

A BOOTSTRAP MULTIGRID EIGENSOLVER*

JAMES BRANNICK[†] AND SHUHAO CAO[‡]

Abstract. This paper introduces bootstrap multigrid methods for solving eigenvalue problems arising from the discretization of partial differential equations. Inspired by the full bootstrap algebraic multigrid setup algorithm that includes an AMG eigensolver, we illustrate how the algorithm can be simplified for the case of a discretized partial differential equation, thereby developing a bootstrap geometric multigrid (BMG) approach. We illustrate numerically the efficacy of the BMG method for (1) recovering eigenvalues having large multiplicity, (2) computing interior eigenvalues, and (3) approximating shifted indefinite eigenvalue problems. Numerical experiments are presented to illustrate the basic components and ideas behind the success of the overall bootstrap multigrid approach. For completeness, we present a simplified error analysis of a two-grid bootstrap algorithm for the Laplace–Beltrami eigenvalue problem.

Key words. eigenvalue problem, bootstrap multigrid, multigrid eigensolver, surface finite element method

MSC codes. 58C40, 65N25, 65N30, 65N55

DOI. 10.1137/20M131151X

1. Introduction. The aim of this paper is to present a bootstrap multigrid (BMG) framework for solving eigenvalue problems arising from discretizing partial differential equations (PDEs). The approach we consider is motivated by the bootstrap algebraic multigrid (BAMG) setup algorithm designed and analyzed in [8, 7, 29, 11, 10]. The key components of the BAMG setup algorithm that motivate our proposed geometric BMG algorithm are the adaptive or bootstrap construction of the coarse spaces coupled with the BAMG multilevel eigensolver used in constructing interpolation (or prolongation). The overall BAMG process simultaneously computes approximations of algebraically smooth error and constructs improved or enriched coarse spaces using these approximations. The resulting BAMG setup algorithm has been shown to provide for an efficient algebraic multigrid (AMG) eigensolver as well as a robust and efficient AMG setup process for solving sparse linear systems [8, 11]. In this paper, we adapt the BAMG setup algorithm to a geometric method, i.e., one that makes explicit use of the PDE and discretization in its design, and draw connections between this BMG approach and existing geometric two-grid and multigrid eigensolvers that have been developed for approximating PDE eigenvalue problems. We illustrate numerically that the resulting bootstrap method is suitable for (1) recovering eigenvalues having large multiplicity, (2) computing interior eigenvalues, and (3) approximating shifted indefinite eigenvalue problems.

*Received by the editors January 8, 2020; accepted for publication (in revised form) by J. H. Adler June 28, 2022; published electronically November 18, 2022.

<https://doi.org/10.1137/20M131151X>

Funding: The authors' research was supported in part by the National Science Foundation under grants DMS-1320608, DMS-1418934, DMS-1620346, and DMS-2136075.

[†]Department of Mathematics, Pennsylvania State University, University Park, State College, PA 16802 USA (brannick@psu.edu).

[‡]Division of Computing, Analytics, and Mathematics, School of Science and Engineering, University of Missouri Kansas City, Kansas City, MO 64110 USA (sch59@umkc.edu).

The problem of interest in this paper to present a BMG eigensolver is the Laplace–Beltrami eigenvalue problem

$$(1.1) \quad -\Delta_\Gamma u = \lambda u,$$

where Δ_Γ denotes the Laplace–Beltrami operator on a two-dimensional, smooth, orientable, and closed surface Γ , $\lambda \in \mathbb{R}^+$ is the eigenvalue to the continuous eigenvalue problem, and $u : \Gamma \rightarrow \mathbb{R}$ denotes the associated eigenfunction. Letting

$$(1.2) \quad a(u, v) := \int_\Gamma \nabla_\Gamma u \cdot \nabla_\Gamma v \, dS, \quad \text{and} \quad b(u, v) := \int_\Gamma uv \, dS,$$

the weak formulation of (1.1) is as follows: Find $u \in H^1(\Gamma)$ and $\lambda \in \mathbb{R}^+$ such that

$$(1.3) \quad a(u, v) = \lambda b(u, v) \quad \text{for any } v \in H^1(\Gamma),$$

where $H^1(\Gamma) := \{v \in L^2(\Gamma) : \nabla_\Gamma v \in L^2(\Gamma)\}$ equipped with the following norms:

$$\|v\|_{H^1(\Gamma)}^2 := \|v\|_{L^2(\Gamma)}^2 + |v|_{H^1(\Gamma)}^2 \quad \text{with} \quad |v|_{H^1(\Gamma)} := \|\nabla_\Gamma v\|_{L^2(\Gamma)}.$$

Throughout this paper, we assume that a surface finite element discretization [20, 21] is used to approximate problem (1.3).

Though we derive the geometric BMG eigensolver with a focus on the finite element approximation to the problem above, we note that the overall strategy is applicable for much wider classes of PDE eigenvalue problems. The Laplace–Beltrami model is selected as our model problem since it is a challenging problem in that its eigenvalues have large multiplicity. Moreover, the Laplace–Beltrami spectrum on a 2-sphere is explicitly known, allowing us to study the proposed BMG approach in detail for a specific problem.

1.1. The BAMG setup. The BAMG setup algorithm that serves as the main motivation for our proposed BMG eigensolver can be viewed as an AMG eigensolver for the algebraic system $Au = \lambda Bu$ with (λ, u) the unknown eigenpairs. As we show below, it is straightforward to derive a BMG approach directly from the BAMG setup algorithm.

In contrast to geometric multigrid methods that begin with a coarse problem (on a coarse mesh) and then aim to construct a sequence of problems on increasingly finer meshes (resulting in more accurate approximations), in AMG the problem is assumed to be given on the finest mesh, and then is sought to be coarsened in an algebraic fashion, i.e., using only algebraic information available from the linear system. Accordingly, we label the finest-grid system matrix as $A_0 = A$. Later on, in our description of the related geometric approaches, we adopt the usual geometric multigrid notation and set the coarsest-grid system to A_0 . When discussing two-level approaches, A_h represents the fine-grid matrix, and A_H denotes the coarse-grid system matrix, obtained using the Galerkin form $A_H = P^T A_h P$ in the AMG setting, or by rediscretizing in the geometric multigrid setting.

Given the fine-grid system based on A_h , the BAMG setup algorithm aims to construct an interpolation operator P , and the corresponding coarse-grid matrix $A_H = P^T A_h P$. A certain approximation property is satisfied by the interpolation operator and, hence, by the corresponding coarse space [8]. Generally, the AMG interpolation operator must accurately approximate (generalized) eigenvectors corresponding to small eigenvalues of the fine-grid system matrix A_h [10], which are referred to as the slow-to-converge error of the AMG smoother. Various techniques exist for

constructing accurate interpolation for problems where such spectral information is available, e.g., classical AMG [9] and smoothed aggregation [36]. The adaptive AMG [12, 13] and BAMG algorithms were designed to construct P for cases where this information is not available, e.g., quantum dynamics applications. The main idea in these approaches is to make use of a multilevel algorithm in the AMG setup to compute slow-to-converge error of relaxation, and then to use these modes to adapt the coarse space. The overall BAMG approach for enriching the coarse spaces uses the combination of a multilevel eigensolver that efficiently and accurately computes approximations to the (generalized) eigenvectors with small eigenvalues of the fine-grid system matrix, i.e., it aims to solve problems of the form

$$(1.4) \quad Au = \lambda Bu,$$

e.g., a surface finite element discretization of problem (1.3). Then, a least squares (LS) process is used to construct interpolation to fit these computed approximations. The overall approach is applied in the usual recursive way to obtain the corresponding multilevel algorithm. We describe the overall process adopting the same notation we used in [8] and recall some numerical results obtained for the Poisson problem as well.

BAMG interpolation is derived to provide the best LS fit to a set of smooth test vectors $\mathcal{V} = \{v^{(1)}, \dots, v^{(k)}\}$. Specifically, each row of interpolation, denoted by p_i , is defined as the minimizer of the local LS functional:

$$(1.5) \quad \mathcal{L}(p_i) = \sum_{\kappa=1}^k \omega_{\kappa} \left((v)^{(\kappa)}_{\{i\}} - \sum_{j \in \Omega_i} (p_i)_j v^{(\kappa)}_{\{j\}} \right)^2,$$

where Ω_i are the sets of interpolation points, $v_{\tilde{\Omega}}$ denotes the canonical restriction of a vector v to the set $\tilde{\Omega} \subset \Omega := \{1, \dots, n\}$, and ω_{κ} denotes the *interpolation weights*. A common choice for the weights is given by $\omega_k = \|v_k\|/\|Av_k\|$, in which case the LS functional can be viewed as a local version of the weak approximation property [8], assuming A is symmetric and positive definite.

The rationale behind the bootstrap multilevel generalized eigensolver is as follows. If an initial multigrid hierarchy is constructed using LS interpolation, given the initial Galerkin operators $A_0 = A, A_1, \dots, A_L$ on each level and the corresponding interpolation operators $P_{l+1}^l, l = 0, \dots, L-1$, define the composite interpolation matrices and corresponding mass matrices as $P_l = P_1^0 \dots P_l^{l-1}$ and $T_l = P_l^T B_0 P_l, l = 1, \dots, L$; then for any level l and any given vector $x_l \in \mathbb{R}^{n_l}$ and $\lambda^l \in \mathbb{R}$ such that $A_l x_l = \lambda^l T_l x_l$ we have

$$(1.6) \quad \text{Rayleigh quotient of } P_l x^l := \frac{\langle x_l, x_l \rangle_{A_l}}{\langle x_l, x_l \rangle_{T_l}} = \frac{\langle P_l x_l, P_l x_l \rangle_{A_0}}{\langle P_l x_l, P_l x_l \rangle_{T_l}} = \lambda^l,$$

where we used that $\langle x_l, x_l \rangle_{A_l} = \langle P_l x_l, P_l x_l \rangle_{A_0}$, which follows from the definition of $A_l = P_l^T A_0 P_l$. This result relates the eigenvectors and eigenvalues of any of the coarse-grid matrices to the eigenvectors and eigenvalues of the finest-grid operator A . Note that the eigenvalue approximations in (1.6) are continuously updated within the algorithm so that the overall approach resembles an inverse Rayleigh-quotient iteration found in eigenvalue computations (cf. [38]). Figure 1.1 provides a schematic outline of a V -cycle version of the BAMG setup algorithm. We note that $\mathcal{V}^r$ and $\mathcal{V}^e$ are the sets of approximate test vectors from relaxations and approximate eigenvectors, respectively, that are used in computing the LS interpolations.

Following are a few general remarks regarding the above BAMG setup algorithm and its conversion to a geometric multigrid method, before we proceed with a general discussion of the BMG eigensolver:

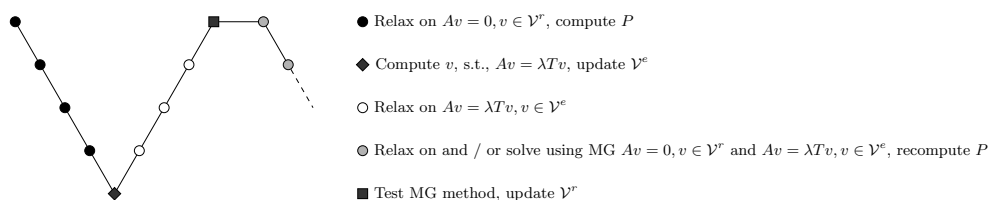


FIG. 1.1. A possible Galerkin BAMG V-cycle setup scheme from [8].

- The BAMG eigensolver only solves the eigenvalue problem on the coarsest level and then uses an iterative method to approximately solve (simpler) shifted linear systems on the corresponding finer levels. This strategy of only solving the eigenvalue problem on the coarsest level is what gives the approach its efficiency and ultimately its near optimality.
- The method continuously updates (enriches) the coarse space using these improved eigen-approximations in the LS interpolation process and then the Galerkin definition of the coarse-level operator. Moreover, it is this enrichment that gives the method the flexibility to construct approximations to both oscillatory and smooth eigenvectors using very coarse meshes.
- The overall procedure uses a relatively large enrichment space since experimentally this gives the best overall results in the BAMG construction of interpolation and, hence, ultimately yields the best overall AMG solver. We propose a similar enrichment strategy that uses multiple vectors in the BMG eigensolver; however, since we are interested in solving the eigenvalue problem and not using the approach as a setup process for defining interpolation, we omit the set of relaxed vectors $\mathcal{V}^e$ in the BMG algorithm and enrich the coarse space using only eigenvector approximations $\mathcal{V}^e$. Correspondingly, the BMG approach begins by solving an eigenvalue problem on the coarsest level A_0 .
- A main cost in the BAMG setup is the need to compute the AMG hierarchy in each iteration, i.e., the cost of computing P_{l+1}^l , and $A_l = (P_{l+1}^l)^T A_{l-1} P_{l+1}^l$ on all levels. In the BMG setting, the coarse spaces are defined using the chosen finite element discretization plus an additional global enrichment space. This difference is, in fact, the main modification that we make to the BAMG eigensolver in order to arrive at the BMG approach.
- Though we include the possibility of using the existing MG solver in the BAMG setup when solving the shifted linear systems on finer meshes, typically only a few relaxation steps are needed. We use many pre- and postsmoothing steps in the BAMG setup in order to reduce the number of setup cycles, which, in turn, offsets the cost of recomputing the AMG hierarchy in each iteration of the BAMG setup. For example, using a standard discretization for the two-dimensional Laplace eigenvalue problem in [8] on a structured mesh, two $V(4,4)$ BAMG setup cycles as depicted in Figure 1.1 on a finest-level mesh of size 64×64 using standard (full) coarsening and Gauss-Seidel relaxation yield comparable result versus direct methods with all geometric information. These partial experimental results from [8, Figure 4.2 and Table 4.5] illustrate the efficacy and potential of the BAMG setup algorithm as an AMG eigensolver.

1.2. Existing geometric multigrid eigensolvers. Generally speaking, in situations where multiple eigen-modes are required, a highly refined uniform mesh is needed for a good approximation for all these modes. In such cases, solving the associated systems of equations on this given fine mesh can be prohibitively expensive. There are many methods taking advantage of a multilevel hierarchy to remedy this cost. For the two-dimensional Laplace eigenvalue problem on the plane, a two-grid eigensolver that reduces the overall costs of solving the resulting finite element systems was proposed in [39] and later improved using a Newton-type iteration in [25]. A similar approach was designed for the Maxwell eigenvalue problem in [40]. These two-grid methods involve a coarse mesh and a fine mesh and the finite element spaces defined on these meshes. In addition, a direct solve, e.g., `eig` in MATLAB, is used to solve the coarse space eigenvalue problem, and then Newton's method is applied on the fine mesh in order to solve the nonlinear eigenvalue problem using the coarse solution as an initial guess, which results in solving a linear source problem. The a priori analysis for the finite element approximations to the Laplace–Beltrami eigenvalue problem is studied in [6, 33]. To the best of our knowledge, there is no known two-grid (or multigrid) eigensolver for the Laplace–Beltrami operator on surfaces in the finite element setting, which is the model problem we design the BMG eigensolver for in this paper.

Though two-grid methods do provide significant improvements when compared with single-grid methods (such as the Arnoldi algorithm) in terms of their computational complexity, they too have drawbacks in practice. First, two-grid methods are generally not optimal since the mesh spacing tends to zero even for the coarse eigenvalue problem, which needs to be solved with high accuracy. For example, in order to resolve eigenpairs corresponding to large eigenvalues in the discrete spectra for our model problem, the coarse mesh used in the two-grid method must be fine enough to capture these oscillatory modes. In practice, we observe a “loss of spectra” phenomenon when using two-grid methods where the coarse mesh is not fine enough. This issue is overcome in our proposed BAMG algorithm by using the idea to enrich the coarse space as in [8] and as discussed above. We note in addition that these two-grid eigensolvers often require solving a linear source problem on the fine mesh that is indefinite so that using optimal solvers such as multigrid can become problematic. As we show numerically in this paper, it is not necessary to solve this indefinite problem directly, and a few sweeps of an iterative solver suffice to obtain a nearly optimal multigrid algorithm. In fact, for certain cases, we show that the shift can be moved to the right-hand side using an interpolated coarse approximation to the eigenfunction of interest.

In [14], multilevel analogues of the two-grid solvers noted above are developed. Specifically, the paper develops multilevel approaches for nearly singular elliptic problems and eigenvalue problems. It should be noted that these methods are able to approximate the components in the eigenspace with small eigenvalues of (1.3) and as presented cannot be used to approximate larger eigenpairs. This is because on the coarsest level the corresponding space is defined using standard finite element basis functions, e.g., nodal Lagrange basis, and thus it is not possible to approximate oscillatory functions.

In a recent paper [27], another multilevel approach was developed in which the coarse eigenvalue problem is solved in an enriched space. This enrichment is achieved by including a single extra function in the coarse space that is obtained by solving a positive definite source problem on a finer mesh. Then, this two-grid correction scheme is used repeatedly to span multiple levels, resembling the bootstrapping procedure

developed in [8] but only for a single enrichment vector. The approach we propose involves a suitable geometric projection from coarse spaces to fine spaces (defined on a sequence of refined and nonnested meshes) and the use of a bootstrap enrichment procedure to iteratively improve the coarse spaces until the desired approximation is computed to sufficient accuracy.

It is known that the Laplace–Beltrami eigenvalues have very high multiplicity on closed surfaces (e.g., see [34, Chapter 3]), which adds to the difficulties associated with solving this system. For example, for the Laplace–Beltrami operator on the 2-sphere, the number of linearly independent eigenfunctions associated with the l th distinct eigenvalue $\lambda = l(l+1)$ is $2l+1$. Thus, the bootstrap approach we propose which enriches the coarsest space with a subspace of linearly independent approximations is of particular interest for this model problem. We note that the idea to enrich the coarse space in designing eigensolvers goes back to [26] and [28]. These authors also analyze an iterative method for computing the smallest eigenpair under the somewhat restrictive condition that the initial guess of the eigenfunction is sufficiently close to the smallest one, namely that its Rayleigh quotient lies between the smallest and second smallest eigenvalues. In [15], the method from [26] is extended to both two-grid and multigrid methods and an algorithm for computing a given number of the smallest eigenpairs is presented. The paper also presents a convergence theory with less restrictive assumptions on the initial guess.

In this paper, we develop the BMG eigensolver for the surface finite element discretization of the shifted Laplace–Beltrami eigenvalue problem. The base two-grid method can be viewed as a generalization of the approaches proposed in [26, 28, 15, 27] in that the coarse space is enriched with a subspace, instead of a single eigenfunction, and we consider computing interior eigenvalues directly by introducing a shift. Alternatively, our proposed approach can be viewed as a simplification of the BAMG algorithm [8] in that we use the finite element spaces to explicitly define the components of the multilevel method, including interpolation and restriction operators among different levels, and the enriched coarsest space eigenvalue problem. Meanwhile, the iterative procedure involving multilevels of mesh refinement resembles the geometric cascadic multigrid in [35] and [23]. Overall, we note that though we focus on this model problem the approach we present here is applicable to much wider classes of problems with relatively few modifications.

This paper is organized as follows. In section 2, we provide some preliminary notation and present the a priori estimate of the surface finite element method for the eigenvalue problems from [6]. In section 3, we introduce the standard two-grid method for Laplace–Beltrami eigenvalue problems on surfaces mimicking the approaches developed for the elliptic eigenvalue problem in [39, 25, 40]. In addition, we prove the convergence of this method for the case of a smooth, closed, and orientable surface. In section 4, we derive the finite element BMG method for the Laplace–Beltrami eigenvalue problem and give details on the approach. In addition, we prove the convergence of the bootstrap two-grid eigensolver for the shifted Laplace–Beltrami eigenvalue problem in the case the enrichment space is defined using several functions. We note that on a surface the nonnestedness of the meshes also introduces a geometric error into the discrete approximations. In this regard, our model problem presents several challenges and our two-grid approach is quite general since the inclusion of a shift also covers the case of computing interior eigenvalues (with high multiplicity). Section 5 contains results of numerical experiments for both the two-grid and multigrid methods applied to the model problem on $\mathbb{S}^2$. Note that by fixing the geometry we are able to study the algorithm in a detailed and systematic way.

2. Notation and preliminary results. In this section, the finite element approximation, together with its a priori error estimate, to the eigenvalue problem (1.3) are presented. Before presenting the details of the discretization we set notation that is used throughout the paper. Here, we need to distinguish between the continuous solution of the eigenvalue problem on the continuous surface, Γ , the continuous solution on the discrete surface (mesh), Γ_h , used to approximate the surface (which introduces a geometric error), and the discrete finite element approximation obtained on the discrete surface which admits a discretization error.

We denote the eigen-modes to the continuous eigenvalue problem on the continuous surface by lowercase letters, e.g., u , and the associated eigenvalues by λ . The associated solution to the continuous eigenvalue problem on the discrete surface Γ_h is denoted by $\bar{u}$ and the associated eigenvalues are given by $\bar{\lambda}$. We distinguish between these two solutions since the solution on the discrete surface will only approximate the true solution since the discrete surface approximation introduces a geometric error. The finite element spaces are similarly defined, where we use instead capital calligraphic letters, e.g., the finite element space on Γ_h is denoted by $\mathcal{V}_h$.

In describing a multigrid approach for solving eigenvalue problems it is convenient to distinguish between the solutions of the nonlinear finite element eigenvalue problem and the solutions to associated linear problems. Assume that a pair of finite element spaces are given, where the coarse finite element approximation space is $\mathcal{V}_H$ and the fine space is $\mathcal{V}_h$. Then, the subscript on u_h denotes that this is the solution to a direct eigensolve of the eigenvalue problem in $\mathcal{V}_h$. A superscript u^h denotes a source problem approximation in the fine space $\mathcal{V}_h$. In a two-grid method, a direct eigensolve solution u_H in a coarser space $\mathcal{V}_H$ (to a nonlinear problem) is used, and a source problem approximation u^h using u_H as data in a finer space $\mathcal{V}_h$ yields an improved approximation. The vector representation of u_h in the canonical finite element basis of $\mathcal{V}_h$ is denoted by U_h , while U^h is for u^h . This nomenclature, where a subscript corresponds to a direct eigensolve and a superscript stands for source problem approximation, is adopted throughout the paper. Matrices and operators are denoted with capital letters, where the actual definition should be clear from context. Moreover, the details on our construction of fine meshes and the associated eigen-spaces are given in the beginning of section 3.

We use $x \lesssim y$ and $z \gtrsim w$ to represent $x \leq c_1 y$ and $z \geq c_2 w$, respectively, where c_1 and c_2 are two constants independent of the mesh size h and eigenvalues. The constants in these inequalities may in certain cases depend on specific eigenvalues and when such dependence exists, it will be stated explicitly.

The surface gradient operator on a two-dimensional smooth orientable surface that can be embedded into $\mathbb{R}^3$ can be defined using extensions $\nabla_\Gamma : H^1(\Gamma) \rightarrow (L^2(\Gamma))^3$ as follows:

$$(2.1) \quad (\nabla_\Gamma f)(\mathbf{x}) := (I - \mathbf{n}(\mathbf{x})\mathbf{n}(\mathbf{x})^\top) \nabla \tilde{f}(\mathbf{x}) = \mathbf{n}(\mathbf{x}) \times (\nabla \tilde{f}(\mathbf{x}) \times \mathbf{n}(\mathbf{x})),$$

where $\tilde{f}$ is a smooth extension of f to a three-dimensional tubular neighborhood U of Γ , $\nabla : H^1(\Omega) \rightarrow (L^2(\Omega))^3$ is the weak gradient operator in $\mathbb{R}^3$, and $\mathbf{n}(\mathbf{x})$ is the unit normal pointing to the outside of this closed surface at point $\mathbf{x}$. The Laplace–Beltrami operator Δ_Γ is then defined in a distributional sense:

$$(2.2) \quad \langle -\Delta_\Gamma f, g \rangle = \int_\Gamma (-\Delta_\Gamma f) g \, dS := \int_\Gamma \nabla_\Gamma f \cdot \nabla_\Gamma g \, dS \quad \forall g \in C^\infty(\Gamma).$$

For a more detailed definition and the technicalities that arise when defining a differential operator on surfaces, we refer readers to [18, 21, 5].

2.1. The eigenvalue problem on the discrete surface. Let $\mathcal{T}_h = \{T\}$ be a triangulation and $\Gamma_h = \cup_{T \in \mathcal{T}_h} T$ be a piecewise planar surface approximating the continuous surface Γ , where T stands for the “flat” triangular element. Γ_h is assumed to be quasi-uniform and regular. The mesh size is then defined as the maximum of the diameter of all the triangles: $h := \max_{T \in \mathcal{T}_h} \text{diam } T$. Furthermore, the set of all vertices is denoted by $\mathcal{N}_h$. For any $\mathbf{z} \in \mathcal{N}_h$, it is assumed that $\mathbf{z} \in \Gamma$, i.e., any vertex in the triangulation lies on the original continuous surface Γ .

Note that the surface gradient on a smooth surface carries over naturally to a discrete surface Γ_h (e.g., see [37]): the unit normal $\mathbf{n}(\mathbf{x})$ is now a constant vector $\mathbf{n}_T$ for each point $\mathbf{x} \in T$. For ease of notation, the surface gradients ∇_{Γ_h} on Γ_h and on Γ will both be denoted by ∇_Γ , where the definition should be clear from the context. With these definitions the bilinear forms on Γ_h are as follows:

$$(2.3) \quad a_h(\bar{u}, \bar{v}) := \int_{\Gamma_h} \nabla_\Gamma \bar{u} \cdot \nabla_\Gamma \bar{v} dS, \quad \text{and} \quad b_h(\bar{u}, \bar{v}) := \int_{\Gamma_h} \bar{u} \bar{v} dS.$$

Note that the subscript used in defining the bilinear forms is used to represent that the continuous problem has been restricted to a discrete surface, Γ_h . Moreover, the fact that this discrete surface Γ_h is piecewise linear affine, which gives a $C^{0,1}$ -surface, implies that the Sobolev space $H^1(\Gamma_h)$ is well-defined (see [20]).

The weak formulation for the eigenvalue problem on the discrete surface Γ_h is now given by the following: find $\bar{u} \in H^1(\Gamma_h)$ and $\bar{\lambda} \in \mathbb{R}^+$ such that

$$(2.4) \quad a_h(\bar{u}, \bar{v}) = \bar{\lambda} b_h(\bar{u}, \bar{v}) \quad \text{for any } \bar{v} \in H^1(\Gamma_h).$$

Using the Poincaré inequality (see [18, Lemma 2.2]) or the compact embedding of $H^1(\Gamma_h)/\mathbb{R} \subset \subset L^2(\Gamma_h)$ when Γ_h is a piecewise linear affine manifold [1, Chapter 2], and the geometric error estimate between (1.2) and (2.3) (e.g., see [21, section 4]), it follows that if the mesh is sufficiently fine (required for the coercivity), then for any $\bar{u}, \bar{v} \in H^1(\Gamma_h)/\mathbb{R}$

$$(2.5) \quad a_h(\bar{u}, \bar{v}) \lesssim \|\bar{u}\|_{H^1(\Gamma_h)} \|\bar{v}\|_{H^1(\Gamma_h)}, \quad \text{and} \quad a_h(\bar{u}, \bar{u}) \gtrsim \|\bar{u}\|_{H^1(\Gamma_h)}^2.$$

If $\bar{v} \neq 0$, $b_h(\bar{v}, \bar{v}) = \|\bar{v}\|_{L^2(\Gamma_h)}^2 > 0$, the coercivity and continuity of (2.5) imply that $a_h(\cdot, \cdot)$ induces a bounded, compact, and self-adjoint operator. By the Hilbert–Schmidt theory and the spectrum theory of the Laplace–Beltrami operator on compact surfaces (every closed surface being compact; see [16]), problem (2.4) is a well-posed self-adjoint eigenvalue problem. The eigenvalues $\{\bar{\lambda}_i\}_{i=0}^\infty$ for problem (2.4) form a discrete sequence, starting from 0, with no accumulation point:

$$0 = \bar{\lambda}_0 < \bar{\lambda}_1 \leq \bar{\lambda}_2 \leq \cdots \rightarrow \infty.$$

Moreover, the eigenfunctions ϕ_k associated with $\bar{\lambda}_k$ are orthogonal in the sense that $b_h(\phi_i, \phi_j) = \delta_{ij}$.

Let $M(\bar{\lambda})$ be the eigenspace spanned by the eigenfunctions associated with $\bar{\lambda}$ for (2.4) defined on the discrete surface Γ_h :

$$(2.6) \quad M(\bar{\lambda}) := \{\bar{u} \in H^1(\Gamma_h) : a_h(\bar{u}, \bar{v}) = \bar{\lambda} b_h(\bar{u}, \bar{v}) \quad \forall \bar{v} \in H^1(\Gamma_h)\}.$$

Similarly, the eigenspace for the continuous eigenvalue problem (1.3) on Γ is given by

$$(2.7) \quad M(\lambda) := \{u \in H^1(\Gamma) : a(u, v) = \lambda b(u, v) \quad \forall v \in H^1(\Gamma)\}.$$

2.2. Finite element approximation. In this subsection, the surface finite element discretization (2.9) of eigenvalue problem (2.4) is established. Here, if the geometric error introduced by the discrete surface is sufficiently small, then the surface finite element approximates the eigenvalue problem (1.3) on the original smooth surface. In the last part of this subsection, the a priori error estimation for the surface finite element eigenvalue problem using a direct eigensolve is presented, giving Lemma 2.2. Note that the orders of the approximation errors for the computed eigenpairs given in this lemma are useful in determining the effectiveness of an iterative procedure to obtain approximate eigenpairs, namely, the two-grid or multigrid method need to compute approximations with the same (or higher) order of approximation error.

The finite element approximation to problem (2.4) $\mathcal{V}_h$ is as follows:

$$(2.8) \quad \mathcal{V}_h = \{\phi_h \in C^0(\Gamma_h) : \phi_h|_T \in P^1(T) \quad \forall T \in \mathcal{T}_h\}.$$

The discretization to problem (2.4) is to find $u_h \in \mathcal{V}_h$ and $\lambda_h \in \mathbb{R}^+$ such that

$$(2.9) \quad a_h(u_h, v_h) = \lambda_h b_h(u_h, v_h) \quad \forall v_h \in \mathcal{V}_h.$$

Note that the finite element approximation problem (2.9) serves as a straightforward conforming discretization to (2.4) on the discrete polygonal surface Γ_h , but not directly to the original eigenvalue problem (1.3) on Γ .

The connection between the approximation on Γ_h and its continuous counterpart on the surface Γ is established through a bijective lifting operator (see [18]), between any triangle $T \subset \Gamma_h$ to a curvilinear triangle on Γ . Then, for any $v \in H^1(\Gamma_h)$, its lifting $\tilde{v}$ to the continuous surface Γ can be defined as follows: for any point $\mathbf{x} \in \Gamma_h$, there is a unique point $\tilde{\mathbf{x}} \in \Gamma$ such that

$$(2.10) \quad \tilde{\mathbf{x}} + d(\mathbf{x})\mathbf{n}(\tilde{\mathbf{x}}) = \mathbf{x}, \quad \text{and} \quad \tilde{v}(\tilde{\mathbf{x}}) = v(\mathbf{x}),$$

where $d(\mathbf{x})$ is the signed distance to Γ at point $\mathbf{x} \in \Gamma_h$ and $d(\mathbf{x})$ is positive when $\mathbf{x}$ is outside of the closed surface Γ with $|d(\mathbf{x})| = \min_{\mathbf{y} \in \Gamma} |\mathbf{x} - \mathbf{y}|$.

LEMMA 2.1 (Lemma 3 in [20], Lemma 4.7 in [21]). *If d defined in (2.10) satisfies $\|d\|_{L^\infty(\Gamma_h)} \lesssim h^2$, then for any $v \in H^1(\Gamma_h)$*

$$(2.11) \quad \begin{aligned} (1 - ch^2) |v|_{H^1(\Gamma_h)} &\leq |\tilde{v}|_{H^1(\Gamma)} \leq (1 + ch^2) |v|_{H^1(\Gamma_h)}, \\ \text{and } (1 - ch^2) \|v\|_{L^2(\Gamma_h)} &\leq \|\tilde{v}\|_{L^2(\Gamma)} \leq (1 + ch^2) \|v\|_{L^2(\Gamma_h)}. \end{aligned}$$

For any $u_h, v_h \in \mathcal{V}_h$,

$$(2.12) \quad |b(\tilde{u}_h, \tilde{v}_h) - b_h(u_h, v_h)| \leq ch^2 \|u_h\|_{L^2(\Gamma_h)} \|v_h\|_{L^2(\Gamma_h)}.$$

When using the linear surface finite element on a piecewise planar mesh on Γ_h , the following estimate holds [6].

LEMMA 2.2. *If the mesh size h is small enough and all the vertices of Γ_h lie on Γ , then for an eigenvalue λ of problem (1.3) with multiplicity m on Γ , there exist m $\lambda_{h,k}$'s that are the eigenvalues of problem (2.9) on Γ_h , and*

$$(2.13) \quad |\lambda_{h,k} - \lambda| \leq C(\lambda)h^2 \quad \forall 1 \leq k \leq m.$$

Moreover, let $M(\lambda_h) = \text{span}\{u_{h,k}\}_{k=1}^m$, where $u_{h,k}$'s are the eigenfunctions associated with λ_h ; then for any eigenfunction $u \in M(\lambda)$,

$$(2.14) \quad \min_{w_h \in M(\lambda_h)} |u - \tilde{w}_h|_{H^1(\Gamma)} \leq C(\lambda)h.$$

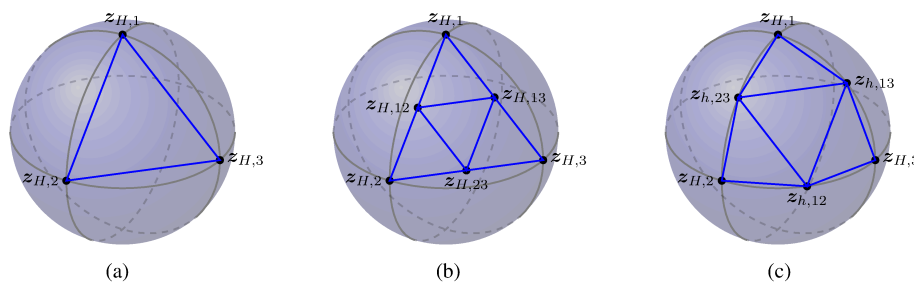


FIG. 3.1. Illustration of the prolongation operator from the coarsest octahedral mesh to one level finer on a 2-sphere: (a) Blue triangle is an element on the coarse mesh to approximate the geometry of a 2-sphere. A function $w \in \mathcal{V}_H$ has value $w(\mathbf{z}_{H,i})$ at $\mathbf{z}_{H,i}$, $i = 1, 2, 3$. (b) Uniform refinement by connecting three midpoints of each edge. Standard prolongation yields the prolonged function's value at $\mathbf{z}_{H,ij}$ which is $(w(\mathbf{z}_{H,j}) + w(\mathbf{z}_{H,i}))/2$. (c) Newly created vertices $\mathbf{z}_{H,ij}$ being projected onto the surface as the vertices for the fine mesh.

3. A two-grid eigensolver. In this section, a two-grid algorithm to accelerate the approximation of (2.9) exploiting a coarse-fine hierarchical structure is presented in Algorithm 3.1. Algorithm 3.1 is an extension to the surface case of the two-grid methods introduced in [25, 40, 39].

Assume that a pair of near hierarchical meshes are given, where the fine mesh $\mathcal{T}_h$ is uniformly refined from the coarse mesh $\mathcal{T}_H$. The newly created vertices, which are the midpoints of the edges of $\mathcal{T}_H$, are projected onto the continuous surface Γ . Denote the finite element approximation space (2.8) on $\mathcal{T}_H$ by $\mathcal{V}_H$ and the fine space by $\mathcal{V}_h$.

Under this setting, we construct a two-grid solution u^h to approximate u_h that is sought through a direct solve. First the subscript $u_H \in \mathcal{V}_H$ is obtained by a direct eigensolve of eigenvalue problem (2.9) (a nonlinear problem). Then the superscript u^h in (3.4) of Algorithm 3.1 is obtained by a source problem approximation using u_H 's prolongation as the right-hand-side data in the finer space $\mathcal{V}_h$.

An important difference in the surface case as considered in this paper, when compared with previous works for the Laplacian eigenvalue problem on the plane, is that the projection operators need extra care. When the mesh is refined, the finite element space on the coarse mesh is not a subspace of the fine mesh. The natural inclusion $\mathcal{V}_H \not\subset \mathcal{V}_h$ does not hold (see Figure 3.1).

Consider the geometric projection operator $\mathcal{G}_{H \rightarrow \hat{H}}$ such that

$$(3.1) \quad \hat{\mathcal{V}}_H := \mathcal{G}_{H \rightarrow \hat{H}} \mathcal{V}_H \subset \mathcal{V}_h.$$

The definition of $\mathcal{G}_{H \rightarrow \hat{H}}$ is then given by, for $\mathbf{z} \in \mathcal{N}_h$ and $\hat{w}_H \in \hat{\mathcal{V}}_H$,

$$(3.2) \quad \hat{w}_H(\mathbf{z}) = \mathcal{G}_{H \rightarrow \hat{H}} w_H(\mathbf{z}) = \begin{cases} w_H(\mathbf{z}) & \text{if } \mathbf{z} \in \mathcal{N}_H, \\ (w_H(\mathbf{z}_{H,1}) + w_H(\mathbf{z}_{H,2}))/2 & \text{if } \mathbf{z} \in \mathcal{N}_h \setminus \mathcal{N}_H, \end{cases}$$

where in the second case, $\mathbf{z}$ is the projected midpoint between the coarse mesh vertices $\mathbf{z}_{H,1}$ and $\mathbf{z}_{H,2}$, using the lifting map (2.10) (see Figure 3.1(c)).

The prolongation operator (or natural inclusion) $\mathcal{P}_h$ can then be defined as

$$\mathcal{P}_h : \mathcal{V}_H \rightarrow \mathcal{V}_h, \quad w_H \mapsto \hat{w}_H.$$

Algorithm 3.1. A two-grid scheme with a spectral shift.

1: Coarse grid eigensolve

Set a fixed shift $\mu_H := \mu \geq 0$, and find (u_H, λ_H) satisfying

$$(3.3) \quad a_H(u_H, v) - \mu_H b_H(u_H, v) = \lambda_H b_H(u_H, v) \quad \text{for any } v \in \mathcal{V}_H.$$

2: Fine grid source approximation

Refine $\mathcal{T}_H$ and perform the geometric projection to get $\mathcal{T}_h$, construct $\mathcal{V}_h$, approximate u^h by solving the following indefinite source problem on $\mathcal{T}_h$:

$$(3.4) \quad a_h(u^h, v) - (\mu_H + \lambda_H) b_h(u^h, v) = b_h(\mathcal{P}_h u_H, v) \quad \text{for any } v \in \mathcal{V}_h.$$

maintaining orthogonality to previous eigenfunctions.

Suppose that the finite element approximation spaces have the following basis set:

$$\mathcal{V}_H = \text{span}\{\phi_{H,i}\}_{i=1}^{N_H}, \text{ and } \mathcal{V}_h = \text{span}\{\phi_{h,i}\}_{i=1}^{N_h}.$$

If $w_H = (W_H)^T \Phi_H$, where $\Phi_H = (\phi_{H,1}, \dots, \phi_{H,N_H})^T$, then

$$\mathcal{V}_h \ni \mathcal{P}_h w_H = \mathcal{P}_h \left((W_H)^T \Phi_H \right) = (P_h W_H)^T \Phi^h,$$

where $P_h \in \mathbb{R}^{N_h \times N_H}$ is the matrix representation of the prolongation operator. Note that the geometric projection is implicitly imposed here.

For example, by (3.2), $\mathcal{P}_h \phi_{H,i}(\mathbf{z}_{H,j}) = \delta_{ij}$, and $\mathcal{P}_h \phi_{H,i}(\mathbf{z}_{h,ij}) = 1/2$ if $i \neq j$, where $\mathbf{z}_{h,ij}$ is a newly created vertex in $\mathcal{N}_h \setminus \mathcal{N}_H$ by projecting the midpoint of $\mathbf{z}_{H,i}$ and $\mathbf{z}_{H,j}$ onto the continuous surface (see Figure 3.1(c)).

Similarly, the restriction operator $\mathcal{P}_H$ is opted to be the transpose of the geometric projection $\mathcal{P}_h$ defined above:

$$\mathcal{P}_H : \mathcal{V}_h \rightarrow \mathcal{V}_H, \quad w_h \mapsto \mathcal{P}_H w_h, \quad \text{and} \quad \mathcal{V}_H \ni \mathcal{P}_H w_h = (P_H W_h)^T \Phi_H,$$

where $P_H = (P_h)^T \in \mathbb{R}^{N_H \times N_h}$ is the matrix of the restriction operator.

Given these definitions, the two-grid method approximating the exact solution of problem (2.9) is given by Algorithm 3.1.

Remark 3.1 (natural extension to a multilevel method). When multiple levels of meshes are available (V_{h_k} for $k = 1, \dots, K$ with $K \geq 3$), Algorithm 3.1 can be naturally extended to be a multilevel method by being applied in a cascading fashion between two adjacent levels. For example, starting from V_{h_1} , when a two-grid eigenpair approximation (u^{h_2}, λ^{h_2}) is obtained, we set $(u_{h_2}, \lambda_{h_2}) \leftarrow (u^{h_2}, \lambda^{h_2})$. Then steps 2 and 3 in Algorithm 3.1 are repeated for level 3 to level K .

Remark 3.2 (approximation accuracy of the source problem). In Algorithm 3.1, the source problem (3.4) can be approximated by a direct or multilevel method. We note that if a multilevel hierarchy exists and the two-grid method is applied in a cascading fashion, then numerically the source problem on $\mathcal{V}_{h_k}$ of the k th level ($k \geq 3$) does not necessarily need to be solved at an accuracy of a direct solver. This implies that a smoother (relaxation method) can be applied to problem (3.4) in

step 2 of Algorithm 3.1, with the approximation from the previous level as an initial guess, instead of a direct solve. Below, in Algorithms 4.1 and 4.2, when the term “approximate” is used for the source problems, the user can choose a direct/multigrid solver or smoother. We illustrate this numerically below in section 5.3.

3.1. Convergence analysis. The main ingredient in proving the convergence of Algorithm 3.1 is to bridge the connection between the geometric projection on the surface [21] with existing two-grid convergence results (e.g., see [25, 40]) for the Laplace eigenvalue problem on the plane. The error introduced by the projection between nonhierarchical spaces that arises in this setting is accounted for in the final estimate we derive.

We use the following lemmas to obtain the convergence estimate of the two-grid method for the Laplace–Beltrami eigenvalue problem. The first lemma, Lemma 3.3 (e.g., see [2, 25]), gives the stability estimate for the discrete shifted problem.

LEMMA 3.3 (discrete inf-sup condition with a shift). *If μ is not an exact eigenvalue to problem (2.9), then there exists a constant $C(\mu)$ such that*

$$\sup_{v \in \mathcal{V}_h} \frac{|a_h(u_h, v) - \mu b_h(u_h, v)|}{|v|_{H^1(\Gamma_h)}} \geq C(\mu) |u_h|_{H^1(\Gamma_h)} \quad \forall u_h \in \mathcal{V}_h.$$

The next lemma, Lemma 3.4, is an important identity used to prove the rate of convergence for the approximation of a certain eigenvalue (e.g., see [2]).

LEMMA 3.4. *Let $(u, \lambda) \in H^1(\Gamma) \times \mathbb{R}^+$ be an eigenpair for problem (1.3); then for any $w \in H^1(\Gamma) \setminus \{0\}$*

$$(3.5) \quad \frac{a(w, w)}{b(w, w)} - \lambda = \frac{a(w - u, w - u)}{b(w, w)} - \lambda \frac{b(w - u, w - u)}{b(w, w)}.$$

We then define the following measure of $\mathcal{V}_h$ ’s approximation power:

$$(3.6) \quad \eta(h) := \sup_{f \in H^1(\Gamma_h), |f|_{H^1} = 1} \inf_{v \in \mathcal{V}_h} |Tf - v|_{H^1(\Gamma_h)},$$

where the operator T is defined as

$$(3.7) \quad a_h(Tf, v) = b_h(f, v) \quad \forall f \in H^1(\Gamma_h) \text{ and } \forall v \in H^1(\Gamma_h).$$

For finite element eigenvalue problems on a convex planar domain $\Omega \subset \mathbb{R}^2$, one can assume that the eigenspace $M(\lambda)$ has certain regularity, e.g., one assumes that $M(\lambda) \subset H^2(\Omega)$, which is the Sobolev space containing functions with second weak derivatives that are L^2 -integrable. As a result, it is shown in [25] that if hierarchical coarse and fine finite element spaces are used, i.e., $V_H \subset \mathcal{V}_h$, the general estimate for the two-grid approximation reads for the linear finite element approximation

$$(3.8) \quad \min_{\alpha \in \mathbb{R}} \|u - \alpha u^h\|_{H^1(\Gamma_h)} \leq C(\lambda) (h + H^4) \quad \text{and} \quad |\lambda - \lambda^h| \leq C(\lambda) (h^2 + H^8).$$

However, in the case of a surface finite element here, Theorem 3.3 in [25] cannot be directly applied due to the facts that (i) only $H^1(\Gamma_h)$, not $H^2(\Gamma_h)$, is well-defined on a piecewise linear triangulation Γ_h (see, e.g., [20, section 2]), where $H^2(\Gamma_h)$ stands for, when being well-defined,

$$(3.9) \quad H^2(\Gamma_h) := \{\bar{v} \in H^1(\Gamma_h) : (\nabla_{\Gamma} \bar{v})_i \in H^1(\Gamma_h), i = 1, 2, 3\};$$

(ii) when refining, the spaces are not hierarchical, i.e., $V_H \not\subset \mathcal{V}_h$ (see Figure 3.1).

In the rest of this subsection, a modified two-grid convergence proof is presented following the method from [40], and similar bounds are obtained as the standard two-grid results in (3.8) given that the geometric error is of $O(h^2)$.

THEOREM 3.5 (convergence of the two-grid method). *Let λ_H be an approximation to the eigenvalue λ of problem (1.3) satisfying the a priori estimate in Lemma 2.2. Consider an approximation u^h obtained from the two-grid method given in Algorithm 3.1 with $\mu = 0$ on Γ_h with h sufficiently small, and $\lambda^h := a_h(u^h, u^h)/b_h(u^h, u^h)$. Then, there exists an eigenfunction $u \in M(\lambda)$ such that the following estimates hold:*

$$(3.10) \quad \min_{\alpha \in \mathbb{R}} |u - \alpha \tilde{u}^h|_{H^1(\Gamma)} \leq C(\lambda)(h + H^4), \quad \text{and} \quad |\lambda - \lambda^h| \leq C(\lambda)(h^2 + H^8).$$

Proof. The proof follows from the ones in [25, 40]. Assume that the coarse approximation λ_H is not an eigenvalue of the discrete eigenvalue problem (2.9) on the fine mesh. Consider an auxiliary solution $\hat{u}^h = (\lambda_h - \lambda_H)u^h$, where u^h solves problem (3.4) with $\mu = 0$; then it can be verified that this $\hat{u}^h$ satisfies

$$a_h(\hat{u}^h, v) - \lambda_H b_h(\hat{u}^h, v) = (\lambda_h - \lambda_H)b_h(\mathcal{P}_h u_H, v) \quad \forall v \in \mathcal{V}_h.$$

In the equation above, (u_h, λ_h) is an eigenpair obtained from the direct solve for problem (2.9), i.e., $a_h(u_h, v) = \lambda_h b_h(u_h, v)$ for any $v \in \mathcal{V}_h$. Taking the difference of these two equations yields that for any $v \in \mathcal{V}_h$

$$(3.11) \quad a_h(u_h - \hat{u}^h, v) - \lambda_H b_h(u_h - \hat{u}^h, v) = (\lambda_h - \lambda_H)b_h(u_h - \mathcal{P}_h u_H, v).$$

Applying the discrete inf-sup stability estimate in Lemma 3.3, we have

$$(3.12) \quad \begin{aligned} C |u_h - \hat{u}^h|_{H^1(\Gamma_h)} &\leq \sup_{v \in \mathcal{V}_h} \frac{|a_h(u_h - \hat{u}^h, v) - \lambda_H b_h(u_h - \hat{u}^h, v)|}{|v|_{H^1(\Gamma_h)}} \\ &= \sup_{v \in \mathcal{V}_h} \frac{|(\lambda_h - \lambda_H)b_h(u_h - \mathcal{P}_h u_H, v)|}{|v|_{H^1(\Gamma_h)}}. \end{aligned}$$

By the triangle inequality and the a priori estimate from Lemma 2.2, we have

$$(3.13) \quad |\lambda_h - \lambda_H| \leq |\lambda - \lambda_h| + |\lambda - \lambda_H| \leq C(\lambda)H^2.$$

Now we restrict the true eigenfunction $u \in M(\lambda)$ with its continuous mapping $\bar{u}$ to the discrete surface Γ_h or Γ_H , and use the definition of $\eta(\cdot)$, the direct solve achieving the best approximation (Lemma 2.2), together with the geometric error (Lemma 2.1):

$$(3.14) \quad \begin{aligned} \sup_{v \in \mathcal{V}_h} \frac{b_h(u_h - \mathcal{P}_h u_H, v)}{|v|_{H^1(\Gamma_h)}} &\leq \sup_{v \in H^1(\Gamma_h)} \frac{b_h(u_h - \bar{u}, v)}{|v|_{H^1(\Gamma_h)}} + \sup_{v \in H^1(\Gamma_h)} \frac{b_h(\bar{u} - \mathcal{P}_h u_H, v)}{|v|_{H^1(\Gamma_h)}} \\ &\leq c_1(\lambda)\eta(h) |u_h - \bar{u}|_{H^1(\Gamma_h)} + c_2(\lambda)\eta(H) |\bar{u} - u_H|_{H^1(\Gamma_H)} + c_3\eta(H)H^2. \end{aligned}$$

Now for $\eta(h)$ and $\eta(H)$ defined in (3.6), since restricting on each element T , the eigenfunction is smooth, then by the interpolation estimate in [18, Lemma 2.2], we have

$$(3.15) \quad \eta(h) = \sup_{f \in H^1(\Gamma_h), |f|_{H^1}=1} \inf_{v \in \mathcal{V}_h} |Tf - v|_{H^1(\Gamma_h)} \leq Ch,$$

where C only depends on the geometry. Using this result, (3.12), and the a priori estimate for the direct solve eigenfunction approximation, we have

$$(3.16) \quad |u_h - \widehat{u}^h|_{H^1(\Gamma_h)} \leq C(\lambda)H^4.$$

Then apply the geometric error estimates (2.11) for both $\widehat{u}^h$ and u_h ,

$$(3.17) \quad |u - \widehat{u}^h|_{H^1(\Gamma)} \leq |u - \widetilde{u}_h|_{H^1(\Gamma)} + c(\lambda)h^2 + |u_h - \widehat{u}^h|_{H^1(\Gamma_h)} \leq C(\lambda)(h + H^4),$$

where h is assumed to be small enough such that the geometric error, which is $O(h^2)$, can be omitted compared with the $O(h)$ term.

Last, using the fact that $\widehat{u}^h = (\lambda_h - \lambda_H)u^h$ we have

$$(3.18) \quad \lambda^h = \frac{a_h(u^h, u^h)}{b_h(u^h, u^h)} = \frac{a_h(\widehat{u}^h, \widehat{u}^h)}{b_h(\widehat{u}^h, \widehat{u}^h)}.$$

By Lemma 3.4, the eigenfunction to be normalized to satisfy $b_h(u^h, u^h) = 1$, the Poincaré inequality, and Lemma 2.1, we can get the estimate for the two-grid approximation:

$$(3.19) \quad \begin{aligned} |\lambda^h - \lambda| &= \left| \frac{a_h(u^h, u^h)}{b_h(u^h, u^h)} - \lambda \right| \\ &\leq |a(\widetilde{u}^h, \widetilde{u}^h) - \lambda| + |a_h(u^h, u^h) - a(\widetilde{u}^h, \widetilde{u}^h)| \\ &\leq |a(\widetilde{u}^h - u, \widetilde{u}^h - u) - \lambda b(\widetilde{u}^h - u, \widetilde{u}^h - u)| + Ch^2 \\ &\leq |\widetilde{u}^h - u|_{H^1(\Gamma)}^2 + \lambda \|\widetilde{u}^h - u\|_{L^2(\Gamma)}^2 + Ch^2 \\ &\leq C(\lambda) |\widetilde{u}^h - u|_{H^1(\Gamma)}^2 + Ch^2 \leq C(\lambda)(h^2 + H^8). \end{aligned} \quad \square$$

Remark 3.6. Theorem 3.5 implies that if the mesh sizes are chosen such that $H \lesssim h^{1/4}$ between neighboring levels, then the optimal linear rate of convergence for the eigenfunction in $|\cdot|_{H^1(\Gamma)}$ and the quadratic convergence for the eigenvalue follow. In our setting, assuming multiple levels of meshes (obtained by uniformly refining the mesh from the previous level) and that we project the vertices onto the surface, $H \lesssim h^{1/4}$ holds and the estimate follows. Assume Algorithm 3.1 is applied in a cascading fashion spanning multiple levels; then the optimal convergence rates of these two algorithms depend on the assumption that the coarse mesh is fine enough, that is, they depend on the assumption that the geometric error is sufficiently small.

4. The bootstrap multigrid method. In this section, we propose a finite element BMG eigensolver based on the BAMG framework [8]. The key ingredient in the bootstrap approach proposed in this section is the idea to continuously enrich the coarse space with computed eigenfunction approximations coming from linear solves on finer meshes.

The motivation of the eigensolve in an enriched coarse space is to overcome the drawbacks of the standard two-grid method. Essentially, the two-grid methods proposed, when using the original geometrically defined coarse space, accelerate the eigensolve on the finest mesh. However, the number of correctly approximated eigenpairs by these two-grid methods depends on the dimension of the coarse space (see the numerical example in section 5.1). With the BMG eigensolver, the desired eigenpairs of the original Laplace–Beltrami operator can be approximated with the same order of accuracy as the direct eigensolve achieves on the finest mesh, assuming certain mesh size relations are satisfied between consecutive levels (see Remark 3.6).

Algorithm 4.1. A two-grid BMG scheme for approximating eigenfunctions near μ using an enriched coarse space.

1: Coarse grid eigensolve

Set a coarse-grid shift $\mu_H = \mu \geq 0$, find $(u_{H,i}, \lambda_{H,i}) \in \mathcal{V}_H \times \mathbb{R}^+$ (for $i = 1, \dots, N$, using direct eigensolve, where $N \leq \dim \mathcal{V}_H$) satisfies

$$(4.1) \quad a_H(u_{H,i}, v) - \mu_H b_H(u_{H,i}, v) = \lambda_{H,i} b_H(u_{H,i}, v) \quad \text{for any } v \in \mathcal{V}_H.$$

2: Choose eigenfunctions for the enrichment

Let the index set for the enrichment candidate eigenfunctions be $\Lambda \subset \{1, \dots, N\}$, where if $i \in \Lambda$, $|\lambda_{H,i} - \mu| < \text{tol}$. Let

$$(4.2) \quad X_H := \text{span}\{u_{H,i}\}_{i \in \Lambda}.$$

3: Fine grid source approximation

Refine $\mathcal{T}_H$ and perform the geometric projection to get $\mathcal{T}_h$. Approximate $u^{h,i} \in \mathcal{V}_h$, where $i \in \Lambda$, using $u_{H,i} \in X_H$ as the source, in

$$(4.3) \quad a_h(u^{h,i}, v) - \mu_H b_h(u^{h,i}, v) = \lambda_{H,i} b_h(\mathcal{P}_h u_{H,i}, v) \quad \text{for any } v \in \mathcal{V}_h.$$

Then orthogonalize $u^{h,i}$'s with respect to the inner product $b_h(\cdot, \cdot)$, let the enrichment space contain the orthogonalized source approximations:

$$(4.4) \quad X_h := \text{span}\{u^{h,i}\}_{i \in \Lambda}.$$

4: Coarse grid eigensolve in the enriched space

Set a new shift μ_h . Find $(u_{h,i}, \lambda_{h,i}) \in V_{H,h} \times \mathbb{R}^+$ satisfying, for $i \in \Lambda$,

$$(4.5) \quad a_{H,h}(u_{h,i}, v) - \mu_h b_{H,h}(u_{h,i}, v) = \lambda_{h,i} b_{H,h}(u_{h,i}, v) \quad \text{for any } v \in V_{H,h},$$

where $V_{H,h} := \mathcal{V}_H + X_h$ is the enriched coarse space. Update $X_H = \text{span}\{u_{h,i}\}_{i \in \Lambda}$.

4.1. The two-grid bootstrap algorithm. A two-grid bootstrap algorithm is outlined in Algorithm 4.1 and illustrated briefly in Figure 4.1(a). The algorithm takes a coarse mesh, a shift μ , and the tolerance tol as inputs.

In the two-grid bootstrap method, the bilinear forms $a_{H,h}(\cdot, \cdot)$ and $b_{H,h}(\cdot, \cdot)$ are defined as follows. Let $w \in \mathcal{V}_H + X_h$ be any function such that $w = w_H + w_h$, where $w_H \in \mathcal{V}_H$, and $w_h \in X_h$. Then, for any test function $v = v_H + v_h \in \mathcal{V}_H + X_h$

$$(4.6) \quad \begin{aligned} a_{H,h}(w, v) &:= a_H(w_H, v_H) + a_h(\mathcal{P}_h w_H, v_h) \\ &\quad + a_h(w_h, \mathcal{P}_h v_H) + a_h(w_h, v_h), \\ \text{and } b_{H,h}(w, v) &:= b_H(w_H, v_H) + b_h(\mathcal{P}_h w_H, v_h) \\ &\quad + b_h(w_h, \mathcal{P}_h v_H) + b_h(w_h, v_h). \end{aligned}$$

Now we rewrite some of the key equations in Algorithm 4.1 in their matrix forms. Problem (4.3) can be rewritten as follows: find $U^{h,i} \in \mathbb{R}^{N_h}$, $i \in \Lambda$,

$$(4.7) \quad (A_h - \mu_H M_h) U^{h,i} = \lambda_H M_h (P_h U_{H,i}).$$

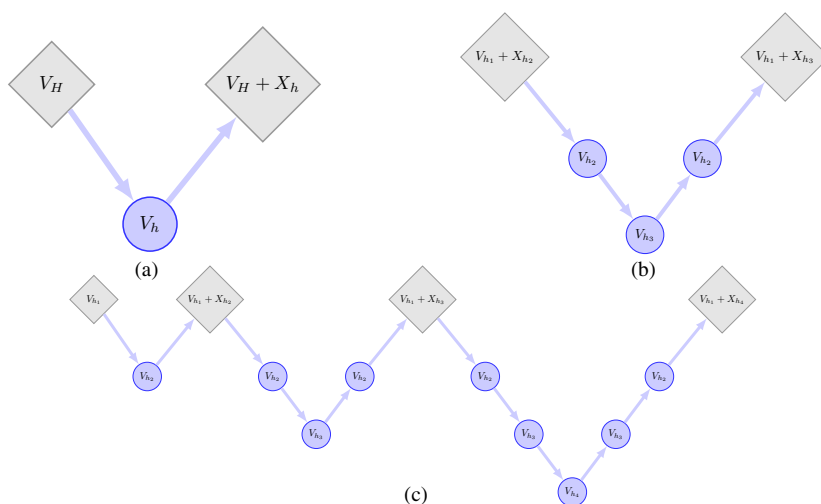


FIG. 4.1. Illustration of Algorithms 4.1, 4.2, and 4.3 in (a), (b), and (c), respectively. A gray diamond box stands for a direct eigensolve on the coarse level (with or without the enrichment), a blue circle stands for a source problem approximation (smoother/solve) on finer levels. The names in the boxes or circles stand for the finite element spaces that are used in the various steps of the algorithm. (a) A two-grid bootstrap algorithm between the coarse and fine levels. (b) A BMG V-cycle iteration between level 1 and level 3. The enrichment space X_{h_2} is updated to X_{h_3} using the approximations in V_{h_3} . (c) A bootstrap full multigrid cycle iteration between level 1 and level 4. From coarse to fine, Rayleigh quotient iteration (4.18) is performed. From fine to coarse, smoothing (4.19) is performed.

Here A_h and M_h are the stiffness matrix and mass matrix for the degrees of freedom on the fine approximation space $\mathcal{V}_h$, respectively. $U^{h,i}$ is the vector representation of $u^{h,i}$ in the canonical finite element basis, and its superscript is inherited from $u^{h,i}$. The $U_{H,i}$ with the subscript is the vector representation of the direct solve solution $u_{H,i}$ in the coarse approximation space $\mathcal{V}_H$.

Step 4 of Algorithm 4.1 is to find $U_{h,i} \in \mathbb{R}^{N_H+|\Lambda|}$, and $\lambda_{h,i} \in \mathbb{R}^+$, $i \in \Lambda$,

$$(4.8) \quad (A_{H,h} - \mu_h M_{H,h})U_{h,i} = \lambda_{h,i} M_{H,h} U_{h,i}.$$

The enriched stiffness and mass matrices $A_{H,h}$ and $M_{H,h}$ are in the following coarse-fine block form: let $U^h := (U^{h,1}, \dots, U^{h,|\Lambda|}) \in \mathbb{R}^{N_h \times |\Lambda|}$ be the block of all the approximations from problem (4.7), and

$$A_{H,h} = \begin{pmatrix} A_H & P_H A_h U^h \\ (U^h)^T A_h P_h & (U^h)^T A_h U^h \end{pmatrix}, \quad M_{H,h} = \begin{pmatrix} M_H & P_H M_h U^h \\ (U^h)^T M_h P_h & (U^h)^T M_h U^h \end{pmatrix}.$$

Remark 4.1 (choice of the shifts). The shifts $\mu_H, \mu_h \geq 0$ in Algorithm 4.1 are added in case the user is interested in a specific range of the eigenvalues. If one is to find the eigenvalues from the smallest one, the shift can be set as $\mu_H = \mu_h = 0$ for all of the enriched coarse eigenvalue problems and fine source approximation problems. To recover interior eigenvalues, the coarse-grid shift μ_H can be set to be a positive number that is near the center of an eigen-cluster of interest. Then the new shift μ_h is updated using the Rayleigh quotients computed from the fine source approximations.

The choice of the set of enrichment functions X_H (test vectors in the BAMG [8] setup) with index set Λ is related to the eigenspace of interest, as well as the

resources one can afford. As the dimension of X_H increases, the sparsity of $A_{H,h}$ deteriorates, resulting in a more expensive fine-grid source problem. Thus, we propose that the user shall choose the coarse space according to the spectrum of interest. Using approximating the Laplace–Beltrami spectrum on a closed surface as an example, consider the simple case that the user wants to recover the l th eigenpair, $(\lambda_{h,l}, u_{h,l})$, where $1 \leq l < \dim \mathcal{V}_H$. The eigensolve in the coarse space contains the discrete approximations $\{(\lambda_{H,i}, u_{H,i})\}_{i=1}^{\dim \mathcal{V}_H}$ to these eigenpairs. Then, the index set for the eigenfunction approximations Λ in (4.2) can be chosen as $\{\hat{k} \in \mathbb{Z} : l - m \leq \hat{k} \leq l\}$. Here, m is greater than or equal to the geometric multiplicity shown in the discrete spectra for the eigenvalue closest to the eigenvalue of interest, say, $\lambda_{H,l}$, and the tolerance `tol` is set to be the maximum distance between the distinctive eigenvalue clusters. Intuitively, these choices are motivated by the fact that the algorithm should “detect” the improvement in the approximations of the eigenpair $(\lambda_{h,l}, u_{h,l})$ of interest. For additional discussion of how to set shifts and how to choose the enrichment candidates, please refer to the examples in section 5.2.

THEOREM 4.2 (error estimates for BMG Algorithm 4.1). *Assume $\mu \geq 0$ is an eigenvalue of neither problem (2.9) nor problem (2.4), and suppose $u_{H,i} \in X_H$ in (4.2) for $i = 1, \dots, \dim X_H$ are linearly independent, and there exists a $u_i \in M(\lambda + \mu)$ with the estimates*

$$(4.9) \quad |\lambda - \lambda_{H,i}| \leq C(\lambda, \mu)H^2, \quad \text{and} \quad |u_i - \tilde{u}_{H,i}|_{H^1(\Gamma)} \leq C(\lambda, \mu)H,$$

$$\sup_{v \in \mathcal{V}_h/\mathbb{R}} \frac{b(u_i - \widetilde{\mathcal{P}_h u_{H,i}}, \tilde{v})}{|v|_{H^1(\Gamma_h)}} \leq C(\lambda, \mu)H |u_i - \tilde{u}_{H,i}|_{H^1(\Gamma)}.$$

After one iteration of Algorithm 4.1 with $\mu_H = \mu_h = \mu$ under a uniform refinement with h sufficiently small, and an exact solve in step 3, there exists a $\hat{u} \in M(\lambda + \mu)$ such that the resultant approximation $(\lambda_{h,i}, u_{h,i})$ has the following error estimates:

$$(4.10) \quad |\lambda - \lambda_{h,i}| \leq C(\lambda, \mu)(h^2 + H^4), \quad \text{and} \quad |\hat{u} - \tilde{u}_{h,i}|_{H^1(\Gamma)} \leq C(\lambda, \mu)(h + H^2).$$

Proof. The proof follows from a similar argument in [27, Theorem 3.4] and uses Lemma 3.3, with the geometric error estimates from Lemma 2.1. For simplicity first we consider $\dim X_H = 1$, then define a projection $\Pi_h : H^1(\Gamma) \rightarrow \mathcal{V}_h$ of a $u \in M(\lambda + \mu)$ with respect to the inner product $a_h(\cdot, \cdot)$ as follows:

$$(4.11) \quad a_h(\Pi_h u, v) = a(u, \tilde{v}), \quad \text{and} \quad \int_{\Gamma_h} \Pi_h u \, dS = \int_{\Gamma} u \, dS = 0 \quad \forall v \in \mathcal{V}_h.$$

For u^h from the exact solve of (4.3), by Lemma 3.3,

$$(4.12) \quad c |u^h - \Pi_h u|_{H^1(\Gamma_h)} \leq \sup_{v \in \mathcal{V}_h} \frac{|a_h(u^h - \Pi_h u, v) - \mu b_h(u^h - \Pi_h u, v)|}{|v|_{H^1(\Gamma_h)}}.$$

The numerator on the right-hand side above becomes

$$\begin{aligned} & |a_h(u^h - \Pi_h u, v) - \mu b_h(u^h - \Pi_h u, v)| \\ &= |a_h(u^h, v) - \mu b_h(u^h, v) - \underbrace{(a_h(\Pi_h u, v) - \mu b_h(\Pi_h u, v))}_{=:(\mathfrak{d})}|. \end{aligned}$$

The first difference $a_h(u^h, v) - \mu b_h(u^h, v) = \lambda_H b_h(\mathcal{P}_h u_H, v)$. For $(\mathfrak{d})$ by (4.11) and $a(u, \tilde{v}) - \mu b(u, \tilde{v}) = \lambda b(u, \tilde{v})$ we can write it as

$$\begin{aligned} (\mathfrak{d}) &= a(u, \tilde{v}) - \mu b_h(\Pi_h u, v) \\ &= \lambda b(u, \tilde{v}) + \mu b(u, \tilde{v}) - \mu b(\widetilde{\Pi_h u}, \tilde{v}) + \mu b(\widetilde{\Pi_h u}, \tilde{v}) - \mu b_h(\Pi_h u, v). \end{aligned}$$

As a result, by the triangle inequality we have

$$(4.13) \quad \begin{aligned} |a_h(u^h - \Pi_h u, v) - \mu b_h(u^h - \Pi_h u, v)| &\leq \underbrace{|\lambda_H b_h(\mathcal{P}_h u_H, v) - \lambda b(u, \tilde{v})|}_{=:(I)} \\ &+ \underbrace{|\mu b(u, \tilde{v}) - \mu b(\widetilde{\Pi_h u}, \tilde{v})|}_{=:(II)} + \underbrace{|\mu b(\widetilde{\Pi_h u}, \tilde{v}) - \mu b_h(\Pi_h u, v)|}_{=:(III)}. \end{aligned}$$

Next we show the estimates for each term on the right of (4.13). For (I) we have

$$(4.14) \quad \begin{aligned} (I) &= |\lambda_H b_h(\mathcal{P}_h u_H, v) - \lambda b(u, \tilde{v})| \\ &\leq |b(\lambda_H \widetilde{\mathcal{P}_h u_H} - \lambda u, \tilde{v})| + |b_h(\lambda_H \mathcal{P}_h u_H, v) - b(\lambda_H \widetilde{\mathcal{P}_h u_H}, \tilde{v})|. \end{aligned}$$

For the second term above in (4.14) which is introduced by approximating the geometry, by letting $q = u^h - \Pi_h u \in \mathcal{V}_h$ below, applying [21, Lemma 4.7], a Poincaré inequality on Γ_h since $\int_{\Gamma_h} (u^h - \Pi_h u) dS = 0$ [32, Remark 5.3], and discarding higher order terms, we have

$$(4.15) \quad \begin{aligned} &|b_h(\lambda_H \mathcal{P}_h u_H, v) - b(\lambda_H \widetilde{\mathcal{P}_h u_H}, \tilde{v})| \\ &\leq \lambda_H \sup_{q \in \mathcal{V}_h / \mathbb{R}} \frac{|b_h(\lambda_H \mathcal{P}_h u_H, q) - b(\lambda_H \widetilde{\mathcal{P}_h u_H}, \tilde{q})|}{|q|_{H^1(\Gamma_h)}} |v|_{H^1(\Gamma_h)} \\ &\leq ch^2 \lambda_H \frac{\|\mathcal{P}_h u_H\|_{L^2(\Gamma_h)} \|q\|_{L^2(\Gamma_h)}}{|q|_{H^1(\Gamma_h)}} |v|_{H^1(\Gamma_h)} \leq ch^2 \lambda_H \|\mathcal{P}_h u_H\|_{L^2(\Gamma_h)} |v|_{H^1(\Gamma_h)}. \end{aligned}$$

For the first term in (4.14), applying the geometric error estimate in Lemma 2.1 such that $|\tilde{v}|_{H^1(\Gamma)} \leq (1 + ch^2) |v|_{H^1(\Gamma_h)}$, we have

$$(4.16) \quad \begin{aligned} b(\lambda_H \widetilde{\mathcal{P}_h u_H} - \lambda u, \tilde{v}) &\leq (1 + ch^2) \sup_{q \in \mathcal{V}_h / \mathbb{R}} \frac{b(\lambda_H \widetilde{\mathcal{P}_h u_H} - \lambda u, \tilde{q})}{|q|_{H^1(\Gamma_h)}} |v|_{H^1(\Gamma_h)} \\ &\leq (1 + ch^2) \left(|\lambda - \lambda_H| \sup_{q \in \mathcal{V}_h / \mathbb{R}} \frac{b(\widetilde{\mathcal{P}_h u_H}, \tilde{q})}{|q|_{H^1(\Gamma_h)}} + \lambda \sup_{q \in \mathcal{V}_h / \mathbb{R}} \frac{b(\widetilde{\mathcal{P}_h u_H} - u, \tilde{q})}{|q|_{H^1(\Gamma_h)}} \right) |v|_{H^1(\Gamma_h)}. \end{aligned}$$

Exploiting the technique in (4.15) again, and combining the estimates above with the assumption in (4.10), we have reached (I) $\leq C(1 + ch^2)^2 H^2 |v|_{H^1(\Gamma_h)}$. For (II), we have

$$(II) := |\mu b(u, \tilde{v}) - \mu b(\widetilde{\Pi_h u}, \tilde{v})| \leq \mu \|u - \widetilde{\Pi_h u}\|_{L^2(\Gamma_h)} \|v\|_{L^2(\Gamma_h)}.$$

We choose $v = u^h - \Pi_h u \in \mathcal{V}_h$ here; thus the Poincaré inequality can be used and $\|v\|_{L^2(\Gamma_h)} \lesssim |v|_{H^1(\Gamma_h)}$. Since the lifting operator does not preserve the integration, the Poincaré inequality (see, e.g., [22, Lemma B.63]) has to be applied on the first term as follows:

$$c \|u - \widetilde{\Pi_h u}\|_{L^2(\Gamma_h)} \leq |u - \widetilde{\Pi_h u}|_{H^1(\Gamma)} + |\Gamma|^{-1} \int_{\Gamma} \widetilde{\Pi_h u} dS.$$

A standard error estimate for the projection $|u - \widetilde{\Pi_h u}|_{H^1(\Gamma)} \lesssim h |\nabla_{\Gamma} u|_{H^1(\Gamma)}$ (e.g., [18, Lemma 2.2]) can be applied to the first term above. The second term above can be estimated by the fact that $\int_{\Gamma_h} \Pi_h u dS = 0$,

$$\int_{\Gamma} \widetilde{\Pi_h u} dS = b(\widetilde{\Pi_h u}, 1) - b_h(\Pi_h u, 1) \leq ch^2 |\Gamma|^{1/2} \|\Pi_h u\|_{L^2(\Gamma_h)}.$$

As a result, we have (II) $\leq C\mu(h + h^2) |v|_{H^1(\Gamma_h)}$. Last, for (III) [21, Lemma 4.7] is applied again as in (4.15) and we have (III) $\leq C\mu h^2 |v|_{H^1(\Gamma_h)}$. Applying the triangle inequality and discarding the higher order term yield

$$|u - \widetilde{u^h}|_{H^1(\Gamma)} \leq |u - \widetilde{\Pi_h u}|_{H^1(\Gamma)} + |\widetilde{\Pi_h u} - \widetilde{u^h}|_{H^1(\Gamma)} \leq C(\lambda, \mu)(h + H^2).$$

When $\dim X_H > 1$, the argument above still applies by [4, Theorem 4.1] since $u^{h,i}$'s are orthogonal with respect to $b_h(\cdot, \cdot)$. Last, we apply the eigenpair approximation results with multiplicity ≥ 1 results in [3, (8.47b), p. 700, and Theorem 9.1], and for each i , we have for a $\widehat{u} \in M(\lambda)$, $\mathcal{V}_{H,h} := \mathcal{V}_H + \text{span}\{u^{h,i}\}$

$$|\widehat{u} - \widetilde{u}_{h,i}|_{H^1(\Gamma)} \leq \inf_{w \in M(\lambda)} \inf_{v \in \mathcal{V}_{H,h}} |w - \widetilde{v}|_{H^1(\Gamma)} \leq |u_i - \widetilde{u^{h,i}}|_{H^1(\Gamma)},$$

which, together with the estimate above, gives the first estimate in (4.9). Now using the same argument as (3.19) yields the eigenvalue estimate in (4.10). $\square$

Remark 4.3 (smoothers versus solve in (4.3)). For simplicity of presentation, we have assumed that (4.3) is solved exactly in the proof of Theorem 4.2. Nevertheless, applying similar estimates for the Gauss–Seidel smoother using, e.g., [35, Lemma 3.9], the estimate in (4.10) can be shown to yield the same order error under appropriate choice of smoothers.

4.2. The bootstrap multigrid cycle. The V-cycle variant of Algorithm 4.1 is given by incorporating this two-grid method into a multilevel setting, as we outline in Algorithm 4.2 and illustrate briefly using three levels in Figure 4.1(b). The algorithm here is a simplified variant of the BAMG approach presented in [8] since the coarse-level systems and the restriction and interpolation operators can be defined using the finite element formulation at hand; in BAMG the restriction and interpolation operators are defined algebraically and the coarse-level system is computed using the Galerkin definition.

For the geometric BMG algorithm in this paper, we construct an “almost” hierarchical sequence of finite element spaces that are geometrically projected to the finest one, with a total number of K meshes: $\widehat{V}_{h_1} \subset \widehat{V}_{h_2} \subset \cdots \subset \widehat{V}_{h_{K-1}} \subset V_{h_K}$. The meshes are obtained using the surface mesh refinement procedure in section 2, and then finite element spaces on the first through $(K - 1)$ th levels are projected onto the K th level by recursively using the projection in (3.1). We note that the coarsest space has been enriched by approximations from all finer spaces at the end of one BMG V-cycle.

In contrast to the conventional fine-coarse-fine multigrid V-cycle for a source problem, the BMG V-cycle (Algorithm 4.2) poses itself as an “inverted” V-cycle as the relaxations process as coarse-fine-coarse. In Figure 4.1(b), the first V_{h_2} node corresponds to the presmoothing stage of the conventional multigrid V-cycle. The Rayleigh quotient iteration problem (4.18) is approximated in V_{h_2} using a smoother. Starting from the V_{h_3} node, together with the second V_{h_2} node, a smoother for the source problem (4.19) is applied on each level, resembling the postsmoothing stage of the conventional multigrid V-cycle. The diamond boxes represent the direct eigensolve in the space $V_{h_1} + X_{h_i}$ ($i = 2, 3$).

Algorithm 4.2. A BMG V-cycle for approximating eigenpairs between level 1 and level $k \geq 2$. *Input:* $(u_{h_{k-1},i}, \mu_{h_{k-1}}, \lambda_{h_{k-1},i})$, *Output:* $(u_{h_k,i}, \mu_{h_k}, \lambda_{h_k,i})$.

1: Coarse grid eigensolve in an enriched space

For a fixed shift $\mu_{h_{k-1}}$, perform a direct eigensolve for the coarse-grid eigenpairs $(u_{h_{k-1},i}, \lambda_{h_{k-1},i}) \in V_{h_1, h_{k-1}} \times \mathbb{R}^+$ for $i \in \Lambda$, where $V_{h_1, h_{k-1}} = V_{h_1} + X_{h_{k-1}}$, and Λ is index set of $X_{h_{k-1}}$ on the $(k-1)$ th level, satisfying, for any $v \in V_{h_1, h_{k-1}}$,

$$(4.17) \quad a_{h_1, h_{k-1}}(u_{h_{k-1},i}, v) - \mu_{h_{k-1}} b_{h_1, h_{k-1}}(u_{h_{k-1},i}, v) = \lambda_{h_{k-1},i} b_{h_1, h_{k-1}}(u_{h_{k-1},i}, v).$$

2: Rayleigh quotient iteration

For $s = 2, \dots, k-1$, approximate for $u^{h_s, i}$ in the following source problem on level s : for any $v \in V_{h_s}$

$$(4.18) \quad a_{h_s}(u^{h_s, i}, v) - \mu_{h_{k-1}} b_{h_s}(u^{h_s, i}, v) = \lambda_{h_{k-1},i} b_{h_s, h_{k-1}}(u^{h_{s-1}, i}, v),$$

where $u^{h_1, i} := u_{h_{k-1},i} \in V_{h_1} + X_{h_{k-1}}$.

3: Smoothing of fine-grid auxiliary source problem

Apply a smoother for $u^{h_k, i} \in V_{h_k}$ from level k back to level 1 for the following problem defined on the finest grid: for any $v \in V_{h_k}$

$$(4.19) \quad a_{h_k}(u^{h_k, i}, v) - \mu_{h_{k-1}} b_{h_k}(u^{h_k, i}, v) = \lambda_{h_{k-1},i} b_{h_k}(\mathcal{P}_{h_k} u^{h_{k-1}, i}, v).$$

Then orthogonalize $u^{h_k, i}$'s with respect to the inner product $b_{h_k}(\cdot, \cdot)$.

4: Coarse shift and enrichment space updates

Update $X_{h_k} = \text{span}\{u^{h_k, i}\}_{i \in \Lambda}$. Update the coarse-grid shift $\mu_{h_{k-1}}$ from the previous cycle to μ_{h_k} based on the Rayleigh quotients of the eigenfunctions in X_{h_k} .

5: Coarse grid eigensolve in the updated enriched space and shift

Find $(u_{h_k}, \lambda_{h_k}) \in V_{h_1, h_k} \times \mathbb{R}^+$ satisfying, for $i \in \Lambda$, and any $v \in V_{h_1, h_k}$,

$$(4.20) \quad a_{h_1, h_k}(u_{h_k, i}, v) - \mu_{h_k} b_{h_1, h_k}(u_{h_k, i}, v) = \lambda_{h_k, i} b_{h_1, h_k}(u_{h_k, i}, v),$$

where $V_{h_1, h_k} = V_{h_1} + X_{h_k}$ is the updated enriched coarse space.

Finally, we present the bootstrap full multigrid (BFMG) method in Algorithm 4.3. The transition from the BMG V-cycle (Algorithm 4.2) to the BFMG (Algorithm 4.3) resembles the conventional full geometric multigrid cycle that applies V-cycles in an incremental fashion, in terms of the levels (meshes) involved. On the coarsest mesh, the eigenvalue problem is directly solved and then the eigenpairs of interest form the right-hand sides of the source problems that are approximated on the finer meshes. The approximated solutions obtained from the source problems are then used to enrich the coarse space. Overall, the algorithm is continuously improving the eigenpair approximations by working mainly on the coarsest level. The algorithm is again illustrated briefly using four levels in Figure 4.1(c).

For the BFMG algorithm (Algorithm 4.3), we remark that the final output on the K th level, which approximates a certain eigenpair of interest, is given by the source problem approximation $(u_{h_K}, \lambda_{h_K}) := (u^{h_K}, \lambda^{h_K})$. The effect of a single BMG V-cycle for this approximation resembles the two-grid method in Algorithm 3.1 in

Algorithm 4.3. BFMG scheme for approximating eigenpairs over K levels.

1: **Coarse eigensolve**

Let $(u_{h_1,i}, \lambda_{h_1,i}) \in V_{h_1} \times \mathbb{R}^+$ ($i \in \Lambda \subset \{1, \dots, \dim V_{h_1}\}$) perform a direct eigensolve for the following problem:

$$a(u_{h_1,i}, v) - \mu_{h_1} b(u_{h_1,i}, v) = \lambda_{h_1,i} b(u_{h_1,i}, v) \quad \text{for any } v \in V_{h_1}.$$

2: **V-cycle iteration**

Set a certain level $k > 1$, perform the V-cycle iteration as in Algorithm 4.2 between level 1 and k .

3: **Rayleigh quotient**

If $k < K$, $k \leftarrow k + 1$. If $k = K$, compute the eigenvalue approximation on the finest level for all $i \in \Lambda$ using the source approximations.

that a direct eigensolve is applied in the (enriched) coarse space and, then, the source approximation is computed on the finer spaces.

Otherwise, if the aim is to recover the entire spectrum, then the mesh can be continuously refined, in which case the final output from the BFMG algorithm (Algorithm 4.3) converges to the true eigenpairs of problem (1.3).

5. Numerical experiments. In this section, we report various results from the finite element approximation of the eigenvalue problem (1.3) on a 2-sphere $\mathbb{S}^2$. The numerical experiments in this section are carried out using the finite element toolbox *iFEM* [17]. The source codes for these experiments are publicly available at <http://github.com/lyc102/ifem/tree/master/research>. The initial coarse mesh is generated by *Distmesh* [30].

In all tests (of both the two-grid methods and the BFMG from Algorithm 4.3), the finer meshes are obtained from a uniform refinement of the coarser mesh. The mesh sizes satisfy that $h_{k-1}^4 \lesssim h_k$ between the coarser mesh at the $(k-1)$ th level and the finer mesh at the k th level. The newly created vertices are then projected onto the continuous surface.

In all the BMG approaches, the dimension of the enrichment space on the coarsest level $\dim X_h$ is fixed unless explicitly stated otherwise. This dimension is usually set as the multiplicity of the largest possible eigenvalue being computed plus some additional overlap with its neighboring eigenvalues in the discrete spectra. We note that in the first two subsections, unless specifically stated otherwise the source problems on the finer levels are solved using a direct method.

The true solutions to the eigenvalue problem (1.1) of the Laplace–Beltrami operator on the 2-sphere are known as the real spherical harmonics (e.g., see [24]). Specifically, the j th eigenvalues, for $l^2 \leq j \leq (l+1)^2 - 1$, counting multiplicity, are $\lambda_j = l(l+1)$ for $l \in \mathbb{Z}^+$. The dimension of the associated eigenspace to the l th distinctive eigenvalue is $2l+1$.

In computation, due to the numbering of the vertices in the triangulation, the eigenfunctions obtained approximate a rotated version of the spherical harmonics represented using Cartesian coordinates. For this reason, we use an a posteriori error estimator to give the error estimate of the eigenfunctions under an $H^1(\Gamma)$ -seminorm. The error estimator we use is a combination of the one in [18] for a noneigenvalue problem and the one from [19] for the eigenvalue problem on a polygonal domain. The local error estimator for a surface triangle $T \in \mathcal{T}_h$ is defined as follows:

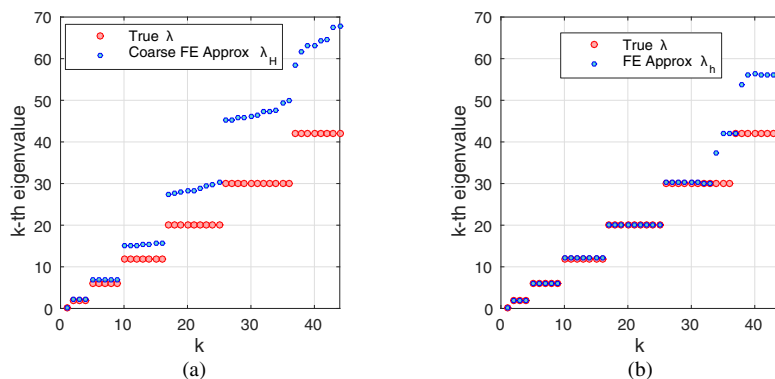


FIG. 5.1. Coarse $\#(\text{DoF}_H) = 54$. The two-grid method results in the approximated eigenspace dimension being less than intended. Blue dots are the Rayleigh quotients computed by the two-grid finite element approximations. Red dots are true eigenvalues. (a) Coarse grid approximation λ_H 's versus the true λ . (b) Two-grid approximation λ_h 's after five levels versus the true λ .

$$\eta_T^2 = h_T^2 \lambda_h^2 \|u_h\|_{L^2(T)}^2 + \frac{1}{2} \sum_{e \in \partial T} h_e \left\| [\nabla_\Gamma u_h \cdot (\mathbf{n}_T \times \boldsymbol{\tau}_e)]_e \right\|_{L^2(e)}^2 + \|\mathbf{B}_h \nabla_\Gamma u_h\|_{L^2(T)}^2. \quad (5.1)$$

For notation please refer to [18, 6, 5]. The global error estimator is obtained by $\eta(u_h) := (\sum_{T \in \mathcal{T}_h} \eta_T^2)^{1/2}$.

5.1. Standard two-grid eigensolver. To begin we apply the two-grid method from Algorithm 3.1 with shift $\mu = 0$ and note that the indefinite source problem on the finer grid is solved directly. Our aim here is to illustrate the spectra loss phenomenon that is caused by a coarse space being too small. To this end, the coarsest mesh that approximates the sphere is set to have only 54 nodes, i.e., $\#(\text{DoF}_H) = 54$. The coarsest spectrum computed by `eigs` can be viewed in Figure 5.1(a). We can observe that the true $\lambda_j = 30$ for $26 \leq j \leq 36$, and the dimension of the eigenspace for $\lambda_j = 30$ is 11. Further, we see that the numerical approximations to the higher end of the spectrum have large errors, that is, we see that the approximated $\lambda_{H,j}$ on the coarse mesh are closer to the next eigenvalue 42 in the spectra than the true eigenvalue 30 that they are supposed to approximate. This causes part of the eigenpairs from the coarse eigenspace $M(\lambda_{H,j})$ for $26 \leq j \leq 36$ to converge to eigenpairs of the true $\lambda_{j'} = 42$ for $37 \leq j' \leq 49$. As a result, as seen in Figure 5.1(b) $\dim(M(\lambda_{h,j})) = 8 < \dim(M(\lambda_j)) = 11$, where $M(\lambda_{h,j})$ is the finite element approximated eigenspace associated with $\lambda_{h,j} \approx 30$.

Note that the spectrum loss becomes even more severe for the approximation of the eigenspace for the true eigenvalue $\lambda_j = 42$ ($37 \leq j \leq 49$). In this case, if we use the same coarse mesh only four eigenfunctions are recovered in the fine space (see Figure 5.1(b)).

When a certain eigenpair is “lost” in the two-grid approximation scheme, a finer coarse mesh can be used in order to recover an improved approximation. This is because on a finer coarse mesh the wavelength of these eigenfunctions can be resolved. As an example consider the case where the coarse mesh $\#(\text{DoF}_H) = 54$ and the 34th eigenvalue ($\lambda_{34} = 30$) is wrongly approximated by the two-grid method (see Figure 5.1(b)). If a finer coarse mesh is used instead with $\#(\text{DoF}_H) = 96$, then the 34th eigenvalue is recovered. A comparison of this case is provided in Figure 5.2.

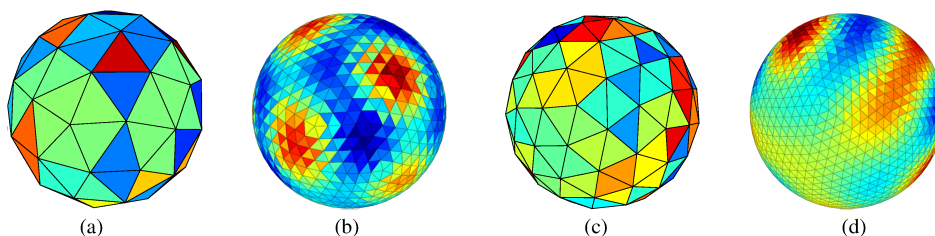


FIG. 5.2. Two-grid method, comparison of finite element approximations of one eigenfunction associated with eigenvalue $\lambda_{34} = 30$ starting from different coarse meshes. (a) $\#DoF = 54$, u_H with $\lambda_{H,34} \approx 47.5871$. (b) After three levels of refinement, $\lambda_{h,34} \approx 397353$, u_h becomes an approximation (both visually and in terms of the Rayleigh quotient) to an eigenfunction associated with $\lambda_j = 42$ ($37 \leq j \leq 49$). (c) $\#DoF = 96$, u_H with $\lambda_{H,34} \approx 40.8393$. (d) After three levels of refinement, $\lambda_{h,34} \approx 30.6344$, correct approximation u_h to an eigenfunction associated with $\lambda_j = 30$ ($26 \leq j \leq 36$).

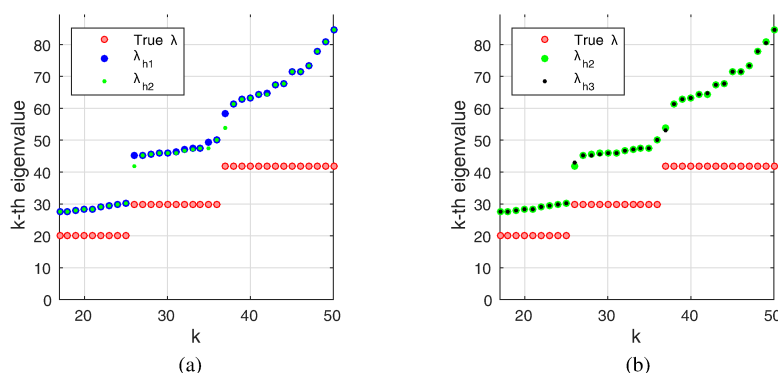


FIG. 5.3. Coarse $\#(DoF_H) = 54$, single enrichment multigrid method results in the eigenpair of interest not being represented in the discrete spectrum. Red dots are true eigenvalues. (a) Coarse eigensolve approximations λ_{h1} , and λ_{h2} obtained from solving a coarse eigenvalue problem in an enriched space $V_{h1} + \{\tilde{u}_{h2}\}$, versus the true λ . (b) λ_{h2} obtained from $V_{h1} + \{\tilde{u}_{h2}\}$, and λ_{h3} obtained from $V_{h1} + \{\tilde{u}_{h3}\}$, versus the true λ .

5.2. Bootstrap multigrid enrichment. An alternative to simply increasing the coarse space by refining the mesh is given by the bootstrap idea to enrich the coarse space with eigenfunctions obtained by approximating source problems on finer meshes as in Algorithm 4.3.

If the dimension of the coarsest enriched space is fixed such that $\dim \mathcal{V}_H + 1$ in all the V-cycles (as in the method from [27]), then approximating larger eigenvalues with multiplicity greater than one still requires a finer coarse space.

To see this we consider the case where the user wants to recover the eigenfunctions that correspond to $\lambda_j = 42$, where $37 \leq j \leq 49$. In this example, the coarsest space $\mathcal{V}_H = V_{h1}$ is chosen as in the previous example where $\#(DoF_H) = 54$. We set the coarse space being enriched as $V_{h1,h2} := V_{h1} + \{u^{37,h2}\}$ and solve the coarse eigenvalue problem using a direct eigensolve `eigs` in MATLAB. The results in Figure 5.3(a) show that the new approximation $\lambda_{h2,37}$ does approximate the 37th eigenvalue to some extent. However, when we perform this procedure again and then solve the coarse eigenvalue problem in the updated enriched space $V_{h1,h3} := V_{h1} + \{u^{37,h3}\}$,

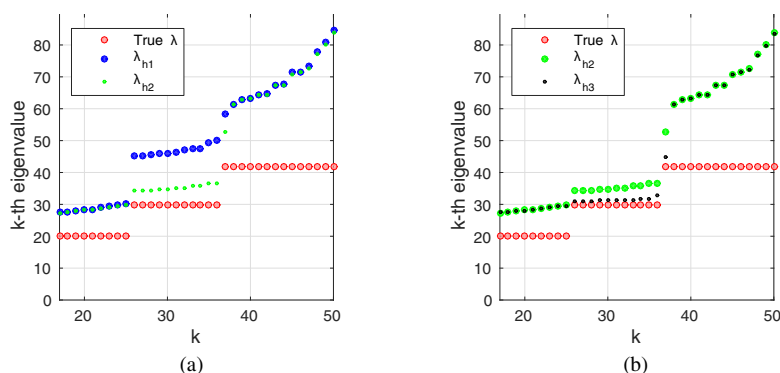


FIG. 5.4. Coarse $\#(\text{DoF}_H) = 54$. The enrichment space is chosen according to the placement of the eigenpair of interest on the discrete spectra. Red dots are true eigenvalues. (a) Coarse-grid eigensolve approximations λ_{h_1} 's and λ_{h_2} 's obtained from the eigensolve in the enriched space $V_{h_1} + \text{span}\{u^{h_2,j'}\}_{26 \leq j' \leq 37}$, versus the true λ . (b) Enriched coarse eigensolve approximations λ_{h_2} 's obtained from $V_{h_1} + \text{span}\{u^{h_2,j'}\}_{26 \leq j' \leq 37}$, and λ_{h_3} 's obtained from $V_{h_1} + \text{span}\{u^{h_3,j'}\}_{26 \leq j' \leq 37}$, versus the true λ .

we observe that the approximation to the eigenpair of interest does not improve (see Figure 5.3(b)).

The reason for this behavior can be explained as follows. First, we see that the approximations to $\lambda_{h_1,j}$ ($26 \leq j \leq 36$) are closer to the true eigenvalue $\lambda_j = 42$ ($37 \leq j \leq 49$) than the corresponding discrete spectra $\lambda_{h_1,j}$ ($37 \leq j \leq 49$). As the coarse space is enriched by a single function $\{u^{h_2,37}\}$, which is obtained by solving a single source problem in V_{h_k} ($k \geq 2$) on the k th level, the new approximation $\lambda_{h_k,37}$ becomes closer to the true eigenvalue $\lambda_{37} = 42$ than it is supposed to approximate. However, it is still not as good an approximation as $\lambda_{h_1,j}$ ($26 \leq j \leq 36$) to the true eigenvalue $\lambda_{37} = 42$. As such, the algorithm mixes these modes and then cannot detect the eigenpair of interest.

Heuristically speaking, for the Laplace–Beltrami eigenvalue problem on $\mathbb{S}^2$, assuming a priori knowledge of the dimension of the eigenspace, it follows that if we seek to approximate the l th distinctive eigenvalue $\lambda_j = l(l+1)$, then the coarse space V_{h_1} should at least be enriched by the subspace $X_{h_k} = \text{span}\{u^{h_k,j'}\}_{j' \in \Lambda}$, where $\Lambda = \{j' \in \mathbb{Z} : (l-1)^2 + 1 \leq j' \leq j\}$. In this example, for $\lambda_{37} = 42$ where $l = 7$, the enrichment space X_{h_k} is $\text{span}\{u^{h_k,j'}\}_{26 \leq j' \leq 37}$, where the $u^{h_k,j'}$'s denote the solutions to the source problems in the finer space V_{h_k} on the k th level ($k \geq 2$) for the 26th to 37th eigenpairs of the discrete spectra. The comparison can be found in Figures 5.4(a) and 5.4(b). Now, because the discrete eigenvalues $\lambda_{h_k,j}$ ($26 \leq j \leq 36$) are better approximated after the coarse space is enriched with multiple eigenfunctions, the algorithm is able to better detect the eigenpair of interest $\lambda_{h_k,j}$ ($j = 37$, $k = 2, 3$) after the second coarse eigensolve, and overall we see that the approximations improve.

If one is interested in recovering all the eigenvalues from the smallest one, then Algorithm 4.2 can be applied with an index-fixed X_{h_k} and for each distinctive eigenvalue one at a time. Here we choose a fixed dimension 20 (greater than the biggest numerical multiplicity observed from the coarse eigensolve). The 20 enrichment candidate functions are the eigenfunctions associated with the 20 eigenvalues nearest to the eigenvalue of interest on the discrete spectra, and these eigenfunctions are kept

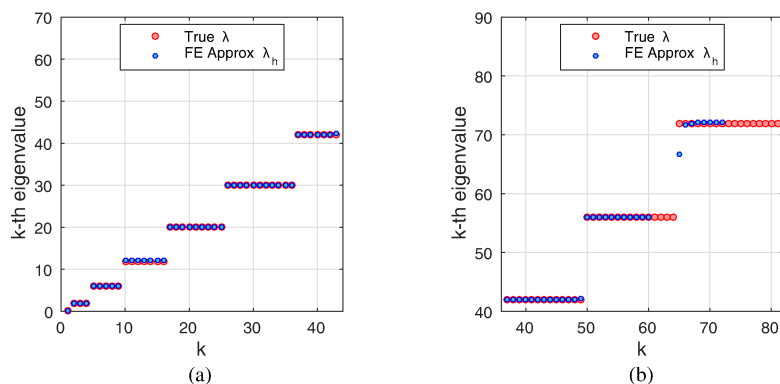


FIG. 5.5. Coarse $\#(\text{DoF}_H) = 54$. BMG with an index-fixed enrichment space of dimension 20 converges as intended (left). BMG with an index-updating enrichment space is able to detect higher eigenvalues that cannot be resolved in the coarse grid (right). Blue dots are the Rayleigh quotient computed by the multigrid finite element approximations. Red dots are true eigenvalues. (a) Multigrid approximations λ_h 's after four levels versus the true λ . (b) Multigrid approximations λ_h 's after six levels versus the true λ .

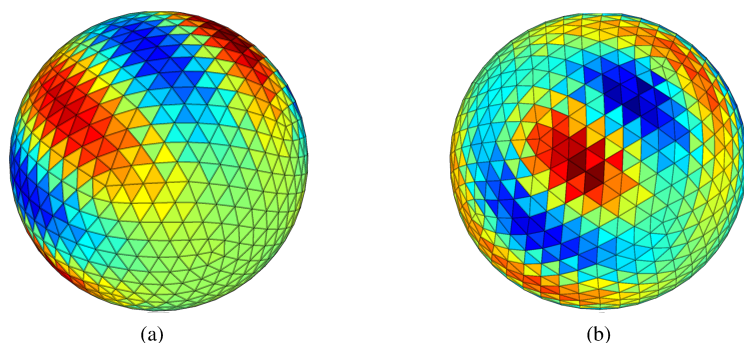


FIG. 5.6. The BMG results of the finite element approximations of the eigenfunction associated with eigenvalues $\lambda_{34} = 30$ and $\lambda_{37} = 42$ after four levels of refinement starting with $\#\text{DoF} = 54$ (figure shows solutions on the third level). For two-grid, these approximations can only be recovered correctly with coarse $\#\text{DoF} = 96$. (a) $\#\text{DoF} = 3330$, $\lambda_{h_4,34} \approx 30.3061$, correct approximation u_h to an eigenfunction associated with $\lambda_j = 30$ ($26 \leq j \leq 36$). (a) $\#\text{DoF} = 3330$, $\lambda_{h_4,42} \approx 42.4909$, correct approximation u_h to an eigenfunction associated with $\lambda_j = 42$ ($37 \leq j \leq 49$).

through the BFMG cycle in X_{h_k} (level $k = 2, 3, 4$). The discrete spectra recovered using this setting of BFMG (Algorithm 4.3) can be found in Figures 5.5(a) and 5.6.

The other way of improving the eigenvalue approximation for a certain range is to set a shift $\mu > 0$ in Algorithm 4.3 to achieve a “zoom in” effect. In Figure 5.7(a), a coarse-grid shift $\mu_{h_1} = 32$ is applied in the coarse solve first in Algorithm 4.1 involving two levels. The enrichment candidate space X_{h_1} is chosen to be a fixed dimension of 20, using the 20 eigenfunctions associated with the 20 eigenvalues closest to 0. After the source solves in V_{h_2} , when performing the eigensolve in an enriched coarse space $V_{h_1,h_2} = V_{h_1} + X_{h_2}$, the new coarse-grid shift μ_{h_2} is set to be the previous shift μ_{h_1} plus the average of the Rayleigh quotients of the functions in X_{h_2} . X_{h_2} contains all the source solutions from problem (4.3) using functions from X_{h_1} as the sources. The new shift μ_{h_2} is then used in the next BMG V-cycle (Algorithm 4.2) involving three levels. This averaging for the new shift is ad hoc. The reason for this choice is that

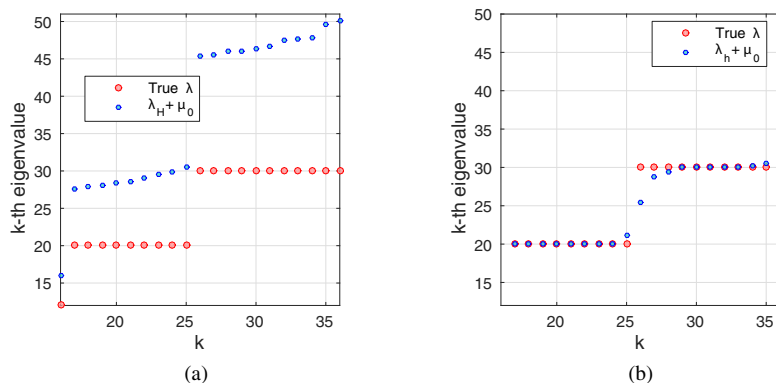


FIG 5.7 Coarse $\#(\text{DoF}_H) = 54$; both the enrichment space and the shift μ are chosen on each level based on the current approximation to the eigenpair of interest on the discrete spectra. Red dots are true eigenvalues. (a) Coarse grid approximation λ_{h_1} 's, the shift $\mu_{h_1} = 32$. (b) λ_{h_4} 's obtained from $V_{h_1} + X_{h_4}$, the shift $\mu_{h_4} = 24.9861$.

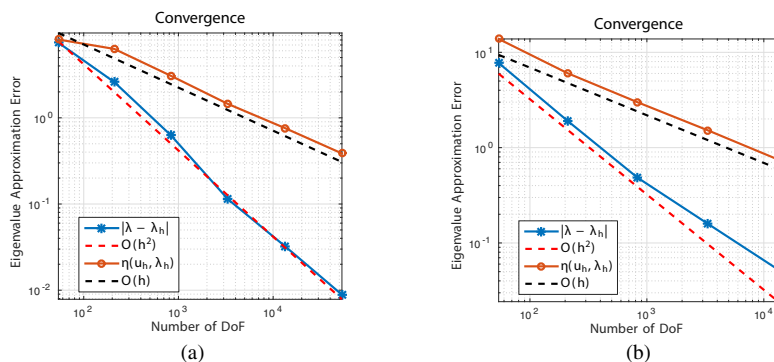


FIG. 5.8. Convergence results of BMG using the shifted problem with an updating μ to approximate interior eigenvalue $\lambda = 20$. The enrichment space's dimension is 20 where the eigenfunctions of the 20 eigenvalues nearest to 0 are used. The coarse eigensolve is performed with a shift updated after each BMG V-cycle (Algorithm 4.2). The convergence is plotted for the smallest eigenvalue among the ones which approximate $\lambda = 20$. (a) The source problem on the current level is solved by a direct solver. (b) The source problem on the current level is approximated by five sweeps of Kaczmarz relaxations.

we observe in practice that performing the BFMG (Algorithm 4.3), when applied continuously among multiple levels, brings the whole discrete spectra closer to the true spectra. Consequently, the averaging after each V-cycle is to compensate for this change. Figure 5.7(b) contains results obtained with applying this procedure twice, between level 3 and level 1 and between level 4 and level 1. The convergence after five levels of refinement can be found in Figure 5.8(a). We note that if the shift is unchanged after each V-cycle, then a similar phenomenon as in Figure 5.3(b) is observed.

Another interesting observation is that by computing the enrichment space according to the eigenpairs that we are attempting to approximate we are able to obtain improved approximations to these interior eigenvalues without having to first compute accurate approximations to the smaller eigenvalues and their corresponding eigenfunctions (see Figure 5.4(b)). Thus, the BFMG (Algorithm 4.3) approach can be used

to compute interior eigenpairs. In contrast, the approach proposed in [15] only computes multiple eigenpairs starting with the smallest. In this way, our method for the shifted Laplace–Beltrami eigenvalue problem is more general and flexible than previous methods that have been developed for the Laplace eigenvalue problem on the plane.

5.3. Solving versus relaxing the source problem. In previous works on designing multigrid eigensolvers (e.g., [27, 31]), the auxiliary problems in the correction step are solved with a direct method on the fine level. In the BMG algorithms (Algorithms 4.1, 4.2 and 4.3), the exact solve is replaced with an iterative solver (a smoother such as symmetric Gauss–Seidel, or Kaczmarz relaxations). We illustrate in the numerical experiments that a direct solve of the shifted and indefinite system is not necessary and that relaxation on the shifted source problems suffices to guarantee a nearly optimal convergence rate for the approximation of the eigenpair the algorithm produces. In the first test of this subsection, we compare the following four ways to deal with the source problem on the fine level:

- (1) Two-grid Algorithm 3.1 with shift $\mu = 0$ applied in a cascading fashion as in Remark 3.1. The shift for finer levels comes from previous levels; direct solve is applied to the shifted (indefinite) problem (3.4) on the finer levels when the algorithm is applied between neighbor levels.
- (2) Same setting as (1). For the shifted (indefinite) problem (3.4), five sweeps of the Kaczmarz smoother per level are applied using the prolongation of the approximation from previous level as an initial guess.
- (3) BFMG Algorithm 4.3 with shift $\mu = 0$. For the unshifted positive definite source problems (4.3) of BMG on the k th level, direct solve is used.
- (4) Same setting as (3). In the BMG V-cycle Algorithm 4.2, one sweep of Gauss–Seidel smoother per level is applied to problem (4.3) using the prolongation of the approximation from previous level as an initial guess.

Note that the simplification in (2) and (4) reduces the overall computational cost of the approach significantly.

To judge whether the source problems need to be approximated to machine precision as in a direct solve in (1) or (3), or some sweeps of a smoother suffice in (2) or (4), a key measure is to check the rates of convergence r when the meshes are continuously refined. The rate of convergence r (notice this is different from the rate of convergence for a linear system solver) satisfies the following:

$$(5.2) \quad |\lambda - \lambda_{h_k}| \sim O(\#(\text{DoF}_k)^{-r}) \sim O(h_k^{2r}),$$

where k stands for the numbering of the levels, and h_k is the mesh size of the k th level. λ_{h_k} is the approximation of the eigenvalues obtained either from a direct eigensolve on level k , Algorithm 3.1 applied cascadingly from level 1 through level k , or Algorithm 4.3. Note that when the mesh is continuously refined such that $H = h_1 > h_2 > \cdots > h_k > \cdots$, the optimal rate of convergence of $O(h_k^2)$ by a direct solve predicted by the a priori estimate (2.13) for the eigenvalue approximation is achieved with $r \approx 1$. Checking if r for an iterative algorithm is close to 1 in turn implies that the algorithm is convergent in terms approximating the true eigenvalues.

In Table 5.1, we compare the rate of convergence for the approximation of the first three distinct eigenvalues for methods (1) through (4) mentioned earlier, as the first three distinct eigenvalues can be recovered by all methods without the loss of spectrum phenomenon in section 5.1.

TABLE 5.1

Comparison of the rates of convergence r as in $O(\#(\text{DoF})^{-r})$ for the first three distinct eigenvalues' approximation of Laplace–Beltrami operator on $\mathbb{S}^2$. TG = two-grid.

λ	TG	TG w/ Kaczmarz	BMG	BMG w/ GS
2	1.0084	0.9656	1.0037	0.9963
6	1.0063	0.9575	1.0005	0.9764
12	1.0084	0.9731	1.0059	0.9801

TABLE 5.2

Comparison of the rates of convergence r in $O(\#(\text{DoF})^{-r})$ for $\lambda = 20$.

λ	Shifted BMG	Shifted BMG w/ Kaczmarz
20	0.9861	0.9054

First, r 's for (1) and (2) are compared in the second and third columns in Table 5.1. Then r 's for (3) and (4) are compared in the fourth and fifth columns in Table 5.1.

The experiments correspond to three choices of the eigenvalues and the results suggest that it is sufficient to solve the unshifted source problem using Gauss–Seidel and the BFMG method (Algorithm 4.3) still yields a nearly optimal rate of convergence with $r \approx 1$. In addition, the promising two-grid results obtained using the Kaczmarz iteration for the indefinite system together with the multilevel results reported in the previous section for the shifted (indefinite) Laplace–Beltrami eigenvalue problem suggest that the algorithm will also work well for symmetric indefinite problems such as the Helmholtz equation.

In the second test of this subsection, we compare the following two methods studied in the last part of section 5.2:

(5) BFMG Algorithm 4.3 with an adaptive shift μ after each BMG V-cycle. For the shifted indefinite source problems (4.19) of on the current finest level, direct solve is used.

(6) Same setting as (5). In the BMG V-cycle Algorithm 4.2, five sweeps of the Kaczmarz smoother per level are applied to problem (4.19) using the prolongation of the approximation from the previous level as an initial guess.

The r 's for (5) and (6) are compared in the second and third columns in Table 5.2. The comparison of the convergence is shown in Figures 5.8(a) and 5.8(b).

The convergence rates of the BFMG (Algorithm 4.3) with uniform refinement are further verified numerically for the Laplace–Beltrami eigenvalue problem on the sphere in Figure 5.9. These results are on par with the a priori two-grid estimates in (3.10). Note that for larger eigenvalues, e.g., $\lambda_j = 30$ (plot on the right), more degrees of freedom are needed in order to obtain the same accuracy as for smaller eigenvalues, e.g., $\lambda = 2$ (plot on the left). Of course, the resolution required to achieve high accuracy for large eigenvalues is expected to be greater since these modes are generally more oscillatory. The key difference in the BMG algorithms (Algorithms 4.1, 4.2, and 4.3) is that with the enriched coarse space it is possible to approximate larger eigenvalues without needing to continuously increase the size of the coarsest system in order to approximate larger eigenpairs, as required by the standard two-grid method (Algorithm 3.1).

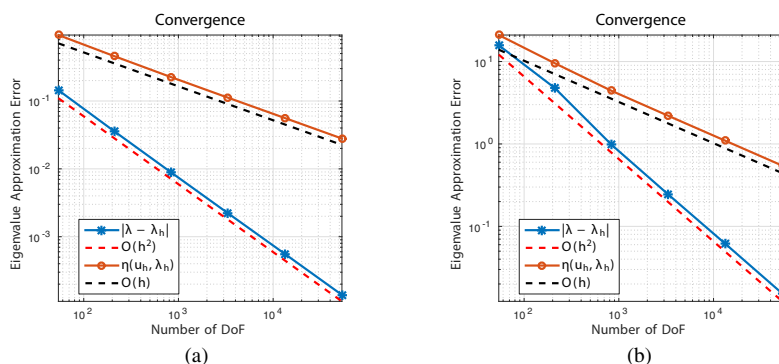


FIG. 5.9. Typical convergence results of BMG: using unshifted problem to approximate $\lambda = 2$, $\lambda = 30$. (a) $\lambda = 2$. The enrichment space's dimension is fixed to be 6 using the 6 eigenfunctions of the lowest 6 eigenvalues from the coarse eigensolve. (b) $\lambda = 30$. The enrichment space's dimension is fixed to be 15 using the 15 eigenfunctions of the 15 eigenvalues starting from the 20th eigenvalue from 0.

6. Conclusions. In this paper, we introduced a bootstrap multigrid (BMG) method for solving PDE eigenvalue problems. As we showed, the BMG algorithm is a geometric version of the BAMG setup algorithm from [8] and it provides an overall framework for the design of multigrid eigensolvers. As an example, we designed and analyzed a finite element BMG method for solving the shifted Laplace–Beltrami eigenvalue problem. Our analysis and numerical experiments focused on this model problem since it is a challenging eigenvalue problem (the eigenvalues have high multiplicity and discretization and refinement of the problem leads to a nonnested sequence of meshes in the multilevel hierarchy) and the true solutions are known, allowing us to systematically study the performance of the algorithm.

First, we extended the standard two-grid method from [25] for the Laplace eigenvalue problem on the plane to the shifted Laplace–Beltrami eigenvalue problem on a surface and we showed that the method gives an optimal $O(h^2)$ convergence, assuming that the coarsening is not too aggressive and that the coarse eigenvalue problem is solved using a direct method. We also showed that if the coarse mesh is not fine enough, then a “spectra loss” phenomena occurs and that approximating larger eigenvalues requires increasingly finer coarse meshes.

To treat this observed spectral loss in the standard two-grid algorithm, we considered the BMG method that enriches the coarse space with a *subspace* consisting of increasingly accurate approximations to the eigenpairs of interest. Moreover, we showed that these approximations can be computed by applying only a few steps of relaxation to related symmetric source problems on finer meshes to obtain reasonably good results. Moreover, our approach is able to approximate eigenpairs with large multiplicity provided that the enrichment space has sufficiently large dimension. Generally, the dimension of the coarsest space needs to depend on the dimension of the eigenspace that the user wants to compute. In addition, we showed that the bootstrap eigensolver can be used to compute a large portion of the eigenspace, starting with the smallest eigenpairs. We also showed that if instead only a few large eigenpairs are sought, then shifted indefinite systems can be solved using the BMG algorithm with the Kaczmarz smoother in order to compute interior eigenpairs directly. Finally,

we proved the convergence of our two-grid BMG approach for the shifted Laplace–Beltrami eigenvalue problem in a simplified setting, illustrating the ability of the method to directly compute interior eigenvalues.

Acknowledgment. The authors appreciate the referees for their suggested revisions of the paper.

REFERENCES

- [1] T. AUBIN, *Some Nonlinear Problems in Riemannian Geometry*, Springer Science & Business Media, New York, 2013.
- [2] I. BABUŠKA AND J. E. OSBORN, *Finite element-Galerkin approximation of the eigenvalues and eigenvectors of self-adjoint problems*, Math. Comp., 52 (1989), pp. 275–297.
- [3] I. BABUŠKA AND J. OSBORN, *Eigenvalue problems*, in Finite Element Methods (Part 1), Handb. Numer. Anal. 2, Elsevier, Amsterdam, 1991, pp. 641–787.
- [4] D. BOFFI, *Finite element approximation of eigenvalue problems*, Acta Numer., 19 (2010), pp. 1–120.
- [5] A. BONITO, A. DEMLOW, AND R. H. NOCHETTO, *Finite element methods for the Laplace–Beltrami operator*, in Handbook of Numerical Analysis, Elsevier, Amsterdam, 2020, Vol. 21, pp. 1–103.
- [6] A. BONITO, A. DEMLOW, AND J. OWEN, *A priori error estimates for finite element approximations to eigenvalues and eigenfunctions of the Laplace–Beltrami operator*, SIAM J. Numer. Anal., 56 (2018), pp. 2963–2988.
- [7] A. BRANDT, *Multiscale scientific computation: Review 2001*, in Multiscale and Multiresolution Methods: Theory and Applications, T. J. Barth, T. F. Chan, and R. Haimes, eds., Springer, Heidelberg, 2001, pp. 1–96.
- [8] A. BRANDT, J. BRANNICK, K. KAHL, AND I. LIVSHITS, *Bootstrap AMG*, SIAM J. Sci. Comput., 33 (2011), pp. 612–632.
- [9] A. BRANDT, S. MCCORMICK, AND J. RUGE, *Algebraic Multigrid (AMG) for Automatic Multigrid Solution with Application to Geodetic Computations*, Technical report, Colorado State University, Fort Collins, 1983.
- [10] J. BRANNICK, F. CAO, K. KAHL, R. D. FALGOUT, AND X. HU, *Optimal interpolation and compatible relaxation in classical algebraic multigrid*, SIAM J. Sci. Comput., 40 (2018), pp. A1473–A1493.
- [11] J. BRANNICK AND K. KAHL, *Bootstrap AMG for the Wilson Dirac system*, SIAM J. Sci. Comput., 36 (2014), pp. 321–347.
- [12] M. BREZINA, R. FALGOUT, S. MACLACHLAN, T. MANTEUFFEL, S. MCCORMICK, AND J. RUGE, *Adaptive smoothed aggregation (α SA)*, SIAM J. Sci. Comput., 25 (2004), pp. 1896–1920.
- [13] M. BREZINA, R. FALGOUT, S. MACLACHLAN, T. MANTEUFFEL, S. MCCORMICK, AND J. RUGE, *Adaptive algebraic multigrid*, SIAM J. Sci. Comput., 27 (2006), pp. 1261–1286.
- [14] Z. CAI, J. MANDEL, AND S. MCCORMICK, *Multigrid methods for nearly singular linear equations and eigenvalue problems*, SIAM J. Numer. Anal., 34 (1997), pp. 178–200.
- [15] T. F. CHAN AND I. SHARAPOV, *Subspace correction multi-level methods for elliptic eigenvalue problems*, Numer. Linear Algebra Appl., 9 (2002), pp. 1–20.
- [16] I. CHAVEL, *Eigenvalues in Riemannian Geometry*, Pure Appl. Math. 115, Academic Press, New York, 1984.
- [17] L. CHEN, *iFEM: An Integrated Finite Element Methods Package in MATLAB*, Technical report, 2009.
- [18] A. DEMLOW AND G. DZIUK, *An adaptive finite element method for the Laplace–Beltrami operator on implicitly defined surfaces*, SIAM J. Numer. Anal., 45 (2007), pp. 421–442.
- [19] R. G. DURÁN, C. PADRA, AND R. RODRÍGUEZ, *A posteriori error estimates for the finite element approximation of eigenvalue problems*, Math. Models Methods Appl. Sci., 13 (2003), pp. 1219–1229.
- [20] G. DZIUK, *Finite Elements for the Beltrami operator on arbitrary surfaces*, in Partial Differential Equations and Calculus of Variations, S. Hildebrandt and R. Leis, eds., Lecture Notes in Math. 1357, Springer, Berlin, 1988, pp. 142–155.
- [21] G. DZIUK AND C. M. ELLIOTT, *Finite element methods for surface PDEs*, Acta Numer., 22 (2013), pp. 289–396.
- [22] A. ERN AND J.-L. GUERMOND, *Theory and Practice of Finite Elements*, Appl. Math. Sci. 159, Springer Science & Business Media, New York, 2013.

- [23] X. HAN, H. XIE, AND F. XU, *A cascadic multigrid method for eigenvalue problem*, J. Comput. Math., 35 (2017), pp. 74–90.
- [24] E. W. HOBSON, *The Theory of Spherical and Ellipsoidal Harmonics*, CUP Archive, 1955.
- [25] X. HU AND X. CHENG, *Acceleration of a two-grid method for eigenvalue problems*, Math. Comp., 80 (2011), pp. 1287–1301.
- [26] M. KASCHIEV, *An iterative method for minimization of the Rayleigh Ritz functional*, Comput. Process. Syst., 6 (1988), pp. 160–170.
- [27] Q. LIN AND H. XIE, *A multi-level correction scheme for eigenvalue problems*, Math. Comp., 84 (2015), pp. 71–88.
- [28] S. MALIASSOV, *On the analog of Schwarz method for spectral problems*, Numer. Methods Math. Model., (1992), pp. 70–79.
- [29] T. MANTEUFFEL, S. MCCORMICK, M. PARK, AND J. RUGE, *Operator-based interpolation for bootstrap algebraic multigrid*, Numer. Linear Algebra Appl., 17 (2010), pp. 519–537.
- [30] P.-O. PERSSON AND G. STRANG, *A simple mesh generator in MATLAB*, SIAM Rev., 46 (2004), pp. 329–345.
- [31] M. R. RACHEVA AND A. B. ANDREEV, *Superconvergence postprocessing for eigenvalues*, Comput. Methods Appl. Math., 2 (2002), pp. 171–185.
- [32] A. REUSKEN, *Analysis of trace finite element methods for surface partial differential equations*, IMA J. Numer. Anal., 35 (2014), pp. 1568–1590.
- [33] A. REUSKEN, *Analysis of finite element methods for surface vector-laplace eigenproblems*, Math. Comp., 91 (2022), pp. 1587–1623.
- [34] R. SCHOEN AND S.-T. YAU, *Lectures on Differential Geometry*, International Press, Cambridge, MA, 1994.
- [35] J. C. URSCHTEL, J. XU, X. HU, AND L. T. ZIKATANOV, *A cascadic multigrid algorithm for computing the Fiedler vector of graph Laplacians*, J. Comput. Math., 33 (2015), pp. 209–226.
- [36] P. VANĚK, J. MANDEL, AND M. BREZINA, *Algebraic multigrid by smoothed aggregation for second and fourth order elliptic problems*, Computing, 56 (1996), pp. 179–196.
- [37] H. WEI, L. CHEN, AND Y. HUANG, *Superconvergence and gradient recovery of linear finite elements for the Laplace-Beltrami operator on general surfaces*, SIAM J. Numer. Anal., 48 (2010), pp. 1920–1943.
- [38] J. H. WILKINSON, ed., *The Algebraic Eigenvalue Problem*, Oxford University Press, New York, 1988.
- [39] J. XU AND A. ZHOU, *A two-grid discretization scheme for eigenvalue problems*, Math. Comp., 70 (1999), pp. 17–25.
- [40] J. ZHOU, X. HU, L. ZHONG, S. SHU, AND L. CHEN, *Two-grid methods for Maxwell eigenvalue problems*, SIAM J. Numer. Anal., 52 (2014), pp. 2027–2047.