dependent phenomena of interest, most notably thermal effects, happen at a much slower time scale. The time-harmonic Maxwell equations in the silica fiber are:

$$\nabla \times \boldsymbol{E} = -i\omega\mu_0 \boldsymbol{H}, \tag{5.1}$$

$$\nabla \times \boldsymbol{H} = i\omega(\varepsilon_0 \boldsymbol{E} + \boldsymbol{P}), \tag{5.2}$$

where $\boldsymbol{E}$ and $\boldsymbol{H}$ are the electric and magnetic field vectors, respectively, $\boldsymbol{P}$ is the induced electric polarization, ω is the angular frequency, and ε_0, μ_0 are the electric permittivity in vacuum and the magnetic permeability in vacuum, respectively. Recall that the complex-valued time-harmonic fields $\boldsymbol{E}, \boldsymbol{H}$ are related to the real-valued time-dependent fields $\boldsymbol{\mathcal{E}}, \boldsymbol{\mathcal{H}}$ through the following ansatz:

$$\boldsymbol{\mathcal{E}}(x, y, z, t) = \mathfrak{Re}\left\{\boldsymbol{E}(x, y, z)e^{i\omega t}\right\}, \tag{5.3}$$

$$\boldsymbol{\mathcal{H}}(x, y, z, t) = \mathfrak{Re}\left\{\boldsymbol{H}(x, y, z)e^{i\omega t}\right\}. \tag{5.4}$$

We assume that the fiber is aligned with its longitudinal axis centered along the z-axis, and the transverse coordinates are x, y. The core and cladding radii are denoted by r_{core} and r_{clad}, respectively, with the corresponding material refractive indices n_{core} and n_{clad}.

The generally nonlinear polarization term $\boldsymbol{P}$ may be modeled as [1]:

$$\boldsymbol{P} = \varepsilon_0 \left(\boldsymbol{\chi}^{(1)} \cdot \boldsymbol{E} + \boldsymbol{\chi}^{(2)} : \boldsymbol{E} \otimes \boldsymbol{E} + \boldsymbol{\chi}^{(3)} \vdots \boldsymbol{E} \otimes \boldsymbol{E} \otimes \boldsymbol{E} + \cdots\right), \tag{5.5}$$

where $\boldsymbol{\chi}^{(k)}$ is the kth-order electric susceptibility, given by a $(k+1)$-rank tensor. A medium with an inversion symmetry at the molecular level (e.g., silica glass: SiO_2) exhibits only negligible second-order susceptibilities [1]. Therefore, in the optical fiber, $\boldsymbol{\chi}^{(2)} \approx \boldsymbol{0}$.

In the context of fiber optics, we are interested in modeling certain nonlinear physical effects; thus it is common to write

$$\boldsymbol{P} = \boldsymbol{P}_{\text{background}} + \boldsymbol{P}_{\text{active gain}} + \boldsymbol{P}_{\text{thermal}} + \boldsymbol{P}_{\text{opt. nonlin.}} + \cdots . \tag{5.6}$$

The background polarization is a first-order susceptibility term and describes the real part of the refractive index. Active gain in the fiber may be modeled as a first-order susceptibility term as well, but in terms of a complex perturbation to the refractive index. The thermal polarization perturbs the real part of the refractive index and may thus, too, be modeled

as a first-order susceptibility. Optical nonlinearities such as SBS or SRS are third-order susceptibility terms.

5.3 Active gain fiber model

In the proposed model, we consider two electric polarization terms: the linear background polarization $\boldsymbol{P}_{\text{background}}$ and the polarization due to active gain $\boldsymbol{P}_{\text{active gain}}$. Thermal polarization will be discussed in the next section. Since there is no term inducing anisotropy in the material refractive index tensor $\boldsymbol{n}$ at this point, we may write the background polarization as:

$$\boldsymbol{P}_k^{\text{background}} = \varepsilon_0(n^2 - 1)\boldsymbol{E}_k, \quad k \in \{s, p\}, \tag{5.7}$$

where index k denotes the appropriate field, s for signal or p for pump. The active gain is modeled as a first-order term or complex perturbation to the refractive index [98, 65],

$$\boldsymbol{P}_k^{\text{active gain}} = i\varepsilon_0 \frac{nc}{\omega_k} g_k(\boldsymbol{E}_{\{s,p\}})\boldsymbol{E}_k, \tag{5.8}$$

where the gain g_k is a function of both electric fields $\boldsymbol{E}_s$ and $\boldsymbol{E}_p$, and c is the speed of light in vacuum. The gain function carries units of m^{-1}, sometimes expressed in dB/m. A positive gain amplifies the field while a negative one causes decay. Considering these first-order susceptibilities only, the time-harmonic Maxwell equations in the fiber are

$$\nabla \times \boldsymbol{E}_k = -i\omega_k \mu_0 \boldsymbol{H}_k, \tag{5.9}$$

$$\nabla \times \boldsymbol{H}_k = i\omega_k(\varepsilon_0 n^2 \boldsymbol{E}_k + \boldsymbol{P}_k^{\text{active gain}}). \tag{5.10}$$

With the gain polarization term (5.8), the Ampère–Maxwell equation (5.10) becomes

$$\nabla \times \boldsymbol{H}_k = i\omega_k \varepsilon_0 n^2 \boldsymbol{E}_k - \varepsilon_0 n c g_k(\boldsymbol{E}_{\{s,p\}})\boldsymbol{E}_k. \tag{5.11}$$

Ion population dynamics. In this effort, the gain function is expressed in terms of rate equations for a simplified version of the ion population dynamics for an ytterbium-doped fiber, as described in [101]. It is assumed that the ytterbium (Yb) dopant is uniformly distributed throughout the fiber core. The electrons of Yb atoms absorb and emit photons at a mean rate that is determined by experimentally measured absorption and emission cross-sections (see Table A.6), denoted as $\sigma_k^{\text{abs}}, \sigma_k^{\text{ems}}$ where $k \in \{s, p\}$ (units of m^2/ion). This simplified two-manifold model only considers a single excited state and ground state for the electrons in the outer most shell of the Yb atom. The amplifier works by primarily absorbing pump photons, sending the electron into the excited state, and then having the majority of those excited electrons stimulated to emit a photon at the signal wavelength, coherent with the signal field, allowing the electron to return to its ground state. This leads to a frequency dependent gain function as follows:

$$g_k(\boldsymbol{r}, z, t) = \sigma_k^{\text{ems}} \mathcal{N}_{\text{excited}}^{\text{Yb}}(\boldsymbol{r}, z, t) - \sigma_k^{\text{abs}} \mathcal{N}_{\text{ground}}^{\text{Yb}}(\boldsymbol{r}, z, t), \quad |\boldsymbol{r}| < r_{\text{core}}, \tag{5.12}$$

where $\boldsymbol{r}$ represents the coordinate directions, (x, y) – Cartesian, or (r, θ) – polar, such that $r \in [0, r_{\text{clad}}]$ and $\theta \in [0, 2\pi)$; and $\mathcal{N}_{\text{ground}}^{\text{Yb}}, \mathcal{N}_{\text{excited}}^{\text{Yb}}$ are the ground-state and excited-state population concentrations of Yb ions expressed in ion/m^3. The total ion population concentration is assumed to be known and must remain constant such that

$$\mathcal{N}_{\text{total}}^{\text{Yb}} = \mathcal{N}_{\text{ground}}^{\text{Yb}} + \mathcal{N}_{\text{excited}}^{\text{Yb}}. \tag{5.13}$$

Therefore,

$$\frac{\partial \mathcal{N}_{\text{excited}}^{\text{Yb}}}{\partial t} = -\frac{\partial \mathcal{N}_{\text{ground}}^{\text{Yb}}}{\partial t}. \tag{5.14}$$

The transient equation for the excited ion population is given by:

$$\frac{\partial \mathcal{N}_{\text{excited}}^{\text{Yb}}}{\partial t} = \sum_{k \in \{s,p\}} \frac{I_k}{\hbar \omega_k} (\sigma_k^{\text{abs}} \mathcal{N}_{\text{ground}}^{\text{Yb}} - \sigma_k^{\text{ems}} \mathcal{N}_{\text{excited}}^{\text{Yb}}) - \frac{\mathcal{N}_{\text{total}}^{\text{Yb}}}{\tau}, \tag{5.15}$$

where $I_k = |\mathfrak{Re}\{\boldsymbol{E}_k \times \boldsymbol{H}_k^*\}|$ is the irradiance, τ is the measured Yb ion upper level radiative lifetime (see Table A.6), $\hbar$ is the reduced Planck constant (see Table A.2), and $I_k/(\hbar\omega_k)$ represents the photon flux.

This model, as has been done in other fiber amplifier models [96, 124, 113], will neglect the time dynamics of the transit time of the light along the length of the fiber ($\sim$10 ns), and the population dynamics of the active gain process, only considering the temporal evolution of the heat deposition and dissipation in the fiber. The characteristic time to steady-state gain is on the order of 10 μs, whereas the characteristic heat diffusion time is on the order of 1 ms, both depending on the fiber configuration.[2] Thus, this effort will use the steady-state solution of the ion population model:

$$\bar{\mathcal{N}}^{\mathrm{Yb}}_{\mathrm{excited}} = \frac{\sum_{k\in\{s,p\}} \frac{I_k}{\hbar\omega_k}\sigma_k^{\mathrm{abs}}}{\frac{1}{\tau} + \sum_{k\in\{s,p\}} \frac{I_k}{\hbar\omega_k}(\sigma_k^{\mathrm{abs}} + \sigma_k^{\mathrm{ems}})}\mathcal{N}^{\mathrm{Yb}}_{\mathrm{total}}, \tag{5.16}$$

which leads to a concise expression for the gain function,

$$\boxed{g_k \approx -\sigma_k^{\mathrm{abs}}\mathcal{N}^{\mathrm{Yb}}_{\mathrm{total}} + (\sigma_k^{\mathrm{abs}} + \sigma_k^{\mathrm{ems}})\bar{\mathcal{N}}^{\mathrm{Yb}}_{\mathrm{excited}}.} \tag{5.17}$$

Non-dimensional Maxwell equations. Non-dimensionalization is essential in the numerical computation of the solution to the Maxwell equations, particularly when the scales involved are very disparate as in the case of optical fiber amplifiers. The non-dimensional Maxwell system (see derivation in Appendix B.1) is given by:

$$\hat{\nabla} \times \hat{\boldsymbol{E}}_k = -i\hat{\omega}_k \hat{\boldsymbol{H}}_k, \tag{5.18}$$

$$\hat{\nabla} \times \hat{\boldsymbol{H}}_k = in^2\hat{\omega}_k \hat{\boldsymbol{E}}_k - nl_0 g_0 \hat{g}_k(\hat{\boldsymbol{E}}_{\{s,p\}})\hat{\boldsymbol{E}}_k, \tag{5.19}$$

where the "hat" symbol indicates non-dimensional variables; note that $l_0 g_0$ is a (positive) dimensionless quantity. See Table A.3 for an overview of the selected dimensional scales

[2] Generally, higher pump and signal irradiances correspond to faster times to steady-state gain.

in our model. The non-dimensional Ampère–Maxwell equation (5.19) reflects that for a positive gain function, the gain term can be interpreted as *negative conductivity*, causing amplification of the propagating field. Conversely, negative gain can be seen as positive conductivity or linear loss.

5.4 Thermal coupling

Thermal polarization. First, we address how the heating affects the solution to the Maxwell system. The effect of the heating is modeled as an isotropic temperature dependence of the refractive index. Let the ambient temperature be denoted by $\bar{T} \equiv T_{\text{ambient}}$, and the refractive index at ambient temperature as $\bar{n} \equiv n(\bar{T})$. Then, let the temperature T and refractive index n at any point $(\boldsymbol{r}, z)$ in the fiber at time t be given by:

$$T(\boldsymbol{r}, z, t) = \bar{T} + \delta T(\boldsymbol{r}, z, t), \tag{5.20}$$

$$n(\boldsymbol{r}, z, t) = \bar{n} + \delta n(\boldsymbol{r}, z, t), \tag{5.21}$$

where δT is the change in temperature, and δn the thermally induced perturbation to the material refractive index. Next, we linearize the refractive index perturbation,

$$n(T) = \bar{n} + \frac{dn}{dT}(\bar{T})\delta T + \frac{d^2n}{dT^2}(\bar{T})\frac{\delta T^2}{2} + \cdots, \tag{5.22}$$

$$\delta n \approx \frac{dn}{dT}(\bar{T})\delta T, \tag{5.23}$$

where dn/dT is the thermo-optic coefficient for silica glass, an experimentally measured value (see Table A.7). In the fiber, we expect temperature changes of up to ca. 100 K from ambient (room) temperature, a regime in which the material refractive index change can be modeled with reasonable accuracy as a linear response to the temperature change. The thermo-optic coefficient is on the order of 10^{-5} K^{-1} for $\mathrm{SiO_2}$, hence one can expect induced refractive index perturbations of about three orders of magnitude smaller than the refractive index n

of the medium: $\delta n \sim \mathcal{O}(10^{-3})$.

In the active gain model, the perturbed refractive index will affect the Maxwell solution through a change in the background and gain polarization:

$$\boldsymbol{P}_k^{\text{background}} = \varepsilon_0(n(T)^2 - 1)\boldsymbol{E}_k, \tag{5.24}$$

$$\boldsymbol{P}_k^{\text{active gain}} = i\varepsilon_0 \frac{n(T)c}{\omega_k} g_k \boldsymbol{E}_k. \tag{5.25}$$

Therefore, we can express thermal polarization explicitly as:

$$\begin{aligned} \boldsymbol{P}_k^{\text{thermal}}(T) = (\boldsymbol{P}_k^{\text{background}}(T) - \boldsymbol{P}_k^{\text{background}}(\bar{T})) \\ + (\boldsymbol{P}_k^{\text{active gain}}(T) - \boldsymbol{P}_k^{\text{active gain}}(\bar{T})). \end{aligned} \tag{5.26}$$

Heat coupling model. The thermal response in the fiber amplifier is modeled by the heat equation,

$$\rho_0 C_p \frac{\partial T}{\partial t} - \nabla \cdot (\boldsymbol{\kappa} \nabla T) = Q, \tag{5.27}$$

where ρ_0, C_p, and $\boldsymbol{\kappa}$ are the mean density, specific heat, and thermal conductivity of silica glass, respectively (see Table A.7). Appropriate boundary and initial conditions are discussed later. We assume that the material is isotropic and homogeneous, so its thermal conductivity is uniform, i.e., $\boldsymbol{\kappa}(\boldsymbol{r}, z) = \kappa \mathbb{I}$, and all of the thermal parameters are assumed to be temperature independent. The right-hand side has the source term $Q = Q(\boldsymbol{r}, z, t)$ that couples the electromagnetic fields to the heat deposition in the fiber.

Heat deposition. The heat source of a stimulated emission dominated amplifier can be modeled by [115, 65]:

$$Q(I_{\{s,p\}}) = -\left(g_p(I_{\{s,p\}})I_p + g_s(I_{\{s,p\}})I_s\right), \tag{5.28}$$

where g_s and g_p are the gain functions for signal and pump fields, respectively, and $I_{\{s,p\}}$ denotes the respective field irradiances. Therefore, Q is explicitly dependent on the solution to the Maxwell equations, and thus is implicitly dependent on the temperature itself. Because gain occurs only inside the fiber core, the heat deposition will, too, only occur in the core.

Non-dimensional heat equation. As discussed previously, we are interested in computing the temperature difference $\delta T = T - \bar{T}$ to obtain the perturbation to the refractive index $\delta n \approx (dn/dT)\delta T$. The heat equation for the temperature difference is given by:

$$\rho_0 C_p \frac{\partial(\delta T)}{\partial t} - \kappa \Delta(\delta T) = Q(I_{\{s,p\}}), \tag{5.29}$$

with appropriate boundary and initial conditions. The non-dimensional heat equation (see derivation in Appendix B.1) is

$$\frac{\partial(\delta \hat{T})}{\partial \hat{t}} - \alpha_0 \hat{\Delta}(\delta \hat{T}) = Q_0 \hat{Q}(\hat{I}_{\{s,p\}}), \tag{5.30}$$

where α_0 denotes a non-dimensional diffusivity scale, and Q_0 a non-dimensional heat deposition scale.

Boundary and initial conditions. We assume that the initial temperature in the fiber is the ambient temperature. In other words, the temperature difference δT is initially zero. At the radial inner cladding boundary with the polymer jacket, we impose zero Dirichlet boundary conditions (ambient temperature), implying efficient cooling at the glass-polymer interface. In future work, it ought to be relatively straightforward to account for the polymer coating and more realistic heat dissipation into the ambient air and/or into the metal spool that fibers are usually coiled around. However, for the primary reason of keeping the computational domain smaller, this effort neglects the more realistic scenario. At the fiber ends, we also impose homogeneous Dirichlet boundary conditions for simplicity because of

the large aspect ratio between the length of the fiber and its radial width, which strongly suggests that most of the heat dissipation occurs through the radial direction rather than in the longitudinal direction. To summarize,

$$\text{Initial condition: } \delta\hat{T}(\boldsymbol{r}, z, 0) = 0; \tag{5.31}$$

$$\text{Boundary conditions: } \delta\hat{T}(\boldsymbol{r}, z, t) = 0, \quad \text{if } \begin{cases} |\boldsymbol{r}| &= r_{\text{clad}}, \\ z &= 0, \\ z &= L. \end{cases} \tag{5.32}$$

Time-stepping scheme. We use implicit Euler time-stepping to advance the heat solution with a (dimensionless) time step $\delta\hat{t}$. The total time is denoted by $\hat{t}_{\max}$. For n uniform time intervals, $\delta\hat{t} = \hat{t}_{\max}/n$. The implicit Euler scheme yields

$$\delta\hat{T}_{n+1} - \delta\hat{t}\alpha_0\hat{\Delta}(\delta\hat{T}_{n+1}) = \delta\hat{T}_n + \delta\hat{t}Q_0\hat{Q}(\hat{I}_{\{s,p\},n}), \tag{5.33}$$

where the argument for the heat deposition (Maxwell fields) is taken from the previous time step; it would not be feasible to implicitly compute the nonlinear source term which itself requires solving a nonlinear Maxwell system. Because of that, the scheme is not fully implicit hence unconditional stability is not given. Instead, a suitable small time step providing stability and sufficient accuracy in time is needed. The scheme can be seen as a fully implicit Euler method with one Picard iteration. Similar time-stepping schemes have been studied in the context of DPG methods for linear problems in [52, 109].

5.5 Short fiber scaling

The equivalent short fiber. A real fiber amplifier is about 5–20 meters long. The number of wavelengths in longitudinal direction is on the order of millions or tens of millions. A 3D vectorial finite element discretization with high order of approximation is computation-

ally expensive: many degrees of freedom are required per wavelength to resolve the fields accurately. A state-of-the-art compute node is currently capable of solving the proposed model for $\mathcal{O}(10^2)$ wavelengths. Due to numerical pollution (cf. Section 4.2), the simulation of a full-length fiber is not feasible at this moment, even with a scalable parallel code. Consequently, the proposed high-fidelity model may either be viewed in the context of a multi-fidelity approach together with simplified models, or we must find a scaling argument that enables us to compute on a short fiber that preserves the physical quantities of interest from a full-length fiber with sufficient accuracy. The latter approach has been proposed for a scaled CMT model [43], since even with much simpler models, simulation of fiber amplifiers remain computationally challenging today. Scaling only in the longitudinal direction makes physical sense for the fiber amplifier problem since the fiber waveguide performance is primarily derived from the transverse distribution of the index of refraction, and the refractive index remains relatively uniform along the fiber length. We corroborate a short fiber scaling argument through numerical results for gain polarization, and we are able to show a more rigorous mathematical argument for the scaling of thermal coupling in the fiber.

Gain scaling. First, we aim to make an argument that gain can be simulated in a short fiber of a length $\tilde{L}$ much smaller than the real fiber length L. It seems natural to introduce an artificial gain scaling term proportional to $L/\tilde{L}$ to amplify the gain proportionally to the shortening in fiber length. Indeed, for a CMT model, it is possible to show a more rigorous gain scaling argument in that way [43]. In the nonlinear Maxwell problem, a convincing mathematical scaling argument may be hard to show, if possible at all, and at this point we restrict ourselves to numerical experiments. We introduce a non-dimensional *short fiber gain amplifier* $\tilde{g}_a$ in the Maxwell system:

$$\boxed{\begin{aligned} \hat{\nabla} \times \hat{\boldsymbol{E}}_k &= -i\hat{\omega}_k \hat{\boldsymbol{H}}_k, \\ \hat{\nabla} \times \hat{\boldsymbol{H}}_k &= in^2 \hat{\omega}_k \hat{\boldsymbol{E}}_k - n\ell_0 g_0 \tilde{g}_a \hat{g}_k(\hat{\boldsymbol{E}}_{\{s,p\}}) \hat{\boldsymbol{E}}_k. \end{aligned}} \tag{5.34}$$

Purely in the context of nonlinear gain, this scaling term could be viewed as an increase of the dopant population concentration $\mathcal{N}_{\text{total}}^{\text{Yb}}$ proportional to $\tilde{g}_a$.

Heat scaling. The goal of the gain scaling is to obtain a field intensity (power distribution) in the short fiber that simulates the laser gain in a real-length fiber. The heat deposition depends only on the field intensity (cf. (5.28)). To simulate the heating in a fiber of length L, given a Maxwell solution on the short fiber of length $\tilde{L}$, a change of coordinates can be used to pull back the heat problem of the real-length fiber to a short domain. Suppose $L = 10$ m $= 10^6 l_0$, and $\tilde{L} = 100$ μm $= 10 l_0$ (recall that l_0 denotes the characteristic length scale—see Table A.3). Let

$$\hat{z} = \frac{L}{\tilde{L}} \tilde{z}, \tag{5.35}$$

so that

$$0 < \hat{z} < 10^6 \;\Leftrightarrow\; 0 < \tilde{z} < 10. \tag{5.36}$$

Denote $\tilde{\alpha}_z := \tilde{L}/L$; here, $\tilde{\alpha}_z = 10^{-5}$. Then,

$$\frac{\partial}{\partial \hat{z}} = \frac{\partial}{\partial \tilde{z}} \frac{\partial \tilde{z}}{\partial \hat{z}} = \tilde{\alpha}_z \frac{\partial}{\partial \tilde{z}} \,, \tag{5.37}$$

$$\hat{\Delta} = \frac{\partial^2}{\partial \hat{x}^2} + \frac{\partial^2}{\partial \hat{y}^2} + \frac{\partial^2}{\partial \hat{z}^2} = \frac{\partial^2}{\partial \hat{x}^2} + \frac{\partial^2}{\partial \hat{y}^2} + \tilde{\alpha}_z^2 \frac{\partial^2}{\partial \tilde{z}^2} \,. \tag{5.38}$$

That is, the scaling yields an anisotropic diffusion operator. Equivalently, we may write

$$\hat{\Delta} = \tilde{\nabla} \cdot \left(\tilde{\Lambda}_z \tilde{\nabla} \right), \tag{5.39}$$

where

$$\tilde{\Lambda}_z = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \tilde{\alpha}_z^2 \end{bmatrix} \text{ and } \tilde{\nabla} = \begin{bmatrix} \partial/\partial \hat{x} \\ \partial/\partial \hat{y} \\ \partial/\partial \tilde{z} \end{bmatrix}. \tag{5.40}$$

We obtain the non-dimensional *short fiber heat equation*:

$$\boxed{\frac{\partial(\delta\hat{T})}{\partial\hat{t}} - \alpha_0\tilde{\nabla}\cdot\left(\tilde{\Lambda}_z\tilde{\nabla}(\delta\hat{T})\right) = Q_0\hat{Q}(\hat{I}_{\{s,p\}}).} \tag{5.41}$$

Intuitively, this makes sense because discrete heat solution points from a full-length fiber are "compressed" by a factor of $\tilde{\alpha}_z$ in z. Consequently, these points should experience very little diffusion in z. Another way to view this rescaling is to say that each element in z is now solving the heat equation for a much longer distance in z, which is justified by the fact that the solution to the heat equation is very smooth. Oscillations in the temperature along z may occur due to wave propagation phenomena such as transverse mode beating. But such phenomena are expected to occur at a scale linked to a certain number of wavelengths in the Maxwell solution (and thus smooth enough within each element). Therefore, the proposed short fiber heat equation is able to capture any physical heating phenomena in the real fiber due to active gain as long as enough wavelengths are computed in the Maxwell problem to exhibit the relevant wave phenomena. While the scaling argument in the Maxwell system was artificial and cannot be guaranteed to reproduce the correct physics without further investigation, the scaled heat equation does reproduce the physical results from a real-length fiber.

5.6 Numerical scheme and results

In this section, we omit the "hat" symbol for non-dimensional quantities; instead, every symbol is now understood to be non-dimensional and the "hat" symbol is overloaded to indicate trace unknowns on the mesh skeleton in the broken DPG formulation. Let Ω denote the bounded, computational fiber domain given by $\Omega := \Omega_t \times (0, L) \subset \mathbb{R}^3$, where $\Omega_t := \{(x, y) : x^2 + y^2 < r_{\text{clad}}^2\}$ is the (cross-sectional) transverse domain, and L is the length of the (fiber) domain; the boundary is denoted by $\Gamma \equiv \partial\Omega$; and Ω_h is a suitable finite element

mesh with mesh skeleton Γ_h. The operator form of the active gain Maxwell problem is:

$$\begin{cases} \nabla \times \boldsymbol{E}_k + i\omega_k \boldsymbol{H}_k = \boldsymbol{0} & \text{in } \Omega, \\ \nabla \times \boldsymbol{H}_k - in^2\omega_k \boldsymbol{E}_k + n\ell_0 g_0 \tilde{g}_a g_k \boldsymbol{E}_k = \boldsymbol{0} & \text{in } \Omega, \\ \boldsymbol{n} \times \boldsymbol{E}_k = \boldsymbol{n} \times \boldsymbol{E}_{0,k} & \text{on } \Gamma, \end{cases} \tag{5.42}$$

where $k \in \{s, p\}$, $\boldsymbol{n}$ is the outward unit normal, and with appropriate boundary data $\boldsymbol{E}_{0,k}$. The corresponding broken ultraweak formulation is:

$$\begin{cases} \boldsymbol{E}_k, \boldsymbol{H}_k \in (L^2(\Omega))^3, \ \hat{\boldsymbol{E}}_k \in \hat{\mathcal{U}}_k, \ \hat{\boldsymbol{H}}_k \in H^{-1/2}(\text{curl}, \Gamma_h), \\ (\boldsymbol{E}_k, \nabla_h \times \boldsymbol{F}) + \langle \boldsymbol{n} \times \hat{\boldsymbol{E}}_k, \boldsymbol{F} \rangle_{\Gamma_h} + i\omega_k (\boldsymbol{H}_k, \boldsymbol{F}) = 0, \qquad \boldsymbol{F} \in H(\text{curl}, \Omega_h), \\ (\boldsymbol{H}_k, \nabla_h \times \boldsymbol{G}) + \langle \boldsymbol{n} \times \hat{\boldsymbol{H}}_k, \boldsymbol{G} \rangle_{\Gamma_h} \\ \qquad -i\omega_k (n^2 \boldsymbol{E}_k, \boldsymbol{G}) + \ell_0 g_0 \tilde{g}_a (n g_k \boldsymbol{E}_k, \boldsymbol{G}) = 0, \qquad \boldsymbol{G} \in H(\text{curl}, \Omega_h), \end{cases} \tag{5.43}$$

where $\hat{\mathcal{U}}_k := \{\hat{\boldsymbol{q}} \in H^{-1/2}(\text{curl}, \Gamma_h) : \boldsymbol{n} \times \hat{\boldsymbol{q}} = \boldsymbol{n} \times \boldsymbol{E}_{0,k} \text{ on } \Gamma\}$.

The fiber parameters used in the numerical simulations for the proposed model are given in Table A.5. Results are shown for a core-pumped, co-pumped, fiber amplifier where the pump and signal fields are both excited with their respective (x-polarized) fundamental mode at the fiber input. That is, $\boldsymbol{E}_{0,k} := (E^x_{0,k}, 0, 0)$, and

$$E^x_{0,k}(r, \theta, z) = \begin{cases} \Psi_{01,k}(r, \theta), & \text{if } z = 0, \\ 0, & \text{otherwise on } \Gamma, \end{cases} \tag{5.44}$$

where $\Psi_{01,k}$ is the transverse electric field of the LP_{01} mode (see (4.15)). At the fiber output, a perfectly matched layer is used to avoid (artificial) reflection of the wave (see Section 2.5.2). At the radial cladding boundary, we prescribe idealized perfect electrical conductor boundary conditions (i.e., vanishing tangential electric field) which is justified by the exponential decay of the core-guided modes in the cladding. The fiber geometry is discretized with isoparametric prismatic and hexahedral elements, as shown in Figure 4.10.

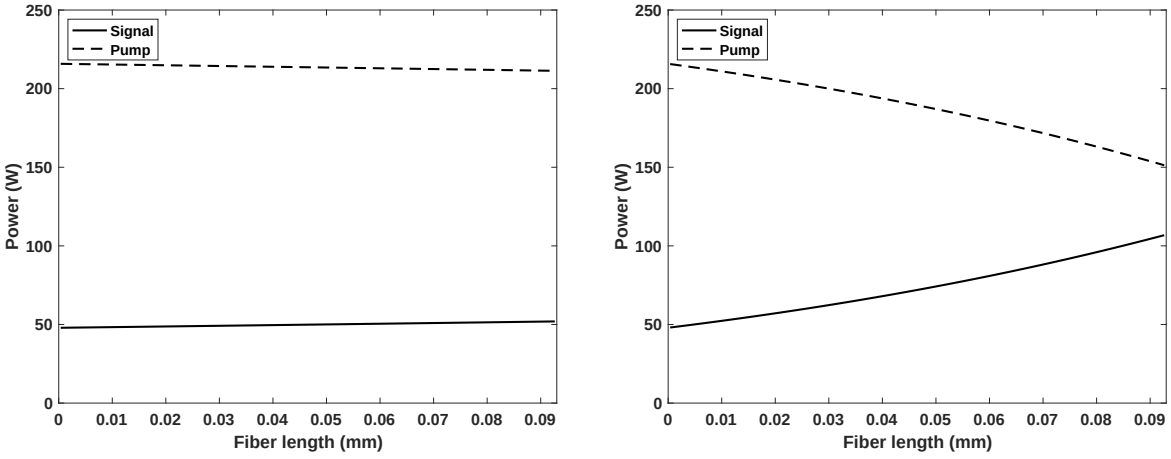
The (uniform) order of approximation is at least $p = 5$ for all numerical simulations. The weakly-coupled Maxwell systems are solved via Picard iteration.

Active gain amplification. First, we investigate the cross-sectional power flux inside the fiber amplifier along the longitudinal (z-)axis. Recall that the power flux through a surface orthogonal to the fiber axis can be computed directly from the trace unknowns in the DPG broken ultraweak formulation (cf. Section 4.1.2). Figure 5.2 illustrates the effect of the artificial gain amplification term $\tilde{g}_a$: it shows the distribution of the signal and pump optical powers along the length of the fiber. For a fiber of about 120 wavelengths, a small (relative to the ratio $L/\tilde{L}$) value of $\tilde{g}_a = 10^2$ causes almost no exchange of power between pump and signal fields. For $\tilde{g}_a = 4 \cdot 10^3$, the gain term suffices to transfer the power of the pump into the signal within the short fiber.

Algorithm 1: Nonlinear gain problem: Picard iteration

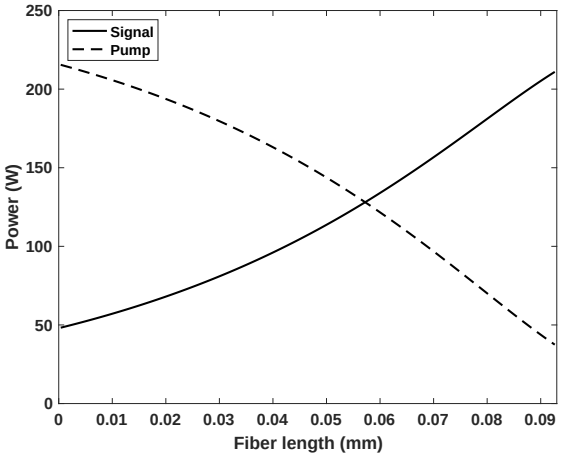
$\boldsymbol{E}_s^{(0)} \leftarrow \mathbf{0}$
$\boldsymbol{E}_p^{(0)} \leftarrow \mathbf{0}$
for $i = 1$ **to** $i_{\max}$ **do**
  $\boldsymbol{E}_s^{(i)} \leftarrow$ solve with $g_s := g_s\big(\boldsymbol{E}_s^{(i-1)}, \boldsymbol{E}_p^{(i-1)}\big)$
  $\boldsymbol{E}_p^{(i)} \leftarrow$ solve with $g_p := g_p\big(\boldsymbol{E}_s^{(i-1)}, \boldsymbol{E}_p^{(i-1)}\big)$
  if $\|\boldsymbol{E}_s^{(i)} - \boldsymbol{E}_s^{(i-1)}\| / \|\boldsymbol{E}_s^{(i)}\| < \varepsilon$ **then**
    break
  end if
end for

The stopping criterion for the nonlinear solver is based on the relative change of the signal field measured in the L^2 norm. Note that the pump field could also be used in the stopping criterion. However, the amplified laser is the primary field of interest in our simulations, hence we have chosen to solely base convergence on the signal field. The implementation of the Picard iteration is given in Algorithm 1. Alternatively, one could use the DPG residual directly as a stopping criterion, but this would add significant computational cost in every iteration. Preferably, the residual should only be computed at the very end of each nonlinear solve in order to determine which elements to refine in an adaptive mesh refinement

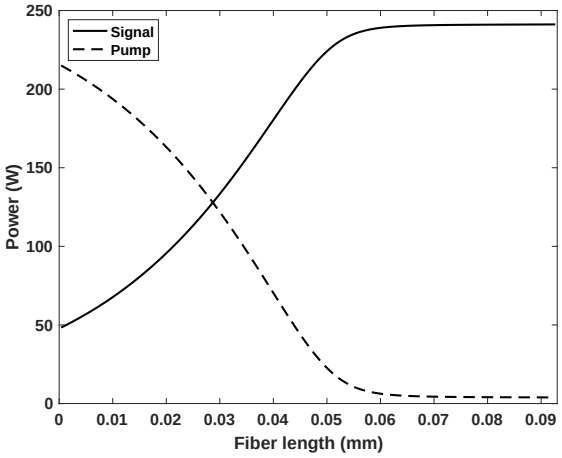


(a) $\tilde{g}_a = 10^2$

(b) $\tilde{g}_a = 10^3$



(c) $\tilde{g}_a = 2 \cdot 10^3$

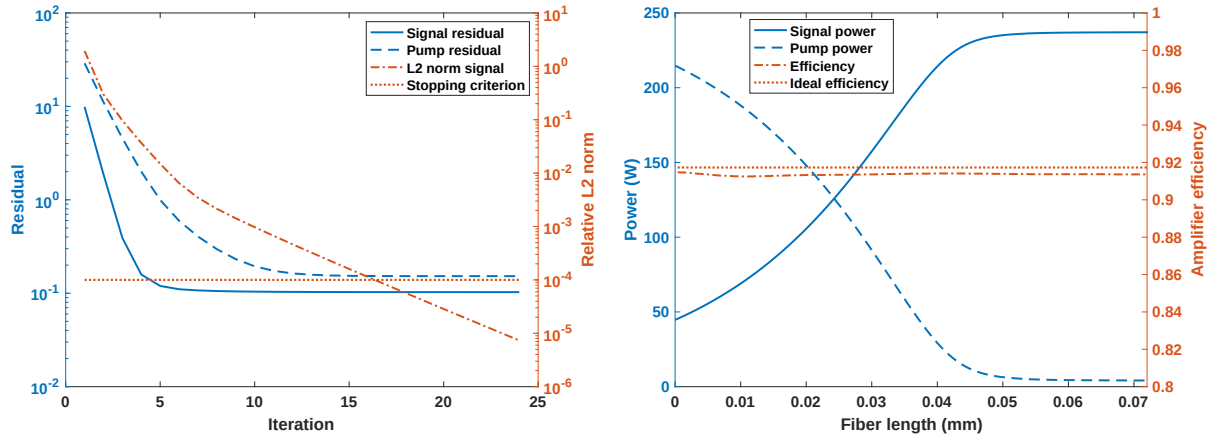


(d) $\tilde{g}_a = 4 \cdot 10^3$

Figure 5.2: Active gain Maxwell simulation for 120 wavelengths: power distribution along the fiber amplifier in the signal and the pump fields for different gain scaling terms $\tilde{g}_a$. Even within a short fiber, the entire pump energy can be transferred into the signal if the gain amplification is scaled artificially.

(see Section 4.3). The convergence of the residual for each field is plotted alongside the convergence of the relative change in L^2 and its stopping criterion ($\varepsilon = 10^{-4}$) in Figure 5.3a. One validation of the model itself is presented in Figure 5.3b, showing the obtained pump efficiency at different points along the fiber, compared to the ideal efficiency $\lambda_p/\lambda_s \approx 91.7\%$. As expected, the optical-to-optical efficiency in the fiber is slightly below this ideal efficiency since the light guided in the core falls into transverse modes which do not uniformly overlap

the core region, which means that the gain is not perfectly/maximally saturated.



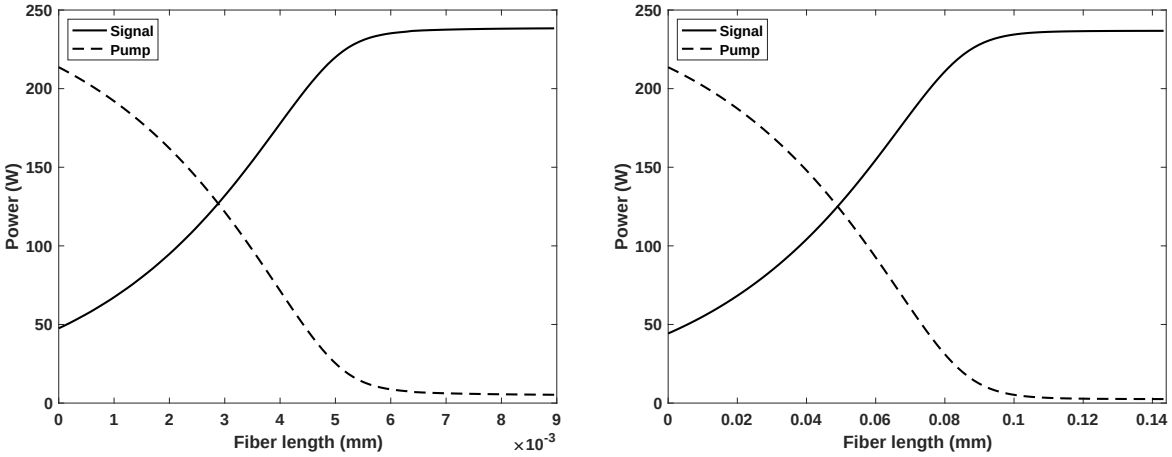
(a) Residual in nonlinear solve

(b) Amplifier efficiency

Figure 5.3: Active gain Maxwell simulation for 120 wavelengths: (a) the DPG residual, measured in the adjoint graph norm, converges in the Picard iterations; and (b) the efficiency of the power transfer from the pump into the signal field is near the ideal amplifier efficiency.

Gain scaling experiment. The goal of introducing the artificial gain scaling term $\tilde{g}_a$ was to preserve certain quantities of interest in the scaled model. A numerical verification for the preservation of the power distribution in fibers of different lengths is shown in Figure 5.4. We compare a fiber of 240 wavelengths with $\tilde{g}_a = 2.5 \cdot 10^3$ to a shorter fiber of 15 wavelengths and $\tilde{g}_a = 4 \cdot 10^4$ (appropriately scaled by a factor of 16). At least qualitatively, the same power distribution is obtained for both fibers.

Table 5.1 summarizes the number of iterations needed in the nonlinear solver until the stopping criterion is reached. As $\tilde{g}_a$ is increased for a fiber of fixed length, the number of iterations increases, indicating the stronger nonlinearity of the system—previously illustrated by Figure 5.2. On the diagonals of the table (starting from top right towards the bottom left) the number of iterations remains constant. These diagonals represent the "gain scaling" of a fiber: moving along them means increasing the fiber length and decreasing $\tilde{g}_a$ by the same factor.



(a) $15\lambda,\ \tilde{g}_a = 4 \cdot 10^4$ (b) $240\lambda,\ \tilde{g}_a = 2.5 \cdot 10^3$

Figure 5.4: Active gain Maxwell simulation: scaling experiment. Approximately the same power distribution (for both signal and pump field) is obtained for two fibers of different length by scaling the gain term $\tilde{g}_a$ appropriately.

Fiber length (λ)	Gain scaling coefficient $\tilde{g}_a$					
	$1.25 \cdot 10^3$	$2.5 \cdot 10^3$	$5 \cdot 10^3$	10^4	$2 \cdot 10^4$	$4 \cdot 10^4$
15	3	4	4	4	8	15
30	4	4	5	9	16	-
60	4	5	10	16	-	-
120	5	10	15	-	-	-

Table 5.1: Active gain Maxwell simulation: number of iterations until the convergence criterion is satisfied for different fiber lengths and gain scaling coefficients. More Picard iterations are needed for stronger gain amplification and/or longer fibers.

While these numerical verifications do not rigorously prove that scaled short fibers reproduce accurate power distributions for real-length fibers where the scaling factor is much larger, they do indicate that the fiber may at least be scaled to some extent in our model while preserving the signal and pump power curves with sufficient accuracy.

DPG primal heat. After solving the initial Maxwell problem at ambient temperature, time is advanced via the time-stepping of the heat equation. The operator form of the heat

problem is:

$$\begin{cases} \frac{\partial(\delta T)}{\partial t} - \alpha_0 \nabla \cdot (\tilde{\Lambda}_z \nabla(\delta T)) &= Q_0 Q(I_{\{s,p\}}) \quad \text{in } \Omega \times (0, t_{\max}], \\ \delta T &= 0 \quad \text{on } \Gamma \times [0, t_{\max}], \\ \delta T &= 0 \quad \text{in } \Omega \times \{0\}. \end{cases} \tag{5.45}$$

To discretize the heat problem, we use the primal DPG formulation which is based on the classic variational formulation and provides a conforming (continuous) discretization of the field variable. The breaking of the test space, however, incurs additional degrees of freedom on the mesh skeleton describing the heat flux. The method is thus more expensive than the weak Galerkin formulation. We have made this choice in anticipation of exploiting adaptivity in our model via the DPG residual in future studies of the thermal perturbations linked to instability phenomena in fiber amplifiers. The DPG broken primal formulation with implicit Euler time-stepping is:

$$\begin{cases} \delta T_{n+1} \in H_0^1(\Omega),\ \hat{\sigma}_{n+1} \in H^{-1/2}(\Gamma_h), \\ (\delta T_{n+1}, v) + \delta t \alpha_0 (\tilde{\Lambda}_z \nabla(\delta T_{n+1}), \nabla_h v) - \delta t \alpha_0 \langle \hat{\sigma}_{n+1}, v \rangle_{\Gamma_h} = \\ \qquad\qquad (\delta T_n, v) + \delta t Q_0 (Q(I_{\{s,p\}}), v), \quad v \in H^1(\Omega_h), \end{cases} \tag{5.46}$$

where $n = 0, 1, \ldots, N-1$, and $\delta t = t_{\max}/N$. The additional (trace) unknown $\hat{\sigma}_{n+1}$ describes the normal heat flux across element boundaries on the mesh skeleton Γ_h; it is discretized as the normal trace of $H(\text{div})$-conforming elements (cf. (2.35)–(2.36) and Section 3.4.2). We equip the test space with the standard energy norm. With this setup, the coupled Maxwell/heat problem now requires the use of elements of the entire H^1–$H(\text{curl})$–$H(\text{div})$–L^2 exact sequence.

Figure 5.5 displays the overall numerical scheme for the coupled problem. The weakly-coupled Maxwell systems are solved via Picard iteration, where the gain polarization is updated once per iteration (as shown in Algorithm 1), and the simulation's time step is ad-

vanced via implicit Euler time-stepping in the heat equation. The fiber material parameters (thermally induced refractive index perturbation) and heat source (heat deposition in the fiber) are updated once per time step.

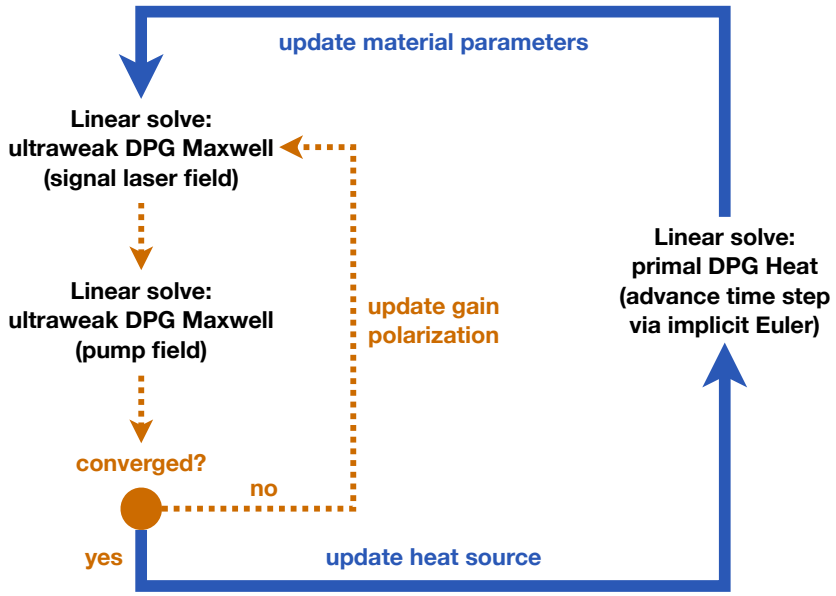
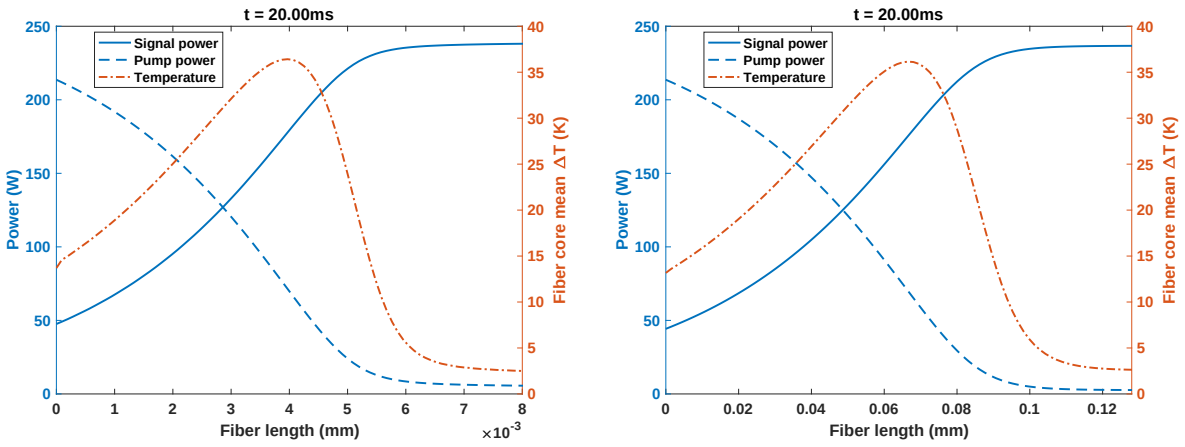


Figure 5.5: Coupled Maxwell/heat simulation: overview of the numerical scheme. Picard iterations are used for linearizing the weakly-coupled Maxwell systems, and implicit Euler time-stepping for the heat equation advances the time step of the simulation. Material parameters and heat deposition in the fiber are updated once per time step.

Heat scaling experiment and induced thermal perturbations. In the numerical experiments, $\delta t = 0.1$ ms has proven to be sufficiently small to provide stability in the heat solve. For a given solution to the nonlinear Maxwell problem, the transient heat equation attains steady-state after circa 15 ms. Figure 5.6 shows the temperature distribution along the fiber amplifier after 20 ms for two different fibers: one with 15 wavelengths (Figure 5.6a) and one with 240 wavelengths (Figure 5.6b). The results indicate that the heat distribution in the fiber can be computed accurately on a short fiber, as expected by the anisotropic diffusion operator scaling argument (cf. Section 5.5). In other words, to obtain an accurate heat curve in the scaled model, one only requires the accurate power distribution (or intensity) of the signal and pump field along the fiber.



(a) $15\lambda,\ \tilde{g}_a = 4 \cdot 10^4,\ \tilde{\alpha}_z = 1.2 \cdot 10^{-6}$ (b) $240\lambda,\ \tilde{g}_a = 2.5 \cdot 10^3,\ \tilde{\alpha}_z = 1.92 \cdot 10^{-5}$

Figure 5.6: Coupled Maxwell/heat simulation: scaling experiment. Approximately the same temperature distribution is obtained after 200 time steps ($\delta t = 0.1$ ms) for two fibers of different length by scaling the anisotropic heat diffusion coefficient $\tilde{\alpha}_z$ appropriately.

To investigate the nonlinear effects induced by thermal perturbations in the fiber amplifier, we compute the solution to the weakly-coupled Maxwell systems in every time step of the heat equation with updated material parameters. The thermally induced perturbations are represented by the change of the material refractive index inside the fiber core and cladding. Figure 5.7 shows the refractive index plotted along x across a slice of the fiber orthogonal to the longitudinal axis (i.e., a fiber cross-section) close to where the peak temperature occurs. The plot demonstrates that the refractive index profile is perturbed significantly compared to the step-index profile at ambient temperature. Consequently, the guided propagating fields may be perturbed in a significant way as the temperature develops inside the fiber amplifier. The implications of this refractive index perturbation are discussed further and analyzed in Section 6.4.

Conclusion. Our nonlinear fiber amplifier model supports both active gain and an integrated thermal response. The simulation incorporates two weakly-coupled time-harmonic 3D vectorial Maxwell systems for the propagating electromagnetic signal and pump fields. This coupling between the fields occurs both through the active gain mechanism in the fiber

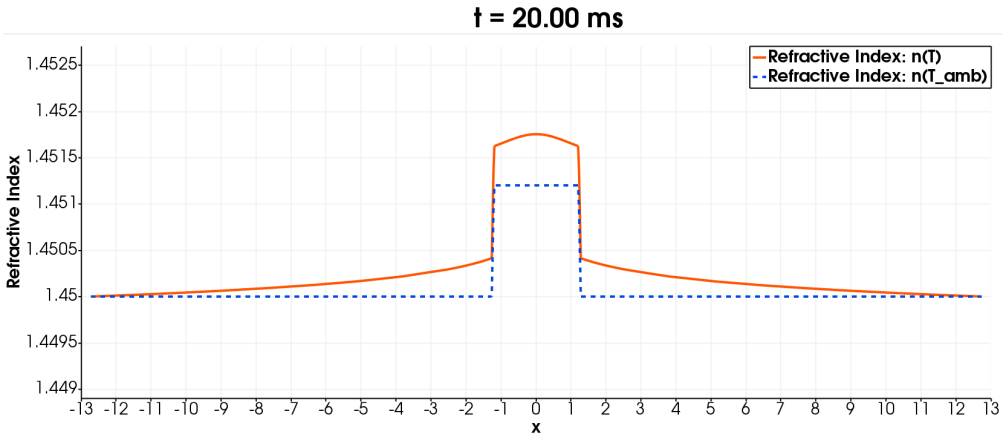


Figure 5.7: Coupled Maxwell/heat simulation: refractive index profile inside the fiber amplifier, plotted along the x-axis in a slice orthogonal to the longitudinal fiber axis. Near the regions with peak temperature, we observe a significant thermally induced perturbation compared to the step-index profile at ambient temperature.

core region and through the thermally induced refractive index perturbations that result from the heat deposition caused by the lasing (see Figure 5.7). The steady-state ytterbium ion population concentrations are updated according to how the irradiances of the pump and signal fields evolve along the length of the fiber. As expected, the pump field experiences loss (negative gain), and the laser signal experiences amplification (positive gain) such that the ideal optical-to-optical efficiency is never surpassed (see Figure 5.3b). Furthermore, this gain via stimulated emission is ultimately what drives the heating in the fiber, resulting in a peak heat load in the region of greatest transfer of energy from the pump field to the signal field (see Figure 5.6).

The proposed model differs from the more common BPM or CMT approaches for numerical simulation of fiber lasers in many ways. Most importantly, our model makes very few assumptions on the propagating fields in order to provide a high-fidelity simulation tool. This approach leads to high computational cost when solving the fiber for many wavelengths in a high-order finite element discretization. The DPG method is used for discretizing both the Maxwell systems and the heat equation, yielding a stable discretization with a built-in error indicator suitable for *hp*-adaptivity. The data from the numerical solution of the

coupled Maxwell/heat 3D fiber amplifier model enables the analysis of the interplay between thermal perturbations of the material and the propagating electromagnetic fields with great accuracy. Therefore, we believe the proposed model is capable of capturing the onset of modal instabilities and eventually providing new insight into the TMI. Additionally, the generality of the model makes it possible to add further relevant nonlinearities, compute counter- or bi-directional pumping configurations, or study birefringent (anisotropic) fibers, all of which are difficult to realize in simpler fiber models due to their inherent assumptions. The high accuracy of our model can also be exploited to validate typical approximations made in lower-fidelity models.

In order to make the computation feasible for a full-length fiber, we introduced a longitudinal scaling for the Maxwell and heat problem. This was accomplished with an artificial material parameter that effectively enhances the gain per unit length, which was shown to preserve certain quantities of interest, e.g., fiber efficiency, within the tested parameter regime. Of particular importance, we were able to scale the heat equation naturally through an anisotropic diffusion operator obtained by a change of coordinates such that the only source of artificial error introduced into the amplifier model by this scaling is through the gain along the fiber.

Chapter 6

Large-Scale Fiber Simulations

In this chapter, we conduct simulations of our 3D fiber amplifier model at large scale. This effort combines the software capabilities developed in Chapter 3 with the mathematical fiber model presented in Chapter 5. All simulations shown in this chapter were computed on TACC's Stampede2 supercomputer on Skylake (SKX) compute nodes.

First, we investigate the computational performance of the fiber amplifier simulation. We discuss optimal configurations for hybrid MPI/OpenMP computation and the speedup obtained from intra-node, shared-memory OpenMP acceleration. Then, we analyze inter-node, distributed-memory MPI weak and strong scaling. A comparison of the nested dissection solver of Section 3.3 to the parallel MUMPS solver is also shown. Weak-scaling results are reported for up to 512 compute nodes with more than 24k cores on Stampede2. These computational aspects of the model are discussed in Section 6.1.

In Section 6.2, we present numerical results of nonlinear active gain amplification in an ytterbium-doped fiber amplifier, both for co-pumped and counter-pumped fiber configurations. Section 6.3 expands upon the results by Nagaraj et al. (2018) for a passive Raman gain amplifier with significantly longer fibers and larger gain amounts. We conclude in Section 6.4 with a discussion on mode coupling in fibers with grated refractive index perturbations and cross-validate our results with a coupled mode theory model simulation.

6.1 Computational performance

Tuning the computational performance of the fiber amplifier model at extreme scale requires careful optimization on both the intra-node, shared-memory level and inter-node, distributed-memory level. We begin by studying the shared-memory parallelism obtained via OpenMP threading.

Remark: All runtime results displayed in this section are based on solving one iteration of the linearized Maxwell problem for the signal laser (the pump solve is equally expensive). In the coupled Maxwell/heat model, the computational complexity is dominated by assembling and solving the Maxwell systems since the heat equation is discretized with much fewer DOFs and only solved once per time step. Therefore, the scaling results presented in this section are representative for the overall scaling of fiber amplifier model computation.

6.1.1 OpenMP acceleration

Stampede2's SKX compute nodes feature 48 distinct compute cores, all of which have shared access to the on-board main memory.[1] Our fiber amplifier FE simulation uses OpenMP threading for various tasks to exploit the available parallelism. Most importantly, whenever a loop over elements is computed, a multi-threaded OpenMP environment enables parallel processing of the elements. For instance, the element assembly, including determination of locally optimal test functions, and the element residual computation are parallelized via OpenMP threading. Other examples include post-processing steps such as computation of the optical power along the fiber, which can be independently processed for each cross-section of the fiber, or compilation of output data for visualization. In all of these tasks, we employ dynamic scheduling of OpenMP threads: each thread computes one element at a time; once it finishes computing the element, the thread is assigned the next element.

[1] Stampede2's SKX compute nodes are two-socket NUMA architectures: each socket has its own main memory attached; all of the main memory can be accessed from cores on either socket, although memory access times differ depending on the memory location.

This is contrary to static scheduling where the workload is distributed among threads a-priori. Dynamic scheduling—despite its scheduling overhead—performs better than static scheduling because the workload per element is variable, depending on the element type (the geometry is discretized with both prisms and hexahedra) and order of approximation. Some tasks, e.g., computation of the residual, additionally require a parallel reduction to calculate the global value from the sum of element contributions. And, in a few cases, critical sections are used to force sequential computation where necessary: for example, we assemble the right-hand side vector from pre-computed element contributions sequentially which is more memory-efficient than performing a reduction over thread-private vectors. The shared-memory parallelization of the linear solve itself is limited to threading inside math kernel libraries called by the solver.

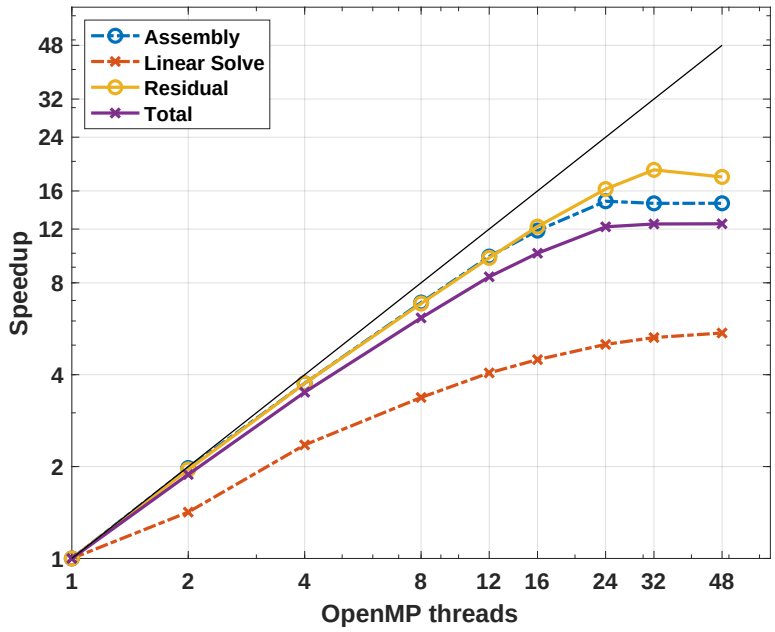


Figure 6.1: Acceleration with intra-node, shared-memory OpenMP parallelism for the fiber amplifier model with uniform polynomial order $p = 5$ for a fixed problem size of 16 wavelengths. The finite element simulation scales near-optimally (except linear solve) up to 12 OpenMP threads. It achieves a total maximum 12x speedup with 24 threads on a 48-core SKX compute node.

Figure 6.1 shows the (strong-scaling) speedup obtained by OpenMP parallelization on

a single compute node for a fiber of 16 wavelengths and uniform fifth-order discretization. Depicted are the speedup for the computationally most expensive steps and the total simulation time. The element assembly and computation of the residual are scaling near-linear initially, achieving a 10x speedup for 12 OpenMP threads. The linear solve via Intel's MKL Pardiso solver achieves a 4x speedup for 12 threads. The overall scaling noticeably deteriorates beyond 12 threads. The maximal speedup for the total simulation time is ca. 12x when using 24 threads (i.e., 50% efficiency). While each SKX compute node has 48 distinct compute cores, fast memory (cache) is generally shared among a few or many cores. Running additional concurrent threads therefore puts more pressure on the memory. By causing more frequent cache misses, this may lead to a deterioration of performance. The additional OpenMP scheduling overhead can also be significant.

6.1.2 Hybrid MPI/OpenMP configuration

In our fiber simulation, the shared-memory parallelism and its practical scaling limitations are an important consideration for optimizing the hybrid MPI/OpenMP performance of the code. This hybrid model combines the distributed-memory MPI parallelism with shared-memory OpenMP parallelism; a configuration can be chosen flexibly with one or multiple MPI processes per compute node. To coordinate inter-node, distributed-memory communication, at least one MPI process must run per compute node. As explained in Section 3.2, *hp*3D distributes degrees of freedom based on a grid partitioning where each MPI process owns one distinct subdomain. OpenMP thread parallelism is exploited within each subdomain by parallelizing the workload over its elements. While using multiple MPI processes per compute node increases the overall memory footprint, it can potentially improve the computational performance significantly. For example, the thread scheduling cost is reduced by using fewer threads per process, and the memory synchronization time is lower when working on distributed data rather than shared data. Regarding the latter point, note that if several MPI processes are used per compute node, the memory associated with one pro-

cess is not shared with the other despite using shared on-board memory. Therefore, memory operations on the logically distributed data can happen concurrently without the need for synchronization (e.g., to keep cache entries valid).

We now investigate the performance of various MPI/OpenMP configurations for the fiber model. These are specified as `MPI x OpenMP` configurations, e.g., `4 x 12` implying that 4 MPI processes are used per compute node with 12 OpenMP threads each. We only consider configurations that have a power-of-two number of MPI processes (for best efficiency in the nested dissection solver) and use exactly one thread per compute core (for optimal intra-node performance), i.e., `MPI x OpenMP = #cores`. With 48 cores per compute node, the following configurations are considered:

$$\texttt{MPI x OpenMP} \in \{\texttt{1 x 48}, \texttt{2 x 24}, \texttt{4 x 12}, \texttt{8 x 6}, \texttt{16 x 3}\}.$$

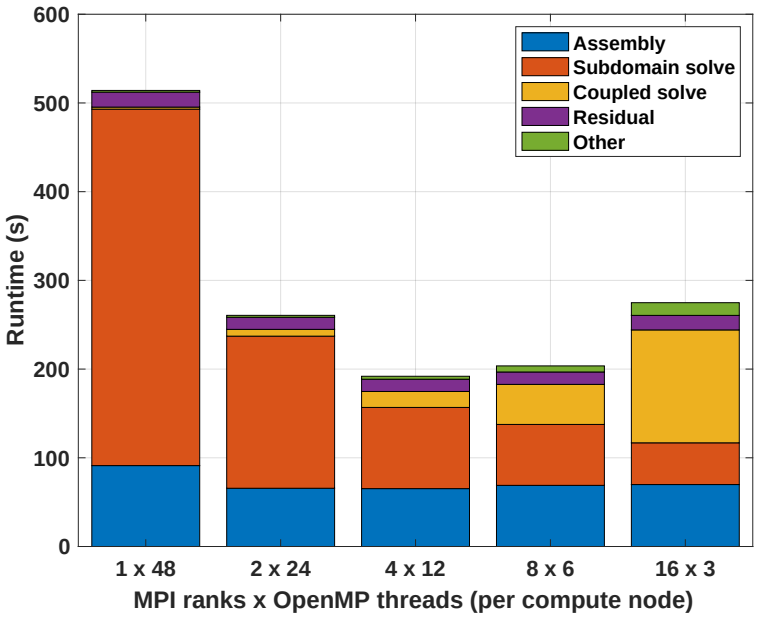


Figure 6.2: Runtime comparison of various hybrid MPI/OpenMP configurations for the fiber amplifier model with uniform polynomial order $p = 5$ for a fixed problem size (128 wavelengths) and number of compute nodes (4 SKX nodes with 48 cores/node). The finite element simulation performs best for 4 MPI processes with 12 OpenMP threads each (per compute node).

MPI	x	OpenMP	Subdomain size (DOFs)	Subdomain solve (s)	Coupled size (DOFs)	Nested levels	Coupled solve (s)
1	x	48	321 380	401.83	5 460	2	2.30
2	x	24	159 780	171.47	12 740	3	7.64
4	x	12	78 980	91.51	27 300	4	17.99
8	x	6	38 580	68.69	56 420	5	45.14
16	x	3	17 470	47.04	114 660	6	127.28

Table 6.1: Solver performance with various hybrid MPI/OpenMP configurations for the fiber amplifier model with uniform polynomial order $p = 5$ for a fixed problem size (128 wavelengths, 4 825 620 DOFs) and number of compute nodes (4 SKX nodes with 48 cores/node). The size of the subdomain and coupled interface problems depends on the choice of MPI processes per compute node.

Figure 6.2 shows the performance for each one of these configurations for the fiber amplifier model with 128 wavelengths and uniform fifth-order discretization, performed on four compute nodes (192 cores total). The plot displays the total runtime in seconds for each configuration, broken down into partial runtimes for the computationally most expensive parts. The runtimes for both assembly and residual computation are nearly identical for all configurations except for `1 x 48` being slightly slower. In the nested dissection solver, however, we observe large differences depending on the configuration. The solver runtime is split into the "subdomain solve" (static condensation of the subdomain interior onto subdomain interfaces—*preliminary step* in Section 3.3) and the "coupled solve" (nested solve of the global coupled interface problem—*steps 1–5* in Section 3.3). The important point to consider is how the choice of `MPI x OpenMP` configuration affects the solver algorithm since each configuration builds a unique nested dissection tree. For a fixed fiber length, using more MPI processes implies having more subdomains but each subdomain becomes smaller, hence the local problem can be solved faster; on the other hand, using more subdomains implies having additional interfaces that are coupled and must be solved with a deeper nested dissection tree. The effect is illustrated by the varying runtimes of the respective solves in Figure 6.2, and it is quantified in terms of degrees of freedom in Table 6.1. Depending on the hybrid configuration, the subdomain interior size varies between 17 470 and 321 380 DOFs

per MPI process while the coupled interface problem has between 5 460 and 114 660 DOFs. Note that these numbers exclude element interior and residual DOFs previously eliminated from the system. Assuming a binary tree structure, the nested dissection solver has between two and six levels depending on the number of subdomains. Other tree configurations (i.e., not binary) are possible as well, and we discuss some options shortly.

To sum up, we generally prefer using as much shared-memory parallelism as possible in our hybrid MPI/OpenMP approach, since every additional MPI process incurs an additional memory footprint. In other words, if two configurations perform the same overall it is preferable to use the one with less MPI processes and more OpenMP threading. However, the OpenMP parallelism does not scale well in our model beyond a certain number of cores (as demonstrated in Figure 6.1). Therefore, we have evaluated different hybrid configurations with more than one MPI process per compute node (see Figure 6.2) and found that performance can be significantly improved this way. In most cases, we have observed that the `MPI x OpenMP = 4 x 12` configuration is optimal and thus report numerical results for this choice, including for the MPI weak- and strong-scaling results presented later in this section.

6.1.3 Nested dissection tree

As pointed out in Section 3.3, there are different parameters for tuning the performance of the nested dissection solver. In particular, we can choose the subproblem size which determines the structure of the separator tree. Recall that `mSUB_PROCS` is the number of MPI processes participating in a single subproblem solve; `mSUB_PROCS` $= 2$ results in a binary tree, `mSUB_PROCS` $= 4$ in a quadtree, and so on. A subproblem has to be at least of size `mSUB_PROCS` $= 2$ in which case it involves two (neighboring) MPI processes solving a single interface problem.

Figure 6.3 depicts the runtime of the coupled solve via nested dissection for three

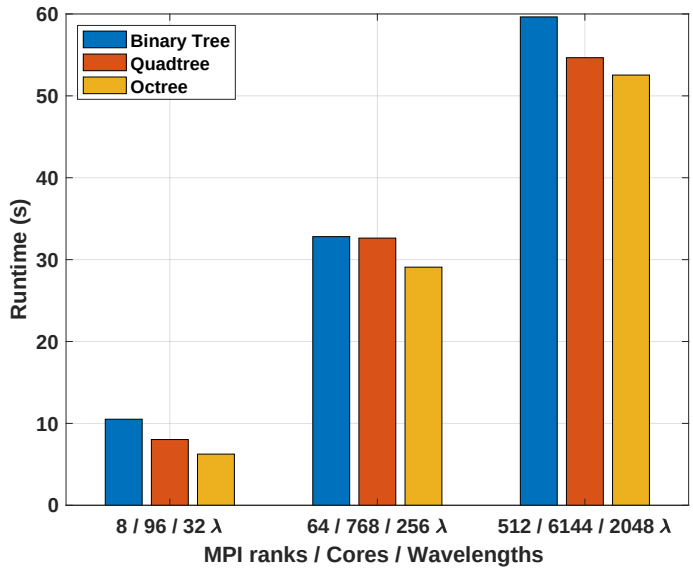


Figure 6.3: Runtime comparison of various tree configurations for the nested dissection solver. Runtimes are for the coupled linear solve of the fiber amplifier model with uniform polynomial order $p = 5$. The nested dissection solver performs best for an octree configuration, where the subproblem size is defined by eight neighboring subdomains (across seven interfaces).

different choices: binary tree, quadtree, and octree. For three different fiber lengths (32, 256, and 2 048 wavelengths), we observe that the runtime tends to decrease for larger subproblem sizes.

Nested tree type	Subproblem size (DOFs)	Nested levels (tree depth)		
		32 λ, 8 ranks	256 λ, 64 ranks	2 048 λ, 512 ranks
Binary tree	1 820	3	6	9
Quadtree	5 460	1.5	3	4.5
Octree	12 740	1	2	3

Table 6.2: Comparison of tree configurations in the nested dissection algorithm. The amount of work per subproblem and the number of subproblems depend on the user choice. For example, computing larger subproblems reduces the depth of the nested dissection tree.

Table 6.2 outlines the tree structure for these solver configurations. The number of "nested levels" represents the depth d of the corresponding tree, defined by $\texttt{mSUB\_PROCS}^d = \texttt{NUM\_PROCS}$. Thus, in some cases the depth is a decimal number. This notation expresses the fact that the root of the tree may consist of a smaller subproblem than the tree's re-

maining subproblems. For example, using eight MPI ranks (`NUM_PROCS` = 8) with quadtree configuration (`mSUB_PROCS` = 4) results in one level with two subproblems (each of size 5 460 DOFs constructed from four neighboring subdomains) and one additional smaller subproblem consisting of the remaining interface of size 1 820 DOFs and constructed by only two MPI processes, hence $d = 1.5$. By this convention, doubling the number of subdomains increases the depth of the binary tree by 1, the quadtree by 0.5, and the octree by 0.33.

From these experiments, we conclude that the fiber model computation with the distributed nested dissection solver can be somewhat accelerated by increasing the subproblem size, and thereby reducing the number of tree levels, as long as the subproblems can be efficiently solved with a parallel direct solver (e.g., MUMPS).

6.1.4 MPI weak scaling

In the effort to compute the fiber model with more wavelengths, weak scaling is the most important performance metric. Generally, we solve as many wavelengths as possible on a single compute node, given memory and time constraints, and then increase the compute resources (and MPI processes) while also increasing the fiber length (to keep the workload per processor approximately constant). The memory requirement per MPI process increases somewhat in this approach due to the redundant storage of the `NODES` data structure (cf. Section 3.2.1), which is quantified later. Otherwise, we primarily focus on the computational time of the fiber simulation in our weak-scaling analysis.

Weak-scaling analysis. Many parts of the fiber model FE simulation scale near-linearly, assuming a well-balanced workload across subdomains. This includes the computation of locally optimal test functions, the residual computation per subdomain, and computation of geometry and Dirichlet degrees of freedom, among others. The efficiency estimate of the linear solve relies on different estimates for different components of the solver: 1) The static condensation of the subdomain interior DOFs is linearly scalable—presuming a perfectly

balanced workload—since the work is independent for every MPI process (or subdomain); 2) The parallel scaling of the coupled interface solve via nested dissection corresponds to a typical divide-and-conquer algorithm, where the parallel efficiency declines as the recursive process proceeds with additional levels (i.e., larger tree depth in nested dissection); and 3) Retrieving the subdomain interior solution (both trace and field variable DOFs) is again an independent task for each subdomain that scales near-linearly when the workload is balanced.

Making a few assumptions about the fiber model discretization, we can estimate the nested dissection solver efficiency more precisely. Let P denote the number of processors (or subdomains) and suppose that we use a nested binary tree configuration (`mSUB_PROCS` $= 2$). Assuming constant workload on each interface between subdomains, (i.e., the same discretization on all fiber cross-sections), we denote the number of unknowns per interface as n_i. Then, solving the (dense) linear system corresponding to a single interface problem via LU factorization requires $\mathcal{O}(n_i^3)$ work. Overall, the computational work of the nested dissection solver increases only linearly with fiber length since the number of interfaces increases linearly and each interface DOF interacts only with neighboring interface DOFs (cf. Figure 3.10). The total amount of computational work for solving the coupled interface problem is $\mathcal{O}(Pn_i^3)$. The computational time, however, does not scale linearly since the interface problems cannot be solved independently in parallel but are coupled. Due to its hierarchical structure, the sequential path of the computation increases with the depth of the nested dissection tree, each level taking $\mathcal{O}(n_i^3)$ time; consequently, the overall time complexity is $\mathcal{O}(\log P\ n_i^3)$.

Weak-scaling results. Before discussing the numerical scaling results, we mention a few points about the weak-scaling test setup. Firstly, we only test inter-node (true distributed-memory) scaling, where the computational performance on one compute node serves as the "baseline". In the hybrid MPI/OpenMP code, this means the baseline uses 4 MPI processes with 12 OpenMP threads each. This way, the effects of faster intra-node communication

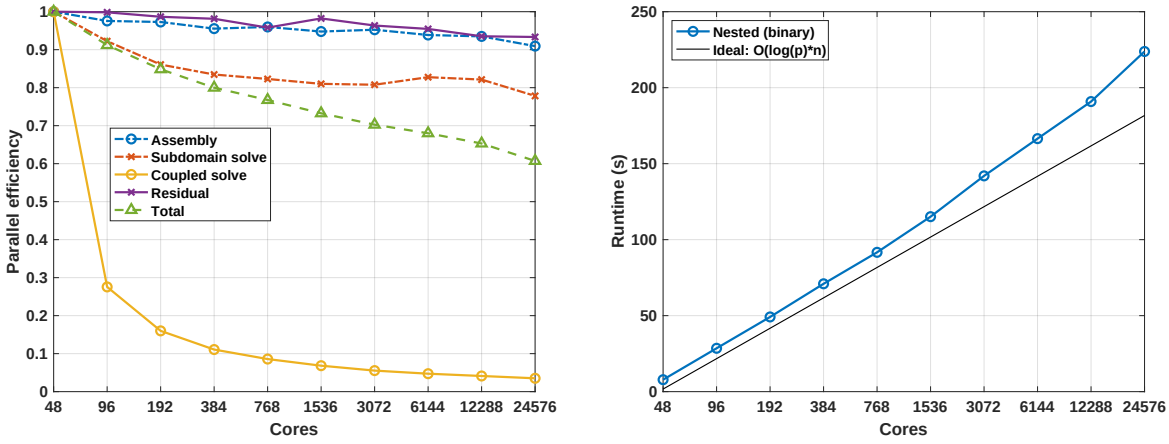
between MPI processes running on the same compute node are removed from the scaling results. Secondly, due to numerical pollution (cf. Section 4.2), increasing the fiber length requires increasing the order of discretization to maintain the same level of accuracy. In our weak-scaling test, however, we must keep the workload per processor constant to obtain valid results, thus the polynomial order used for all fiber lengths is the same. The discretization has uniform order $p = 6$ with 20 elements in the fiber cross-section (4 prisms, 16 hexahedra) and two elements per wavelength in the longitudinal direction. This suffices to yield accurate results for several thousand wavelengths. The baseline on one compute node is a fiber model with 32 wavelengths, whereas the largest instance, computed on 512 nodes, is a fiber of 16 384 wavelengths.

Number of wavelengths	Compute nodes	MPI ranks	Cores	DOFs per Maxwell solution	`NODES` memory (MB)
32	1	4	48	1 994 520	0.5
64	2	8	96	3 989 016	1.0
128	4	16	192	7 978 008	2.0
256	8	32	384	15 955 992	4.0
512	16	64	768	31 911 960	8.0
1 024	32	128	1 536	63 823 896	16.0
2 048	64	256	3 072	127 647 768	32.0
4 096	128	512	6 144	255 295 512	64.0
8 192	256	1 024	12 288	510 591 000	128.0
16 384	512	2 048	24 576	1 021 181 976	256.0

Table 6.3: Setup of weak-scaling tests for the fiber amplifier model with uniform polynomial order $p = 6$. The memory per MPI process increases due to the `NODES` data structure; this amount of storage for global data is modest, however, compared to the memory needed for storing the (distributed) solution DOFs of just one electromagnetic field.

Table 6.3 shows an overview of the weak-scaling setup. In addition to the fiber lengths and compute resources, it also displays the memory required (per MPI process) to store the global `NODES` array. As detailed in Section 3.2, the redundant storage imposes certain scalability constraints. However, as can be seen in Table 6.3, given the large amount of main memory (192 GB per compute node), these constraints are not currently a practical concern

for the fiber amplifier simulation. In the largest instance for a fiber of 16 384 wavelengths, each MPI process has to store only 256 MB of global mesh data. As suggested in the analysis in Section 3.2, this global data is significantly less than the distributed degree-of-freedom data which amounts to more than 16 GB for the electromagnetic field of the signal laser. We note that this number, and the DOFs shown in Table 6.3, do not include distributed geometry DOFs, residual DOFs, or solution DOFs for the electromagnetic field of the pump light and the temperature field of the heat solution. Additionally, two copies of all solution DOFs must be stored in order to enable the iterative solution process of the fixed point iteration for nonlinear laser gain computation and the time-stepping scheme used to discretize the transient heat problem.



(a) Parallel efficiency of the main subroutines (b) Runtime of coupled interface solve

Figure 6.4: Stampede2 weak-scaling results for the fiber amplifier model with uniform polynomial order $p = 6$. The nested dissection solver scales near-optimally up to 512 compute nodes (24 576 cores) and 2 048 MPI processes.

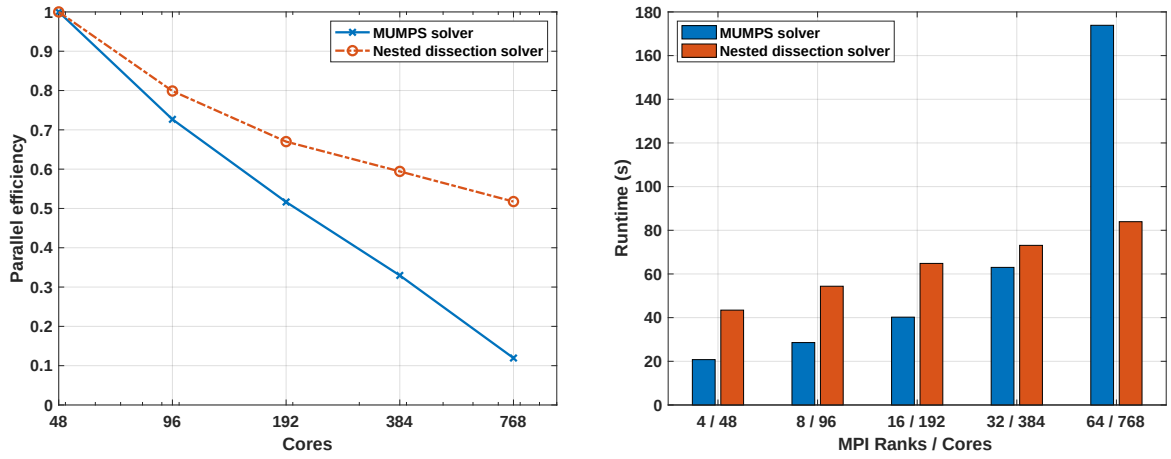
Figure 6.4 shows weak scaling of the fiber amplifier simulation with up to 512 compute nodes (24 756 cores) on Stampede2. The presented runtimes are averaged over at least ten samples in each case, and the nested dissection solver is built with a binary tree for all cases. Figure 6.4a depicts the parallel efficiency for the computationally most expensive steps of the simulation and overall efficiency. The obtained total efficiency declines gradually, at 80 percent for 8 compute nodes, 70 percent for 64 compute nodes, and 60 percent for 512

compute nodes. In other words, the total simulation time for a fiber of 16 384 wavelengths on 512 compute nodes is a bit less than double the simulation time for a fiber of 32 wavelengths on one compute node.

While the element matrix assembly, residual computation, and subdomain solve should in principle obtain near-linear scaling, we observe that in practice computing on many processors yields a (reproducible) variation in the computation time across all processors despite the equally distributed workload. Since all processors must wait for the slowest one to finish its computation, this leads to a slow but gradual efficiency decline: the simulation achieves ca. 90–95 percent efficiency for assembly and residual computation, and ca. 75–85 percent efficiency in the subdomain interior solve with the sparse direct MUMPS solver. The parallel efficiency of the coupled interface solve via nested dissection is expectedly poor because of the logarithmic increase in computation time, but this modest logarithmic increase enables good weak scaling overall. Figure 6.4b illustrates that the parallel algorithm performs close to the expected optimal estimate, with the runtime being proportional to $n \log_2 P$, where $n = 20$ seconds is the approximate computation time per level in the nested dissection tree.

Comparison with MUMPS solver. Before moving on to strong-scaling results, we compare the weak-scaling performance of the nested dissection solver with the distributed MUMPS solver. The test setup is similar to the previous one shown in Table 6.3, starting with 32 wavelengths on one compute node but using a lower polynomial order $p = 5$ to accommodate MUMPS's larger memory requirements. Figure 6.5 is a comparison of the linear solver performance: Figure 6.5a shows the parallel efficiency and Figure 6.5b the solver runtime in seconds. While neither of the parallel direct solvers can obtain linear scaling, the observed efficiency of the nested dissection solver is superior to MUMPS's parallel efficiency. When comparing the runtime, however, we can see that the MUMPS solver performs very well for small instances of the fiber model and has in fact a lower runtime for up to 256 wavelengths (on 8 compute nodes). The nested binary tree construction appears to add more

complexity to the solve than is necessary for these small problem instances. This is consistent with our findings in Section 6.1.3 where we observed a better performance when computing larger subproblems (e.g., an octree configuration). Nonetheless, Figure 6.5b clearly depicts the strongly differing asymptotic trendline for the two solvers: while the nested dissection solver runtime increases by a constant amount with each larger fiber instance (consistent with the logarithmic runtime complexity estimate), the MUMPS solver time appears to increase exponentially and is much higher for a fiber of 512 wavelengths computed on 16 nodes. We note that the MUMPS solver fails for very large fiber instances due to memory constraints.



(a) Parallel efficiency (linear solve)

(b) Runtime (linear solve)

Figure 6.5: Stampede2 weak-scaling results for the fiber amplifier model with uniform polynomial order $p = 5$. Comparison of the nested dissection solver with the MUMPS solver.

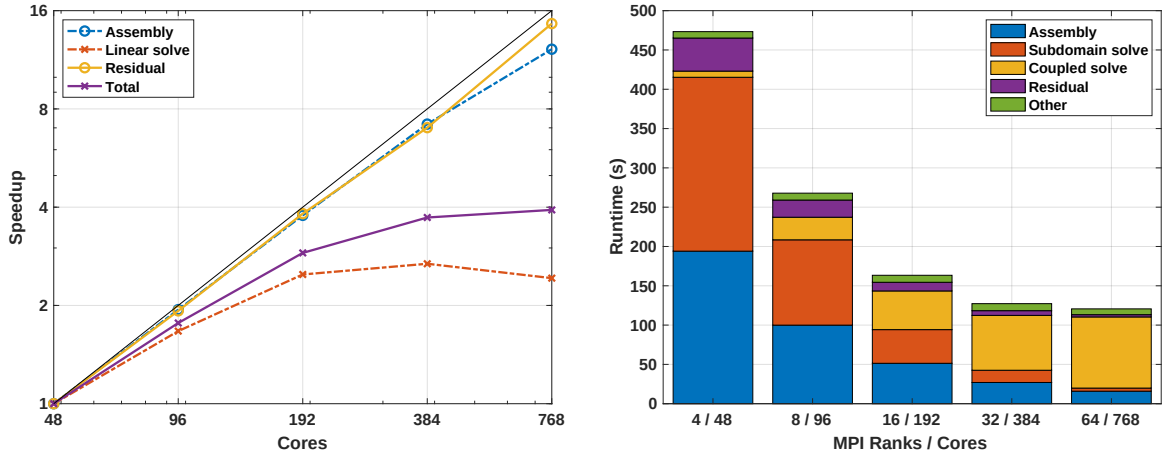
6.1.5 MPI strong scaling

As a metric for the fiber amplifier model performance, strong scaling is somewhat less important than the previously discussed performance metrics since we are primarily aiming to increase the maximum fiber length of the simulation. However, in some cases, good strong scaling can be helpful in accelerating a fiber amplifier simulation for a fixed fiber length (e.g., to run many different configurations). Therefore, we investigate how much the simulation can be accelerated in such a case and how the strong-scaling performance compares to the

ideal (linear) speedup.

The nested dissection algorithm is, by construction, not well-suited for strong scaling. While the subdomain-local computation time decreases linearly with additional processors, we obtain more subdomain interfaces that are coupled and must be solved with a deeper nested dissection tree. The runtime of the coupled portion of the solve therefore increases logarithmically with the number of processors. However, in the fiber simulation, this may still allow for a moderate speedup overall as long as the coupled solve time does not dominate the total runtime.

In our strong-scaling test, the fiber length is fixed to 32 wavelengths with the same uniform sixth-order discretization that was used in the weak-scaling test. As the number of processors grows, the subdomains in the fiber grow smaller each time, i.e., the slabs defined by cross-sectional cuts along the fiber axis become narrower and include fewer elements to compute. The number of compute nodes is increased gradually from one node (four subdomains) to 16 nodes (64 subdomains) at which point each subdomain resolves only half a wavelength along the fiber axis.



(a) Parallel speedup

(b) Runtime

Figure 6.6: Stampede2 strong-scaling results for the fiber amplifier model with uniform polynomial order $p = 6$ and fixed fiber length (32 wavelengths). Parallel speedup of the main subroutines is near-optimal except for the linear solve, leading to a maximum 4x speedup with 16 compute nodes.

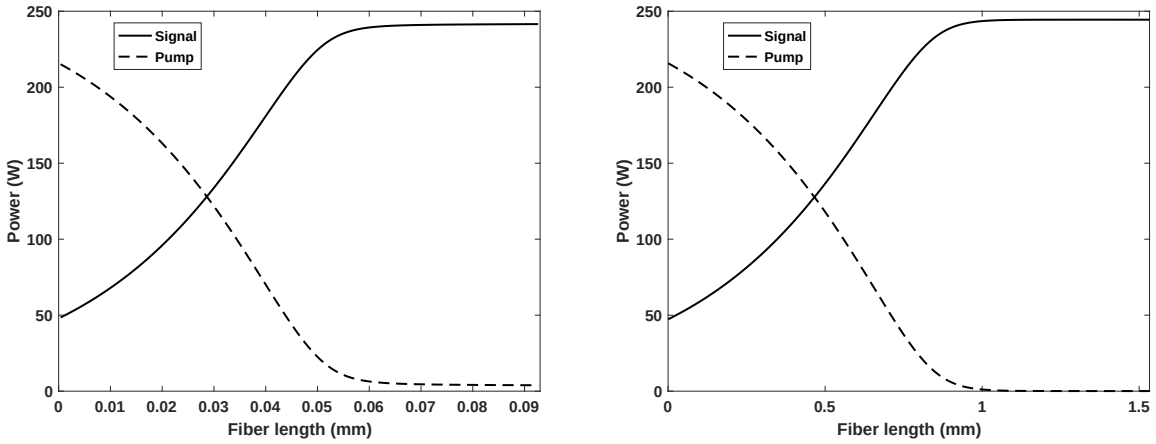
Figure 6.6 shows the strong-scaling results for the fiber amplifier model. In Figure 6.6a, the parallel speedup is plotted. As expected, the speedup for the element matrix assembly and residual computation are almost linear, whereas the solver fails to gain any speedup beyond 32 MPI processes and in fact becomes slower when increasing parallelism further. The solver dilemma is clearly depicted in Figure 6.6b which shows the total runtime of the simulation broken down into the main subroutines. As can be seen, the local "subdomain solve" scales very well whereas the "coupled solve" of the subdomain interfaces increases with additional MPI processes. At 16 compute nodes (64 MPI processes), the coupled solve time dominates the total runtime, thus eliminating any chance of further speedup by increasing the amount of parallelism. Ultimately, however, the observed overall speedup of 1.8 and 2.9 with two and four compute nodes, respectively, may be good enough to justify additional compute resources for accelerating simulation of the fiber model with fixed length.

6.2 Active gain simulations

We now utilize the distributed, parallel code to simulate various large instances of the nonlinear 3D fiber amplifier model. In these simulations, we look at the signal and pump power distribution along the fiber, the amplifier efficiency, and the convergence of the nonlinear solver. The simulations are conducted for an ytterbium-doped, core-pumped, large-mode-area, step-index fiber amplifier such as the one described in Chapter 5 (see model parameters in Tables A.5 and A.6). We employ a high-order discretization to obtain accurate results for nonlinear gain in long fibers. To minimize numerical pollution errors, we use an anisotropic discretization with sixth-order polynomials in the radial direction and seventh-order polynomials in the longitudinal direction (i.e., $(p_x, p_y, p_z) = (6, 6, 7)$), yielding even more degrees of freedom (per wavelength) than were shown in Table 6.3. These are the first results for our nonlinear gain model with more than one thousand wavelengths. We also show simulations of counter-pumped configurations of our model.

6.2.1 Co-pumped configurations

In the co-pumped fiber amplifier model, both the signal and the pump light are injected at the same fiber end. For a core-pumped amplifier, we inject the respective (core-guided) fundamental mode of each light source but provide significantly more power for the pump; the signal laser input is ca. 48 W and the pump input is approximately 216 W, yielding a maximal amplified signal output of about 246 W if the entire pump light is transferred into the signal at optimal efficiency (91.73%). Figure 6.7 displays the optical power in each field for fibers of 128 and 2 048 wavelengths. The artificial gain coefficient $\tilde{g}_a$ is again selected such that sufficient amplification is possible within an unrealistically short fiber. The results indicate that the gain scaling, previously shown for fibers from 15 to 240 wavelengths (see Figure 5.4), extends to several thousand wavelengths—i.e., short fibers appear to (qualitatively) preserve the power distribution of longer, more realistic fibers if their shorter size is accounted for by a larger gain coefficient $\tilde{g}_a$. We can also observe from Figure 6.7 that, based on our previous estimate for the maximal signal output, the optimal efficiency is nearly attained (precise values of the attained efficiencies are shown in the subsequent section—see Table 6.4).



(a) 128 wavelengths, $\tilde{g}_a = 4.0 \cdot 10^3$ (b) 2 048 wavelengths, $\tilde{g}_a = 2.5 \cdot 10^2$

Figure 6.7: Active gain simulation of a co-pumped amplifier configuration for fibers of different lengths. In both cases, 128 and 2 048 wavelengths, the pump power is almost entirely transferred into the signal laser, provided that the artificial gain amplification $\tilde{g}_a$ is large enough.

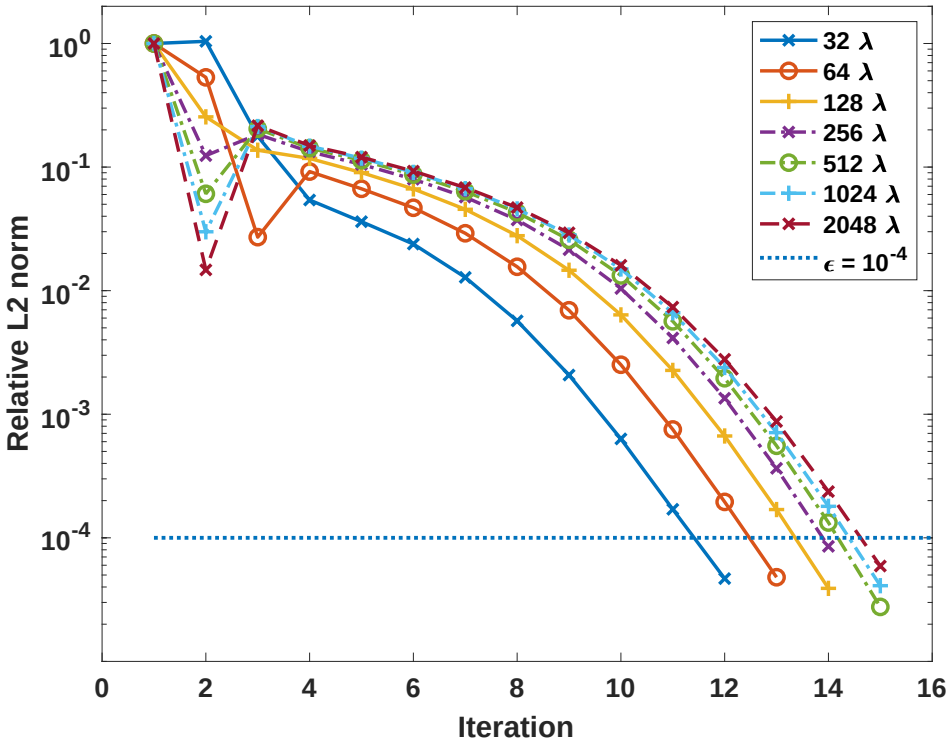


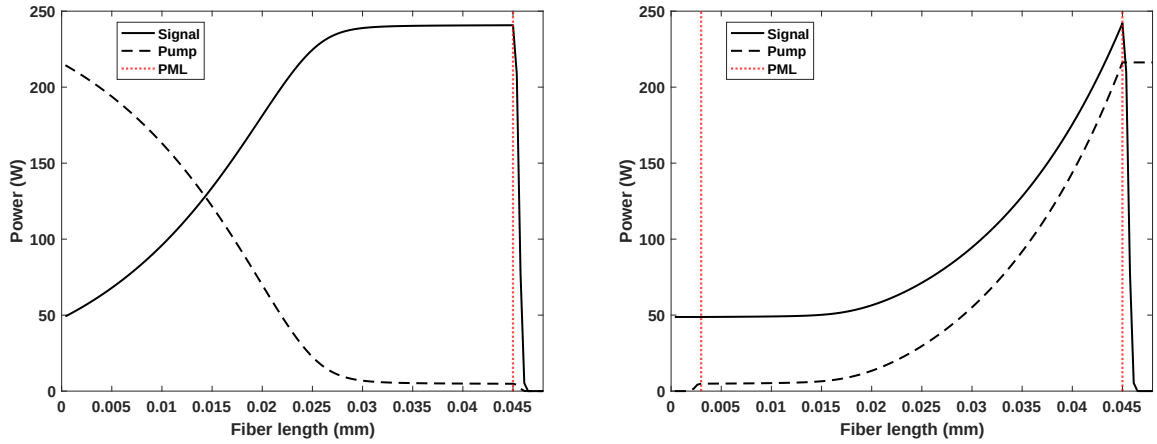
Figure 6.8: Convergence of co-pumped, active gain amplifier simulation for fibers of different lengths. The nonlinear solver converges within 12–15 iterations in all cases, but the number of iterations tends to increase slowly with increasing fiber length.

The convergence of the nonlinear solver for fibers from 32 wavelengths to 2 048 wavelengths is shown in Figure 6.8. In all cases, convergence is attained within 12–15 iterations; however, the number tends to increase slowly with increasing fiber length. Except for some oscillations initially (≤ 4 iterations), the relative change in the L^2 norm of the signal field decreases monotonically for all fibers.

6.2.2 Counter-pumped configurations

So far, we only considered *co-pumped* fiber configurations for which the signal and pump lights are injected at the same fiber end. Another possible configuration is the *counter-pumped* amplifier, where signal and pump lights are injected at opposing fiber ends. These counter-pumped configurations are of practical interest because they tend to have a higher

threshold for nonlinear mode instability than the co-pumped configurations [76]. As previously mentioned, a counter-pumped configuration can be difficult to realize in simpler fiber amplifiers models (e.g., scalar BPM models) due to their inherent assumptions. In our model, counter-pumping is natively supported and requires only minor adjustments of the boundary conditions. The computational complexity is the same as for a co-pumped fiber amplifier. In this section, we show the first numerical results for a counter-pumped configuration of our active gain fiber amplifier model. In principle, the model can also handle a bi-directional pump configuration, where pump light is injected at both fiber ends. This configuration will, however, incur additional computational cost if the second pump source is modeled by a separate field requiring one additional linear solve per iteration.



(a) Co-pumped configuration

(b) Counter-pumped configuration

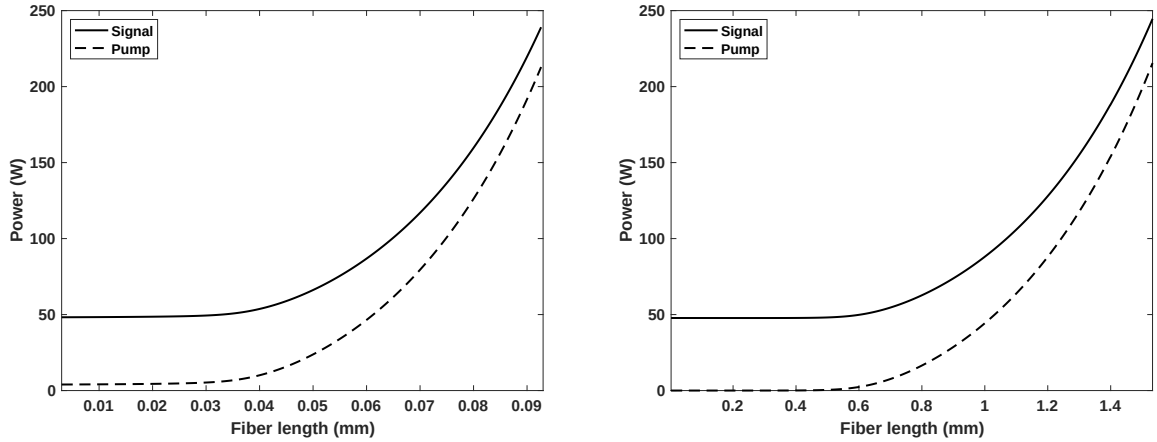
Figure 6.9: Active gain simulation for a fiber of 64 wavelengths, $\tilde{g}_a = 8.0 \cdot 10^3$. In the counter-pumped configuration, the pump light is injected at the fiber exit and the absorbing boundary layer acts at the fiber input where the signal light is injected.

PML in counter-pumped amplifier. In the counter-pumped configuration, the pump light is injected at the fiber exit; we account for this by changing the boundary conditions for the pump field appropriately. In the pump solve, the inhomogeneous tangential electric field boundary condition is now prescribed at the fiber exit, while the PML is implemented at the fiber input where the signal light is injected. In a way, the amplified laser light "sees" an infinitely extending fiber in $+z$-direction whereas the pump light "sees" an infinitely extending

fiber in the $-z$-direction. Nagaraj et al. [98] used this idea for a counter-pumped Raman gain amplifier model and demonstrated its feasibility in a short fiber simulation with low amount of energy transfer. Figure 6.9 shows that it extends to the case of counter-pumping in the active gain fiber amplifier model with full energy transfer from pump to signal. In Figures 6.9a and 6.9b, the dotted lines each mark the beginning of a PML; in Figure 6.9b—the counter-pumped case—the "right" layer only acts on the signal field, and the "left" layer only acts on the pump field.

Gain scaling and amplifier efficiency. In the counter-pumped case, a similar gain scaling coefficient $\tilde{g}_a$ needs to be applied as for the co-pumped configuration in order to fully transfer the pump power to the signal field within a fiber of the same length. For example, this is shown for two fibers with 128 and 2 048 wavelengths in Figure 6.10, where in each case the same $\tilde{g}_a$ is used as in the previous co-pumped configuration (cf. Figure 6.7). Furthermore, the power distribution along the fiber is nearly the same for these two fibers of different lengths, indicating that the gain may be scaled—at least to some extent—via an artificial coefficient to account for shorter fiber lengths also in the counter-pumped configuration.

To measure the total optical-to-optical efficiency in the counter-pumped amplifier model, we compute the gain in signal power (output minus input) divided by the loss of pump power (input minus output). We can do the same for the co-pumped configuration. In both cases, the efficiencies obtained for fibers of 32 to 2 048 wavelengths are very close to the ideal efficiency (see Table 6.4). The amplified laser output, also shown in Table 6.4, is similar for all configurations, with nearly the entire pump power transferred to the signal. This is only possible because we are considering a core-pumped model in which essentially all pump light is injected in the fiber core where it can be converted via the stimulated emission process (see Section 5.1). For a cladding-pumped configuration, any pump light that does not enter the fiber core within a certain fiber length cannot amplify the signal through this process.



(a) 128 wavelengths, $\tilde{g}_a = 4.0 \cdot 10^3$ (b) 2 048 wavelengths, $\tilde{g}_a = 2.5 \cdot 10^2$

Figure 6.10: Active gain simulation of a counter-pumped amplifier configuration for fibers of different lengths. The counter-pumped fiber amplifier model achieves full power transfer from the pump into the signal field on both short and long fibers, where the amount of gain can be adjusted to account for a shorter fiber length.

Model parameters		Co-pumped amplifier		Counter-pumped amplifier	
Number of wavelengths	Gain scaling coefficient $\tilde{g}_a$	Efficiency (%)	Output (W)	Efficiency (%)	Output (W)
32	$1.6 \cdot 10^4$	91.47	240.3	91.53	239.1
64	$8.0 \cdot 10^3$	91.41	240.7	91.43	242.1
128	$4.0 \cdot 10^3$	91.39	241.4	91.36	242.0
256	$2.0 \cdot 10^3$	91.40	242.2	91.44	242.7
512	$1.0 \cdot 10^3$	91.43	243.3	91.34	243.7
1 024	$5.0 \cdot 10^2$	91.43	244.3	91.29	244.6
2 048	$2.5 \cdot 10^2$	91.39	244.2	91.28	244.8

Table 6.4: Fiber amplifier efficiency in active gain simulations for co-pumped and counter-pumped fibers of different lengths. The observed optical-to-optical efficiency is close to the optimal efficiency of 91.73% in all cases. The output powers are 239–245 W for an input of approximately 48 W (signal) and 216 W (pump).

The convergence of the nonlinear scheme is slower for the counter-pumped configuration (see Figure 6.11) than for the co-pumped case (cf. Figure 6.8). We observe that 21–24 iterations are required to reach the same stopping criterion ($\varepsilon = 10^{-4}$). However, there is no clear tendency toward an increasing number of iterations with increasing fiber length.

The generally larger number of iterations needed for convergence appears to come from a more oscillatory nature of the temporary solutions obtained during the Picard iterations; this results in a slower, non-monotonic decay of the corresponding L^2 norm measuring the relative difference between the iterates.

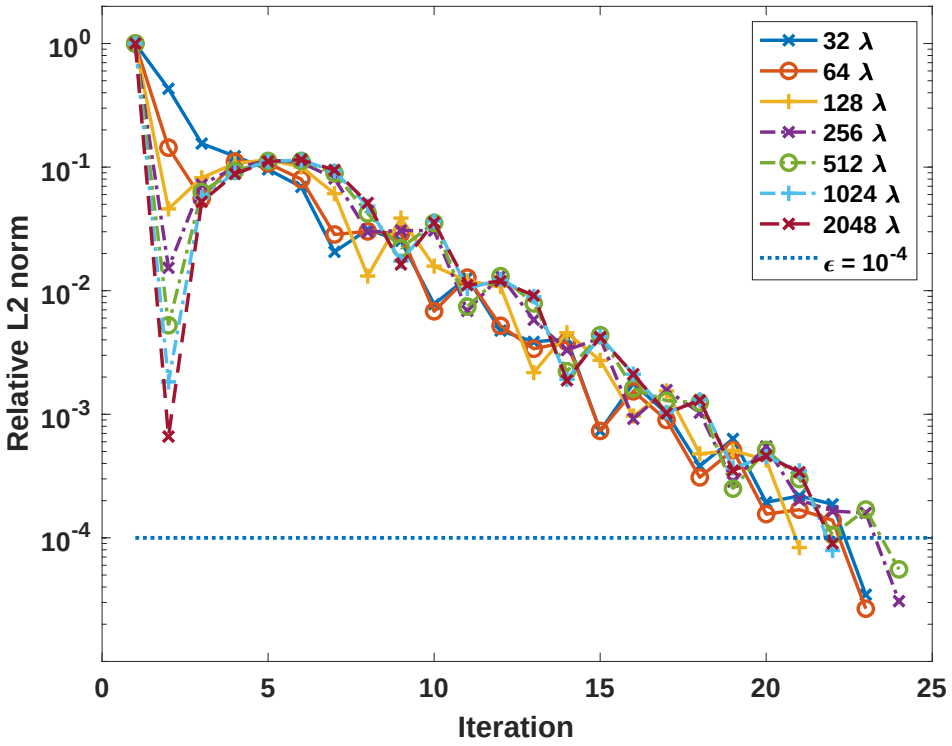


Figure 6.11: Convergence of counter-pumped, active gain amplifier simulation for fibers of different lengths. The nonlinear solver converges within 21–24 iterations in all cases, more than for the co-pumped case, but there is no tendency of increasing number of iterations with increasing fiber length.

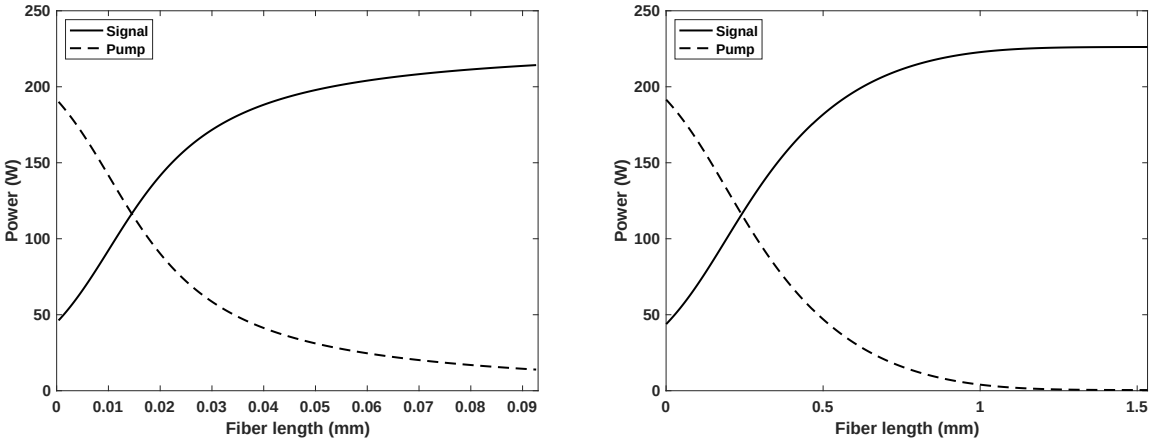
6.3 Raman gain simulations

In this section, we conduct simulations of a three-dimensional Raman gain fiber amplifier model that was first introduced by Nagaraj et al. (2018). The model is also based on the solution of two weakly-coupled time-harmonic Maxwell systems but with a coupling term modeling passive Raman gain rather than gain via an active dopant. We have made minor changes to the original model, including the use of a different non-dimensionalization and

gain amplification term in our derivation (see details in Appendix B.4), as well as significant computational advancements enabling simulation of much longer fiber models.

Contrary to the stimulated emission process in the active gain fiber amplifier, the passive Raman gain amplifier relies on a different mechanism to amplify a laser. We cite the paper from Nagaraj et al. [98]:

> *Raman scattering is an inelastic optical nonlinearity that occurs as incident light (the pump), at a sufficiently high intensity, vibrates the molecules of the medium, resulting in optical phonons and scattered photons (the Stokes field), usually of a lower frequency than the incident photons. This process can start from noise, but in this model the Raman scattering is stimulated by having a seeded signal field offset in frequency from the pump field so as to achieve peak Raman gain and coinciding perfectly with the Stokes field frequency.*



(a) 125 wavelengths, $\tilde{g}_R = 1.6 \cdot 10^6$ (b) 2 000 wavelengths, $\tilde{g}_R = 1.0 \cdot 10^5$

Figure 6.12: Raman gain simulation of a co-pumped amplifier for fibers of different lengths. The Raman gain model delivers nearly full power transfer from the pump into the signal (Stokes field) for both short and long fibers if the artificial gain amplification $\tilde{g}_R$ is large enough.

In other words, we consider two fields with a different frequency: the incident light (pump field) with 1 064 nm wavelength and the Stokes field (signal) with 1 116 nm wavelength. We emphasize that while the underlying principle of nonlinear Raman scattering

with this model was demonstrated in [98], the numerical simulations were limited to fewer than one hundred wavelengths and a relatively low amplification of the Stokes field with no validation of the efficiency or other quantities. The finite element simulations also exhibited significant numerical pollution in the form of a power decay of the propagating field. All of these points have been addressed, and our new simulations extend to the regime of several thousand wavelengths for the nonlinear Raman gain amplification model with high accuracy. As before, we employ an anisotropic high-order discretization, $(p_x, p_y, p_z) = (6, 6, 7)$, to minimize the pollution errors.

Figure 6.12 shows the power distribution of a core-pumped, co-pumped Raman gain amplifier simulation for two fibers of 125 and 2 000 wavelengths. By using a gain scaling term $\tilde{g}_R$ for the Raman model, we are able to simulate the amplification of the Stokes field to maximal power within very short fibers. However, we note that the gain scaling does not perfectly account for the length scaling; for example, in the 125 wavelength fiber, some pump power remains at the output whereas in the 2 000 wavelength fiber, all pump power is transferred into the signal. It appears the longer fiber experiences somewhat stronger Raman scattering than the short fiber which is not (fully) accounted for by the linearly scaled gain coefficient $\tilde{g}_R$.

Model parameters		Raman gain amplifier	
Number of wavelengths	Gain scaling coefficient $\tilde{g}_R$	Efficiency (%)	Output (W)
31	$6.4 \cdot 10^6$	95.36	211.7
62	$3.2 \cdot 10^6$	95.35	213.2
125	$1.6 \cdot 10^6$	95.35	214.3
250	$8.0 \cdot 10^5$	95.36	216.1
500	$4.0 \cdot 10^5$	95.36	219.6
1 000	$2.0 \cdot 10^5$	95.38	223.2
2 000	$1.0 \cdot 10^5$	95.42	226.2

Table 6.5: Fiber amplifier efficiency in Raman gain simulations for fibers of different lengths. The observed optical-to-optical efficiency is near the optimal efficiency of 95.34% in all cases. The output powers are 212–226 W for an input of approximately 45 W (signal/Stokes field) and 191 W (pump).

Table 6.5 shows the optical-to-optical efficiency of the Raman amplifier simulation. Similar to the active gain amplifier model, the Raman gain model delivers near-optimal efficiencies for all tested fibers.

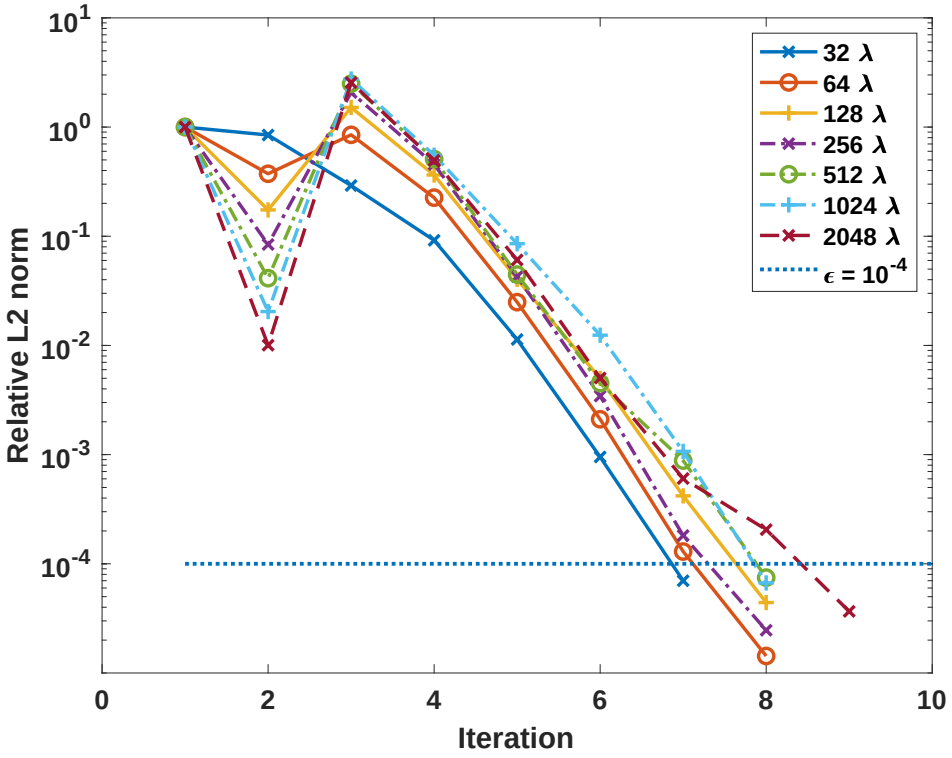


Figure 6.13: Convergence of co-pumped, Raman gain amplifier simulation for fibers of different lengths. The nonlinear solver converges within 7–9 iterations in all cases, less than for the active gain amplifier model.

The Raman gain model converges quicker than the active gain model. As illustrated in Figure 6.13, convergence is attained within 7–9 iterations and the relative change in the iterates decays monotonically after the first few iterations.

Lastly, we remark that the Raman gain fiber amplifier model also supports counter- and bi-directional pumping configurations in the same way as the active gain model. Analogous to the counter-pumped active gain model, the counter-pumped Raman gain simulation converges slower than the co-pumped configuration.

6.4 Refractive index grating

In this section, we use our model to study aspects of the thermal effects occurring in high-power fiber amplifiers. Increasing the power of the fiber amplifier usually causes a higher thermal load. The resulting thermal effects are a significant factor to consider in designing and operating fiber amplifiers. Mainly, there are two effects in active gain fibers due to the thermal load [76]: 1) mode shrinking, and 2) the transverse mode instability (TMI). Mode shrinking refers to the shrinking of the *mode field diameter*—the characteristic width of the Gaussian profile of the guided fundamental mode (FM) (see Figure 4.1)—due to a thermally induced change of the refractive index in the fiber core (see Figure 5.7). This effect has several consequences: it increases the intensity of the light in the fiber core and thus contributes to the onset of nonlinear effects, and it also increases the effective core numerical aperture, thereby allowing for additional higher-order modes (HOMs) to propagate. Because of this thermally induced mode shrinking, even fibers that were designed to operate single-mode may effectively become multimode waveguides which can lead to degradation of the output beam quality if significant energy is transferred into the HOMs [68].

The nonlinear TMI is characterized by a chaotic fluctuation of the output beam at high power [44]. This fluctuation comes from an energy transfer between guided transverse modes of the fiber, i.e., a coupling of the FM and the HOMs. While TMI mitigation strategies are still actively researched, there is already a wide scientific consensus explaining the origin of this thermally induced nonlinear effect [76]. In short, there are two requirements for the TMI to occur. First, two simultaneously propagating guided transverse modes in the fiber amplifier create a periodic *modal interference pattern* (MIP) with a period—the *mode beat length*—that depends on the modes' propagation constants (see Figure 6.14). Due to this MIP, the intensity of the light also fluctuates periodically along the fiber longitudinal direction (see Figure 6.15). This in turn leads to a periodic fluctuation of the excited ion population density in the active fiber core, hence to a fluctuation in the conversion of pump

to signal photons that causes the heat deposition via the quantum defect. In other words, the MIP is imprinted on the heat load in the fiber. Through the fluctuating heat load, the thermo-optical effect then causes a periodic *refractive index grating* (RIG) along the fiber longitudinal direction [77]. This thermally induced RIG may lead to a mode coupling that allows guided transverse modes to exchange energy. The second condition that needs to be fulfilled for such a mode coupling is a phase shift between the MIP and the RIG [114]. The exact origins of this phase shift are still debated but may be due to noise in the fiber laser system at high power [76].

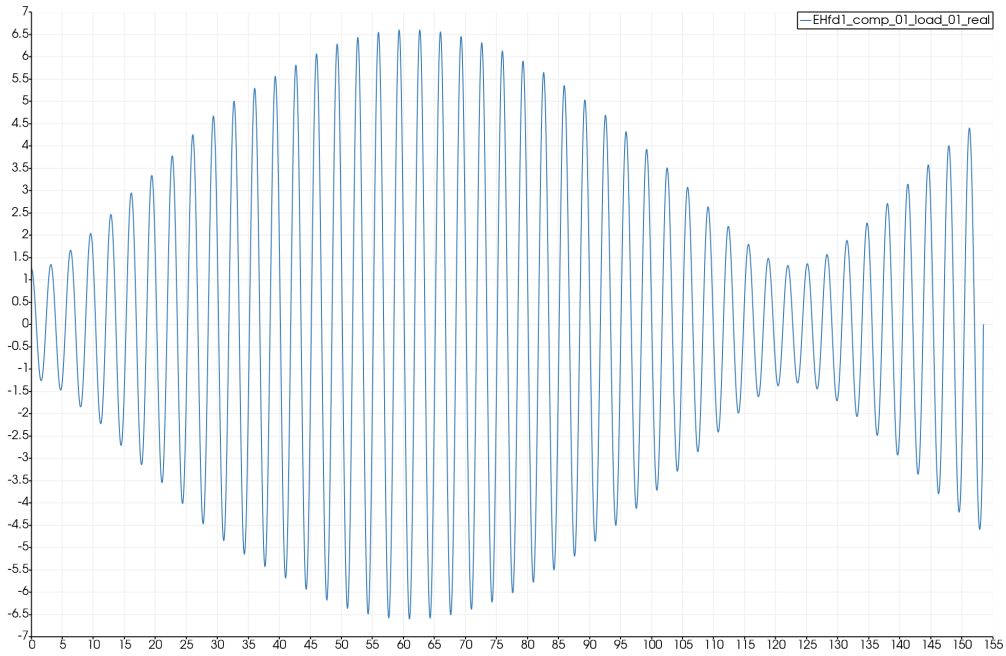
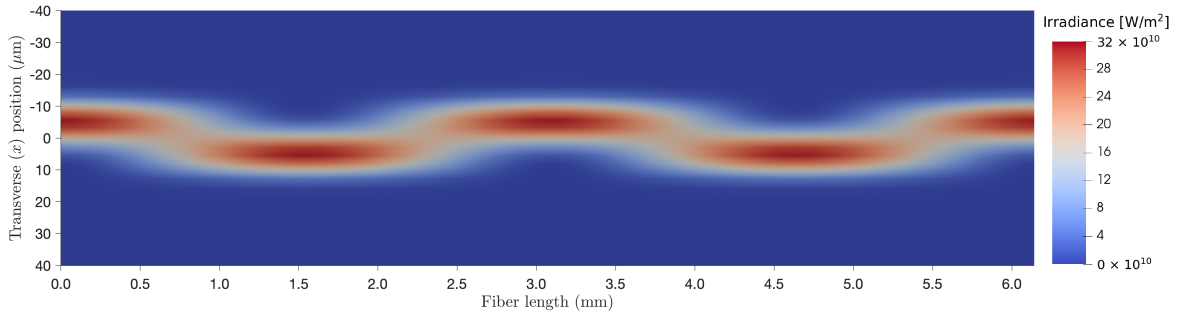


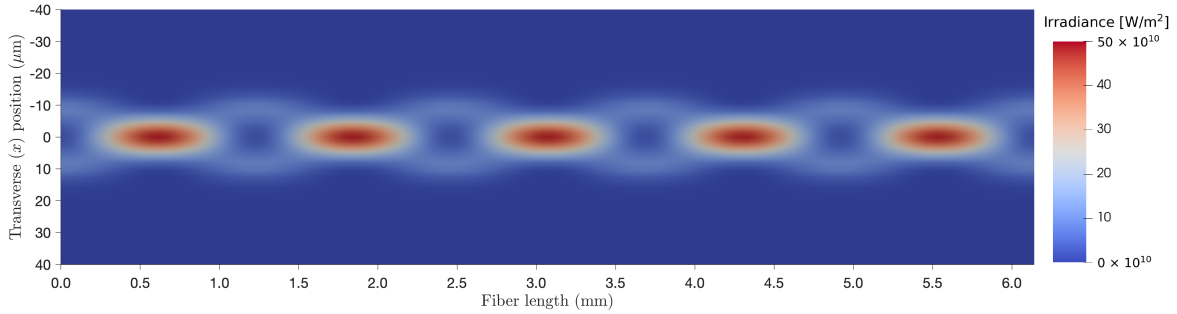
Figure 6.14: Mode beating in $\Re\{E_x\}$ component between the LP_{01} and LP_{02} modes plotted along the longitudinal axis of the fiber. The mode beat length is ca. 1.23 mm (approximately 1 677 wavelengths). Note that the waves in the plot originate from a small sample of points representing full 2 090 wavelengths over that distance.

As mentioned above, the thermally-induced RIG has a period equal to the length of the mode beat between the FM and the HOMs. For weakly-guiding large-mode-area fiber amplifiers, the mode beat length between the FM and a guided HOM is typically several thousand wavelengths. Capturing a RIG over multiple mode beat lengths with our nonlinear full vectorial Maxwell model is therefore challenging due to the computational cost

of simulating a fiber of that length. To test and validate mode coupling in the Maxwell model without computing the full nonlinear coupled model (i.e., without laser gain and heating), we prescribe an artificial grating of the refractive index that has some of the characteristics of the RIG induced by heating.



(a) Asymmetric $\mathrm{LP}_{01}/\mathrm{LP}_{11}$ mode beating with a period of ca. 4 221 wavelengths (3.10 mm)



(b) Symmetric $\mathrm{LP}_{01}/\mathrm{LP}_{02}$ mode beating with a period of ca. 1 677 wavelengths (1.23 mm)

Figure 6.15: Mode beating of the FM and the HOMs in a 6.1 mm long fiber (approximately 8 320 wavelengths). The mode beating is illustrated by the periodic irradiance beating plotted in a slice orthogonal to the transverse (y) axis in the fiber center. In each case, the propagating light has an optical power of 50 W and is composed of 80% FM and 20% HOM.

Experimental setup. We consider a large-mode-area step-index fiber (see parameters in Table A.5) that supports four guided modes: $\{\mathrm{LP}_{01}, \mathrm{LP}_{11}, \mathrm{LP}_{21}, \mathrm{LP}_{02}\}$.[2] Let Δk_{lp} denote the difference between the propagation constants of the FM (LP_{01}) and a guided LP_{lp} HOM.

[2]The two possible rotations of the radially asymmetric modes LP_{11} and LP_{21} are denoted by $\mathrm{LP}_{11a}, \mathrm{LP}_{11b}$ and $\mathrm{LP}_{21a}, \mathrm{LP}_{21b}$, respectively. See Appendix B.2 for a definition of each mode.

For this particular fiber, the (normalized) propagation constants of the modes are:

$$\{k_{01}, k_{11}, k_{21}, k_{02}\} \approx \{85.6833, 85.6630, 85.6380, 85.6322\}. \tag{6.1}$$

Then,

$$\{\Delta k_{11}, \Delta k_{21}, \Delta k_{02}\} \approx \{0.0203, 0.0453, 0.0511\}. \tag{6.2}$$

The corresponding (normalized) mode beat lengths are (cf. Figure 6.15),

$$2\pi/\{\Delta k_{11}, \Delta k_{21}, \Delta k_{02}\} \approx \{309.517, 138.702, 122.959\}, \tag{6.3}$$

with a length scale of 10^{-5} m (see Table A.3). The number of wavelengths of the FM per beat length is:

$$k_{01}/\{\Delta k_{11}, \Delta k_{21}, \Delta k_{02}\} \approx \{4220.85, 1891.46, 1676.78\}. \tag{6.4}$$

We introduce an artificial long-period grating of the refractive index in the following way:

$$\tilde{n}(x, y, z) = n(x, y) + \phi(x, y)\delta n \sin(\Delta k_{lp} z + \varphi_0), \tag{6.5}$$

where $n(x, y)$ is the unperturbed refractive index, δn is the perturbation amplitude, and $\phi : \Omega_t \ni (x, y) \to [0, 1]$ determines the perturbation profile in the fiber transverse domain $\Omega_t := \{(x, y) : x^2 + y^2 < r_{\text{clad}}^2\}$; the period of the grating is the beat length $2\pi/\Delta k_{lp}$ between the FM and a guided LP_{lp} HOM, $(l, p) \in \{(1, 1), (2, 1), (0, 2)\}$, with initial phase φ_0. The perturbation profile $\phi(x, y)$ determines the region in which the perturbation is active in the fiber cross-section. We consider an indicator-function $\phi(x, y)$ that defines a perturbation profile where the material refractive index is perturbed inside the perturbation region, and

outside of which it is not perturbed. Let

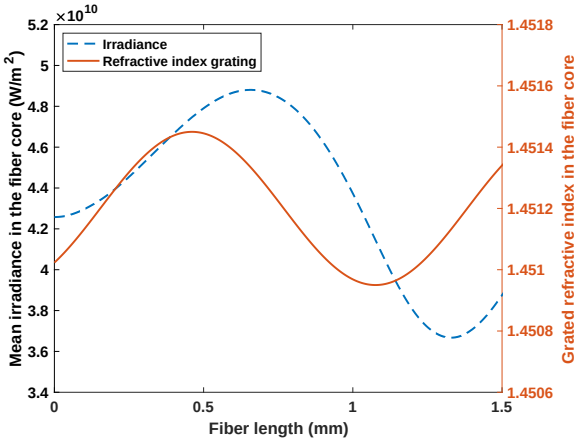
$$B((\tilde{x},\tilde{y}),\tilde{r}) := \left\{(x,y) : \sqrt{(x-\tilde{x})^2+(y-\tilde{y})^2} < \tilde{r}\right\} \tag{6.6}$$

for any cross-section (transverse domain) of the fiber; then,

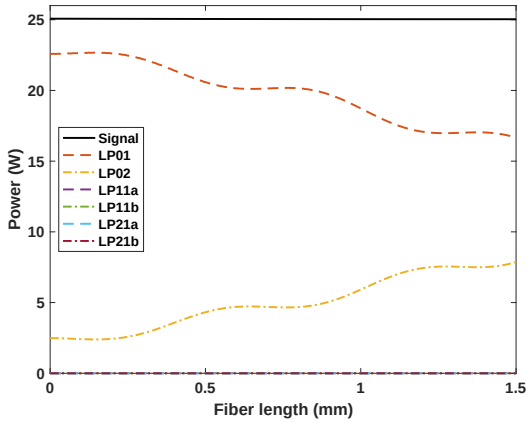
$$\phi(x,y) := \begin{cases} 1, & (x,y) \in B((\tilde{x},\tilde{y}),\tilde{r}), \\ 0, & \text{otherwise}, \end{cases} \tag{6.7}$$

defines a circular perturbation region of radius $\tilde{r}$, centered at $(\tilde{x},\tilde{y})$, within the cross-section of the fiber.

Phase shift. The phase shift between the MIP and the RIG determines whether an energy transfer between the modes can occur and in which direction energy can be transferred (i.e., either from FM to HOM or from HOM to FM) [114, 119, 76]. More precisely, if a negative phase shift occurs (MIP right-shifted of the RIG, i.e., shifted toward fiber output), then the FM can transfer energy to the HOM, whereas if a positive phase shift occurs (MIP left-shifted of the RIG, i.e., shifted toward fiber input), then the HOM can transfer energy to the FM [114]. To test the mode coupling dependency on the MIP–RIG phase shift in the Maxwell model, we consider a radially symmetric perturbation to induce coupling between the LP_{01} and LP_{02} modes. The perturbation profile of the long-period grating is given by (6.7) with $(\tilde{x},\tilde{y}) = (0,0)$ and $\tilde{r} = 0.5 r_{\mathrm{core}}$. The perturbation amplitude is $\delta n = 2.5 \cdot 10^{-4}$, and the grating period is the mode beat length $2\pi/\Delta k_{02}$. The total input power is 25 W. In a realistic fiber amplifier, a small percentage of the input power is usually in HOMs. In order to better illustrate the mode coupling, we simulate a seed ratio with a somewhat larger relative higher-order modal content of 90% LP_{01} and 10% LP_{02}.

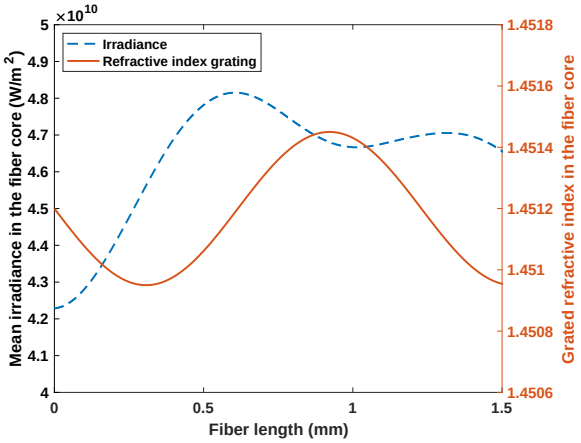


(a) MIP–RIG negative phase shift

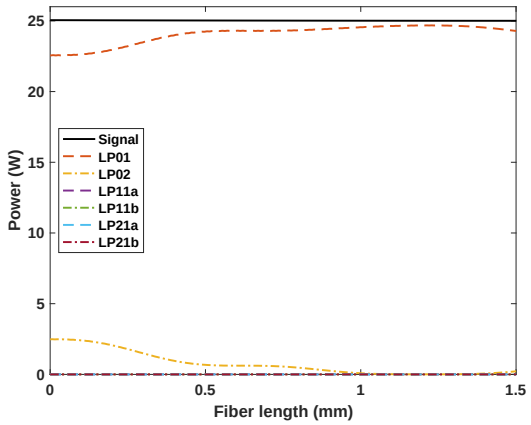


(b) FM to HOM energy transfer

Figure 6.16: Coupling of the LP_{01} and LP_{02} modes via a radially symmetric refractive index grating for a 1.5 mm long fiber (approximately 2 048 wavelengths). (a) A negative phase shift between the MIP and the RIG causes (b) an energy transfer from FM to HOM.



(a) MIP–RIG positive phase shift



(b) HOM to FM energy transfer

Figure 6.17: Coupling of the LP_{01} and LP_{02} modes via a radially symmetric refractive index grating for a 1.5 mm long fiber (approximately 2 048 wavelengths). (a) A positive phase shift between the MIP and the RIG causes (b) an energy transfer from HOM to FM.

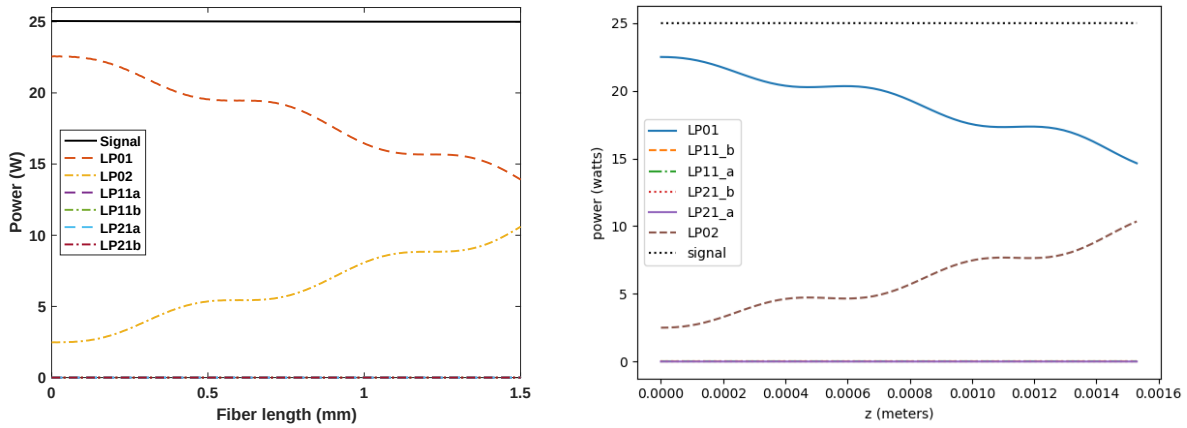
First, the mode coupling is tested for a negative phase shift in a 1.5 mm long fiber (ca. 2048 wavelengths). Figure 6.16a illustrates this phase shift by plotting the prescribed refractive index in the perturbed region and the mean irradiance in the fiber core along the longitudinal fiber axis. The fluctuation of the mean irradiance corresponds to the periodic MIP of the LP_{01} and LP_{02} modes (cf. Figure 6.15b). In Figure 6.16b, the approximate modal

content in the fiber is displayed.[3] As expected, an energy transfer occurs from the FM to the HOM. Conversely, a positive phase shift (Figure 6.17a) induces an energy transfer from the HOM to the FM (Figure 6.17b).

Cross-validation with CMT model. As previously discussed, there exist fiber amplifiers models that are much more computationally efficient than our 3D vectorial Maxwell simulation. One particularly efficient and widely used category of models is based on coupled mode theory (CMT). In the CMT approach, the transverse electric field that is assumed to be strictly polarized in one direction is decomposed into a discrete set of propagating guided modes. These modes are then explicitly coupled to one another via coupling coefficients. We cross-validate our 3D Maxwell simulation with a CMT model developed by T. Goswami et al. (Portland State University) in collaboration with J. Grosek (Air Force Research Laboratory). For this validation, large-mode-area fibers with similar properties are used, and the same RIG is prescribed in both fiber models. The fibers are then seeded with a similar input (total power and modal content). Whereas the 3D Maxwell model solves a computationally expensive full vectorial boundary value problem, the CMT model very efficiently computes the modal content along a fiber of the same length. In the 3D Maxwell simulation, the modal content is computed by projecting the transverse electric field onto guided transverse modes for different fiber cross-sections (see Section 4.1.3). The optical power carried in each mode along the fiber can then be compared for these independent and entirely different models.

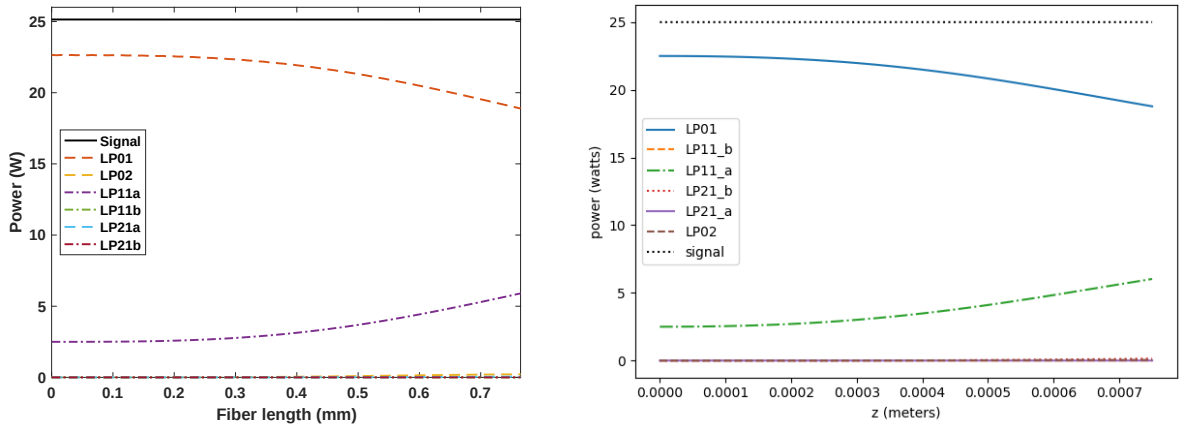
First, we use a radially symmetric long-period grating that couples the LP_{01} and LP_{02} modes similar to the previous example with a negative phase shift (MIP right-shifted of RIG by $\pi/2$). This coupling causes significant energy transfer from the FM to the HOM. Figure 6.18 depicts the modal content of the propagating field in a 1.5 mm long fiber (ca. 2 048 wavelengths). Along the entire length of the fiber, the simulations yield qualitatively comparable results for the Maxwell model (Figure 6.18a) and the CMT model (Figure 6.18b).

[3] Recall that in the 3D Maxwell model, the modal content is computed via L^2 projection of the transverse electric field onto transverse modes for different fiber cross-sections (cf. Section 4.1.3).



(a) Maxwell model (b) CMT model (by T. Goswami)

Figure 6.18: Coupling of the LP_{01} and LP_{02} modes via a radially symmetric refractive index grating for a 1.5 mm long fiber (approximately 2 048 wavelengths). The Maxwell model and the CMT model yield similar modal content along the entire length of the fiber.



(a) Maxwell model (b) CMT model (by T. Goswami)

Figure 6.19: Coupling of the LP_{01} and LP_{11} modes via a radially asymmetric refractive index grating for a 0.75 mm long fiber (approximately 1 024 wavelengths). The Maxwell model and the CMT model yield similar modal content along the entire length of the fiber.

Next, we prescribe an asymmetric perturbation to induce coupling between the LP_{01} and LP_{11} modes. The perturbation region is given by (6.7) with $(\tilde{x}, \tilde{y}) = (-0.4r_{\mathrm{core}}, 0)$ and $\tilde{r} = 0.6r_{\mathrm{core}}$, and the perturbation amplitude is $\delta n = 2.5 \cdot 10^{-4}$. The fiber is seeded with 90% FM and 10% LP_{11} mode. Figure 6.19 shows the modal content along the fiber length

(ca. 0.75 mm) for the Maxwell model (Figure 6.19a) and the CMT model (Figure 6.19b). In both simulations, significant energy is transferred from the FM to the HOM. Qualitatively, both models yield the same results for the modal content. This result is remarkable considering the fact that each model relies on a completely different set of assumptions (or more precisely, the CMT model makes significantly more assumptions than the Maxwell model). Therefore, this cross-validation gives confidence to the quality of the approximations made by the CMT model, which accurately reproduces the results of the higher-fidelity Maxwell model for the modal energy transfer due to the grated refractive index in the fiber.

Chapter 7

Conclusions and Future Directions

7.1 Conclusions

Contributions. The first accomplishment of this dissertation is the development of a parallel MPI/OpenMP implementation of the three-dimensional FE software *hp*3D. This effort focused on efficiently distributing the workload in the stiffness and load assembly process based on a (mesh) distribution for DOFs, including dynamic load balancing to accommodate adaptive refinements. The solve of the linear system was parallelized through interfacing with third-party libraries (MUMPS, PETSc), and we also developed a distributed nested dissection solver that is particularly suitable to solving waveguide problems. Of particular importance, the parallel software supports all of the advanced FE technologies implemented in *hp*3D, including: 1) isotropic and anisotropic *hp*-adaptive mesh refinements; 2) hybrid meshes with tetrahedral, hexahedral, prismatic, and pyramidal elements; 3) conforming discretizations of the H^1–$H(\text{curl})$–$H(\text{div})$–L^2 energy spaces; and 4) coupled multiphysics variables. The support of these FE technologies makes the scalable parallel *hp*3D software a unique analysis tool applicable to complex multiphysics problems. For this reason, the developed software capabilities advance the state of the art in *hp*-adaptive FE computation.

Using robustly stable DPG FE discretizations for the time-harmonic Maxwell equations, we studied the pollution effect in optical waveguides. Our study corroborates theo-

retical results [89, 39, 90] and provides guidance on how to mitigate pollution in waveguide problems. We also showed that the built-in error indicator of the DPG method can be effectively used to guide *hp*-adaptivity for capturing high-order modes in waveguide simulations.

Additionally, this dissertation presented a high-fidelity 3D fiber amplifier model based on a conforming DPG FE discretization for the time-harmonic Maxwell equations. We were able to make significant modeling advancements, demonstrating how to incorporate nonlinear laser amplification and thermal effects into the full vectorial Maxwell model for an ytterbium-doped fiber amplifier, while maintaining the computational feasibility. The model natively supports co-, counter-, and bi-directional pumping configurations, as well as anisotropic materials. Our parallelization effort has enabled large-scale numerical simulations of the fiber amplifier model with several thousand wavelengths. The parallel *hp*3D code was shown to scale efficiently in these simulations for up to 512 compute nodes (24 576 cores) and more than one billion DOFs.

Limitations. The computational advancements of this effort enabled us to scale the size of the full 3D fiber model simulation from computing less than 100 wavelengths to solving circa 10 000 wavelengths. Even though this represents a significant advancement, it is far short of simulating a full-length fiber of 5–20 meters that would require resolving several million wavelengths. A simulation of this size remains currently infeasible with our model mainly due to the numerical pollution effect that drives the need for higher-order discretization as the fiber length is increased. To simulate laser gain in a short fiber, we introduced an artificial longitudinal scaling for our model. While this approach may have limitations, we showed that it qualitatively preserves certain quantities of interest (power distribution, amplifier efficiency, and thermal load); a similar approach was used in [43].

Another limitation of the 3D model is the accurate propagation of cladding fields for many wavelengths. In the same way that the fiber core guides light, the silica-glass cladding is itself also a waveguide; this is exploited in cladding-pumped amplifiers. Since the fiber

cladding is much larger than the fiber core, it is difficult to find an efficient discretization that captures propagating modes in both the fiber core and cladding geometry. Increasing the mesh resolution in the fiber cladding with additional high-order elements mitigates this issue but increases the computational complexity of the simulation. We note that other fiber amplifier models typically do not model cladding modes explicitly. Instead, many models simply approximate the pump field as a guided plane wave [114, 96, 43].

Implications for fiber amplifier modeling. The state of the art in numerical simulation of fiber amplifiers mostly consists of BPM and CMT approaches. These models are much more computationally efficient than the 3D vectorial Maxwell model developed in this dissertation. However, to achieve computational efficiency, BPM and CMT models make a variety of assumptions that may limit their ability to accurately capture some of the nonlinear optical phenomena in fiber amplifiers. While our 3D Maxwell model is computationally expensive, it provides unique opportunities to simulate those nonlinearities because the model makes as few assumptions as possible. As a high-fidelity model, it also effectively complements other, typically lower-fidelity, fiber amplifier models. It does so, for instance, by providing a validation tool for lower-fidelity models. Therefore, we believe that our model and implementation will contribute to finding mitigation strategies for undesired nonlinear effects in optical fiber amplifiers.

7.2 Future work

Fiber amplifier simulation. In future work, our 3D Maxwell model could be extended to the full problem domain (e.g., the polymer coating), more complex fiber geometries (e.g., microstructure fibers), and/or to include other realistic effects (e.g., SBS). One additional feature of interest is the effect of coiling the fiber around a (cooling) spool. This bending of the fiber is in principle already supported by the model but has not been studied extensively. As the complexity of the model increases, a more sophisticated nonlinear solver may be

required. Future numerical tests of this model will also attempt to show the onset TMI power threshold in the scaled short fiber, as well as the correct relative changes to this threshold for known TMI mitigation techniques (e.g., reducing the core numerical aperture). To investigate mode instabilities, it may be necessary that the artificially scaled fiber is long enough to capture the interference patterns between the guided modes, meaning at least multiple mode beat lengths.

Adaptive multilevel solver. Because of its elongated geometry, the fiber amplifier model can be efficiently computed with our parallel nested dissection solver. In general though, fast and robust iterative solution schemes for high-frequency wave problems are still needed despite much progress in the last decade [48, 56]. A recently proposed DPG-based adaptive multilevel solver [103, 104] was shown to efficiently and reliably solve difficult problems with localized wave solutions. This multilevel solver was developed in *hp*3D and builds on the *hp*-adaptive FE technology of the code. However, the current implementation does not support distributed-memory computation. An MPI implementation of this solver is under development in *hp*3D and will enable solution of currently intractable problems in 3D wave simulation.

Open-source code and *hp* FE book volume III. Both the *hp*3D library and individual application codes are currently maintained in a private GitHub repository. We plan to publish the *hp*3D software under a permissive, open-source BSD-3 license. To make the software more accessible to users, we aim to extend its documentation, including code documentation, descriptions of model problems, and an extensive user manual. Building on the existing books [26, 37], we are working towards publishing the third volume of the *hp* FE book series, describing new features [57, 83, 50, 102] and parallelization of the *hp*3D code, in particular, as well as giving in-depth descriptions of multiphysics applications that have been implemented in *hp*3D.

Appendices

Appendix A

Tabulated Values

A.1 SI units

Symbol	Description
m	Meter
kg	Kilogram
s	Second
rad	Radian
A	Ampere
C	Coulomb
F	Farad
H	Henry
J	Joule
K	Kelvin
N	Newton
T	Tesla
V	Volt
W	Watt
Wb	Weber

Table A.1: SI units

A.2 Physical constants

Symbol	Description	Value	Unit
ε_0	Vacuum permittivity	$8.854 \cdot 10^{-12}$	F/m
μ_0	Vacuum permeability	$4\pi \cdot 10^{-7}$	H/m
c	Speed of light in vacuum	$2.998 \cdot 10^{8}$	m/s
$\hbar$	Reduced Planck's constant	$1.055 \cdot 10^{-34}$	J·s

Table A.2: Physical constants

A.3 Fiber amplifier model parameters

Symbol	Description	Value	Unit
l_0	Length	10^{-5}	m
I_0	Irradiance	10^{10}	W/m^2
σ_0	Absorption/Emission cross-section	10^{-26}	m^2/ion
ν_0	Ion population concentration	10^{25}	ion/m^3
T_0	Temperature	1	K
t_0	Time	10^{-3}	s

Table A.3: Selected dimensional scales

Symbol	Description	Derivation	Value	Unit
ω_0	Angular frequency	c/l_0	$2.998 \cdot 10^{13}$	rad/s
E_0	Electric field	$\sqrt{I_0/(c\varepsilon_0)}$	$1.941 \cdot 10^{6}$	V/m
H_0	Magnetic field	$\sqrt{c\varepsilon_0 I_0}$	$5.152 \cdot 10^{3}$	A/m
P_0	Power	$I_0 l_0^2$	1	W
g_0	Active gain	$\sigma_0 \nu_0$	10^{-1}	1/m
$g_{0,R}$	Raman gain	$g_R I_0$	10^{-3}	1/m

Table A.4: Derived dimensional scales

Symbol	Description	Value	Unit
r_{core}	Core radius	12.7	μm
r_{clad}	Cladding radius	127	μm
n_{core}	Refractive index in fiber core	1.4512	-
n_{clad}	Refractive index in fiber cladding	1.4500	-
NA	Core numerical aperture	0.059	-
λ_s	Signal wavelength	1 064	nm
λ_p	Pump wavelength	976	nm
$V(\omega_s)$	Normalized signal frequency	4.43	-

Table A.5: Step-index fiber parameters

Parameter	Description	Value	Unit
σ_s^{abs}	Absorption cross-section (signal light)	$6 \cdot 10^{-27}$	m^2/ion
σ_s^{ems}	Emission cross-section (signal light)	$3.58 \cdot 10^{-25}$	m^2/ion
σ_p^{abs}	Absorption cross-section (pump light)	$1.429 \cdot 10^{-24}$	m^2/ion
σ_p^{ems}	Emission cross-section (pump light)	$1.776 \cdot 10^{-24}$	m^2/ion
$\mathcal{N}_{\text{total}}^{\text{Yb}}$	Total ion population concentration	$6 \cdot 10^{25}$	ion/m^3
τ	Upper level radiative lifetime	$8 \cdot 10^{-4}$	s

Table A.6: Active gain model parameters (ytterbium-doped fiber)

Parameter	Description	Value	Unit
C_p	Specific heat capacity	703	J/(kg·K)
ρ_0	Mean density	2 201	kg/m^3
κ	Thermal conductivity	1.38	W/(m·K)
dn/dT	Thermo-optic coefficient	$1.285 \cdot 10^{-5}$	1/K

Table A.7: Heat coupling model parameters (silica glass)

Parameter	Description	Value	Unit
g_R	Bulk Raman gain coefficient	10^{-13}	m/W
λ_s	Stokes field wavelength	1 116	nm

Table A.8: Raman gain model parameters

A.4 Bessel functions

n	m					
	0	1	2	3	4	5
1	2.4048	3.8317	5.1356	6.3802	7.5883	8.7715
2	5.5201	7.0156	8.4172	9.7610	11.0647	12.3386
3	8.6537	10.1735	11.6198	13.0152	14.3725	15.7002
4	11.7915	13.3237	14.7960	16.2235	17.6160	18.9801

Table A.9: n-th roots of m-th order Bessel functions J_m

n	m					
	0	1	2	3	4	5
1	3.8317	1.8412	3.0542	4.2012	5.3175	6.4156
2	7.0156	5.3314	6.7061	8.0152	9.2824	10.5199
3	10.1735	8.5363	9.9695	11.3459	12.6819	13.9872
4	13.3237	11.7060	13.1704	14.5858	15.9641	17.3128

Table A.10: n-th roots of m-th order Bessel functions derivatives J'_m

Appendix B

Fiber Amplifier Model Computation

B.1 Non-dimensionalization of the governing equations

As in many other applications, non-dimensionalization is essential in the numerical computation of the solution to the Maxwell equations. In particular, when the scales involved are very disparate as in the case of optical fiber amplifiers. We proceed by first non-dimensionalizing the linear time-harmonic Maxwell equations (1.16)–(1.17), then consider the active gain model (5.9)–(5.10).

Non-dimensional Maxwell equations. First, we define the non-dimensional variables $\hat{x}$, $\hat{y}$, $\hat{z}$, $\hat{\omega}_k$, $\hat{\boldsymbol{E}}_k$, $\hat{\boldsymbol{H}}_k$ by

$$x = l_0\hat{x},\ y = l_0\hat{y},\ z = l_0\hat{z}, \tag{B.1}$$

$$\omega_k = \omega_0\hat{\omega}_k, \tag{B.2}$$

$$\boldsymbol{E}_k = E_0\hat{\boldsymbol{E}}_k, \tag{B.3}$$

$$\boldsymbol{H}_k = H_0\hat{\boldsymbol{H}}_k, \tag{B.4}$$

where l_0, ω_0, E_0, and H_0 are appropriate dimensional scales, and the index $k \in \{s, p\}$ refers to the signal or pump field, respectively. See Table A.3 for an overview of the selected

dimensional scales in the fiber amplifier model. Then,

$$\frac{\partial}{\partial x_i} = \frac{\partial}{\partial \hat{x}_i}\frac{\partial \hat{x}_i}{\partial x_i} = l_0 \frac{\partial}{\partial \hat{x}_i}, \quad i = 1, 2, 3, \tag{B.5}$$

where $\{x_1, x_2, x_3\} \equiv \{x, y, z\}$; consequently,

$$\nabla \times = \frac{1}{l_0} \hat{\nabla} \times . \tag{B.6}$$

Plugging these relations into (1.16)–(1.17) yields

$$\hat{\nabla} \times \hat{\boldsymbol{E}}_k = -i \frac{H_0 l_0 \omega_0 \mu_0}{E_0} \mu_{r,k} \hat{\omega}_k \hat{\boldsymbol{H}}_k, \tag{B.7}$$

$$\hat{\nabla} \times \hat{\boldsymbol{H}}_k = i \frac{E_0 l_0 \omega_0 \varepsilon_0}{H_0} \varepsilon_{r,k} \hat{\omega}_k \hat{\boldsymbol{E}}_k, \tag{B.8}$$

where $\mu_{r,k} = \mu_k/\mu_0$, $\varepsilon_{r,k} = \varepsilon_k/\varepsilon_0$ are the (dimensionless) relative permeability and permittivity, respectively. By definition, $\mu_{r,k} = \varepsilon_{r,k} = 1$ in vacuum. Let

$$\frac{H_0 l_0 \omega_0 \mu_0}{E_0} = 1, \tag{B.9}$$

$$\frac{E_0 l_0 \omega_0 \varepsilon_0}{H_0} = 1; \tag{B.10}$$

then,

$$\omega_0 = c/l_0, \quad H_0 = \varepsilon_0 c E_0. \tag{B.11}$$

We can choose two problem-dependent dimensional scales. In the fiber amplifier model, we select an intensity scale I_0 that determines the field magnitudes through $I_0 = E_0 H_0$, as well as a length scale l_0; these also define the power scale $P_0 = I_0 l_0^2$. Now, we can write down

the non-dimensional version of the equations:

$$\hat{\nabla} \times \hat{\boldsymbol{E}}_k = -i\mu_{r,k}\hat{\omega}_k \hat{\boldsymbol{H}}_k, \tag{B.12}$$

$$\hat{\nabla} \times \hat{\boldsymbol{H}}_k = i\varepsilon_{r,k}\hat{\omega}_k \hat{\boldsymbol{E}}_k. \tag{B.13}$$

Next, consider the active gain model (5.9)–(5.10). Let $\hat{g}_k$ denote the non-dimensional gain function with scale g_0. The dimensional scale g_0 can be calculated from appropriate scalings for absorption and emission cross-sections and the ion population concentration,

$$\sigma_k^{\{\text{abs,ems}\}} = \sigma_0 \hat{\sigma}_k^{\{\text{abs,ems}\}}, \quad \mathcal{N}^{\text{Yb}}_{\{\text{total,ex,gr}\}} = \nu_0 \hat{\mathcal{N}}^{\text{Yb}}_{\{\text{total,ex,gr}\}}, \tag{B.14}$$

so that $g_0 = \sigma_0 \nu_0$, yielding the non-dimensional active gain Maxwell system:

$$\hat{\nabla} \times \hat{\boldsymbol{E}}_k = -i\hat{\omega}_k \hat{\boldsymbol{H}}_k, \tag{B.15}$$

$$\hat{\nabla} \times \hat{\boldsymbol{H}}_k = in^2\hat{\omega}_k \hat{\boldsymbol{E}}_k - nl_0 g_0 \hat{g}_k(\hat{\boldsymbol{E}}_{\{s,p\}})\hat{\boldsymbol{E}}_k. \tag{B.16}$$

Non-dimensional wavelength. To calculate the number of wavelengths for a given fiber domain, we must take the non-dimensionalization of the spatial component and the frequency component into account. By definition, the wavelength $\lambda_{0,k}$ (in vacuum) is

$$\lambda_{0,k} = \frac{2\pi c}{\omega_k} = \frac{2\pi c}{\omega_0 \hat{\omega}_k} = \frac{2\pi}{\hat{\omega}_k} l_0, \tag{B.17}$$

thus the non-dimensional wavelength is $\hat{\lambda}_{0,k} = \lambda_{0,k}/l_0 = 2\pi/\hat{\omega}_k$.

Inside a medium with uniform refractive index n,

$$\hat{\lambda}_k = \frac{\hat{\lambda}_{0,k}}{n} = \frac{2\pi}{\hat{\omega}_k n}, \tag{B.18}$$

so the *number of wavelengths per unit length* l_0 is

$$\frac{1}{\hat{\lambda}_k} = \frac{\hat{\omega}_k n}{2\pi}, \tag{B.19}$$

where $2\pi/\hat{\lambda}_k$ can be identified as the non-dimensional wavenumber.

Non-dimensional heat equation. We must also non-dimensionalize the heat equation for the coupled problem. Let

$$\delta T = T_0 \delta\hat{T}, \tag{B.20}$$

$$t = t_0 \hat{t}, \tag{B.21}$$

with an appropriate temperature scale T_0 and time scale t_0. Then, the heat equation (5.29) becomes

$$\frac{\rho_0 C_p T_0}{t_0} \frac{\partial(\delta\hat{T})}{\partial\hat{t}} - \frac{\kappa T_0}{l_0^2} \hat{\Delta}(\delta\hat{T}) = g_0 I_0 \hat{Q}(\hat{I}_{\{s,p\}}). \tag{B.22}$$

Rearranging the coefficients yields

$$\frac{\partial(\delta\hat{T})}{\partial\hat{t}} - \underbrace{\frac{\kappa t_0}{l_0^2 \rho_0 C_p}}_{\equiv\alpha_0} \hat{\Delta}(\delta\hat{T}) = \underbrace{\frac{g_0 I_0 t_0}{\rho_0 C_p T_0}}_{\equiv Q_0} \hat{Q}(\hat{I}_{\{s,p\}}), \tag{B.23}$$

where α_0 denotes a non-dimensional diffusivity scale, and Q_0 denotes a non-dimensional heat deposition scale. Choosing $T_0 = 1$ K, $t_0 = 1$ ms, $l_0 = 10$ μm, $I_0 = 10^{10}$ W/m^2, $g_0 = 0.1$ m^{-1} (cf. Table A.3), we obtain $\alpha_0 \approx 8.92$ and $Q_0 \approx 0.646$ for the silica glass material (see parameters in Table A.7).

B.2 Definition of guided LP modes

This section summarizes the linearly polarized guided modes for the step-index fiber with parameters given by Table A.5. This particular fiber supports four core-guided LP modes: $\mathrm{LP}_{01}, \mathrm{LP}_{11}, \mathrm{LP}_{21}$, and LP_{02}. Two of the modes, LP_{01} and LP_{02}, are radially symmetric. The LP_{11} and LP_{21} modes are asymmetric and consist of two (L^2-orthogonal) fields each, rotated by 90 degrees and 45 degrees, respectively; these rotated fields are denoted by $\{\mathrm{LP}_{11a}, \mathrm{LP}_{11b}\}$ and $\{\mathrm{LP}_{21a}, \mathrm{LP}_{21b}\}$. Six additional corresponding y-polarized LP modes exist. The plots in Figure B.1 show the magnitude of the electric fields (colors) as well as the orientation (glyphs) of the x-polarized light for the asymmetric modes.

The x-polarized electric fields (E_x-components) of the (unnormalized) guided modes are denoted by $\{\psi_{01}, \psi_{11a}, \psi_{11b}, \psi_{21a}, \psi_{21b}, \psi_{02}\}$, respectively. Let $x = r\cos\theta$, $y = r\sin\theta$, where $\theta \in [0, 2\pi)$; then,

$$\psi_{01}(r,\theta) = \begin{cases} J_0(\gamma_{01}r)/J_0(\gamma_{01}r_{\mathrm{core}}), & r \le r_{\mathrm{core}}, \\ K_0(\beta_{01}r)/K_0(\beta_{01}r_{\mathrm{core}}), & r > r_{\mathrm{core}}, \end{cases} \tag{B.24}$$

$$\psi_{11a}(r,\theta) = \cos(\theta) \begin{cases} J_1(\gamma_{11}r)/J_1(\gamma_{11}r_{\mathrm{core}}), & r \le r_{\mathrm{core}}, \\ K_1(\beta_{11}r)/K_1(\beta_{11}r_{\mathrm{core}}), & r > r_{\mathrm{core}}, \end{cases} \tag{B.25}$$

$$\psi_{11b}(r,\theta) = \sin(\theta) \begin{cases} J_1(\gamma_{11}r)/J_1(\gamma_{11}r_{\mathrm{core}}), & r \le r_{\mathrm{core}}, \\ K_1(\beta_{11}r)/K_1(\beta_{11}r_{\mathrm{core}}), & r > r_{\mathrm{core}}, \end{cases} \tag{B.26}$$

$$\psi_{21a}(r,\theta) = \cos(2\theta) \begin{cases} J_2(\gamma_{21}r)/J_2(\gamma_{21}r_{\mathrm{core}}), & r \le r_{\mathrm{core}}, \\ K_2(\beta_{21}r)/K_2(\beta_{21}r_{\mathrm{core}}), & r > r_{\mathrm{core}}, \end{cases} \tag{B.27}$$

$$\psi_{21b}(r,\theta) = \sin(2\theta) \begin{cases} J_2(\gamma_{21}r)/J_2(\gamma_{21}r_{\mathrm{core}}), & r \le r_{\mathrm{core}}, \\ K_2(\beta_{21}r)/K_2(\beta_{21}r_{\mathrm{core}}), & r > r_{\mathrm{core}}, \end{cases} \tag{B.28}$$

$$\psi_{02}(r,\theta) = \begin{cases} J_0(\gamma_{02}r)/J_0(\gamma_{02}r_{\mathrm{core}}), & r \le r_{\mathrm{core}}, \\ K_0(\beta_{02}r)/K_0(\beta_{02}r_{\mathrm{core}}), & r > r_{\mathrm{core}}, \end{cases} \tag{B.29}$$

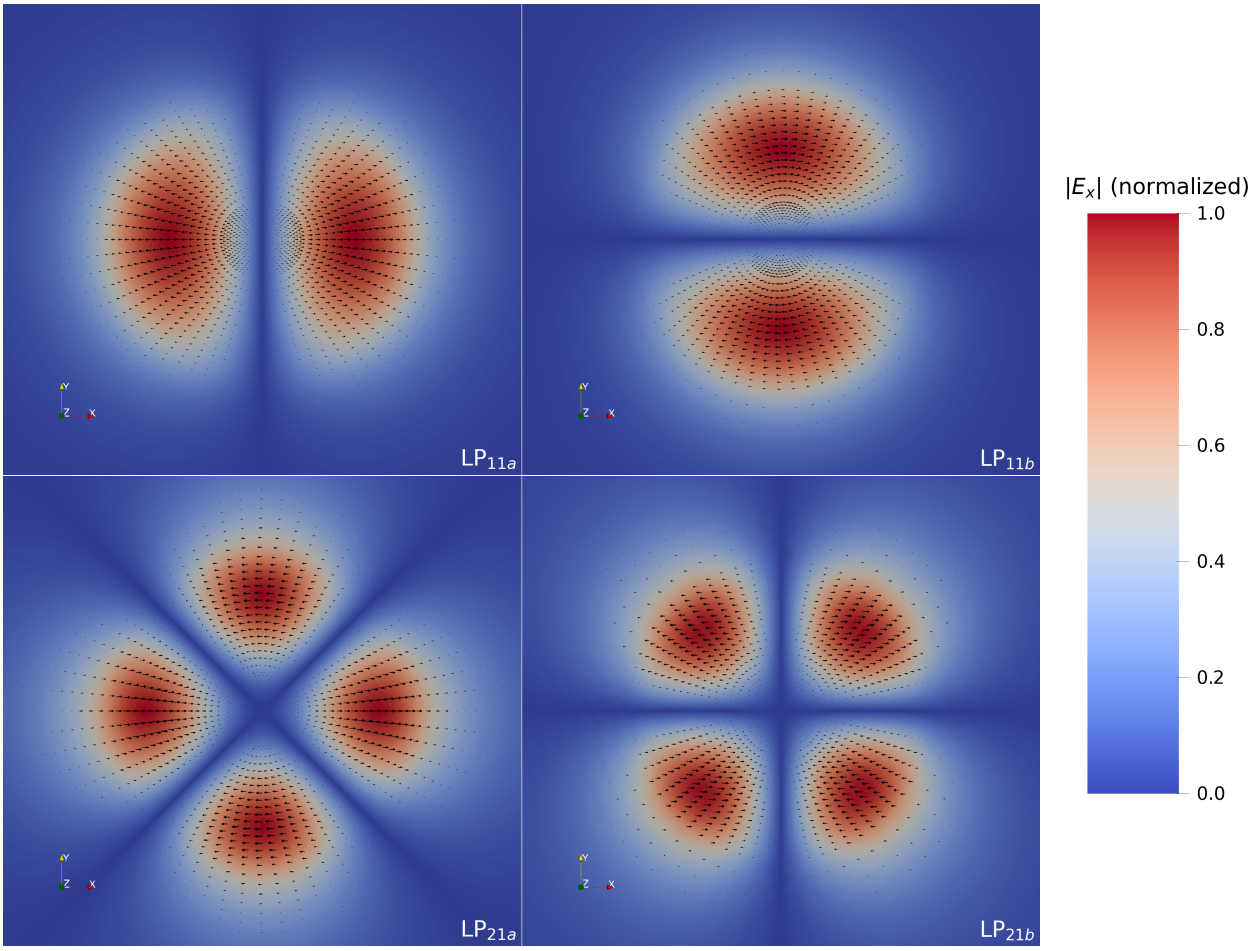


Figure B.1: Electric field magnitude and orientation of x-polarized asymmetric guided LP modes. The step-index fiber with parameters given in Table A.5 supports four guided modes: $\mathrm{LP}_{01}, \mathrm{LP}_{11}, \mathrm{LP}_{21}$, and LP_{02}. The radially asymmetric LP_{11} and LP_{21} modes have two possible rotations each.

with the (normalized) coefficients,

$$\{\gamma_{01}, \beta_{01}\} \approx \{1.53131, 3.12978\} \ , \tag{B.30}$$

$$\{\gamma_{11}, \beta_{11}\} \approx \{2.41319, 2.51336\} \ , \tag{B.31}$$

$$\{\gamma_{21}, \beta_{21}\} \approx \{1.81660, 1.42726\} \ , \tag{B.32}$$

$$\{\gamma_{02}, \beta_{02}\} \approx \{3.33123, 1.02145\} \ . \tag{B.33}$$

The ratio of the power that is confined to the core region, denoted Γ and previously

defined in (4.20), varies significantly between these modes. The respective confinement ratios for the core-guided LP modes are

$$\{\Gamma_{01}, \Gamma_{11}, \Gamma_{21}, \Gamma_{02}\} \approx \{96.11, 88.77, 74.79, 59.58\}\%. \tag{B.34}$$

B.3 Perfectly matched layer: implementation

Section 2.5.2 gave a brief introduction to the idea of using absorbing boundary layers in wave propagation problems posed in unbounded domains. While optical fibers have in principle finite length, the computation of the Maxwell model is too expensive to model an entire fiber. Therefore, we consider an artificially bounded fiber domain that can be thought of as a small section of an optical fiber. At the end of this bounded domain, we want to avoid artificial reflections of the wave caused by imposing (artificial) boundary conditions. Following [6], we summarize the key points behind complex coordinate stretching in PMLs for the DPG method and apply this technique to the Maxwell problem in the fiber domain.

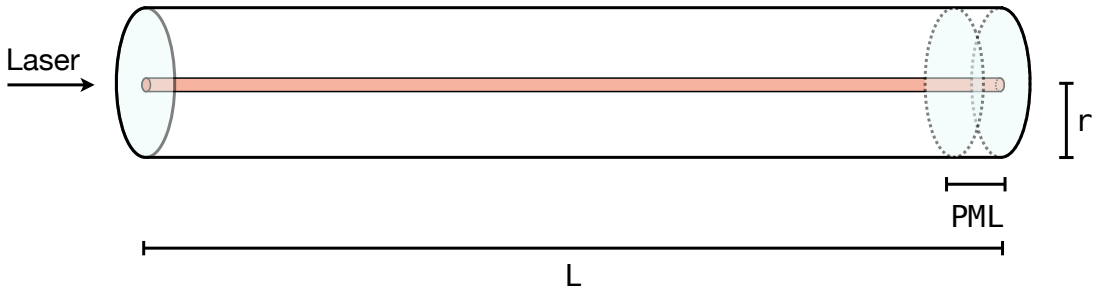


Figure B.2: Illustration of the PML region in the fiber amplifier model. A small layer with complex coordinate stretching at the fiber end is sufficient to absorb the outgoing wave.

Complex coordinate stretching in a fiber waveguide. Consider a bounded fiber domain $\Omega := \Omega_t \times (0, L) \subset \mathbb{R}^3$, where $\Omega_t := \{(x, y) : x^2 + y^2 < r^2\}$ is the transverse domain for a fiber of radius r, and L is the length of the (fiber) domain. Then, we define a computational fiber domain $\Omega_c := \Omega_t \times (0, l) \subset \Omega$, where $l < L$ is the length of the (computational) domain of interest. The PML region is given by $\Omega_{\text{PML}} := \Omega \backslash \Omega_c$. An illustration of the setup is shown

in Figure B.2. We define the following (uniaxial) complex stretching map:

$$\mathbb{R}^3 \ni (x, y, z) \longrightarrow (x, y, \tilde{z}) \in \mathbb{C}^3, \tag{B.35}$$

where

$$\tilde{z} = \begin{cases} z & , \text{ if } 0 < z < l, \\ z - if(z, \omega) & , \text{ if } l \leq z < L, \end{cases} \tag{B.36}$$

and $f(z, \omega) > 0$ is the stretching function. Since the coordinate stretching is only done inside the PML region, the solution in the domain of interest Ω_c coincides with its unstretched counterpart. Meanwhile, inside the PML region, the positive stretching function causes (outgoing) guided waves of the form $\boldsymbol{E}(x, y)e^{-ikz}$ to decay exponentially.

In our numerical experiments, we have used the stretching function

$$f(z, \omega) = \frac{C}{\omega} \left(\frac{z - l}{L - l} \right)^n, \tag{B.37}$$

where $C = 25, n = 3$, and the length of the PML region, $L - l$, is usually on the order of a few wavelengths of the fundamental mode.

Ultraweak DPG Maxwell formulation with PML. Let $\boldsymbol{J}$ denote the (diagonal) Jacobian corresponding to the (uniaxial) stretching defined in (B.35)–(B.36); $\boldsymbol{J}$ is given by:

$$\boldsymbol{J} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \partial\tilde{z}/\partial z \end{bmatrix}. \tag{B.38}$$

In our implementation of the PML, we follow [6]; that is, we first use Piola transformations for the exact sequence spaces to pull-back the Maxwell equations from the stretched coordinate system into the regular Cartesian coordinates, then develop the variational for-

mulation. The strong form of the pulled-back equations is:

$$\begin{cases} J^{-1}\boldsymbol{J}\nabla\times\boldsymbol{E}+i\omega\boldsymbol{\mu}\boldsymbol{J}^{-T}\boldsymbol{H}=\boldsymbol{0}, \\ J^{-1}\boldsymbol{J}\nabla\times\boldsymbol{H}-i\omega\boldsymbol{\varepsilon}\boldsymbol{J}^{-T}\boldsymbol{E}=\boldsymbol{0}, \end{cases} \tag{B.39}$$

where $J=\det\boldsymbol{J}$. We consider anisotropic, heterogeneous material parameters (i.e., $\boldsymbol{\mu}$ and $\boldsymbol{\varepsilon}$ are non-uniform second-rank tensors) which are needed to model birefringent fibers, for example. The pulled-back equations can now be used to derive the broken ultraweak variational formulation:

$$\begin{cases} \boldsymbol{E},\boldsymbol{H}\in(L^2(\Omega))^3,\ \hat{\boldsymbol{E}}\in\hat{\mathcal{U}},\ \hat{\boldsymbol{H}}\in H^{-1/2}(\mathrm{curl},\Gamma_h), \\ (\boldsymbol{E},\nabla_h\times\boldsymbol{F})+\langle\boldsymbol{n}\times\hat{\boldsymbol{E}},\boldsymbol{F}\rangle_{\Gamma_h}+(i\omega J\boldsymbol{J}^{-1}\boldsymbol{\mu}\boldsymbol{J}^{-T}\boldsymbol{H},\boldsymbol{F})=0, \quad \boldsymbol{F}\in H(\mathrm{curl},\Omega_h), \\ (\boldsymbol{H},\nabla_h\times\boldsymbol{G})+\langle\boldsymbol{n}\times\hat{\boldsymbol{H}},\boldsymbol{G}\rangle_{\Gamma_h}-(i\omega J\boldsymbol{J}^{-1}\boldsymbol{\varepsilon}\boldsymbol{J}^{-T}\boldsymbol{E},\boldsymbol{G})=0, \quad \boldsymbol{G}\in H(\mathrm{curl},\Omega_h), \end{cases} \tag{B.40}$$

where $\hat{\mathcal{U}}:=\big\{\hat{\boldsymbol{q}}\in H^{-1/2}(\mathrm{curl},\Gamma_h):\boldsymbol{n}\times\hat{\boldsymbol{q}}=\boldsymbol{n}\times\boldsymbol{E}_0 \text{ on } \Gamma\big\}$. In the fiber problem, we impose PEC boundary conditions ($\boldsymbol{n}\times\boldsymbol{E}_0=\boldsymbol{0}$) at the fiber exit ($z=L$), assuming that the (forward traveling) wave has decayed to zero (numerically) in the PML region ($l\le z<L$).

The stretching also translates into a modified computation for the adjoint graph norm in the test space (cf. Section 2.5.1). Denoting $\mathfrak{v}=(\boldsymbol{F},\boldsymbol{G})\in\mathcal{V}(\Omega_h)\equiv(H(\mathrm{curl},\Omega_h))^2$, the (quasi-optimal) test norm is given by:

$$\begin{aligned} \|\mathfrak{v}\|^2_{\mathcal{V}(\Omega_h)} &= \|\nabla_h\times\boldsymbol{F}+i\omega(J\boldsymbol{J}^{-1}\boldsymbol{\varepsilon}\boldsymbol{J}^{-T})^*\boldsymbol{G}\|^2 \\ &\quad+\|\nabla_h\times\boldsymbol{G}-i\omega(J\boldsymbol{J}^{-1}\boldsymbol{\mu}\boldsymbol{J}^{-T})^*\boldsymbol{F}\|^2+\alpha(\|\boldsymbol{F}\|^2+\|\boldsymbol{G}\|^2). \end{aligned} \tag{B.41}$$

Numerical example. To show the effectiveness of the PML in the fiber simulation, we consider the nonlinear gain problem (Chapter 5). In a short fiber ($L=1.2$) with moderate active gain, the signal laser is amplified along the length of the computational domain of interest ($0<z<l,\ l=0.9$) as shown in Figure B.3 (ca. 12 wavelengths). As the wave enters the artificial PML region ($l\le z<L$), it decays exponentially and vanishes completely within

just a few wavelengths, justifying the PEC boundary conditions at $z = 1.2$. The pump field, not shown in Figure B.3, exhibits a similar decay in the PML region but uses a stretching function corresponding to the pump frequency ω_p.

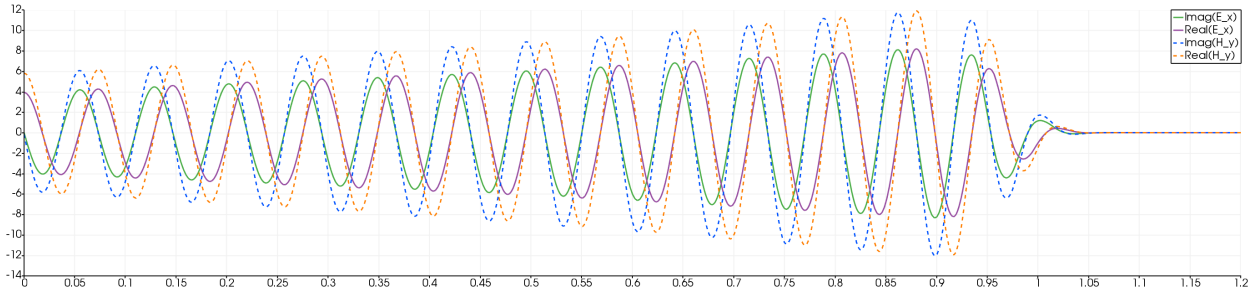


Figure B.3: Electric and magnetic field components plotted over the longitudinal fiber axis in the nonlinear active gain problem. The amplified signal field is decaying exponentially in the PML region.

B.4 Raman gain amplification

In this section, we state the Raman gain model proposed in [98] and our modifications that include a different non-dimensionalization and adjusted gain scaling for computing on short fibers. This modified model is the one used in the Raman gain simulations in Section 6.3. Similar to the active gain model, the Raman gain model couples the two propagating fields (signal/Stokes field and pump field) via a gain function g. Although Raman scattering is an optical nonlinearity typically modeled as a third-order susceptibility, the Raman gain polarization can be approximated by the following first-order complex-valued perturbation to the refractive index [98]:

$$\boldsymbol{P}_k^{\text{Raman gain}} = i\varepsilon_0 \frac{nc}{\omega_k} g_k \boldsymbol{E}_k, \tag{B.42}$$

where g_k is the Raman gain function (with unit m^{-1}). This gain function is given by:

$$g_k = g_k(I_l) := \Upsilon_R^k g_R I_l, \quad k, l \in \{s, p\},\ l \neq k, \tag{B.43}$$

where

$$\Upsilon_R^k = \begin{cases} -\omega_p/\omega_s & , \ k = p, \\ 1 & , \ k = s, \end{cases} \tag{B.44}$$

and g_R is the bulk Raman gain coefficient (an experimentally determined value). For example, in a real silica fiber we may assume $g_R \approx 10^{-13}$ m/W [118]. It is helpful to assume a realistic value which can be embedded into our model non-dimensionalization. A-posteriori we introduce a (non-dimensional) artificial gain scaling coefficient $\tilde{g}_R$ that enhances the Raman gain per unit length. By using $\tilde{g}_R$, we can then more clearly see by how much the gain in the short fiber needs to be artificially scaled compared to a realistic fiber. This is a similar approach as we took for the active gain model (cf. (5.34)) and differs from the approach taken in [98]. Let $\hat{g}_k = \hat{g}_k(\hat{I}_l)$ denote the non-dimensional Raman gain function with a dimensional scale $g_{0,R} := g_R I_0$. Proceeding analogous to the derivation of the non-dimensional active gain Maxwell system (B.15)–(B.16), we obtain:

$$\hat{\nabla} \times \hat{\boldsymbol{E}}_k = -i\hat{\omega}_k \hat{\boldsymbol{H}}_k, \tag{B.45}$$

$$\hat{\nabla} \times \hat{\boldsymbol{H}}_k = in^2 \hat{\omega}_k \hat{\boldsymbol{E}}_k - nl_0 g_{0,R} \tilde{g}_R \hat{g}_k(\hat{I}_l) \hat{\boldsymbol{E}}_k, \tag{B.46}$$

where $\hat{g}_k(\hat{I}_l) = \Upsilon_R^k \hat{I}_l$, and $\tilde{g}_R$ is the artificial scaling of the Raman gain coefficient used for a short fiber.

Bibliography

[1] G.P. Agrawal. *Nonlinear fiber optics*. 5th ed. Academic Press, 2012.

[2] J. Ahrens, B. Geveci, and C. Law. "Paraview: an end-user tool for large data visualization". In: *The Visualization Handbook*. Elsevier, 2005.

[3] M. Ainsworth, P. Monk, and W. Muniz. "Dispersive and dissipative properties of discontinuous Galerkin finite element methods for the second-order wave equation". In: *J. Sci. Comput.* 27.1-3 (2006), pp. 5–40.

[4] P.R. Amestoy et al. "A fully asynchronous multifrontal solver using distributed dynamic scheduling". In: *SIAM J. Matrix Anal. Appl.* 23.1 (2001), pp. 15–41.

[5] T. Arbogast and J.L. Bona. *Functional analysis for the applied mathematician*. Lecture notes; The University of Texas at Austin. 2010.

[6] A.V. Astaneh, B. Keith, and L. Demkowicz. "On perfectly matched layers for discontinuous Petrov–Galerkin methods". In: *Comput. Mech.* 63.6 (2019), pp. 1131–1145.

[7] A.V. Astaneh et al. "High-order polygonal discontinuous Petrov–Galerkin (PolyDPG) methods using ultraweak formulations". In: *Comput. Methods Appl. Mech. Engrg.* 332 (2018), pp. 686–711.

[8] I. Babuška. "Error-bounds for finite element method". In: *Numer. Math.* 16.4 (1971), pp. 322–333.

[9] I.M. Babuška and S.A. Sauter. "Is the pollution effect of the FEM avoidable for the Helmholtz equation considering high wave numbers?" In: *SIAM J. Numer. Anal.* 34.6 (1997), pp. 2392–2423.

[10] C. Bacuta et al. "Analysis of non-conforming DPG methods on polyhedral meshes using fractional Sobolev norms". In: *Comput. Math. Appl.* (2020).

[11] J. Badger, S. Henneking, and L. Demkowicz. "Sum factorization for fast integration of DPG matrices on prismatic elements". In: *Finite Elem. Anal. Des.* 172 (2020), p. 103385.

[12] S. Balay et al. *PETSc users manual*. Tech. rep. ANL-95/11 - Revision 3.13. Argonne National Laboratory, 2020. URL: `https://www.mcs.anl.gov/petsc`.

[13] A.T. Barker et al. "A scalable preconditioner for a primal discontinuous Petrov–Galerkin method". In: *SIAM J. Sci. Comput.* 40.2 (2018), A1187–A1203.

[14] J.P. Berenger. "A perfectly matched layer for the absorption of electromagnetic waves". In: *J. Comput. Phys.* 114.2 (1994), pp. 185–200.

[15] F. Bertrand et al. "Recent advances in least-squares and discontinuous Petrov–Galerkin finite element methods". In: *Comput. Methods Appl. Math.* 19.3 (2019), pp. 395–397.

[16] T.D. Blacker, W.J. Bohnhoff, and T.L. Edwards. *CUBIT mesh generation environment. Volume 1: users manual*. Tech. rep. Sandia National Laboratories, 1994.

[17] J. Bramble and J. Pasciak. "Analysis of a finite PML approximation for the three dimensional time-harmonic Maxwell and acoustic scattering problems". In: *Math. Comput.* 76.258 (2007), pp. 597–614.

[18] D. Broersen, W. Dahmen, and R. Stevenson. "On the stability of DPG formulations of transport equations". In: *Math. Comput.* 87.311 (2018), pp. 1051–1082.

[19] T. Bui-Thanh and O. Ghattas. "A PDE-constrained optimization approach to the discontinuous Petrov–Galerkin method with a trust region inexact Newton-CG solver". In: *Comput. Methods Appl. Mech. Engrg.* 278 (2014), pp. 20–40.

[20] C. Carstensen, L. Demkowicz, and J. Gopalakrishnan. "Breaking spaces and forms for the DPG method and applications including Maxwell equations". In: *Comput. Math. Appl.* 72.3 (2016), pp. 494–522.

[21] J. Chan, L. Demkowicz, and R. Moser. "A DPG method for steady viscous compressible flow". In: *Comput. Fluids* 98 (2014), pp. 69–90.

[22] Z. Chen and X. Xiang. "A source transfer domain decomposition method for Helmholtz equations in unbounded domain". In: *SIAM J. Numer. Anal.* 51.4 (2013), pp. 2331–2356.

[23] C. Chevalier and F. Pellegrini. "PT-Scotch: a tool for efficient parallel graph ordering". In: *Parallel Comput.* 34.6-8 (2008), pp. 318–331.

[24] W.C. Chew and W.H. Weedon. "A 3D perfectly matched medium from modified Maxwell's equations with stretched coordinates". In: *Microw. Opt. Technol. Lett.* 7.13 (1994), pp. 599–604.

[25] A. Cohen, W. Dahmen, and G. Welper. "Adaptivity and variational stabilization for convection-diffusion equations". In: *ESAIM Math. Model. Numer. Anal.* 46.5 (2012), pp. 1247–1273.

[26] L. Demkowicz. *Computing with hp Finite Elements. I. One and Two Dimensional Elliptic and Maxwell Problems*. Chapman & Hall/CRC Press, Taylor and Francis, 2006.

[27] L. Demkowicz. *Mathematical theory of finite elements*. Lecture notes; The University of Texas at Austin. 2020.

[28] L. Demkowicz. "Polynomial exact sequences and projection-based interpolation with application to Maxwell equations". In: *Mixed Finite Elements, Compatibility Conditions, and Applications*. Springer, 2008, pp. 101–158.

[29] L. Demkowicz and J. Gopalakrishnan. "A class of discontinuous Petrov–Galerkin methods. II. Optimal test functions". In: *Numer. Methods Partial Differ. Equ.* 27.1 (2011), pp. 70–105.

[30] L. Demkowicz and J. Gopalakrishnan. "A class of discontinuous Petrov–Galerkin methods. Part I: the transport equation". In: *Comput. Methods Appl. Mech. Engrg.* 199.23-24 (2010), pp. 1558–1572.

[31] L. Demkowicz and J. Gopalakrishnan. "Analysis of the DPG method for the Poisson equation". In: *SIAM J. Numer. Anal.* 49.5 (2011), pp. 1788–1809.

[32] L. Demkowicz and J. Gopalakrishnan. "Discontinuous Petrov–Galerkin (DPG) method". In: *Encyclopedia of Computational Mechanics Second Edition* (2017), pp. 1–15.

[33] L. Demkowicz, J. Gopalakrishnan, and A.H. Niemi. "A class of discontinuous Petrov–Galerkin methods. Part III: adaptivity". In: *Appl. Numer. Math.* 62.4 (2012), pp. 396–427.

[34] L. Demkowicz and N. Heuer. "Robust DPG method for convection-dominated diffusion problems". In: *SIAM J. Numer. Anal.* 51.5 (2013), pp. 2514–2537.

[35] L. Demkowicz, J.T. Oden, and W. Rachowicz. "Toward a universal *hp* adaptive finite element strategy. Part 1: constrained approximation and data structure". In: *Comput. Methods Appl. Mech. Engrg.* 77 (1989), pp. 79–112.

[36] L. Demkowicz et al. "A spacetime DPG method for the Schrödinger equation". In: *SIAM J. Numer. Anal.* 55.4 (2017), pp. 1740–1759.

[37] L. Demkowicz et al. *Computing with hp Finite Elements. II. Frontiers: Three Dimensional Elliptic and Maxwell Problems with Applications*. Chapman & Hall/CRC, 2007.

[38] L. Demkowicz et al. *The double adaptivity paradigm (how to circumvent the discrete inf–sup conditions of Babuška and Brezzi)*. Tech. rep. 19-07. Oden Institute, 2019.

[39] L. Demkowicz et al. "Wavenumber explicit analysis of a DPG method for the multidimensional Helmholtz equation". In: *Comput. Methods Appl. Mech. Engrg.* 213 (2012), pp. 126–138.

[40] A. Deraemaeker, I. Babuška, and P. Bouillard. "Dispersion and pollution of the FEM solution for the Helmholtz equation in one, two and three dimensions". In: *Int. J. Numer. Methods Eng.* 46.4 (1999), pp. 471–499.

[41] K. Devine et al. "Zoltan data management services for parallel dynamic applications". In: *Comput. Sci. Eng.* 4.2 (2002), pp. 90–97.

[42] W. Dörfler. "A convergent adaptive algorithm for Poisson's equation". In: *SIAM J. Numer. Anal.* 33.3 (1996), pp. 1106–1124.

[43] D. Drake et al. "Simulation of optical fiber amplifier gain using equivalent short fibers". In: *Comput. Methods Appl. Mech. Engrg.* (2019), p. 112698.

[44] T. Eidam et al. "Experimental observations of the threshold-like onset of mode instabilities in high power fiber amplifiers". In: *Opt. Express* 19.14 (2011), pp. 13218–13224.

[45] B. Engquist and L. Ying. "Sweeping preconditioner for the Helmholtz equation: hierarchical matrix representation". In: *Comm. Pure Appl. Math.* 64.5 (2011), pp. 697–735.

[46] B. Engquist and L. Ying. "Sweeping preconditioner for the Helmholtz equation: moving perfectly matched layers". In: *Multiscale Model. Simul.* 9.2 (2011), pp. 686–710.

[47] J. Ernesti and C. Wieners. "Space-time discontinuous Petrov–Galerkin methods for linear wave equations in heterogeneous media". In: *Comput. Methods Appl. Math.* 19.3 (2019), pp. 465–481.

[48] O.G. Ernst and M.J. Gander. "Why it is difficult to solve Helmholtz problems with classical iterative methods". In: *Numerical Analysis of Multiscale Problems. Lect. Notes Comput. Sci. Eng.* Vol. 83. Springer, 2012, pp. 325–363.

[49] F. Fuentes, L. Demkowicz, and A. Wilder. "Using a DPG method to validate DMA experimental calibration of viscoelastic materials". In: *Comput. Math. Appl.* 325 (2017), pp. 748–765.

[50] F. Fuentes et al. "Orientation embedded high order shape functions for the exact sequence elements of all shapes". In: *Comput. Math. Appl.* 70.4 (2015), pp. 353–458.

[51] T. Führer and N. Heuer. "Robust coupling of DPG and BEM for a singularly perturbed transmission problem". In: *Comput. Math. Appl.* 74.8 (2017), pp. 1940–1954.

[52] T. Führer, N. Heuer, and J.S. Gupta. "A time-stepping DPG scheme for the heat equation". In: *Comput. Methods Appl. Math.* 17.2 (2017), pp. 237–252.

[53] T. Führer, N. Heuer, and A.H. Niemi. "An ultraweak formulation of the Kirchhoff–Love plate bending model and DPG approximation". In: *Math. Comput.* 88.318 (2019), pp. 1587–1619.

[54] T. Führer, N. Heuer, and F.J. Sayas. "An ultraweak formulation of the Reissner–Mindlin plate bending model and DPG approximation". In: *Numer. Math.* 145.2 (2020), pp. 313–344.

[55] M.J. Gander, I.G. Graham, and E.A. Spence. "Applying GMRES to the Helmholtz equation with shifted Laplacian preconditioning: what is the largest shift for which wavenumber-independent convergence is guaranteed?" In: *Numer. Math.* 131.3 (2015), pp. 567–614.

[56] M.J. Gander and H. Zhang. "A class of iterative solvers for the Helmholtz equation: factorizations, sweeping preconditioners, source transfer, single layer potentials, polarized traces, and optimized Schwarz methods". In: *SIAM Rev.* 61.1 (2019), pp. 3–76.

[57] P. Gatto. "Modeling bone conduction of sound in the human head using hp-finite elements". PhD thesis. The University of Texas at Austin, 2012.

[58] A. George. "Nested dissection of a regular finite element mesh". In: *SIAM J. Numer. Anal.* 10.2 (1973), pp. 345–363.

[59] F. Gonthier et al. "Mode coupling in nonuniform fibers: comparison between coupled-mode theory and finite-difference beam-propagation method simulations". In: *J. Opt. Soc. Am. B* 8.2 (1991), pp. 416–421.

[60] J. Gopalakrishnan, I. Muga, and N. Olivares. "Dispersive and dissipative errors in the DPG method with scaled norms for Helmholtz equation". In: *SIAM J. Sci. Comput.* 36.1 (2014), A20–A39.

[61] J. Gopalakrishnan and W. Qiu. "An analysis of the practical DPG method". In: *Math. Comput.* 83.286 (2014), pp. 537–552.

[62] J. Gopalakrishnan and J. Schöberl. "Degree and wavenumber [in] dependence of Schwarz preconditioner for the DPG method". In: *Spectral and High Order Methods for Partial Differential Equations ICOSAHOM 2014*. Springer, 2015, pp. 257–265.

[63] J. Gopalakrishnan and P. Sepúlveda. "A space-time DPG method for the wave equation in multiple dimensions". In: *Space-Time Methods: Applications to Partial Differential Equations*. De Gruyter, 2019, pp. 117–140.

[64] D.J. Griffiths. *Introduction to electrodynamics*. 3rd ed. Prentice Hall, 1999.

[65] J. Grosek. *Coupled mode theory fiber amplifier model*. Tech. rep. Air Force Research Laboratory, 2018.

[66] J. Grosek et al. "Laser simulation at the Air Force Research Laboratory". In: *XXI International Symposium of High Power Laser Systems and Applications 2016*. International Society for Optics and Photonics. 2017, 102540N.

[67] K.R. Hansen et al. “Thermally induced mode coupling in rare-earth doped fiber amplifiers”. In: *Opt. Lett.* 37.12 (2012), pp. 2382–2384.

[68] K.R. Hansen et al. “Thermo-optical effects in high-power ytterbium-doped fiber amplifiers”. In: *Opt. Express* 19.24 (2011), pp. 23965–23980.

[69] S. Henneking and L. Demkowicz. “A numerical study of the pollution error and DPG adaptivity for long waveguide simulations”. In: *Comput. Math. Appl.* (2020).

[70] S. Henneking, J. Grosek, and L. Demkowicz. “Model and computational advancements to full vectorial Maxwell model for studying fiber amplifiers”. In: *Comput. Math. Appl.* 85 (2021), pp. 30–41.

[71] N. Heuer and M. Karkulik. “A robust DPG method for singularly perturbed reaction-diffusion problems”. In: *SIAM J. Numer. Anal.* 55.3 (2017), pp. 1218–1242.

[72] F. Ihlenburg and I. Babuška. “Finite element solution of the Helmholtz equation with high wave number part I: the h-version of the FEM”. In: *Comput. Math. Appl.* 30.9 (1995), pp. 9–37.

[73] 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”. In: *SIAM J. Numer. Anal.* 34.1 (1997), pp. 315–358.

[74] J.D. Jackson. *Classical electrodynamics*. 3rd ed. John Wiley & Sons, 1999.

[75] C. Jauregui, J. Limpert, and A. Tünnermann. “High-power fibre lasers”. In: *Nat. Photonics* 7.11 (2013), p. 861.

[76] C. Jauregui, C. Stihler, and J. Limpert. “Transverse mode instability”. In: *Adv. Opt. Photonics* 12.2 (2020), pp. 429–484.

[77] C. Jauregui et al. “Temperature-induced index gratings and their impact on mode instabilities in high-power fiber laser systems”. In: *Opt. Express* 20.1 (2012), pp. 440–451.

[78] G. Karypis and V. Kumar. "A fast and high quality multilevel scheme for partitioning irregular graphs". In: *SIAM J. Sci. Comput.* 20.1 (1998), pp. 359–392.

[79] B. Keith, F. Fuentes, and L. Demkowicz. "The DPG methodology applied to different variational formulations of linear elasticity". In: *Comput. Methods Appl. Mech. Engrg.* 309 (2016), pp. 579–609.

[80] B. Keith et al. "An ultraweak DPG method for viscoelastic fluids". In: *J. Non-Newton. Fluid Mech.* 247 (2017), pp. 107–122.

[81] B. Keith et al. "Discrete least-squares finite element methods". In: *Comput. Methods Appl. Mech. Engrg.* 327 (2017), pp. 226–255.

[82] M.S. Khaira, G.L. Miller, and T.J. Sheffler. *Nested dissection: a survey and comparison of various nested dissection algorithms*. Tech. rep. CMU-CS-92-106R. Carnegie-Mellon University. Department of Computer Science, 1992.

[83] K. Kim. "Finite element modeling of electromagnetic radiation and induced heat transfer in the human body". PhD thesis. The University of Texas at Austin, 2013.

[84] A. Kobyakov, M. Sauer, and D. Chowdhury. "Stimulated Brillouin scattering in optical fibers". In: *Adv. Opt. Photonics* 2.1 (2010), pp. 1–59.

[85] W. Leng and L. Ju. "A diagonal sweeping domain decomposition method with source transfer for the Helmholtz equation". In: *arXiv preprint arXiv:2002.05327* (2020).

[86] J. Li and L. Demkowicz. "An Lp-DPG method for the convection-diffusion problem". In: *Comput. Math. Appl.* (2020).

[87] X. Li and X. Xu. "Domain decomposition preconditioners for the discontinuous Petrov–Galerkin method". In: *ESAIM Math. Model. Numer. Anal.* 51.3 (2017), pp. 1021–1044.

[88] J.M. Melenk. "On generalized finite element methods". PhD thesis. University of Maryland at College Park, 1995.

[89] J.M. Melenk and S.A. Sauter. "Wavenumber explicit convergence analysis for Galerkin discretizations of the Helmholtz equation". In: *SIAM J. Numer. Anal.* 49.3 (2011), pp. 1210–1243.

[90] J.M. Melenk and S.A. Sauter. "Wavenumber-explicit hp-FEM analysis for Maxwell's equations with transparent boundary conditions". In: *Found. Comput. Math.* (2020), pp. 1–117.

[91] C. Michler et al. "Improving the performance of perfectly matched layers by means of hp-adaptivity". In: *Numer. Methods Partial Differ. Equ.* 23.4 (2007), pp. 832–858.

[92] J. Mora and L. Demkowicz. "Fast integration of DPG matrices based on sum factorization for all the energy spaces". In: *Comput. Methods Appl. Math.* 19.3 (2019), pp. 523–555.

[93] I. Muga and K.G. van der Zee. "Discretization of linear problems in Banach spaces: Residual minimization, nonlinear Petrov–Galerkin, and monotone mixed methods". In: *SIAM J. Numer. Anal.* 58.6 (2020), pp. 3406–3426.

[94] J. Muñoz-Matute, D. Pardo, and L. Demkowicz. "A DPG-based time-marching scheme for linear hyperbolic problems". In: *Comput. Methods Appl. Mech. Engrg.* 373 (2021), p. 113539.

[95] J. Muñoz-Matute, D. Pardo, and L. Demkowicz. "Equivalence between the DPG method and the Exponential Integrators for linear parabolic problems". In: *J. Comput. Phys.* (2020), p. 110016.

[96] S. Naderi et al. "Investigations of modal instabilities in fiber amplifiers through detailed numerical simulations". In: *Opt. Express* 21.13 (2013), pp. 16111–16129.

[97] S. Nagaraj, S. Petrides, and L. Demkowicz. "Construction of DPG Fortin operators for second order problems". In: *Comput. Math. Appl.* 74.8 (2017), pp. 1964–1980.

[98] S. Nagaraj et al. "A 3D DPG Maxwell approach to nonlinear Raman gain in fiber laser amplifiers". In: *J. Comput. Phys.* 2 (2019), p. 100002.

[99] A.H. Niemi, J. Bramwell, and L. Demkowicz. "Discontinuous Petrov–Galerkin method with optimal test functions for thin-body problems in solid mechanics". In: *Comput. Methods Appl. Mech. Engrg.* 200.9-12 (2011), pp. 1291–1300.

[100] J.T. Oden et al. "Toward a universal *hp* adaptive finite element strategy. Part 2: a posteriori error estimation". In: *Comput. Methods Appl. Mech. Engrg.* 77 (1989), pp. 113–180.

[101] H.M. Pask et al. "Ytterbium-doped silica fiber lasers: versatile sources for the 1-1.2 μm region". In: *IEEE J. Sel. Top. Quantum Electron.* 1.1 (1995), pp. 2–13.

[102] S. Petrides. "Adaptive multilevel solvers for the discontinuous Petrov–Galerkin method with an emphasis on high-frequency wave propagation problems". PhD thesis. The University of Texas at Austin, 2019.

[103] S. Petrides and L. Demkowicz. "An adaptive DPG method for high frequency time-harmonic wave propagation problems". In: *Comput. Math. Appl.* 74.8 (2017), pp. 1999–2017.

[104] S. Petrides and L. Demkowicz. "An adaptive multigrid solver for DPG methods with applications in linear acoustics and electromagnetics". In: *Comput. Math. Appl.* 87 (2021), pp. 12–26.

[105] W. Rachowicz, J.T. Oden, and L. Demkowicz. "Toward a universal *hp* adaptive finite element strategy. Part 3: design of *hp* meshes". In: *Comput. Methods Appl. Mech. Engrg.* 77 (1989), pp. 181–212.

[106] W. Rachowicz, A. Zdunek, and W. Cecot. "A discontinuous Petrov–Galerkin method for compressible viscous flows in three dimensions". In: *AIP Conference Proceedings*. Vol. 2239. 1. AIP Publishing LLC. 2020, p. 020044.

[107] D.J. Richardson, J. Nilsson, and W.A. Clarkson. "High power fiber lasers: current status and future perspectives". In: *J. Opt. Soc. Am. B* 27.11 (2010), B63–B92.

[108] N.V. Roberts, T. Bui-Thanh, and L. Demkowicz. "The DPG method for the Stokes problem". In: *Comput. Math. Appl.* 67.4 (2014), pp. 966–995.

[109] N.V. Roberts and S. Henneking. "Time-stepping DPG formulations for the heat equation". In: *Comput. Math. Appl.* (2020).

[110] K. Saitoh and M. Koshiba. "Full-vectorial finite element beam propagation method with perfectly matched layers for anisotropic optical waveguides". In: *J. Light. Technol.* 19.3 (2001), pp. 405–413.

[111] J. Salazar, J. Mora, and L. Demkowicz. "Alternative enriched test spaces in the DPG method for singular perturbation problems". In: *Comput. Methods Appl. Math.* 19.3 (2019), pp. 603–630.

[112] J. Schöberl. "NETGEN an advancing front 2D/3D-mesh generator based on abstract rules". In: *Comput. Visual Sci.* 1.1 (1997), pp. 41–52.

[113] A.V. Smith and J.J. Smith. "Increasing mode instability thresholds of fiber amplifiers by gain saturation". In: *Opt. Express* 21.13 (2013), pp. 15168–15182.

[114] A.V. Smith and J.J. Smith. "Mode instability in high power fiber amplifiers". In: *Opt. Express* 19.11 (2011), pp. 10180–10192.

[115] A.V. Smith and J.J. Smith. "Mode instability thresholds for Tm-doped fiber amplifiers pumped at 790 nm". In: *Opt. Express* 24.2 (2016), pp. 975–992.

[116] A.V. Smith and J.J. Smith. "Spontaneous Rayleigh seed for stimulated Rayleigh scattering in high power fiber amplifiers". In: *IEEE Photonics J.* 5.5 (2013), pp. 7100807–7100807.

[117] R.G. Smith. "Optical power handling capacity of low loss optical fibers as determined by stimulated Raman and Brillouin scattering". In: *Appl. Opt.* 11.11 (1972), pp. 2489–2494.

[118] R. Stegeman et al. “Raman gain measurements in bulk glass samples”. In: *J. Opt. Soc. Am. B* 22.9 (2005), pp. 1861–1867.

[119] C. Stihler et al. “Phase-shift evolution of the thermally-induced refractive index grating in high-power fiber laser systems induced by pump-power variations”. In: *Opt. Express* 26.15 (2018), pp. 19489–19497.

[120] C.C. Stolk, M. Ahmed, and S.K. Bhowmik. “A multigrid method for the Helmholtz equation with optimized coarse grid corrections”. In: *SIAM J. Sci. Comput.* 36.6 (2014), A2819–A2841.

[121] M. Taus et al. “L-Sweeps: a scalable, parallel preconditioner for the high-frequency Helmholtz equation”. In: *J. Comput. Phys.* 420 (2020), p. 109706.

[122] The HDF Group. *Hierarchical data format version 5*. URL: `http://www.hdfgroup.org/HDF5`.

[123] P. Tsuji, B. Engquist, and L. Ying. “A sweeping preconditioner for time-harmonic Maxwell’s equations with finite elements”. In: *J. Comput. Phys.* 231.9 (2012), pp. 3770–3783.

[124] B. Ward. “Modeling of transient modal instability in fiber amplifiers”. In: *Opt. Express* 21.10 (2013), pp. 12053–12067.

[125] B. Ward, C. Robin, and I. Dajani. “Origin of thermal modal instabilities in large mode area fiber amplifiers”. In: *Opt. Express* 20.10 (2012), pp. 11407–11422.

[126] P.L. Wizinowich et al. “The W.M. Keck Observatory laser guide star adaptive optics system: overview”. In: *Publ. Astron. Soc. Pac.* 118.840 (2006), p. 297.

[127] J. Zitelli et al. “A class of discontinuous Petrov–Galerkin methods. Part IV: the optimal test norm and time-harmonic wave propagation in 1D”. In: *J. Comput. Phys.* 230.7 (2011), pp. 2406–2432.