LOCAL FOURIER ANALYSIS OF P-MULTIGRID FOR HIGH-ORDER FINITE ELEMENT OPERATORS*

JEREMY L THOMPSON[†], JED BROWN[‡], AND YUNHUI HE[§]

Abstract. Multigrid methods are popular for solving linear systems derived from discretizing PDEs. Local Fourier Analysis (LFA) is a technique for investigating and tuning multigrid methods. P-multigrid is popular for high-order or spectral finite element methods, especially on unstructured meshes. In this paper, we introduce LFAToolkit.jl, a new Julia package for LFA of high-order finite element methods. LFAToolkit.jl analyzes preconditioning techniques for arbitrary systems of second order PDEs and supports mixed finite element methods. Specifically, we develop LFA of p-multigrid with arbitrary second-order PDEs using high-order finite element discretizations and examine the performance of Jacobi and Chebyshev smoothing for two-grid schemes with aggressive p-coarsening. A natural extension of this LFA framework is the analysis of h-multigrid for finite element discretizations or finite difference discretizations that can be represented in the language of finite elements. With this extension, we can replicate previous work on the LFA of h-multigrid for arbitrary order discretizations using a convenient and extensible abstraction. Examples in one, two, and three dimensions are presented to validate our LFA of p-multigrid for the Laplacian and linear elasticity.

Key words. local Fourier analysis, p-multigrid, high-order, finite elements

1. Introduction. Multigrid methods [4, 6, 29] are popular for solving linear systems derived from discretizing PDEs. Local Fourier Analysis (LFA) [3, 33] is a powerful tool for predicting multigrid performance and tuning of multigrid components by examining the spectral radius and/or condition number of the *symbol* of the underlying operator, which enables sharp predictions for large-scale problems.

High-order finite element methods offer accuracy advantages over low-order finite elements [13, 24, 25]; however, high-order finite elements are less common because a linear operator or the Jacobian of a non-linear operator rapidly loses sparsity in a sparse matrix representation. Matrix-free formulations of these methods [7, 14, 20], provide efficient implementations of these methods on modern hardware [8, 15, 21].

LFA of h-multigrid with high-order finite elements was developed in [17] for Lagrange bases with uniformly spaced nodes. P-multigrid, developed by Rønquist and Patera [28], is based on decreasing the order of the bases in high-order or spectral finite element methods rather than aggregating elements to coarsen the mesh. While it is possible [12] to use high-order finite element methods with matrix-free implementation of h-multigrid preconditioning on complex problems, p-multigrid can be a more natural fit for high-order methods on unstructured meshes. The benefits of using *p*-multigrid for matrix-free operators on unstructured meshes for solid mechanics engineering applications is discussed in [9].

In this paper, we develop LFA of p-multigrid with arbitrary second-order PDEs

†Department of Computer Science, University of Colorado, Boulder, CO (jeremy@jeremylt.org).

‡Department of Computer Science, University of Colorado, Boulder, CO (jed@jedbrown.org).

§Department of Applied Mathematics, University of Waterloo, Waterloo, ON, Canada (yunhui.he@uwaterloo.ca).

using high-order finite element discretizations and investigate Jacobi and Chebyshev smoothing for two-grid schemes by way of LFAToolkit.jl [30], a new Julia package for LFA of high-order finite element methods. Several software packages have been developed for LFA of h-multigrid methods [26, 19, 33], however, to the best of our knowledge, no packages implement LFA of p-multigrid methods, especially with arbitrary PDEs. Although [32] discussed LFA of p-multigrid for the discontinuous Galerkin method, this formulation cannot be extended to the continuous Galerkin method. Our LFA formulation for p-multigrid with high-order finite element discretizations for the continuous Galerkin method can be generalized to reproduce previous work on h-multigrid for high-order finite elements [17] and can be extended to LFA for h-multigrid of finite difference discretizations that can be represented via finite element discretizations as well as LFA of block smoothers [23, 10]. Our work here could be represented in terms of the aLFA framework [19]; however we choose a representation that will be familiar for the high-order finite element and spectral elements community. We give details on the mathematical representation of LFA of grid-transfer operators for p-multigrid for the continuous Galerkin method, which seems absent in current literature. Moreover, our LFA framework for p-multigrid generalizes and unifies classical LFA for h-multigrid, which would be a rich area for research.

We investigate the performance of Jacobi and Chebyshev semi-iterative smoothers for p-multigrid with aggressive coarsening for the scalar Laplacian in one and two dimensions, and we validate our LFA of p-multigrid with the scalar Laplacian in three dimensions against numerical experiments. Our analysis demonstrates that the performance of p-multigrid with these two smoothers degrades as we coarsen more aggressively. Finally, we analyze the performance of p-multigrid with a Chebyshev smoother for three dimensional linear elasticity to demonstrate ability of our LFA framework to handle more complex PDEs.

This paper is organized as follows. In section 2 we outline the notation for LFA used in this paper. In section 3 we develop the notation for LFA of second-order PDEs with arbitrary order bases, dimension, and number of components used in LFAToolkit.jl, and we use this notation to develop LFA of p-multigrid and the Jacobi and Chebyshev smoothers. Section 4 contains numerical results investigating the performance of Jacobi and Chebyshev smoothing for p-multigrid on the one, two, and three dimensional scalar Laplacian and three dimensional linear elasticity. Finally, section 5 contains concluding remarks.

1.1. Reproducibility. The numerical results for LFA in this paper were generated using the Julia package LFAToolkit.jl [30]. This package is under active development and may be found on GitHub at jeremylt/LFAToolkit.jl. This repository contains Julia scripts and interactive Jupyter notebooks that can replicate the tables and plots in this paper.

Our numerical experiments demonstrating actual convergence rates of p-multigrid were conducted using libCEED [8] with PETSc [2] p-multigrid example found in the libCEED repository.

2. Definitions and Notation. Consider a scalar Toeplitz operator L_h on an infinite one dimensional uniform nodal grid G_h,

$$L_h \mathrel{\hat{=}} [s_\kappa]_h \, (\kappa \in V) \,;$$

(2.1)
$$L_h w_h \, (x) = \sum_{\kappa \in V} s_\kappa w_h \, (x + \kappa h) \,,$$

where $V \subset \mathbb{Z}$ is a finite index set, $s_\kappa \in \mathbb{R}$ are constant coefficients, and $w_h(x)$ is a l^2 function on G_h. In the terminology of stencils, s_κ are stencil weights that are nonzero on the neighborhood $\kappa \in V$.

Since L_h is Toeplitz, it can be diagonalized by the standard Fourier modes $\varphi(\theta, x) = e^{\iota\theta x/h}$, with $\iota^2 = -1$, which alias according to $\varphi(\theta + 2\pi, x) = \varphi(\theta, x)$ on all grid points $x \in h\mathbb{Z}$.

DEFINITION 2.1 (Symbol of L_h). *If for all grid functions $\varphi(\theta, x)$ we have*

$$(2.2) \qquad L_h \varphi(\theta, x) = \tilde{L}_h(\theta)\,\varphi(\theta, x),$$

then we define $\tilde{L}_h(\theta) = \sum_{\kappa \in V} s_\kappa e^{\iota\theta\kappa}$ as the symbol of L_h, where $\iota^2 = -1$.

This definition can be extended to a $q \times q$ linear system of operators by

$$(2.3) \qquad \mathbf{L}_h = \begin{bmatrix} L_h^{1,1} & \cdots & L_h^{1,q} \\ \vdots & \vdots & \vdots \\ L_h^{q,1} & \cdots & L_h^{q,q} \end{bmatrix},$$

where $L_h^{i,j}$, $i, j \in \{1, 2, \ldots, q\}$ are given by scalar Toeplitz operators describing how component j appears in the equation for component i. The components here may represent different fields and/or nodes of a finite element basis. The symbol of $\mathbf{L}_h$, denoted $\tilde{\mathbf{L}}_h$, is a $q \times q$ matrix-valued function of θ given by $\left(\tilde{\mathbf{L}}_h\right)_{i,j} = \tilde{L}_h^{i,j}(\theta)$. For a system of equations representing an error propagation operator in a relaxation scheme, the spectral radius of the symbol matrix determines how rapidly the scheme decreases error at a target frequency.

These definitions are readily extended to d dimensions by taking the neighborhood $V \subset \mathbb{Z}^d$ and letting $\theta \in [-\pi, \pi)^d$. For standard coarsening from a fine grid with a mesh size h to a coarse grid with a mesh size of $H = 2h$, low frequencies are given by $\theta \in T^{\text{low}} = [-\pi/2, \pi/2)^d$ and high frequencies are given by $\theta \in [-\pi, \pi)^d \setminus T^{\text{low}}$ or equivalently by periodicity, $\theta \in T^{\text{high}} = [-\pi/2, 3\pi/2)^d \setminus T^{\text{low}}$.

3. Local Fourier Analysis for P-Multigrid. We now develop the LFA formulation used in LFAToolkit.jl, first in one dimension and then in multiple dimensions, followed by LFA of polynomial smoothers and LFA of p-multigrid with high-order finite elements.

3.1. High-Order Finite Elements. We will use the representation of the weak form of linear second-order PDEs described in [7], which is given by

$$(3.1) \qquad \langle v, f(u) \rangle = \int_\Omega \begin{bmatrix} v^T & \nabla v^T \end{bmatrix} \begin{bmatrix} f_{0,0} & f_{0,1} \\ f_{1,0} & f_{1,1} \end{bmatrix} \begin{bmatrix} u \\ \nabla u \end{bmatrix} = \int_\Omega fv, \quad \forall v \in \mathcal{V}$$

for some suitable $\mathcal{V} \subseteq H_0^1(\Omega)$. In this equation, $f_{i,j}$ may come from a linear PDE or the linearization of a non-linear problem. Boundary terms have been omitted, as they are not present on the infinite uniform grid G_h.

This omission of boundary conditions corresponds to periodic boundary conditions on a finite domain; however, in practice LFA on the infinite uniform grid offers good predictions for finite domains with other boundary conditions [33].

Selecting a finite element basis, we can discretize this weak form and produce

$$(3.2) \qquad \mathbf{Au} = \mathbf{b}.$$

Using the algebraic representation of PDE operators given in [7], the PDE operator $\mathbf{A}$ is of the form

(3.3) $$\mathbf{A} = \mathbf{G}^T \mathbf{A}_e \mathbf{G}, \quad \text{with } \mathbf{A}_e = \mathbf{B}^T \mathbf{D} \mathbf{B},$$

where $\mathbf{G}$ represents the element assembly operator, $\mathbf{B}$ is a basis operator which computes the values and derivatives of the basis functions at the quadrature points, and $\mathbf{D}$ is a block diagonal operator which provides the pointwise application of the bilinear form on the quadrature points, to include quadrature weights and the change in coordinates between the physical and reference space.

Consider the specific case of a block Toeplitz operator representing an arbitrary second order scalar PDE in one dimension with Lagrange basis of polynomial degree p given in the form of (3.2) and (3.3). The basis $\mathbf{B}$ is defined on the mesh grid with points given by x_i, for $i \in \{1, 2, \ldots, p+1\}$ on one element. The nodes on the left and right boundaries of the element map to the same Fourier mode when localized to nodes unique to a single finite element, so we can compute the symbol matrix of size $p \times p$ as

(3.4) $$\tilde{\mathbf{A}} = \mathbf{Q}^T \left(\mathbf{A}_e \odot \left[e^{\iota(x_j - x_i)\theta/h} \right] \right) \mathbf{Q},$$

where $\odot$ represents pointwise multiplication of the elements, h is the length of the element, and $i, j \in \{1, 2, \ldots, p+1\}$. In the pointwise product $\mathbf{A}_e \odot \left[e^{\iota(x_j - x_i)\theta/h} \right]$, the (i, j) entry is given by $(\mathbf{A}_e)_{i,j} e^{\iota(x_j - x_i)\theta/h}$. $\mathbf{Q}$ is a $(p+1) \times p$ matrix that localizes Fourier modes to each finite element, given by

(3.5) $$\mathbf{Q} = \begin{bmatrix} \mathbf{I} \\ \mathbf{e}_0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 \\ 1 & 0 & \cdots & 0 \end{bmatrix}.$$

We refer to $\mathbf{Q}$ as the Fourier mode localization operator.

The computation of this symbol matrix extends to more complex PDE with multiple components and in higher dimensions. Multiple components are supported by extending the $p \times p$ system of Toeplitz operators in (3.4) to a $(n \cdot p) \times (n \cdot p)$ system of operators, where n is the number of components in the PDE. The localization operator for a multi-component PDE is given by $\mathbf{Q}_n = \mathbf{I}_n \otimes \mathbf{Q}$, where I_n is the identity matrix with size $n \times n$. In general, we omit the subscript indicating the number of components for the localization operator.

The infinite uniform grid G_h is extended into higher dimensions by taking the direct sum of the one dimensional grid. Tensor products are used to extend the one dimensional bases into higher dimensions. The basis evaluation operators in two dimensions are given by

(3.6)
$$\mathbf{B}_{\text{interp2d}} = \mathbf{B}_{\text{interp1d}} \otimes \mathbf{B}_{\text{interp1d}},$$
$$\mathbf{B}_{\text{grad2d}} = \begin{bmatrix} \mathbf{B}_{\text{grad1d}} \otimes \mathbf{B}_{\text{interp1d}} \\ \mathbf{B}_{\text{interp1d}} \otimes \mathbf{B}_{\text{grad1d}} \end{bmatrix},$$

where $\mathbf{B}_{interp*d}$ and $\mathbf{B}_{grad*d}$ are the basis interpolation and gradient operators, respectively.

In a similar fashion, the localization operator for Fourier modes in two dimensions is given by

$$(3.7) \qquad \mathbf{Q}_{2\mathrm{d}} = \mathbf{Q} \otimes \mathbf{Q}.$$

As with tensor product bases, an analogous computation can be done in higher dimensions. Again, we generally omit the subscript indicating the dimension of the Fourier mode localization operator.

DEFINITION 3.1. *The symbol matrix of a finite element operator for an arbitrary second order PDE with any basis order, dimension, and number of components is given by*

$$(3.8) \qquad \tilde{\mathbf{A}} = \mathbf{Q}^T \left(\mathbf{A}_e \odot \left[e^{\imath (\mathbf{x}_j - \mathbf{x}_i) \cdot \boldsymbol{\theta} / \mathbf{h}} \right] \right) \mathbf{Q},$$

where $\odot$ represents pointwise multiplication of the elements, $\mathbf{h}$ is the length of the element in each dimension, $\boldsymbol{\theta}$ is the target frequency in each dimension, $i, j \in \{1, 2, \ldots, n \cdot (p + 1)^d\}$, n is the number of components, p is the polynomial degree of the discretization, and d is the dimension of the finite element basis. $\mathbf{A}_e$ is the finite element operator for the element and $\mathbf{Q}$ is the localization of Fourier modes on an element.

Note that this LFA notation is applicable to any second-order PDE with a weak form that can be represented by (3.3). This representation is used in LFAToolkit.jl, where the users provide the finite element basis $\mathbf{B}$, the node to mode mapping $\mathbf{Q}$, and the pointwise representation of the weak form $\mathbf{D}$, and the software can provide the LFA of the PDE with various preconditioners. The nodes may be defined by points $\mathbf{x}_i$ or more generally, as dual basis functions that may be applied to the Fourier modes; however, we restrict ourselves to nodal bases in this paper.

If the pointwise representation of the weak form $\mathbf{D}$ has a tensor product structure, then the element operator $\mathbf{A}_e$ and therefore the symbol $\tilde{\mathbf{A}}$ will have a tensor product structure, as shown in [17]. We omit this optimization in this analysis and LFAToolkit.jl in favor of supporting second-order PDEs that do not have a tensor product decomposition.

3.2. Polynomial Smoothers. Multigrid methods require a fine grid smoother, typically applied both before and after the coarse grid correction is computed. In this section, we compute the symbol of the error propagation operator for two polynomial smoothers, Jacobi and Chebyshev.

The error propagation operator for a smoother is given by $\mathbf{S} = \mathbf{I} - \mathbf{M}^{-1}\mathbf{A}$, where $\mathbf{M}^{-1}$ is given by the particular smoother under investigation. The LFA-predicted convergence factor is given by the maximum spectral radius of the symbol across all frequencies

$$(3.9) \qquad \mu(\omega) = \max_{\boldsymbol{\theta} \in T} \rho\left(\tilde{\mathbf{S}}(\nu, \omega, \boldsymbol{\theta}) \right),$$

where $\rho\left(\tilde{\mathbf{S}}(\nu, \omega, \boldsymbol{\theta}) \right)$ denotes the spectral radius of the matrix symbol $\tilde{\mathbf{S}}$. Our goal is to select tuning parameters for the polynomial smoothers to minimize the LFA-predicted convergence factor of two-grid p-multigrid cycles with agressive coarsening.

3.2.1. Jacobi Smoother. With Jacobi smoothing, $\mathbf{M}^{-1}$ is given by a weighted inverse of the true operator diagonal, $\omega \operatorname{diag}(\mathbf{A})^{-1}$. Following the derivation from

(a) Spectrum of Jacobi for $p = 2$ (b) Spectrum of Jacobi for $p = 4$
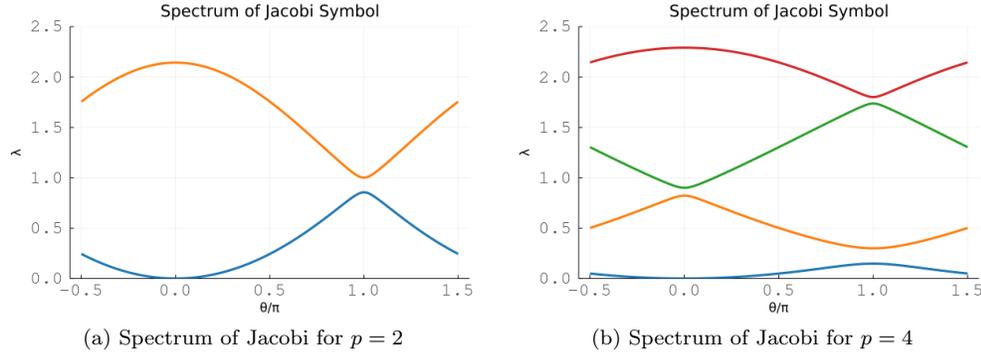
Fig. 3.1: Jacobi smoothing for high-order finite elements for the 1D Laplacian.

subsection 3.1, the symbol of the Jacobi error propagation operator therefore is given by

$$(3.10) \qquad \tilde{\mathbf{S}}(\omega, \boldsymbol{\theta}) = \mathbf{I} - \tilde{\mathbf{M}}^{-1}(\omega, \boldsymbol{\theta}) \tilde{\mathbf{A}}(\boldsymbol{\theta}) = \mathbf{I} - \omega \left(\mathbf{Q}^T \operatorname{diag}(\mathbf{A}_e) \mathbf{Q} \right)^{-1} \tilde{\mathbf{A}}(\boldsymbol{\theta}),$$

where $\tilde{\mathbf{M}}^{-1}$ has been simplified by the fact that $e^{i(\mathbf{x}_i - \mathbf{x}_i) \cdot \boldsymbol{\theta}/\mathbf{h}} = 1$.

If multiple pre or post-smoothing passes are used by our multigrid algorithm, we take the product of the symbol matrix with itself to represent repeated application of the Jacobi smoother.

DEFINITION 3.2. *The symbol of the error propagation operator for Jacobi smoothing is given by*

$$(3.11) \qquad \tilde{\mathbf{S}}(\nu, \omega, \boldsymbol{\theta}) = \left(\mathbf{I} - \omega \left(\mathbf{Q}^T \operatorname{diag}(\mathbf{A}_e) \mathbf{Q} \right)^{-1} \tilde{\mathbf{A}}(\boldsymbol{\theta}) \right)^{\nu},$$

where ν is the number of smoothing passes and ω is the smoothing parameter.

Using Definition 3.2, we plot the eigenvalues of $\tilde{\mathbf{M}}^{-1}\tilde{\mathbf{A}}(\boldsymbol{\theta})$ over the interval $\theta \in [-\pi/2, 3\pi/2)$ in Figures 3.1a and 3.1b for the one dimensional Laplacian with a 2nd and 4th degree H^1 Lagrange basis functions on Gauss-Lobatto points. The eigenvalues have been colored to identify the characteristic curves.

3.2.2. Chebyshev Smoother. Damped Jacobi with parameter ω is based on the degree 1 monic polynomial with root at $\alpha = 1/\omega > 0$. For $\nu = 1$ smoothing passes, this is the minimal degree 1 monic polynomial on the interval $[\alpha - c, \alpha + c]$ for any positive c. However, for $\nu \geq 2$, the Jacobi smoother is a degree ν monic polynomial that is not minimal on the interval (because it "concentrates" at the center α). Chebyshev smoothers are based on a stable recurrence for the polynomial of any degree that is minimial on the specified target interval of the positive real axis, which should contain the eigenvalues of all modes the smoother is responsible for. It is well known that polynomial smoothers allow more aggressive coarsening than Jacobi [5]. For further discussion of the error propagation properties of the Chebyshev semi-iterative method, see [16].

We use the Jacobi preconditioned operator, $(\operatorname{diag} \mathbf{A})^{-1} \mathbf{A}$, in this iteration instead of the finite element operator $\mathbf{A}$, similar to the discussion in [1].

The terms in the Chebyshev semi-iterative method can be modeled by the three term recurrence relation given by

$$(3.12) \qquad \mathbf{u}_k = -\left(\mathbf{r}_{k-1} + \alpha \mathbf{u}_{k-1} + \beta_{k-2} \mathbf{u}_{k-2}\right)/\gamma_{k-1},$$

with left-preconditioned residual $\mathbf{r}_{k-1} = (\operatorname{diag} \mathbf{A})^{-1} (\mathbf{b} - \mathbf{A}\mathbf{u}_{k-1})$. The target interval $[\lambda_{\min}, \lambda_{\max}] = [\alpha - c, \alpha + c]$ should be selected to be as narrow as possible while containing the eigenvalues of the preconditioned operator $(\operatorname{diag} \mathbf{A})^{-1} \mathbf{A}$ that the smoother is responsible for. (Other polynomial methods should be used for operators that are not symmetric positive definite or nearly so.) The parameters β and γ are given by the recurrence

$$(3.13) \qquad \begin{array}{cc} \beta_0 = -\frac{c^2}{2\alpha} & \gamma_0 = -\alpha \\ \beta_k = \frac{c}{2} \frac{T_k(\eta)}{T_{k+1}(\eta)} = \left(\frac{c}{2}\right)^2 \frac{1}{\gamma_k} & \gamma_k = \frac{c}{2} \frac{T_{k+1}(\eta)}{T_k(\eta)} = -(\alpha + \beta_{k-1}). \end{array}$$

In this equation, $T_i(\zeta) = 2\zeta T_{i-1}(\zeta) - T_{i-2}(\zeta)$ are the classical Chebyshev polynomials, which are evaluated at the point $\eta = -\alpha/c$. First order Chebyshev is equivalent to Jacobi smoothing with the classical choice of parameter value of $\omega = 1/\alpha$, so we should expect relatively poor smoothing for $k = 1$.

The residual in the Chebyshev semi-iterative method can therefore be modeled by the three term recurrence

$$(3.14) \qquad \mathbf{r}_k = \left((\operatorname{diag} \mathbf{A})^{-1} \mathbf{A}\mathbf{r}_{k-1} - \alpha \mathbf{r}_{k-1} - \beta_{k-2}\mathbf{r}_{k-2}\right)/\gamma_{k-1}.$$

Using the recurrence relation given in (3.14), we can define the error propagation of the kth order Chebyshev smoother in terms of the error in the first term:

$$(3.15) \qquad \begin{aligned} \mathbf{E}_0 &= \mathbf{I} \\ \mathbf{E}_1 &= \mathbf{I} - \frac{1}{\alpha} (\operatorname{diag} \mathbf{A})^{-1} \mathbf{A} \\ \mathbf{E}_k &= \left((\operatorname{diag} \mathbf{A})^{-1} \mathbf{A}\mathbf{E}_{k-1} - \alpha \mathbf{E}_{k-1} - \beta_{k-2}\mathbf{E}_{k-2}\right)/\gamma_{k-1} \end{aligned}$$

With this recursive definition of the error propagation operator, we can define the symbol for Chebyshev smoothing.

DEFINITION 3.3. *The symbol of the error propagation operator for a kth order Chebyshev smoother based on the Jacobi preconditioned operator is given by*

$$(3.16) \qquad \tilde{\mathbf{S}}(\nu, k, \boldsymbol{\theta}) = \left(\tilde{\mathbf{E}}_k\right)^\nu,$$

where ν is the number of smoothing passes and $\tilde{\mathbf{E}}_k(\boldsymbol{\theta})$ is given by the recursive definition

$$(3.17) \qquad \begin{aligned} \tilde{\mathbf{E}}_0(\boldsymbol{\theta}) &= \mathbf{I} \\ \tilde{\mathbf{E}}_1(\boldsymbol{\theta}) &= \mathbf{I} - \frac{1}{\alpha} \tilde{\mathbf{A}}_J \tilde{\mathbf{A}}(\boldsymbol{\theta}) \\ \tilde{\mathbf{E}}_k(\boldsymbol{\theta}) &= \left(\tilde{\mathbf{A}}_J \tilde{\mathbf{A}}(\boldsymbol{\theta}) \tilde{\mathbf{E}}_{k-1}(\boldsymbol{\theta}) - \alpha \tilde{\mathbf{E}}_{k-1}(\boldsymbol{\theta}) - \beta_{k-2} \tilde{\mathbf{E}}_{k-2}(\boldsymbol{\theta})\right)/\gamma_{k-1} \end{aligned}$$

with $\tilde{\mathbf{A}}_J = \left(\mathbf{Q}^T \operatorname{diag}(\mathbf{A}_e) \mathbf{Q}\right)^{-1}$ giving the symbol of the Jacobi operator.

Using Definition 3.3, we plot the eigenvalues of $\tilde{\mathbf{M}}^{-1}\tilde{\mathbf{A}}(\boldsymbol{\theta})$ in Figures 3.2a and 3.2b for the one dimensional Laplacian with a 2nd and 4th degree H^1 Lagrange basis functions on Gauss-Lobatto points. In both plots, we use the computation of the

(a) Spectrum of Chebyshev for $p = 2$

(b) Spectrum of Chebyshev for $p = 4$
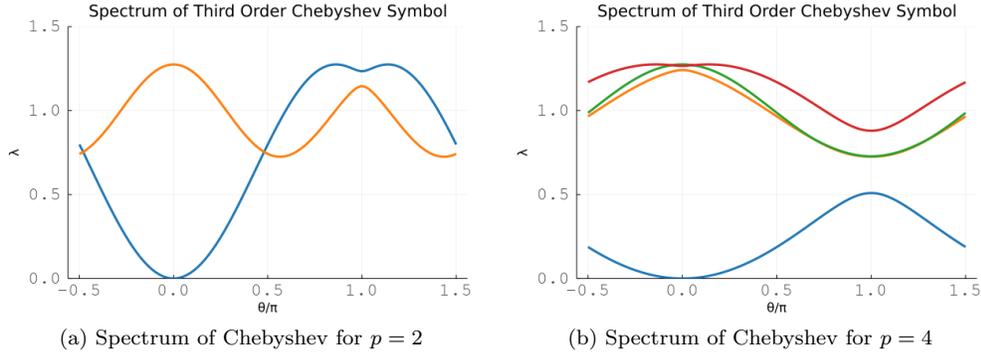
Fig. 3.2: Chebyshev smoothing for high-order finite elements for the 1D Laplacian.

extremal eigenvalues discussed below. These eigenvalues have been colored to identify the characteristic curves. We see improved smoothing when compared to the Jacobi smoothing in Figures 3.1a and 3.1b.

The Chebyshev semi-iterative method is known to be sensitive to the quality of the estimates of the extremal eigenvalues $\lambda_{\min}$ and $\lambda_{\max}$. Large eigenvalues are associated with higher frequencies for differential operators so $\lambda_{\min}$ for effective smoothing is not the minimal eigenvalue, which is hard to compute and goes to zero under grid refinement. PETSc [2] estimates the eigenvalues of the preconditioned operator via Krylov iterations, yielding an estimate of the maximal eigenvalue, $\hat{\lambda}_{\max}$, which is used to set the target interval according to $\lambda_{\min} = 0.1\hat{\lambda}_{\max}$ and $\lambda_{\max} = 1.1\hat{\lambda}_{\max}$. This default lower bound was chosen empirically for rapid coarsening with smoothed aggregation AMG and the upper bound incorporates a safety due to $\hat{\lambda}_{\max}$ generally being an underestimate of the true maximum eigenvalue.

In LFAToolkit.jl, we also want to target the upper part of the spectrum of the error; we estimate the spectral radius of the symbol of the Jacobi preconditioned operator by sampling the the frequencies at a small number of values to estimate $\hat{\lambda}_{\max}$. We then take $\lambda_{\min} = 0.1\hat{\lambda}_{\max}$ and $\lambda_{\max} = 1.0\hat{\lambda}_{\max}$.

3.3. P-Multigrid. With this representation of the symbol of high-order PDE operators, we can derive the symbol of the p-multigrid two-grid error propagation operator.

The two-grid multigrid error propagation operator is given by

$$(3.18) \qquad \mathbf{E}_{2\mathrm{mg}} = \mathbf{S}_f \left(\mathbf{I} - \mathbf{P}_{\mathrm{ctof}} \mathbf{A}_c^{-1} \mathbf{R}_{\mathrm{ftoc}} \mathbf{A}_f \right) \mathbf{S}_f,$$

where $\mathbf{A}_f$ represents the action of the PDE operator on the fine grid, $\mathbf{S}_f$ represents the fine grid smoother error propagation operator, and $\mathbf{A}_c^{-1}$ represents the coarse grid solve, which may be another multigrid cycle. $\mathbf{P}_{\mathrm{ctof}}$ and $\mathbf{R}_{\mathrm{ftoc}}$ represent the grid prolongation and restriction operators, respectively.

Some multigrid implementations allow the number of pre and post smoothing passes to be set independently. This derivation of LFA for p-multigrid allows independently setting these parameters; however, we omit this option in the notation for simplicity.

This error propagation operator can represent both h-multigrid and p-multigrid, depending upon the grid transfer operators and coarse grid representation chosen, but

we focus on p-multigrid grid transfer operators in this paper.

3.3.1. Grid Transfer Operator. In p-multigrid, grid transfer operators can be represented elementwise and can thus be easily represented in the form of (3.3).

The prolongation operator from the coarse to the fine grid interpolates low-order basis functions at the nodes for the high-order basis functions. Figure 3.3 shows the evaluation of second order basis function on the Gauss-Lobatto nodes for a fourth order basis.
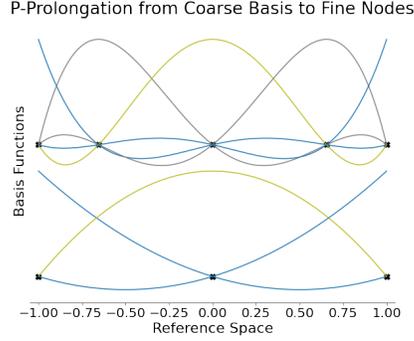
Fig. 3.3: P-Prolongation from Coarse Basis to Fine Basis Points in 1D.

The prolongation operator from the coarse to the fine grid can be represented by

$$\mathbf{P}_{\text{ctof}} = \mathbf{P}_f^T \mathbf{P}_e \mathbf{P}_c$$

(3.19)

$$\mathbf{P}_e = \mathbf{ID}_{\text{scale}} \mathbf{B}_{\text{ctof}}$$

where $\mathbf{B}_{ctof}$ is an interpolation operator from the coarse grid basis to the fine grid basis, $\mathbf{P}_f$ is the fine grid element assembly operator, $\mathbf{P}_c$ is the coarse grid element assembly operator, and $\mathbf{D}_{\text{scale}}$ is a scaling operator to account for node multiplicity across element interfaces. Restriction from the fine grid to the coarse grid is given by the transpose, $\mathbf{R}_{\text{ftoc}} = \mathbf{P}_{\text{ctof}}^T$.

Following the derivation from subsection 3.1, we can derive the symbols of $\mathbf{P}_{\text{ctof}}$ and $\mathbf{R}_{\text{ftoc}}$.

DEFINITION 3.4. *The symbol of the p-prolongation is given by*

(3.20) $$\tilde{\mathbf{P}}_{ctof}(\boldsymbol{\theta}) = \mathbf{Q}_f^T \left(\mathbf{P}_e \odot \left[e^{\imath(\mathbf{x}_{j,c} - \mathbf{x}_{i,f}) \cdot \boldsymbol{\theta}/\mathbf{h}} \right] \right) \mathbf{Q}_c,$$

where $i \in \{1, 2, \ldots, n \cdot (p_{fine} + 1)^d\}$, $\mathbf{h}$ is the length of the element in each dimension, $j \in \{1, 2, \ldots, n \cdot (p_{coarse} + 1)^d\}$, n is the number of components, p_{fine} and p_{coarse} are the polynomial degrees of the fine and coarse grid discretizations, respectively, and d is the dimension of the finite element basis. The matrices $\mathbf{Q}_f$ and $\mathbf{Q}_c$ are the localization mappings for the fine and coarse grid, respectively, and the element p-prolongation operator is given by $\mathbf{P}_e = \mathbf{D}_{scale} \mathbf{B}_{ctof}$. The nodes $\mathbf{x}_{j,c}$ and are $\mathbf{x}_{i,f}$ are on the coarse and fine grids, respectively.

DEFINITION 3.5. *The symbol of p-restriction is given by the expression*

(3.21) $$\tilde{\mathbf{R}}_{ftoc}(\boldsymbol{\theta}) = \mathbf{Q}_c^T \left(\mathbf{R}_e \odot \left[e^{\imath(\mathbf{x}_{j,f} - \mathbf{x}_{i,c}) \cdot \boldsymbol{\theta}/\mathbf{h}} \right] \right) \mathbf{Q}_f,$$

where $i \in \{1, 2, \ldots, n \cdot (p_{coarse} + 1)^d\}$, $\mathbf{h}$ is the length of the element in each dimension, $j \in \{1, 2, \ldots, n \cdot (p_{fine} + 1)^d\}$, n is the number of components, p_{fine} and p_{coarse} are the polynomial degrees of the fine and coarse grid discretizations, respectively, and d is the dimension of the finite element basis. The matrices $\mathbf{Q}_f$ and $\mathbf{Q}_c$ are the localization mappings for the fine and coarse grid, respectively, and the element p-restriction operator is given by $\mathbf{R}_e = \mathbf{P}_e^T = \mathbf{B}_{ctof}^T \mathbf{D}_{scale}$. The nodes $\mathbf{x}_{j,f}$ and are $\mathbf{x}_{i,c}$ are on the fine and coarse grids, respectively.

3.3.2. Multigrid Error Propagation Symbol. Based on the previous discussion, we can combine the expressions in p-multigrid to give the symbol of the error propagation operator.

DEFINITION 3.6. *The symbol for the p-multigrid error propagation operator is given by*
(3.22)
$$\tilde{\mathbf{E}}_{2mg}(\nu, \omega, \boldsymbol{\theta}) = \tilde{\mathbf{S}}_f(\nu, \omega, \boldsymbol{\theta}) \left[\mathbf{I} - \tilde{\mathbf{P}}_{ctof}(\boldsymbol{\theta}) \tilde{\mathbf{A}}_c^{-1}(\boldsymbol{\theta}) \tilde{\mathbf{R}}_{ftoc}(\boldsymbol{\theta}) \tilde{\mathbf{A}}_f(\boldsymbol{\theta}) \right] \tilde{\mathbf{S}}_f(\nu, \omega, \boldsymbol{\theta}),$$

where $\tilde{\mathbf{A}}$ is the symbol of the fine grid operator, $\tilde{\mathbf{S}}_f(\nu, \omega, \boldsymbol{\theta})$ is the symbol of the smoother, and $\tilde{\mathbf{A}}_c^{-1}(\boldsymbol{\theta})$ is the symbol of the coarse grid solve. $\tilde{\mathbf{P}}_{ctof}(\boldsymbol{\theta})$ and $\tilde{\mathbf{R}}_{ftoc}(\boldsymbol{\theta})$ represent the symbol of the grid prolongation and restriction operators, respectively.

Note again that this derivation is applicable for any PDE with a weak form that can be represented by (3.3). This expression can be extended to represent multi-level schemes by recursively applying $\tilde{\mathbf{E}}_{2mg}$ until the coarsest grid is reached.

3.4. Extension to H-Multigrid. By using macro-elements, [22, 10], our LFA of p-multigrid can be extended to LFA of h-multigrid. For 1D analysis, a macro-element is single element comprising of a pair of sub-elements with separate quadrature spaces, which is equivalent to partially assembling the finite element operator on two element subdomains. Prolongation between a coarse grid element and a fine grid macro-element is given by evaluating the coarse grid basis functions on the fine grid macro-element nodes with the multiplicity correction, as shown in subsection 3.3.1. LFA in higher dimensions is given by tensor-products of 1D basis operations, as before.
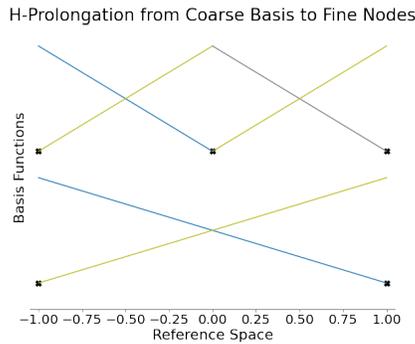
Fig. 3.4: H-Prolongation from Coarse Basis to Fine Basis Points in 1D.

In Figure 3.4, we see an example of interpolation from a coarse grid basis to a fine grid macro-element basis. The linear shape functions are evaluated at the nodes of

the fine grid macro-element, which is a pair of linear sub-elements. On the left linear micro-element, there are two basis functions which are zero over the domain of the right sub-element, and the reverse is also true. The element level operator, $\mathbf{A}_e$, on the fine grid macro-element assembles the element level operator for each sub-element into the action of the PDE operator on the full macro-element.

For example, the fine and coarse grid element operators for the one dimensional scalar Laplacian on quadratic elements are given by

$$(3.23) \qquad \mathbf{A}_{e,f} = \frac{2}{3h} \begin{bmatrix} 7 & -8 & 1 & 0 & 0 \\ -8 & 16 & -8 & 0 & 0 \\ 1 & -8 & 14 & -8 & 1 \\ 0 & 0 & -8 & 16 & -8 \\ 0 & 0 & 1 & -8 & 7 \end{bmatrix},$$

where h is the length of the macro-element, and

$$(3.24) \qquad \mathbf{A}_{e,c} = \frac{1}{3h} \begin{bmatrix} 7 & -8 & 1 \\ -8 & 16 & -8 \\ 1 & -8 & 7 \end{bmatrix}.$$

The corresponding symbols are computed as in Definition 3.1 above.

Following the derivation from subsection 3.3.1, we can derive the symbols of grid transfer operators for h-multigrid.

DEFINITION 3.7. *The symbol of the h-prolongation operator is given by*

$$(3.25) \qquad \tilde{\mathbf{P}}_{ctof}(\boldsymbol{\theta}) = \mathbf{Q}_f^T \left(\mathbf{P}_e \odot \left[e^{\imath(\mathbf{x}_{j,c} - \mathbf{x}_{i,f}) \cdot \boldsymbol{\theta}/\mathbf{h}} \right] \right) \mathbf{Q}_c,$$

where $i \in \left\{ 1, 2, \ldots, n\,(mp+1)^d \right\}$, $j \in \left\{ 1, 2, \ldots, n\,(p+1)^d \right\}$, $\mathbf{h}$ is the length of the macro-element, d is the dimension of the finite element bases, n is the number of components, m is the coarsening factor or the number of micro-elements in each fine grid macro-element, p is the polynomial degree of the discretization, and d is the dimension of the finite element basis. The matrices $\mathbf{Q}_f$ and $\mathbf{Q}_c$ are the localization mappings for the fine and coarse grid, respectively, and the macro-element h-prolongation operator is given by $\mathbf{P}_e = \mathbf{ID}_{scale}\mathbf{B}_{ctof}$.

DEFINITION 3.8. *The symbol of h-restriction operator is given by*

$$(3.26) \qquad \tilde{\mathbf{R}}_{ftoc}(\boldsymbol{\theta}) = \mathbf{Q}_c^T \left(\mathbf{R}_e \odot \left[e^{\imath(\mathbf{x}_{j,f} - \mathbf{x}_{i,c}) \cdot \boldsymbol{\theta}/\mathbf{h}} \right] \right) \mathbf{Q}_f,$$

where $i \in \left\{ 1, 2, \ldots, n\,(p+1)^d \right\}$, $j \in \left\{ 1, 2, \ldots, n\,(mp+1)^d \right\}$, $\mathbf{h}$ is the length of the macro-element, d is the dimension of the finite element bases, n is the number of components, m is the coarsening factor or the number of micro-elements in each fine grid macro-element, p is the polynomial degree of the discretization, and d is the dimension of the finite element basis. The matrices $\mathbf{Q}_f$ and $\mathbf{Q}_c$ are the localization mappings for the fine and coarse grid, respectively, and the macro-element h-restriction operator is given by $\mathbf{R}_e = \mathbf{P}_e^T = \mathbf{B}_{ctof}^T \mathbf{D}_{scale} \mathbf{I}$.

By representing the fine grid with macro-elements and the prolongation operator with this interpolation, this LFA of p-multigrid exactly reproduces the results of He and MacLachlan [17] for LFA of high-order h-multigrid.

Our LFA of h-multigrid presented here is based on a specific basis of the Fourier space used in [22] rather than the commonly used Fourier modes, see [31, 33], where the symbol of each component in multigrid methods is formed based on different harmonic frequencies. The symbols of grid-transfer operators described in Definitions 3.7 and 3.8 give a general way for multigrid coarsening with factor m, and this framework is simpler, especially for the high-order discretizations described in [17]. Also, our LFA for h-multigrid is not restricted to finite element discretizations with uniformly spaced nodes.

On uniform rectangular meshes, linear finite elements produce the same discetized operator as finite differencing. The nine-point stencil for the Laplace operator in two dimensions is given by

$$(3.27) \qquad \frac{1}{3} \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

with a corresponding local Fourier analysis symbol given by

$$(3.28) \qquad \tilde{A}\left(\theta_1, \theta_2\right) = \frac{2}{3}\left(4 - \cos\left(\theta_1\right) - \cos\left(\theta_2\right) - 2\cos\left(\theta_1\right)\cos\left(\theta_2\right)\right).$$

The assembled matrix for a single linear element in two dimensions is given by

$$(3.29) \qquad \mathbf{A}_e = \frac{1}{3} \begin{bmatrix} 2 & -1/2 & -1/2 & -1 \\ -1/2 & 2 & -1 & -1/2 \\ -1/2 & -1 & 2 & -1/2 \\ -1 & -1/2 & -1/2 & 2 \end{bmatrix},$$

with a corresponding local Fourier analysis symbol given by

$$(3.30) \qquad \begin{aligned} \tilde{A}\left(\theta_1, \theta_2\right) &= \mathbf{Q}^T \left(\mathbf{A}_e \odot \left[e^{i(\mathbf{x}_j - \mathbf{x}_i)\cdot\boldsymbol{\theta}}\right]\right) \mathbf{Q} \\ &= \frac{2}{3}\left(4 - \cos\left(\theta_1\right) - \cos\left(\theta_2\right) - 2\cos\left(\theta_1\right)\cos\left(\theta_2\right)\right), \end{aligned}$$

where $\mathbf{Q} = \begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix}^T$.

We can use the LFA of multigrid given by Definition 3.6 with the prolongation and restriction symbols given by Definitions 3.7 and 3.8 to reproduce LFA of h-multigrid methods for finite differencing where the stencil can be represented by a finite element discretization. This LFA of arbitrary second-order PDEs with high-order finite element discretizations agrees with previous work on LFA of PDE operators derived with finite differencing with analogous stencils.

As mentioned before, our focus of this work is p-multigrid, so we will not expand the discussion of h-multigrid further method here. Applying this LFA framework to h-multigrid or hp-multigrid methods are topics for future research.

4. Numerical Results. In this section, we present numerical results for this analysis for the scalar Laplacian in one and two dimensions with H^1 Lagrange bases on Gauss-Lobatto points with Gauss-Legendre quadrature. Next, we validate these results with numerical experiments for the scalar Laplacian in three dimensions. Lastly, we consider linear elasticity in three dimensions.

(a) Convergence for $p = 4$ to $p = 2$, $\nu = 1$ (b) Convergence for $p = 4$ to $p = 1$, $\nu = 1$

(c) Convergence for $p = 4$ to $p = 2$, $\nu = 2$ (d) Convergence for $p = 4$ to $p = 1$, $\nu = 2$
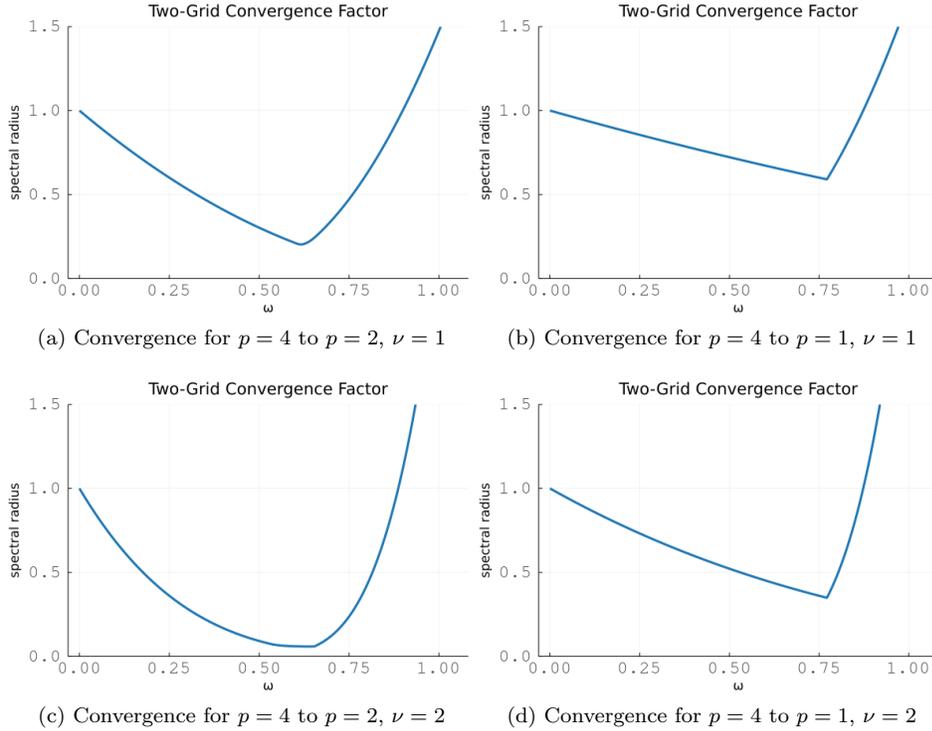
Fig. 4.1: Two-grid analysis for Jacobi smoothing for high-order finite elements for the 1D Laplacian.

4.1. Scalar Laplacian - 1D Convergence Factors.

4.1.1. Jacobi Smoothing. In Figure 4.1a and Figure 4.1b, we plot the two-grid convergence factor for p-multigrid with a single iteration of Jacobi pre and post-smoothing for the one dimensional Laplacian as a function of the Jacobi smoothing parameter ω, and in Figure 4.1c and Figure 4.1d we plot the two-grid convergence factor for p-multigrid with two iterations of Jacobi pre and post-smoothing for the one dimensional Laplacian as a function of the Jacobi smoothing parameter ω. On the left we show conservative coarsening from quartic to quadratic elements and on the right we show more aggressive coarsening from quartic to linear elements. As expected, the two-grid convergence factor decreases as we coarsen more rapidly. Also, the effect of underestimating the optimal Jacobi smoothing parameter, ω, is less pronounced than the effect of overestimating the smoothing parameter, especially with a higher number of pre and post-smooths.

In contrast to the previous work on h-multigrid for high-order finite elements, [17], poorly chosen values of $\omega < 1.0$ can result in a spectral radius of the p-multigrid error propagation symbol that is greater than 1, indicating that application of p-multigrid with Jacobi smoothing at these parameter values will result in increased error.

The results in Table 4.1 provide the LFA convergence factor and optimal values of ω for two-grid high-order p-multigrid for a variety of basis polynomial degrees and coarsening factors.

p_{fine} to p_{coarse}	$\nu = 1$		$\nu = 2$		$\nu = 3$	
	$\rho_{\min}$	ω_{opt}	$\rho_{\min}$	ω_{opt}	$\rho_{\min}$	ω_{opt}
$p = 2$ to $p = 1$	0.137	0.63	0.060	0.69	0.041	0.72
$p = 4$ to $p = 2$	0.204	0.62	0.059	0.64	0.045	0.70
$p = 4$ to $p = 1$	0.591	0.77	0.350	0.77	0.207	0.77
$p = 8$ to $p = 4$	0.250	0.60	0.068	0.60	0.033	0.63
$p = 8$ to $p = 2$	0.668	0.73	0.446	0.73	0.298	0.73
$p = 8$ to $p = 1$	0.874	0.78	0.764	0.78	0.668	0.78
$p = 16$ to $p = 8$	0.300	0.57	0.090	0.57	0.035	0.58
$p = 16$ to $p = 4$	0.719	0.69	0.517	0.69	0.371	0.69
$p = 16$ to $p = 2$	0.906	0.73	0.820	0.73	0.743	0.73
$p = 16$ to $p = 1$	0.968	0.74	0.936	0.74	0.906	0.74

Table 4.1: Two-grid convergence factor and optimal Jacobi parameter for the 1D Laplacian.

(a) Convergence for $p = 4$ to $p = 2$, $\nu = 1$ (b) Convergence for $p = 4$ to $p = 1$, $\nu = 1$
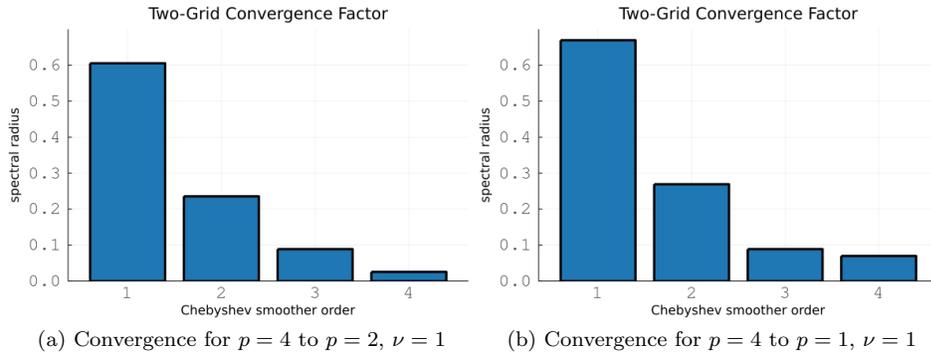
Fig. 4.2: Two-grid analysis for Chebyshev smoothing for high-order finite elements for the 1D Laplacian.

Traditional estimates of the optimal smoothing parameter based upon extremal eigenvalues of the preconditioned operator are incompatible with this LFA framework. Optimal parameter estimation is an open question for high-order p-multigrid, but optimization techniques, such as those discussed in [11], can be used to tune these parameters, especially for more complex PDEs.

4.1.2. Chebyshev Smoothing. In Figures 4.2a and 4.2b we plot the two-grid convergence factor for p-multigrid with Chebyshev pre and post-smoothing for the one dimensional Laplacian as a function of the Chebyshev order, k. On the left we show conservative coarsening from quartic to quadratic elements and on the right we show more aggressive coarsening from quartic to linear elements. As expected, the two-grid convergence factor degrades as we coarsen more rapidly.

The results in Table 4.2 provide the LFA-predicted convergence factor and optimal values of k for two-grid high-order p-multigrid for a variety of coarsening rates and

p_{fine} to p_{coarse}	$k = 1$	$k = 2$	$k = 3$	$k = 4$
$p = 2$ to $p = 1$	0.545	0.220	0.063	0.017
$p = 4$ to $p = 2$	0.576	0.222	0.089	0.025
$p = 4$ to $p = 1$	0.623	0.269	0.089	0.070
$p = 8$ to $p = 4$	0.638	0.244	0.074	0.022
$p = 8$ to $p = 2$	0.657	0.260	0.097	0.059
$p = 8$ to $p = 1$	0.881	0.674	0.510	0.393
$p = 16$ to $p = 8$	0.664	0.253	0.075	0.022
$p = 16$ to $p = 4$	0.714	0.328	0.135	0.059
$p = 16$ to $p = 2$	0.907	0.741	0.602	0.496
$p = 16$ to $p = 1$	0.970	0.912	0.857	0.809

Table 4.2: Two-grid convergence factor with Chebyshev smoothing for 1D Laplacian.

orders of Chebyshev smoother. From this table, we can see that the effectiveness of higher order Chebyshev smoothers degrades as we coarsen more aggressively, but Chebyshev smoothing still provides better two-grid convergence than multiple pre and post-smoothing Jacobi iterations.

$\lambda_{\min} = 0.2\hat{\lambda}_{\max}$				
p_{fine} to p_{coarse}	$k = 1$	$k = 2$	$k = 3$	$k = 4$
$p = 4$ to $p = 2$	0.410	0.093	0.043	0.024
$p = 4$ to $p = 1$	0.611	0.250	0.106	0.071
$p = 8$ to $p = 4$	0.435	0.081	0.016	0.007
$p = 8$ to $p = 1$	0.891	0.739	0.623	0.529
$p = 16$ to $p = 8$	0.443	0.081	0.015	0.006
$p = 16$ to $p = 1$	0.973	0.931	0.894	0.861
$\lambda_{\min} = 0.3\hat{\lambda}_{\max}$				
p_{fine} to p_{coarse}	$k = 1$	$k = 2$	$k = 3$	$k = 4$
$p = 4$ to $p = 2$	0.279	0.070	0.042	0.031
$p = 4$ to $p = 1$	0.638	0.332	0.184	0.104
$p = 8$ to $p = 4$	0.289	0.050	0.023	0.012
$p = 8$ to $p = 1$	0.899	0.777	0.682	0.599
$p = 16$ to $p = 8$	0.294	0.055	0.020	0.010
$p = 16$ to $p = 1$	0.975	0.942	0.913	0.885

Table 4.3: Two-grid convergence factor with Chebyshev smoothing for 1D Laplacian with modified lower eigenvalue bound.

The results in Table 4.3 provide the LFA-predicted convergence factor for two-grid high-order p-multigrid for a variety of coarsening rates and orders of Chebyshev smoother with different scaling factors for the minimum eigenvalue estimate used

(a) Convergence for $p = 4$ to $p = 1$, $\nu = 1$ (b) Convergence for $p = 4$ to $p = 1$, $\nu = 2$
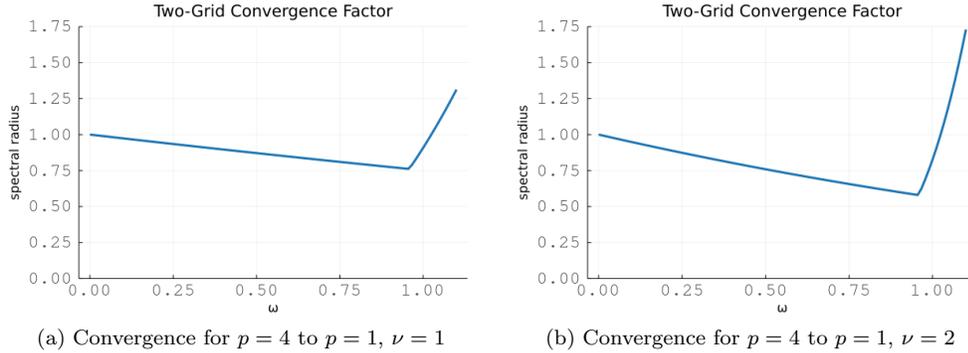
Fig. 4.3: Convergence for high-order finite elements for the 2D Laplacian.

in the Chebyshev iterations. Increasing the lower eigenvalue estimate results in the Chebyshev method better targeting high frequency error modes, which results in improved two-grid convergence when halving the polynomial degree of the basis functions. However, increasing the lower eigenvalue estimate results in worse two-grid convergence for aggressive coarsening directly to linear elements.

4.2. Scalar Laplacian - 2D Convergence Factors.

4.2.1. Jacobi Smoothing. In Figure 4.3a and Figure 4.3b we show the two-grid convergence factor for p-multigrid with one and two iterations of Jacobi smoothing, respectively, for the two dimensional Laplacian as a function of the Jacobi smoothing parameter ω.

p_{fine} to p_{coarse}	$\nu = 1$		$\nu = 2$		$\nu = 3$	
	$\rho_{\min}$	ω_{opt}	$\rho_{\min}$	ω_{opt}	$\rho_{\min}$	ω_{opt}
$p = 2$ to $p = 1$	0.230	0.95	0.091	0.99	0.061	1.03
$p = 4$ to $p = 2$	0.388	0.82	0.151	0.82	0.078	0.83
$p = 4$ to $p = 1$	0.763	0.95	0.582	0.95	0.444	0.95
$p = 8$ to $p = 4$	0.646	0.79	0.418	0.79	0.272	0.79
$p = 8$ to $p = 2$	0.858	0.84	0.737	0.84	0.633	0.84
$p = 8$ to $p = 1$	0.952	0.87	0.907	0.87	0.864	0.87

Table 4.4: Two-grid convergence factor and optimal Jacobi parameter for 2D Laplacian.

The results in Table 4.4 provide the LFA-predicted convergence factor and optimal values of ω for two-grid high-order p-multigrid for a variety of basis polynomial degrees and coarsening factors.

4.2.2. Chebyshev Smoothing. The results in Table 4.5 provide the LFA-predicted convergence factor for two-grid high-order p-multigrid for a variety of coarsening rates and orders of Chebyshev smoother. The two-grid convergence factor still degrades and the effectiveness of higher order Chebyshev smoothers is again reduced

p_{fine} to p_{coarse}	$k = 1$	$k = 2$	$k = 3$	$k = 4$
$p = 2$ to $p = 1$	0.621	0.252	0.075	0.039
$p = 4$ to $p = 2$	0.607	0.281	0.085	0.047
$p = 4$ to $p = 1$	0.768	0.424	0.219	0.127
$p = 8$ to $p = 4$	0.669	0.278	0.110	0.055
$p = 8$ to $p = 2$	0.864	0.633	0.456	0.336
$p = 8$ to $p = 1$	0.956	0.873	0.795	0.730
$p = 16$ to $p = 8$	0.855	0.613	0.435	0.319
$p = 16$ to $p = 4$	0.938	0.822	0.719	0.634
$p = 16$ to $p = 2$	0.976	0.928	0.882	0.842
$p = 16$ to $p = 1$	0.992	0.975	0.959	0.944

Table 4.5: Two-grid convergence factor with Chebyshev smoothing for 2D Laplacian.

$\lambda_{\min} = 0.2\hat{\lambda}_{\max}$				
p_{fine} to p_{coarse}	$k = 1$	$k = 2$	$k = 3$	$k = 4$
$p = 4$ to $p = 2$	0.450	0.137	0.067	0.050
$p = 4$ to $p = 1$	0.786	0.525	0.362	0.255
$p = 8$ to $p = 4$	0.668	0.330	0.172	0.106
$p = 8$ to $p = 1$	0.960	0.899	0.848	0.801
$\lambda_{\min} = 0.3\hat{\lambda}_{\max}$				
p_{fine} to p_{coarse}	$k = 1$	$k = 2$	$k = 3$	$k = 4$
$p = 4$ to $p = 2$	0.407	0.106	0.073	0.059
$p = 4$ to $p = 1$	0.803	0.590	0.447	0.341
$p = 8$ to $p = 4$	0.691	0.409	0.256	0.164
$p = 8$ to $p = 1$	0.963	0.915	0.874	0.835

Table 4.6: Two-grid convergence factor with Chebyshev smoothing for 2D Laplacian with modified lower eigenvalue bound.

The results in Table 4.6 provide the LFA-predicted convergence factor and optimal values of k for two-grid high-order p-multigrid for a variety of coarsening rates and orders of Chebyshev smoother with different scaling factors for the minimum eigenvalue estimate used in the Chebyshev iterations. As in one dimension, increasing the lower eigenvalue estimate results in better two-grid convergence when conventional coarsening by halving the polynomial degree of the basis functions but results in worse two-grid convergence with aggressive coarsening.

4.3. Scalar Laplacian - 3D Convergence Factors. In this section, we compare the LFA two-grid convergence factors to numerical results. Our numerical experiments were conducted using the libCEED [8] with PETSc [2] multigrid example found

in the libCEED repository. PETSc provides the mesh management, linear solvers, and multigrid preconditioner while libCEED provides the matrix-free operator evaluation.

We recover the manufactured solution given by

$$(4.1) \qquad f(x, y, z) = xyz \sin(\pi x) \sin(\pi(1.23 + 0.5y)) \sin(\pi(2.34 + 0.25z))$$

on the domain $[-3, 3]^3$ with Dirichlet boundary conditions for finite element discretizations with varying orders on approximately 8 million degrees of freedom for a variety of test cases.

Although LFA is defined on infinite grids, which most naturally translate to periodic problems, LFA-predicted convergence factors are also accurate for appropriate problems with other boundary conditions on finite grids after appropriate reformulation of the analysis [27]. Since it is not feasable to tune multigrid individually for each set of boundary conditions in engineering domains, we explore the sharpness and robustness of infinite-grid LFA by comparing directly to the test problem with Dirichlet boundary conditions.

4.3.1. Jacobi Smoothing. Since the Chebyshev smoothing is based upon the Jacobi preconditioned operator, it is important to validate the LFA of the Jacobi smoothing before considering Chebyshev smoothing. We use simple Jacobi smoothing with a weight of $\omega = 1.0$ to validate the LFA.

p_{fine} to p_{coarse}	LFA	libCEED
$p = 2$ to $p = 1$	0.312	0.301
$p = 4$ to $p = 2$	1.436	1.402
$p = 4$ to $p = 1$	1.436	1.401
$p = 8$ to $p = 4$	1.989	1.885
$p = 8$ to $p = 2$	1.989	1.874
$p = 8$ to $p = 1$	1.989	1.875

Table 4.7: LFA and experimental two-grid convergence factors with Jacobi smoothing for 3D Laplacian with $\omega = 1.0$.

The results in Table 4.7 provide the LFA and experimental convergence factors for the test problem. As expected, the high-order fine grid problems diverge with a smoothing factor of $\omega = 1.0$; however, the LFA provides reasonable upper bounds on the true convergence factor seen in the experimental results.

4.3.2. Chebyshev Smoothing. We used the LFA estimates of the maximal eigenvalue to set the extremal eigenvalues used the Chebyshev iteration in PETSc, using $\lambda_{\min} = 0.1\hat{\lambda}_{\max}$ and $\lambda_{\max} = 1.0\hat{\lambda}_{\max}$, where $\hat{\lambda}_{\max}$ is the estimated maximal eigenvalue of the symbol of the Jacobi preconditioned operator.

Table 4.8 shows that the LFA predictions agree well with the experimental convergence factors. As with the one and two dimensional results, rapid coarsening of the polynomial degree of the bases decreases the effectiveness of higher order Chebyshev smoothing.

4.4. Linear Elasticity - 3D Convergence Factors. To demonstrate the suitability of this LFA formulation for more complex PDEs, we consider linear elasticity

p_{fine} to p_{coarse}	$k = 2$		$k = 3$		$k = 4$	
	LFA	libCEED	LFA	libCEED	LFA	libCEED
$p = 2$ to $p = 1$	0.253	0.222	0.076	0.058	0.041	0.033
$p = 4$ to $p = 2$	0.277	0.251	0.111	0.097	0.062	0.050
$p = 4$ to $p = 1$	0.601	0.587	0.416	0.398	0.295	0.276
$p = 8$ to $p = 4$	0.398	0.391	0.197	0.195	0.121	0.110
$p = 8$ to $p = 2$	0.748	0.743	0.611	0.603	0.506	0.469
$p = 8$ to $p = 1$	0.920	0.914	0.871	0.861	0.827	0.814

Table 4.8: LFA and experimental two-grid convergence factors with Chebyshev smoothing for 3D Laplacian.

in three dimensions. The strong form of the static balance of linear momentum at small strain for the three dimensional linear elasticity problem is given by [18] as

$$(4.2) \qquad \nabla \cdot \boldsymbol{\sigma} + \boldsymbol{g} = \boldsymbol{0},$$

where $\boldsymbol{\sigma}$ is the stress function and $\boldsymbol{g}$ is the forcing function. This strong form has the corresponding weak form

$$(4.3) \qquad \int_\Omega \nabla \mathbf{v} : \boldsymbol{\sigma} \, dV - \int_{\partial\Omega} \mathbf{v} \cdot (\boldsymbol{\sigma} \cdot \hat{\mathbf{n}}) \, dS - \int_\Omega \mathbf{v} \cdot \mathbf{g} \, dV = 0, \forall \mathbf{v} \in \mathcal{V}$$

for some displacement $\mathbf{u} \in \mathcal{V} \subset H^1(\Omega)$, where : denotes contraction over both components and dimensions.

Linear elasticity constitutive modeling is based upon the Lamé parameters,

$$(4.4) \qquad \lambda = \frac{E\nu}{(1+\nu)(1-2\nu)}, \quad \mu = \frac{E}{2(1+\nu)},$$

where E is the Young's modulus and ν is the Poisson's ratio for the materiel.

In the linear elasticity constitutive model, the symmetric strain tensor is given by

$$(4.5) \qquad \boldsymbol{\epsilon} = \frac{1}{2}\left(\nabla \mathbf{u} + \nabla \mathbf{u}^T\right),$$

and the linear elasticity constitutive law is given by $\boldsymbol{\sigma} = \mathsf{C} : \boldsymbol{\epsilon}$ where

$$(4.6) \qquad \mathsf{C} = \begin{bmatrix} \lambda + 2\mu & \lambda & \lambda & & & \\ \lambda & \lambda + 2\mu & \lambda & & & \\ \lambda & \lambda & \lambda + 2\mu & & & \\ & & & \mu & & \\ & & & & \mu & \\ & & & & & \mu \end{bmatrix}.$$

We can represent this PDE in the form given by (3.3) and therefore investigate the LFA of p-multgrid for this PDE.

The results in Table 4.9 provide the LFA convergence factor and optimal values of k for two-grid high-order p-multigrid for a variety of coarsening rates and orders of Chebyshev smoother. The LFA two-grid convergence factor for three dimensional linear elasticity is much poorer than the LFA two-grid convergence factor for the three dimensional scalar Laplacian, and the effect of rapid coarsening in the polynomial degree of the basis functions is far more pronounced for linear elasticity.

p_{fine} to p_{coarse}	$k = 1$	$k = 2$	$k = 3$	$k = 4$
$p = 2$ to $p = 1$	0.815	0.521	0.321	0.204
$p = 4$ to $p = 2$	0.878	0.666	0.499	0.382
$p = 4$ to $p = 1$	0.969	0.908	0.850	0.800
$p = 8$ to $p = 4$	0.939	0.826	0.725	0.643
$p = 8$ to $p = 2$	0.981	0.943	0.906	0.873
$p = 8$ to $p = 1$	0.995	0.984	0.973	0.963

Table 4.9: Two-grid convergence factor with Chebyshev smoothing for 3D linear elasticity.

5. Conclusions. In this paper we developed LFA of p-multigrid with arbitrary second-order PDEs using high-order finite element discretizations by using an operator representation for efficient application of matrix-free implementations. We introduced LFAToolkit.jl [30], a new Julia package for LFA of high-order finite element methods, and used LFAToolkit.jl to investigate p-multigrid method with Jacobi and Chebyshev semi-iterative method smoothing and aggressive coarsening strategies. We observed that the performance of p-multigrid with these two polynomial smoothers degrades as we coarsen more aggressively.

The LFA of p-multigrid framework presented here can be extended to the LFA of h-multigrid methods and reproduces previous work in this area. Also, we briefly described how this work can be extended to LFA of h-multigrid for finite difference discretization stencils that can be represented as finite difference methods.

The LFA formulation in LFAToolkit.jl accurately predicts the performance of two-grid p-multigrid with polynomial smoothers for the scalar Laplacian in three dimensions and can provide two-grid convergence factors for more complex PDEs, such as linear elasticity in three dimensions.

REFERENCES

[1] M. Adams, M. Brezina, J. Hu, and R. Tuminaro, *Parallel multigrid smoothing: polynomial versus Gauss-Seidel*, Journal of Computational Physics, 188 (2003), pp. 593–610, https://doi.org/10.1016/S0021-9991(03)00194-3.

[2] S. Balay, S. Abhyankar, M. F. Adams, J. Brown, P. Brune, K. Buschelman, L. Dalcin, A. Dener, V. Eijkhout, W. D. Gropp, D. Karpeyev, D. Kaushik, M. G. Knepley, D. A. May, L. C. McInnes, R. T. Mills, T. Munson, K. Rupp, P. Sanan, B. F. Smith, S. Zampini, H. Zhang, and H. Zhang, *PETSc users manual*, Tech. Report ANL-95/11 - Revision 3.15, Argonne National Laboratory, 2021, https://www.mcs.anl.gov/petsc.

[3] A. Brandt, *Multi-level adaptive solutions to boundary-value problems*, Mathematics of Computation, 31 (1977), pp. 333–390.

[4] A. Brandt, *Guide to multigrid development*, in Multigrid methods, Springer, 1982, pp. 220–312.

[5] J. Brannick, X. Hu, C. Rodrigo, and L. Zikatanov, *Local Fourier analysis of multigrid methods with polynomial smoothers and aggressive coarsening*, Numerical Mathematics: Theory, Methods and Applications, 8 (2015), p. 1–21, https://doi.org/10.4208/nmtma.2015.w01si.

[6] W. L. Briggs, V. E. Henson, and S. F. McCormick, *A multigrid tutorial*, SIAM, 2000.

[7] J. Brown, *Efficient nonlinear solvers for nodal high-order finite elements in 3D*, Journal of Scientific Computing, 45 (2010), pp. 48–63, https://doi.org/10.1007/s10915-010-9396-8.

[8] J. Brown, A. Abdelfattah, V. Barra, N. Beams, J. S. Camier, V. Dobrev, Y. Dudouit, L. Ghaffari, T. Kolev, D. Medina, W. Pazner, T. Ratnayaka, J. L. Thompson, and

S. Tomov, *libCEED: Fast algebra for high-order element-based discretizations*, Journal of Open Source Software, 6 (2021), p. 1945, https://doi.org/10.21105/joss.02945.

[9] J. Brown, V. Barra, N. Beams, L. Ghaffari, M. Knepley, W. Moses, R. Shakeri, K. Stengel, J. L. Thompson, and J. Zhang, *Performance portable solid mechanics via matrix-free p-multigrid*, 2022, https://doi.org/10.48550/ARXIV.2204.01722, https://arxiv.org/abs/2204.01722.

[10] J. Brown, Y. He, and S. MacLachlan, *Local Fourier analysis of balancing domain decomposition by constraints algorithms*, SIAM Journal on Scientific Computing, 41 (2019), pp. S346–S369, https://doi.org/10.1137/18M1191373.

[11] J. Brown, Y. He, S. MacLachlan, M. Menickelly, and S. M. Wild, *Tuning multigrid methods with robust optimization and local Fourier analysis*, SIAM Journal of Scientific Computing, 43 (2021), pp. A109–A138, https://doi.org/10.1137/19M1308669.

[12] D. Davydov, J.-P. Pelteret, D. Arndt, M. Kronbichler, and P. Steinmann, *A matrix-free approach for finite-strain hyperelastic problems using geometric multigrid*, International Journal for Numerical Methods in Engineering, (2020), https://doi.org/10.1002/nme.6336.

[13] L. Demkowicz, J. T. Oden, W. Rachowicz, and O. Hardy, *Toward a universal $h - p$ adaptive finite element strategy, Part 1. constrained approximation and data structure*, Computer Methods in Applied Mechanics and Engineering, 77 (1989), pp. 79–112, https://doi.org/10.1016/0045-7825(89)90129-1.

[14] M. O. Deville, P. F. Fischer, and E. H. Mund, *High-order methods for incompressible fluid flow*, Cambridge University Press, 2002.

[15] P. Fischer, M. Min, T. Rathnayake, S. Dutta, T. Kolev, V. Dobrev, J.-S. Camier, M. Kronbichler, T. Warburton, K. Swirydowicz, and J. Brown, *Scalability of high-performance PDE solvers*, International Journal of High Performance Computing Applications, 34 (2020), https://doi.org/10.1177/1094342020915762.

[16] M. H. Gutknecht and S. Röllin, *The Chebyshev iteration revisited*, Parallel Computing, 28 (2002), pp. 263–283, https://doi.org/10.1016/S0167-8191(01)00139-9.

[17] Y. He and S. MacLachlan, *Two-level Fourier analysis of multigrid for higher-order finite-element discretizations of the Laplacian*, Numerical Linear Algebra with Applications, 27 (2020), p. e2285, https://doi.org/10.1002/nla.2285.

[18] T. J. Hughes, *The finite element method: linear static and dynamic finite element analysis*, Courier Corporation, 2012.

[19] K. Kahl and N. Kintscher, *Automated local Fourier analysis (aLFA)*, BIT Numerical Mathematics, 60 (2020), pp. 651–686, https://doi.org/10.1007/s10543-019-00797-w.

[20] D. A. Knoll and D. E. Keyes, *Jacobian-free Newton–Krylov methods: a survey of approaches and applications*, Journal of Computational Physics, 193 (2004), pp. 357–397, https://doi.org/10.1016/j.jcp.2003.08.010.

[21] M. Kronbichler and K. Ljungkvist, *Multigrid for matrix-free high-order finite element computations on graphics processors*, ACM Transactions on Parallel Computing (TOPC), 6 (2019), pp. 1–32, https://doi.org/10.1145/3322813.

[22] P. Kumar, C. Rodrigo, F. J. Gaspar, and C. W. Oosterlee, *On local Fourier analysis of multigrid methods for PDEs with jumping and random coefficients*, SIAM Journal on Scientific Computing, 41 (2019), pp. A1385–A1413, https://doi.org/10.1137/18M1173769.

[23] S. P. MacLachlan and C. W. Oosterlee, *Local Fourier analysis for multigrid with overlapping smoothers applied to systems of PDEs*, Numerical Linear Algebra with Applications, 18 (2011), pp. 751–774, https://doi.org/10.1002/nla.762.

[24] J. T. Oden, L. Demkowicz, W. Rachowicz, and T. Westermann, *Toward a universal $h - p$ adaptive finite element strategy, Part 2. a posteriori error estimation*, Computer methods in applied mechanics and engineering, 77 (1989), pp. 113–180, https://doi.org/10.1016/0045-7825(89)90130-8.

[25] W. Rachowicz, J. T. Oden, and L. Demkowicz, *Toward a universal $h - p$ adaptive finite element strategy part 3. design of $h - p$ meshes*, Computer Methods in Applied Mechanics and Engineering, 77 (1989), pp. 181–212, https://doi.org/10.1016/0045-7825(89)90131-X.

[26] H. Rittich, *Extending and automating Fourier analysis for multigrid methods*, PhD thesis, Universität Wuppertal, Fakultät für Mathematik und Naturwissenschaften, 2018.

[27] C. Rodrigo, F. J. Gaspar, and L. T. Zikatanov, *On the validity of the local Fourier analysis*, Journal of Computational Mathematics, 37 (2019), pp. 340–348, https://doi.org/10.4208/jcm.1803-m2017-0294.

[28] E. M. Rønquist and A. T. Patera, *Spectral element multigrid. I. formulation and numerical results*, Journal of Scientific Computing, 2 (1987), pp. 389–406.

[29] K. Stüben and U. Trottenberg, *Multigrid methods: Fundamental algorithms, model problem analysis and applications*, in Multigrid methods, Springer, 1982, pp. 1–176.

[30] J. L. Thompson, *LFAToolkit.* https://github.com/jeremylt/LFAToolkit.jl, 2021.
[31] U. Trottenberg, C. W. Oosterlee, and A. Schüller, *Multigrid*, Academic Press, Inc., San Diego, CA, 2001. With contributions by A. Brandt, P. Oswald and K. Stüben.
[32] J. J. van der Vegt and S. Rhebergen, *Discrete Fourier analysis of multigrid algorithms*, Memorandum Department of Applied Mathematics, University of Twente, Enschede, The Netherlands, (2011).
[33] R. Wienands and W. Joppich, *Practical Fourier analysis for multigrid methods*, CRC press, 2004.