Computing Planetary Interior Normal Modes with A Highly Parallel Polynomial Filtering Eigensolver

Jia Shi
Department of Earth, Environmental and Planetary Sciences, Rice University
Houston, TX, USA
jia.shi@rice.edu

Ruipeng Li
Center for Applied Scientific Computing
Lawrence Livermore National Laboratory
Livermore, CA, USA
li50@llnl.gov

Yuanzhe Xi
Department of Mathematics and Computer Science, Emory University
Atlanta, GA, USA
yxi26@emory.edu

Yousef Saad
Department of Computer Science and Engineering, University of Minnesota
Minneapolis, MN, USA
saad@umn.edu

Maarten V. de Hoop
Department of Computational and Applied Mathematics, Rice University
Houston, TX, USA
mdehoop@rice.edu

***Abstract*—A highly parallel algorithm has been developed and exploited to compute the planetary normal modes of the elastic-gravitational system, which is approximated via the mixed finite element method on unstructured tetrahedral meshes. The eigenmodes of the relevant generalized eigenvalue problem were extracted by a Lanczos approach combined with polynomial filtering. In contrast with the standard shift-and-invert and the full-mode coupling algorithms, the polynomial filtering technique is ideally suited for solving large-scale 3-D interior eigenvalue problems since it significantly enhances the memory and computational efficiency without loss of accuracy. The parallel efficiency and scalability of this approach are demonstrated on Stampede2 at the Texas Advanced Computing Center. To our knowledge, this is the first time that the direct calculation of the normal modes of 3-D strongly heterogeneous planets, in particular, Earth and Mars, is made feasible via a combination of multiple matrix-free methods and a separation of the essential spectra.**

***Index Terms*—Eigenvalues and Eigenfunctions; Geophysics**

I. Introduction

The study of planetary normal modes is important for studying the dynamic response to sources including earthquakes along faults, meteorite impacts, postseismic relaxation, as well as for analyzing and computing surface and body waves [1], [2]. The relatively low-lying, but numerous normal modes in the spectrum of a planet contain essential information about its large-scale structure and provide constraints on heterogeneity in geochemical composition, temperature, and anisotropy. For a review of Earth's free oscillations, see [3]. Due to the presence of a fluid outer core, an essential spectrum needs to be dealt with. The eigenfrequencies can be well measured using modern seismometers at the surface of the Earth, e.g., [4]. The analysis of normal modes has provided evidence that Earth's inner core is solid [5] and anisotropic [6]–[9], and constraints on density variations of Earth's mantle [10]. More recently, the density structure at the Earth's core-mantle boundary was studied using, in particular, Stoneley mode splitting observations [11]. Moreover, the normal modes associated with low-lying eigenfrequencies are used to characterize large earthquakes; for the great Sumatra earthquake, see [4].

It is believed that the normal modes provide the only way to study internal planetary density and attenuation, which is difficult for modern wave-equation based inversion algorithms, such as the adjoint tomography [12], or waveform inversion [13]. Additionally, with the InSight (Interior exploration using Seismic Investigations, Geodesy and Heat Transport) [14] mission to Mars, which will deploy a single seismic station on its surface [15] in November 2018, the study of normal modes on this planet becomes of great interest while the discontinuities and general heterogeneities are unknown. It is expected that the investigations of Mars will provide evidence for early stages of planets that are lost on Earth due to its active tectonics and large-scale mantle convection. It is indeed expected that a large interval of eigenfrequencies will be observable [16]. The fast computation of a large interval of eigenfrequencies of a large family of generally heterogeneous models will open a unique way to explore Mars' interior. The computational ability of these normal modes will enable researchers to explore planetary structures, including, density [10], wave speeds [1], [16], inner core [7], core-mantle boundary [17], etc., even with limited receivers.

However, the current standard approaches to computing the point spectrum and the associated normal modes have several limitations. Assuming spherical symmetry, the problem becomes one-dimensional and the computations in such models [18]–[20] are still a common practice; these are then typically used in a perturbation theory to include weak, angular heterogeneities [1]. Using the normal modes in a spherically symmetric model as a basis in the matrix representation leads to the "full-mode coupling" approach [21]–[23]. Such a Rayleigh-Ritz type of approach works well under the assumption of weak angular heterogeneities, or perhaps very low eigenfrequencies, but in more strongly heterogeneous models certainly apparent at higher eigenfrequencies, the matrix that needs to be diagonalized becomes dense. Moreover, the separation of the essential spectrum needs to be carried out.

When considering lateral variations in the density and elastic moduli, the finite element method (FEM) is, in general, needed, which demands an entirely different computational strategy. In this work, we overcame several long-standing complications in the previous work [18] by characterizing and separating the essential spectrum and introducing a mixed FEM approach, and reconstructed the elastic-gravitational system on the entire planet without simplification to resolve the pollution issue of the associated generalized eigenvalue problem (GEP) from the essential spectrum. The three major challenges in solving the GEP resulting from our newly discretized elastic-gravitational system on a 3-D planet, are the following:

1) **A large-scale 3-D GEP.** Since it is a 3-D planetary-scale problem, the degrees of freedom for the displacement will reach several billion, depending on the target eigenfrequency interval. For an Earth model, we typically need five hundred thousand elements to capture the normal modes at 1.0 mHz and five hundred billion elements for 100.0 mHz.
2) **Essential spectrum and computations of a great number of the interior normal modes.** The existence of the fluid regions results in the presence of an essential spectrum [24]. Although the computation of, for example, the geostrophic modes is possible, these are not relevant to seismic studies. We developed a separation of the essential spectrum, leading to an additional term that involves solving a sparse linear system internally.
3) **No explicit formulation of the stiffness matrix.** Since the separation needs an additional term to deal with the essential spectrum, it is not always feasible to store the stiffness matrix explicitly for large-scale 3-D problems. However, the separation can be handled more efficiently via a matrix-free scheme. For relatively low eigenfrequencies, self-gravitation needs to be computed. This leads to an additional dense matrix that requires special methods to perform matrix-vector multiplications.

Our primary goal is to set new standards for the normal modes of a rotating self-gravitating planetary model. We also aim to calculate the normal modes ranging from seconds to rotational modes with years in the timescale, such as the Chandler wobble of the Earth [25], [26] or the lunar tide mode [27]. Traditionally, interior normal modes of the resulting large and sparse GEP have been computed by invoking standard shift-and-invert approaches, but these often lead to computational and memory bottlenecks for large 3-D problems. Here, a novel, highly parallel and memory efficient polynomial filtered Lanczos algorithm has been developed and exploited for solving problems of much larger size than those solved by traditional approaches. A matrix-free scheme whereby the matrix is invoked only via matrix-vector multiplications, allows us to accommodate a fluid outer core and self-gravitation. The parallel efficiency and scalability of the proposed algorithm have been demonstrated on Stampede2 at the Texas Advanced Computing Center (TACC). The method developed here armed with several novel high-powered algorithms for calculating eigenvalues can carry out the calculations accurately. The flexibility of our computational framework allows us to cope with multiphysics efficiently. Furthermore, our work will also lead to a much faster method, i.e., normal mode summation [1], to calculate synthetic seismogram than the modern seismic wave simulator [28], which won the Gordon Bell award in 2003. We show for the first time, to the best of our knowledge, that direct computation of the normal modes of 3-D heterogeneous planets is feasible via a combination of multiple effective computational techniques. We demonstrate the performance of our proposed algorithm and code up to several hundred million elements, but it is still far from the limit on current super-computational platforms.

The outline of the paper is as follows. In Section II, we formulate the GEP from the elastic-gravitational system associated with different eigensolvers. In Section III, we introduce our novel approaches to solve for interior normal modes. In Section IV, we present the experimental results on up to 12,288 cores as well as the accuracy, efficiency, and scalability. We conclude and discuss the future work in Section V.

II. Problem formulation and Discretization

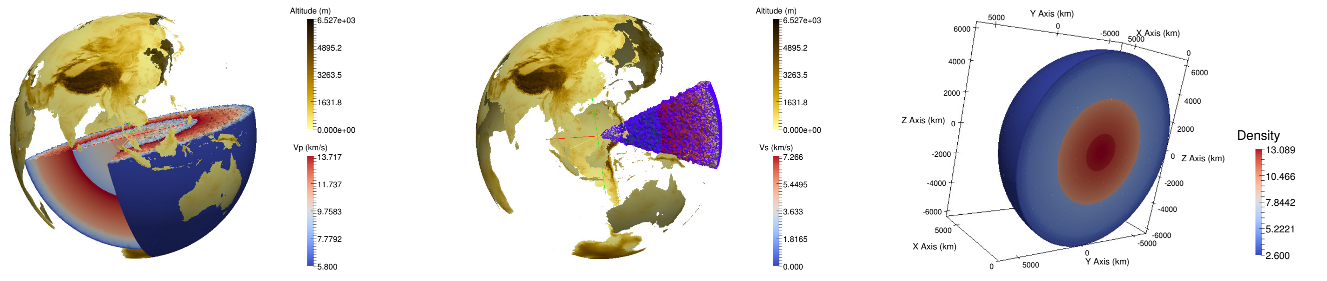
In this section, we introduce the GEP that is derived from the elastic-gravitational system of a non-rotating planet [1], [29] with different algorithms for solving interior normal modes. We also review the state-of-the-art eigensolvers and discuss the challenges encountered when using these algorithms to compute interior normal modes of the GEPs that are obtained from a 3-D model.

A. Modeling a general planet

Since Earth contains both solid and fluid regions, we use it as one of our examples and then generalize our approach to study other terrestrial planets. Our planet also contains several discontinuities, such as the core-mantle and the inner-core boundaries, which correspond to geochemical transitions. To describe the geometry of a general planet, especially their internal discontinuities, we utilize fully unstructured tetrahedral meshes to build our planetary models, see Figure 1 as an example. The meshes are generated by using DistMesh [30] and TetGen [31]. We then model the elastic-gravitational system of a non-rotating planet.

To study planetary normal modes, we include linear elasticity, compressible fluid, the fluid-solid and free surface boundary conditions. Discretization of the classical formula [1] leads to computational difficulties, since non-seismic modes from compressible fluid pollute the computations of the point spectrum. In this work, we use a displacement-pressure representation [33] explicitly for describing the oscillations of these non-seismic modes. Effectively, we separate out the essential spectrum, which contains the non-seismic modes. Hence, we build the geometry for the target planet and apply Continuous Galerkin approximation for the redesigned weak formulae. To

Fig. 1. The Preliminary reference Earth Model (PREM) [32]: (a) P wave speed model; (b) S wave speed model; (c) density model with axes. The red-blue colormaps in (a,b,c) indicate the speed or density values. The gold colormaps in (a,b) indicate the altitudes in meters.



(a) V_p (b) V_s (c) Density

solve the resulted GEP, the system needs to fit the requirements of the algorithms for solving interior eigenvalue problems.

B. Standard shift-and-invert methods

Standard approaches for computing eigenvalues of a GEP, $Ax = \lambda Mx$, that are located well inside the spectrum often resort to exploiting spectral transformations, i.e., shift-and-invert strategies [34]. They apply projection methods such as subspace iterations or the Lanczos algorithm to a transformed matrix of the form $(A - \sigma M)^{-1}$, where σ is the shift that is chosen near the desired eigenvalues. State-of-the-art algorithms based on Cauchy integral rational filters such as FEAST [35] and z-Pares [36]–[38] or the least-squares rational filters [39] have been designed with the same principles for extracting interior eigenvalues. These algorithms bear similarities with the shift-and-invert schemes in that they also require solving linear systems with $A - \sigma M$ when applying the filtered matrix, where σ denotes the pole used in the rational filter. However, in practice, solving such linear systems often leads to a computational bottleneck, especially on a highly parallel supercomputer with distributed memory. When computing interior eigenvalues, the shifted matrix $A - \sigma M$ is highly indefinite, and this significantly limits the applicability of iterative methods to solve the linear systems. In general, finding efficient parallel preconditioners remains a highly challenging task. Therefore, parallel sparse direct solvers are usually applied to these linear systems. Note here that this requires forming all the matrices explicitly. We introduce an additional variable p to separate out the essential spectra. We then apply the Continuous Galerkin mixed FEM [40], [41] to discretize the redesigned system and obtain a GEP:

$$\begin{bmatrix} A_{sg} & 0 & E_{\text{FS}} \\ 0 & A_f & A_{\text{dg}} \\ E_{\text{FS}}^{\mathsf{T}} & A_{\text{dg}}^{\mathsf{T}} & A_p \end{bmatrix} \begin{bmatrix} u^s \\ u^f \\ p \end{bmatrix} = \omega^2 \begin{bmatrix} M_s & 0 & 0 \\ 0 & M_f & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} u^s \\ u^f \\ p \end{bmatrix}, \quad (1)$$

where the matrix components and their corresponding weak formulae are shown in Table I. The interpretations of different variables are denoted in Table II and $\mathfrak{S}$ represents a symmetrization operator as in [29]. However, (1) is not a conventional eigenvalue problem, since its mass matrix is positive semi-indefinite. It usually requires techniques such as the null space purification [42].

Sparse direct methods are usually employed in this context to solve the shifted linear systems, however, the efficiency and scalability issues of which in a highly parallel environment often substantially lower the overall performance of the outer eigensolver. In the following, we highlight the major difficulties that were encountered when we applied parallel sparse direct solvers to the linear systems. First and foremost, the matrix $A - \sigma M$ needs to be formed explicitly to be factorized. This, however, is not always feasible. In the planetary normal mode computations, it would be very inefficient to form the matrix A explicitly and the only affordable operation is to multiply A with some vectors. Second, it is well-known that factorizations of the matrices from large-scale 3-D problems can be both time and memory consuming. Based on our experimental results, parallel direct solvers would often run out of memory in the compute node for the large-scale problems that are presented in Section IV-B. Third, even if the matrix can be successfully factorized, the solve phase of direct solvers often runs into scalability issues with increased problem sizes and larger numbers of processes. This is due to the inherent sequential nature of the triangular solves, which is also illustrated in Section IV-B. We remark here that these issues are also commonly experienced in many other applications.

C. Polynomial filtering for the GEP

To address the above-mentioned difficulties, polynomial filtering techniques [43]–[45] can be a very appealing alternative as they do not involve solving linear systems with the highly indefinite shifted matrices. Instead, the bulk of the computations are carried out in the form of SpMVs (sparse matrix-vector multiplication), which are generally much easier to be parallelized than solving the indefinite linear systems, so that these types of methods are amenable to parallel processing.

In a nutshell, the idea of polynomial filtering for extracting eigenvalues from a desired interval is to use a polynomial to amplify the eigenvalues inside the interval and dampen the eigenvalues that are outside, just as in the shift-and-invert scheme but without inverting $(A - \sigma M)$. For a GEP $Ax = \lambda Mx$, applying matrix polynomial $\rho(M^{-1}A)$ requires solving linear systems with M. As a consequence, polynomial filtering algorithms cannot be applied to the GEP of the form (1) directly, since the matrix M is singular. However,

Table I. MATRICES IN (2) AND (3) WITH THEIR CORRESPONDING WEAK FORMULAE, WHERE THE VARIABLES ARE SHOWN IN TABLE II. SINCE THE SYSTEM OF A NON-ROTATING PLANET IS HERMITIAN, FOR A BILINEAR EXPRESSION $L(u, v)$, WE USE $\mathfrak{S}\{L(u, v)\} := \frac{1}{2}(L(u, v) + L(v, u))$.

matrices	physical interpretations	corresponding weak formulae
A_{sg}	solid stiffness matrix with reference gravity	$\int_{\Omega^{\mathrm{S}}} (c_{ijkl}\partial_{x_k} u_l^s)\partial_{x_i} v_j^s \,\mathrm{d}x + \int_{\Sigma^{\mathrm{FS}}} \mathfrak{S}\{[\rho^0]^f \nu_i^{s\to f} u_i^s g_j v_j^s\} \,\mathrm{d}x + \int_{\Omega^{\mathrm{S}}} \mathfrak{S}\{(\rho^0 u_i^s g_i)\partial_{x_j} v_j^s - \rho^0 u_i^s(\partial_{x_i} g_j) v_j^s - \rho^0 u_i^s(\partial_{x_i} v_j^s) g_j\} \,\mathrm{d}x$
A_f	non-seismic modes in the fluid regions	$\int_{\Omega^{\mathrm{F}}} \rho^0 N^2 \frac{g_i u_i^f g_j v_j^f}{\lVert g \rVert^2} \,\mathrm{d}x$
A_p	fluid pressure matrices with/without reference gravity	$\int_{\Omega^{\mathrm{F}}} -p/\kappa v^p \,\mathrm{d}x, \quad \int_{\Omega^{\mathrm{F}}} -p^e/\kappa v^p \,\mathrm{d}x$
$A_{\mathrm{dg}}, A_{\mathrm{d}}$	fluid stiffness matrices with/without reference gravity	$\int_{\Omega^{\mathrm{F}}} \left[(\partial_{x_j} p) v_j^f - p\rho^0\kappa^{-1} g_j v_j^f\right] \mathrm{d}x, \quad \int_{\Omega^{\mathrm{F}}} (\partial_{x_j} p^e) v_j^f \,\mathrm{d}x$
$A_{\mathrm{dg}}^{\mathsf{T}}, A_{\mathrm{d}}^{\mathsf{T}}$	constraints with/without reference gravity	$\int_{\Omega^{\mathrm{F}}} \left[u_j^f \partial_{x_j} v^p - \rho^0\kappa^{-1} u_j^f g_j\right] v^p \,\mathrm{d}x, \quad \int_{\Omega^{\mathrm{F}}} u_j^f \partial_{x_j} v^p \,\mathrm{d}x$
A_s	solid stiffness matrix without gravity	$\int_{\Omega^{\mathrm{S}}} (c_{ijkl}\partial_{x_k} u_l^s)\partial_{x_i} v_j^s \,\mathrm{d}x$
$E_{\mathrm{FS}}, E_{\mathrm{FS}}^{\mathsf{T}}$	fluid-solid boundary conditions for the solid/fluid regions	$\int_{\Sigma^{\mathrm{FS}}} p^{(e)} \nu_j^{s\to f} v_j^s \,\mathrm{d}x, \quad \int_{\Sigma^{\mathrm{FS}}} -u_j^s \nu_j^{f\to s} v^p \,\mathrm{d}x$
M_s, M_f	solid and fluid mass matrix	$\int_{\Omega^{\mathrm{S}}} \rho^0 u_j^s v_j^s \,\mathrm{d}x, \quad \int_{\Omega^{\mathrm{F}}} \rho^0 u_j^f v_j^f \,\mathrm{d}x$

Table II. VARIABLES USED IN TABLE I AND THEIR INTERPRETATIONS. NOTE THE BRUNT-VÄISÄLÄ FREQUENCY $N^2 = (\nabla\rho^0/\rho^0 - g\rho^0/\kappa)\cdot g$ MEASURES THE STABILITY OF THE LIQUID OUTER CORE.

variables	interpretations
Ω^{S}, Ω^{F}, Σ^{FS}	solid/fluid regions, fluid-solid boundaries
ρ^0, $[\rho^0]^f$	density and density along Σ^{FS} at the fluid side
c, κ	elastic stiffness tensor, bulk moduli
g, N^2	reference gravity, Brunt-Väisälä frequency
u^s, u^f	displacement fields in the solid/fluid regions
p, p^e	pressure fields with/without reference gravity
v^s, v^f, v^p	test functions for u^s, u^f and pressure
$\nu^{s\to f}$	normal vector from solid to fluid regions
$\nu^{f\to s}$	normal vector from fluid to solid regions

a simple Schur complement trick can fix this problem by first substituting p via $p = -A_p^{-1} E_G^{\mathsf{T}} u$, where $u = [(u^s)^{\mathsf{T}}, (u^f)^{\mathsf{T}}]^{\mathsf{T}}$ assuming that A_p is invertible, and then transforming (1) to

$$\left(A_G - E_G A_p^{-1} E_G^{\mathsf{T}}\right) u = \omega^2 M u, \tag{2}$$

where

$$A_G = \begin{bmatrix} A_{sg} & 0 \\ 0 & A_f \end{bmatrix},\ E_G = \begin{bmatrix} E_{\mathrm{FS}} \\ A_{\mathrm{dg}} \end{bmatrix},\ M = \begin{bmatrix} M_s & 0 \\ 0 & M_f \end{bmatrix},$$

Note that p in (1) is not a variable of the Hamiltonian and does not play a role in the orthonormal condition and system (2) provides the authentic eigenvalue problem with the same orthonormal condition as (1).

For most of the high-frequency modes, we do not need to include the reference gravity. We can then substitute the pressure field via $p^e = A_p^{-1} E^{\mathsf{T}} u$ and obtain a simpler system:

$$(A_S - E A_p^{-1} E^{\mathsf{T}}) u = \omega^2 M u, \tag{3}$$

where

$$A_S = \begin{bmatrix} A_s & 0 \\ 0 & 0 \end{bmatrix},\ E = \begin{bmatrix} E_{\mathrm{FS}} \\ A_{\mathrm{d}} \end{bmatrix},$$

where the matrix components and their corresponding weak formulae are also shown in Table I. We obtain the discretized systems (2) for relatively low normal modes and (3) for all the higher ones. If the planet does not contain fluid regions, we can further simplify (2) and (3) to

$$A_{sg} u^s = \omega^2 M_s u^s \quad \text{and} \quad A_s u^s = \omega^2 M_s u^s, \tag{4}$$

respectively. More details about these simplifications can be found in [33]. Without loss of generalities, we write (2, 3) and their simplified systems in (4) as the GEP:

$$Ax = \lambda M x, \tag{5}$$

where A represents the elastic-gravitational components, M indicates the orthogonality condition of the system, x denotes the displacement field and λ denotes ω^2. One of the most significant advantages of the form (2) over (1) is that the linear system solutions associated with M and A_p can be solved with a highly efficient and parallel iterative method. This will be discussed in detail in Section III-B. Additionally, the problem size has also been reduced and the singular part in the mass matrix of (1) is removed. Now it is feasible to design our fully parallel computational scheme that entirely relies on SpMVs to solve the interior normal modes of the GEP (5).

III. PARALLEL POLYNOMIAL FILTERED LANCZOS ALGORITHMS FOR THE GEP

In this section, we present a fully parallel polynomial filtered Lanczos algorithm that combines several efficient parallel approaches to solve the GEP (5) in a matrix-free framework, where all the matrices are required only in the form of matrix-by-vector products. Current state-of-the-art polynomial filtering codes, such as FILTLAN [44] and EVSL [46], are implemented only for a single computation node parallelized with OpenMP, and the polynomial degrees exploited there are relatively low. However, as we point out in Section II-C, the polynomial filtering technique will provide significant advantages over the standard shift-and-invert methods for large-scale 3-D problems when the factorizations of the shifted matrices are too expensive. Hence, we implemented, to the best of our knowledge, the first fully parallel polynomial filtered Lanczos algorithm. The three levels of parallelism in the proposed approach, enabled by a hybrid parallel computing paradigm, are listed the following:

1) The coarsest level of parallelism is obtained by the *spectrum slicing* approach [45], [47], where the desired part of the spectrum is first partitioned into intervals with roughly equal numbers of eigenvalues. Then, a parallel polynomial filtered Lanczos solver processes on each interval individually and the eigenvalues in each interval are extracted independently from other ones. This level of parallelism is implemented with the Message Passing Interface (MPI) by splitting the global communicator into subcommunicators corresponding to the intervals.
2) The second level of parallelism is enabled by *domain decomposition (DD)*, where the global mesh is first decomposed in such a way that the couplings between subdomains are minimized. The global matrices and vectors are distributed across the processes within each subcommunicator corresponding to the DD and replicated across the subcommunicators with the same distribution.
3) The finest level of parallelism is supported by shared memory multithreaded computations running on multicore processors. Specifically, the BLAS/LAPACK subroutines and the *scalar* SpMV routines, i.e., local SpMVs on a single process, are threaded with OpenMP.

Additionally, for this application, an efficient parallel solver is needed for the right-hand-side matrix M and with A_p in (2) and (3). A simple and efficient parallel iterative method is introduced and used after adequately scaling the matrices.

A. Lanczos algorithms with polynomial filtering

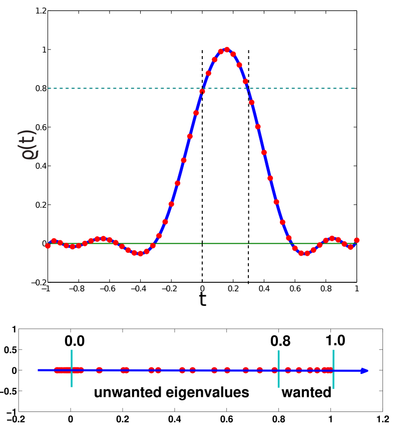
The polynomial filter should be able to map the wanted part of the spectrum of the original matrix, independent of its location in the spectrum, to the largest eigenvalues of the filtered matrix, so that the transformed problem becomes more tractable for standard projection methods such as subspace iterations or the Lanczos algorithm. In this paper, we adopt the polynomial filtering algorithms recently developed [45] due to their simplicity and robustness. These polynomial filtering techniques have recently been implemented in a (scalar) software package named EVSL [46].

The polynomial filter is constructed through a Chebyshev polynomial expansion of the Dirac delta function based at some well-selected point inside the target interval. Since Chebyshev polynomials are defined over the interval $[-1,\ 1]$, a linear transformation is needed to map the eigenvalues of $M^{-1}A$ onto this interval. Figure 2 illustrates a polynomial filter $\rho(t)$ for the eigenvalues in the interval $[0, 0.3]$, where the wanted eigenvalues of $\tilde{A}$ in (6) are mapped to the eigenvalues of $\rho(\tilde{A})$ that are larger than 0.8. This is achieved by applying a simple linear transformation:

$$\tilde{A} := \frac{A - cM}{d},\ c = \frac{\lambda_{\max} + \lambda_{\min}}{2},\ d = \frac{\lambda_{\max} - \lambda_{\min}}{2}, \quad (6)$$

where $\lambda_{\max}$ and $\lambda_{\min}$ are the largest and smallest eigenvalues of $M^{-1}A$, respectively. These two extreme eigenvalues can be efficiently estimated by performing a few steps of the Lanczos algorithm. Similarly, the target interval $[\xi, \eta]$ should be transformed into $[\xi, \eta] := [(\xi - c)/d, (\eta - c)/d]$. In the rest of this paper, the matrix A and the interval (ξ, η) are assumed to be shifted and scaled in such a way.

Fig. 2. An illustration of polynomial filtering techniques for symmetric eigenvalue problems.



Polynomial filter $\rho(t)$ for interval $(0, 0.3)$. The eigenvalues of the filtered matrix are the red dotted points along the blue solid curve

The spectrum of filtered matrix $\rho(A)$, where the wanted eigenvalues in $(0, 0.3)$ are mapped to the largest eigenvalues

Let $T_j(t)$ denote the Chebyshev polynomial of the first kind of degree j. The k-th degree polynomial filter function $\rho(t)$ for the interval $[\xi, \eta]$ centered at γ takes the form

$$\rho(t) = \frac{\sum_{j=0}^{k} \mu_j T_j(t)}{\sum_{j=0}^{k} \mu_j T_j(\gamma)}, \quad (7)$$

where the expansion coefficients are

$$\mu_j = \begin{cases} \frac{1}{2} & \text{if } j = 0 \\ \cos(j \cos^{-1}(\gamma)) & \text{otherwise} \end{cases}. \quad (8)$$

The construction of $\rho(t)$ is simpler than other existing approaches because it involves only two parameters: the degree k and the center γ. They are determined systematically as follows. From some lowest degree, we keep increasing k until the values $\rho(\xi)$ and $\rho(\eta)$ are less than or equal to a preselected threshold τ, which is 0.8 by default, and by construction $\rho(t)$ takes the value 1 at the center γ. It is also useful to make sure that the obtained filter is balanced such that its values at the boundaries are the same, i.e., $\rho(\xi) = \rho(\eta)$. This can be achieved by applying Newton's method to solve the equation

$$\rho(\hat{t}_j) - \rho(\hat{t}_{j+1}) = 0, \quad (9)$$

where the unknown is γ. The Lanczos σ-damping [48, Chap. 4] multipliers, denoted by σ_j, are used to reduce the oscillations near the boundaries of the interval, and these are given by

$$\sigma_j^k = \begin{cases} 1 & \text{if } j = 0 \\ \frac{\sin(j\theta_k)}{j\theta_k}, \quad \theta_k = \frac{\pi}{k+1} & \text{otherwise} \end{cases}.$$

For a GEP, $Ax = \lambda Mx$, the base form of filtering is

$$\rho(M^{-1}A)x = \rho(\lambda)x. \quad (10)$$

This is now an eigenvalue problem in the standard form but the matrix involved is nonsymmetric. Multiplying both sides by M yields the following problem:

$$Kx = \rho(\lambda)Mx, \quad \text{with} \quad K = M\rho(M^{-1}A), \tag{11}$$

where K is symmetric and M is symmetric positive definite. Thus, we can apply the Lanczos algorithm to matrix pencil (K, M). In the polynomial filtered non-restart Lanczos algorithm, each step of the iterative process consists of a Lanczos step with K

$$\beta_{i+1}z_{i+1} = Kv_i - \alpha_i z_i - \beta_i z_{i-1}, \tag{12}$$

followed by a full reorthogonalization against all the previous vectors v_j's, where $z_i \equiv Mv_i$ is an auxiliary sequence saved to avoid the cost of the multiplications with M in the inner products. A test for convergence is triggered for every N_{cycle} steps after the first N_{test} iterations, by checking if the sum of the Ritz values that are greater than τ, the "bar" value of the filter, no longer varies in several consecutive checks. After the Lanczos iterations, the obtained Ritz pairs (θ_i, u_i) are first tested if the Ritz values θ_i are greater than τ and then projected back to the original problem $Ax = \lambda Mx$. Finally, a Ritz pair (λ_i, u_i) is accepted if λ_i falls into the desired interval (ξ, η) and the residual norm is smaller than the given tolerance.

Overall, the multiplication with the filtered matrix K in each Lanczos step (12) often represents the most expensive computation, especially when the degree of the polynomial filter is high. A few practical details in applying K follow. Computing

$$Kv_i = M\rho(M^{-1}A)v_i = \rho(AM^{-1})z_i, \tag{13}$$

requires k solves with M and k SpMV with A where k is the degree of the polynomial filter ρ.

B. Solution methods for linear systems with M

The cost of solving linear systems with M obtained from large 3-D problems by sparse direct methods can be expensive. For the same reasons as discussed in Section II-B, direct solvers are often ruled out as a viable option for the large-scale 3-D problems. Additionally, as shown in (13), multiplying the filtered matrix with a vector requires to perform k solves with M, as many as the multiplications with A, where k is the degree of the polynomial. As will be shown in Section IV, some situations require a degree as high as a few thousand. Therefore, it is crucial to have a scalable solution method for applying M^{-1}, and this essentially precludes the use of parallel sparse direct solvers.

On the other hand, a notable characteristic of the mass matrix M in the context of FEM discretizations is that it is often very well conditioned with a condition number that can be bounded by small constants, independent of mesh sizes after a proper scaling is applied [49]–[51]. This is a notable difference with the stiffness matrix A. As an important consequence this will entail a significant computational advantage by allowing the use of simple iterative methods for applying M^{-1}. In this work, we adopt a simple diagonal scaling approach that was also exploited in [47], [52]. Suppose D is the diagonal matrix that consists of the diagonal entries of M, and let $\hat{A} = D^{-1/2}AD^{-1/2}$, $\hat{M} = D^{-1/2}MD^{-1/2}$ and $\hat{x} = D^{1/2}x$. We solve the following problem,

$$\hat{A}\hat{x} = \lambda\hat{M}\hat{x}, \tag{14}$$

that is congruent to $Ax = \lambda Mx$, using the polynomial filtered Lanczos algorithm, in which the linear systems with $\hat{M}$ are solved by performing a fixed number of Chebyshev iterations. An appealing property of Chebyshev iterations is that there are no inner products involved, which makes this algorithm very efficient for distributed memory architectures [53]. Systems with A_p in (2) and (3), are handled similarly since A_p has the same property as the mass matrix M.

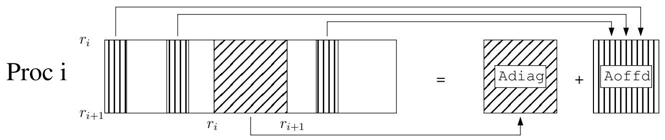
C. Parallel implementations

Here we present the parallel implementation of the methods discussed so far, using primarily a DD framework and exploiting sparse matrix-vector multiplications.

1) Domain decomposition: To fully parallelize our computations, two quantities, solid and fluid vertices, need to be load balanced simultaneously due to the multiphysics nature. We transform an algebraic elements-vertices mesh into a vertices-to-vertices graph and utilize multi-constraint graph partitioning algorithm (ParMetis, [54]) to compute a DD such that the edge-cut is minimized and that each subdomain contains approximately the same amount of solid and fluid vertices. Processors communicate with each other to obtain needed vertices and their related information. Based on the vertices-to-vertices graph at each processor, we construct the local matrix nonzero patterns. To generate local row-based matrices, we include all the elements that are connected to local vertices, build local submatrices element-wise and add them into the local Compressed Sparse Row (CSR) format matrix.

2) Parallel sparse matrix-vector multiplications: As shown in the previous sections, in our polynomial filtered Lanczos algorithm, all the major matrix operations boil down to SpMVs with either the matrix A or M. Therefore, the efficiency of parallel SpMV plays a critical the overall performance of the algorithm. There is a rich body of research on distributed memory SpMV, see, e.g., [55]–[61]. Global sparse matrices are distributed according to the DD based on graph partitioning algorithms to reduce the communication volume involved in the SpMV. Each process owns the entire rows of the local domain which are further split into an "on-process" block and an "off-process" block. Figure 3 illustrates the storage of local rows. Global vectors are distributed conformably to the sparse matrices. We remark that this parallel matrix and vector distribution scheme is widely used by many other linear solver and eigensolver packages such as pARMS [62], hypre [63] and PETSc [64]. A clear advantage of splitting the local rows into the two blocks is that in the parallel SpMV, the computations involved in the multiplication with the on-process block can be overlapped with the communications. The main steps of parallel SpMV implemented with MPI is sketched in Algorithm 1, where the non-blocking communications are first initiated and

Fig. 3. The storage of local rows of a distributed matrix



the local SpMV with the on-process submatrix are performed immediately afterwards so that the communication cost can be (partially) hidden by overlapping with the local computations. The SpMV with the off-process submatrix is executed when the communications finish.

Algorithm 1 Parallel SpMV, $y = Av$

```
1: Copy the elements to send from $v$ to *sendbuf*
2: Post nonblocking MPI_Isend(*sendbuf*, ...)
3: Post nonblocking MPI_Recv(*recvbuf*, ...)
4: Compute scalar SpMV: $y = \text{SpMV}(A_{\text{on-proc}}, v)$
5: MPI_Waitall
6: Compute scalar SpMV: $y = y + \text{SpMV}(A_{\text{off-proc}}, \textit{recvbuf})$
```

IV. Computational experiments

The experiments in Section IV-A and IV-B were conducted on Comet, a supercomputer at the San Diego Supercomputer Center, equipped with an Intel Xeon E5-2680v3 CPU with 24 cores and 128 GB memory on each node. The experiments in Section IV-C and IV-D were carried out on Stampede2 (Phase 1), a supercomputer at the TACC, each node of which has an Intel Xeon Phi Knights Landing (KNL) CPU with 68 cores and 96 GB memory, and on Stampede2 (Phase 2) which has an Intel Xeon Skylake (SKX) CPU with 48 cores and 192 GB memory on each node. The code was written in Fortran 90 and compiled with Intel Fortran Compiler with Intel MKL for high-performance BLAS/LAPACK and scalar SpMV routines.

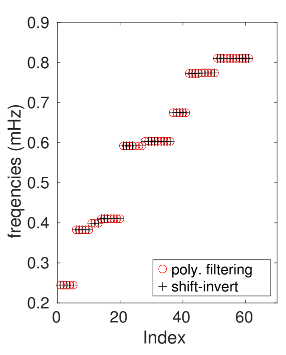
We used PREM [32] to generate 3-D Earth models and a simple solid model to mimic the Martian structure, since Mars is considered as a purely solid planet in many recent tests [65], [66]. The proposed algorithm was compared with the shift-and-invert Lanczos methods available from PARPACK [42], [67], where parallel direct solver MUMPS [68] was used for solving the involved linear systems with the shifted matrix, which was first reordered by the nested dissection approach [69] from ParMetis [54] to reduce fill-ins.

A. Numerical accuracy

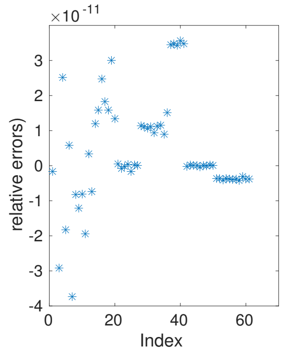
We first demonstrate the accuracy of the eigenvalues computed by the proposed polynomial filtered Lanczos algorithm by comparing them with the ones computed by PARPACK on a relatively small problem from a two-million-element Earth model. We verified that there were no missing or spurious eigenvalues computed by the proposed approach. In Figure 4, we show that the eigenvalues computed by both algorithms are very close.

Since our test 3-D Earth model is spherically symmetric, we can apply the spherical harmonics to reduce the problem to one

Fig. 4. Comparison of the solutions between shift-and-invert and polynomial filtered Lanczos methods.



(a) Eigenfrequencies



(b) Relative errors

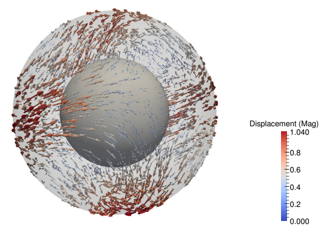
dimension. The normal modes from a spherically symmetric Earth model can be represented via the spherical harmonics [1] into two different major categories, i.e., the spheroidal and toroidal modes, which correspond to primarily different seismic waves. Figure 5 illustrates our computed modes: (a1)–(a4) show different toroidal modes, which have zero displacement inside liquid outer core; (b1)–(b4) show different spheroidal modes. The computed normal modes precisely match the semi-analytic solution of a spherically symmetric Earth model.

B. Memory and computational efficiency

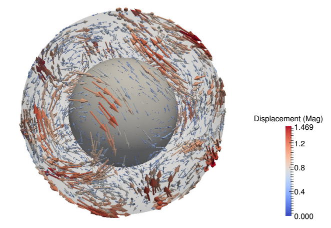
As discussed in Section II-B, standard shift-and-invert methods can be very memory demanding for factoring $A - \sigma M$ from large-scale 3-D problems. Figure 6 shows a comparison of the memory usage between the shift-and-invert Lanczos method and the proposed polynomial filtering Lanczos method. In Figure 6(a), we present the memory requirement of the two methods for solving the GEP (5) of five different solid models, and in Figure 6(b) for solving the GEP (2) of four different Earth models. The problem sizes were roughly doubled when we doubled the number of processes, so that the degrees of freedom on each process was kept about the same. For the shift-and-invert method, we show the maximum of the peak memory utilization required by MUMPS over all the processes (labeled "peak" in the figure) and the average memory consumption for storing the factors (labeled "avg"). The shift-and-invert method was not scalable regarding the memory requirement. In contrast, the polynomial filtering algorithm was much more efficient and scaled perfectly with the problem sizes in terms of the memory consumption, where the memory was mostly used for storing the Lanczos vectors that are evenly distributed across the processes.

In Table III, we also report the time consumption of different methods. The first and second two experiments compute 85 and 262 normal modes, respectively. For a smaller number of normal modes, the polynomial filtering approach shows its computational advantage while the problem sizes are larger than 2.5 million. However, for computing a larger number of normal modes, the standard shift-and-invert method becomes less effective for many iterations due to the inherent sequential nature of the triangular solves as we mentioned in Section II-B.

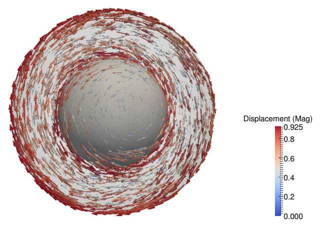
Fig. 5. Illustration of different normal modes from a three-million-element Earth model. (a1) – (a4) illustrate different toroidal modes; (b1) – (b4) illustrate different spheroidal modes. The colors in (a1) – (a4) represent the magnitudes of the displacement field and the colors in (b1) – (b4) represent the radial components of the displacement field. The subtitles ${}_nT_l$ and ${}_nS_l$ represent the notation of the semi-analytic solutions from a spherically symmetric Earth model [1], [20], where the subscriptions n is the overtone number and l is the spherical-harmonic degree.



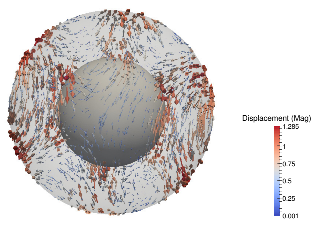
(a1) ${}_0T_2$



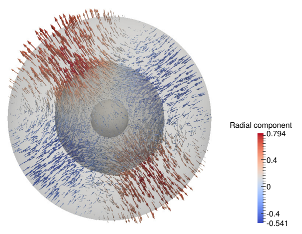
(a2) ${}_0T_5$



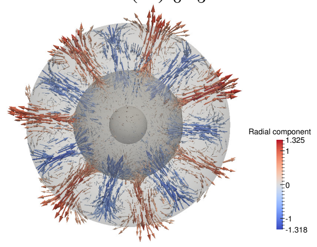
(a3) ${}_1T_1$



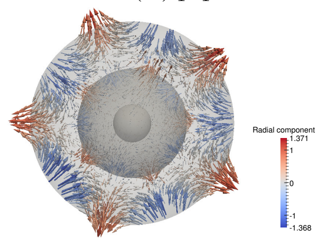
(a4) ${}_1T_3$



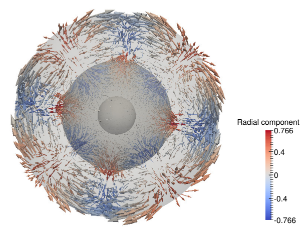
(b1) ${}_0S_2$



(b2) ${}_0S_7$

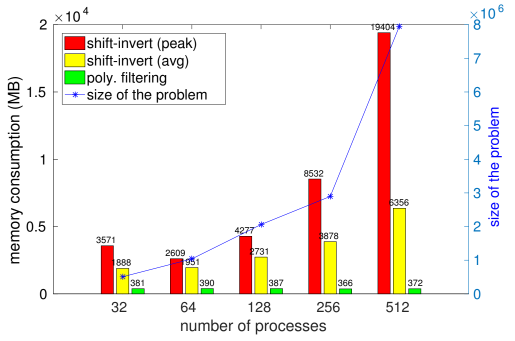


(b3) ${}_1S_5$

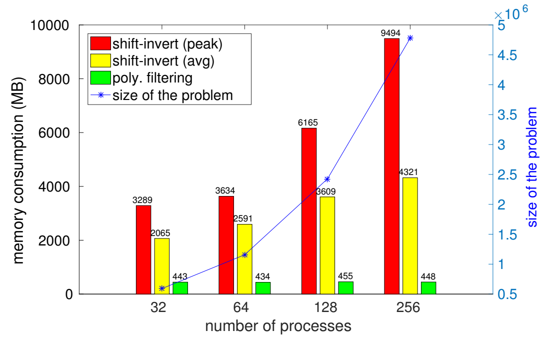


(b4) ${}_2S_4$

Fig. 6. Weak scalability tests of the memory consumption for the shift-and-invert and the polynomial filtered Lanczos methods.



(a) Comparison of different solid models



(b) Comparison of different Earth models

C. *Weak scalability analysis*

We shall start our performance study with the weak scalability analysis of the proposed algorithm for a set of eight solid models and a set of seven Earth models that are tabulated in Table IV, where the number of computation nodes ('nn') and the number of processes ('np') used on both the SKX and KNL CPUs are listed for each problem, and all the cores of a CPU were used with one process per core. The number of elements is provided in the table, as well as the problem sizes, which range from 2 million to 252 million for the solid models and from 2 million to 128 million for the Earth models.

Table III. Time and memory costs of different approaches.

size	methods	nn/np/tds	#eigs	mem.(MB)	total (s)
2,425,349	shift-invert	4/32/3	85	17,013	11040.19
	poly. filter	4/96/1	85	384	11723.11
4,778,004	shift-invert	8/64/3	85	24,077	fail
	shift-invert	16/64/6	85	19,062	33699.19
	poly. filter	8/192/1	85	384	25851.98
1,157,131	shift-invert	2/16/3	262	15,189	11612.50
	poly. filter	2/48/1	262	384	13508.17
2,425,349	shift-invert	4/32/3	262	22,274	33525.62
	poly. filter	4/96/1	262	384	25660.18

1) SpMV: We report the performance of our parallel SpMV implementation on the two Stampede2 machines. As the most important computational kernel in the proposed algorithm, the scalability and parallel efficiency of parallel SpMV are fundamental to the overall performance of the solver. The times for computing one SpMV with matrix A and with matrix M for the problems given in Table IV and the corresponding parallel efficiencies are shown in Figure 7. The parallel SpMV exhibited good weak scalability as expected in the tests, where the problem sizes on each process remained roughly constant. For the Earth model, the matrix $A \equiv A_G - EA_p^{-1}E^{\mathsf{T}}$ in (2) was not explicitly formed, and the SpMV with A was performed by multiplications with A_G, E and E^{T} and a solve with A_p, which was carried out by Chebyshev iterations in the same way as M. The parallel efficiencies in the sense of weak scalability, which is defined as T_{n_0}/T_{np}, are reasonably high, where n_p is the number of processes, n_0 is the smallest number of processes used, and T_{n_p} and T_{n_0} are the associated

Table IV. Test cases for different solid and Earth models.

Exp	SKX nn/np	KNL nn/np	# of elm.	size of A
C1	2/96	2/136	2,067,539	1,038,084
C2	4/192	4/272	4,133,442	2,060,190
C3	8/384	8/544	8,079,387	3,894,783
C4	16/768	16/1088	16,036,734	7,954,392
C5	32/1536	32/2176	32,090,054	15,809,076
C6	64/3072	64/4352	64,187,247	31,138,518
C7	128/6144	128/8704	127,634,594	61,381,362
C8	256/12288	-/-	252,990,545	120,336,519

(a) Solid models with different sizes

Exp	SKX nn/np	# of elm.	size of A_G	size of A_p
E1	2/96	1,972,263	1,086,702	70,429
E2	4/192	4,094,031	2,265,129	160,220
E3	8/384	8,000,777	4,466,349	311,655
E4	16/768	16,436,247	9,037,671	658,285
E5	32/1536	32,468,819	17,579,616	1,251,220
E6	64/3072	64,764,730	34,115,040	2,373,354
E7	128/6144	128,501,841	66,028,227	4,367,668

(b) Earth models of different sizes

Table V. Computation of a fraction of the spectrum

Exp	$(\lambda_{\min}, \lambda_{\max})$	$\frac{\xi-\eta}{2d}$ (%)	(deg, #it.)	#eigs	total (s)	avg (s)
C1	(-8.4e-7, .617)	3.743e-3	(757,92)	57	1546.95	27.14
C2	(-2.0e-5, 1.11)	3.743e-3	(757,172)	57	2101.43	36.87
C3	(-1.5e-5, 1.11)	3.743e-3	(757,212)	60	2085.34	34.76
C4	(-1.0e-4, 3.52)	3.743e-3	(757,212)	82	4420.75	53.91
C5	(-1.9e-4, 6.31)	3.743e-3	(757,412)	155	7685.94	49.59
C6	(-2.7e-4, 9.43)	3.743e-3	(757,712)	271	12006.64	44.30

(a) Solid models

Exp	$(\lambda_{\min}, \lambda_{\max})$	$\frac{\xi-\eta}{2d}$ (%)	(deg, #it)	#eigs	total (s)	avg (s)
E1	(-7.0e-7, 2.86)	8.157e-4	(1144,152)	50	2647.56	52.95
E2	(-2.6e-6, 7.27)	8.157e-4	(1140,412)	148	8530.22	57.63
E3	(-7.1e-6, 17.1)	8.157e-4	(1139,1232)	497	26237.35	52.79
E4	(-1.5e-5, 32.0)	8.157e-4	(1138,2672)	1176	60298.53	51.27
E5	(-4.1e-5, 68.5)	8.157e-4	(1138,7170)	3411	172818.0	50.66

(b) Earth models

Table VI. Computation of all the eigenvalues in fixed intervals

Exp	$(\lambda_{\min}, \lambda_{\max})$	(ξ, η)	(deg, #it)	#eigs	total (s)
C1	(-8.4e-6, .617)	(3.9e-7,8.9e-5)	(271,652)	259	2143.87
C2	(-2.0e-5, 1.11)	(3.9e-7,8.9e-5)	(365,672)	259	3318.60
C3	(-1.5e-5, 1.11)	(3.9e-7,8.9e-5)	(364,672)	259	3247.78
C4	(-1.1e-4, 3.52)	(3.9e-7,8.9e-5)	(649,672)	259	6224.22
C5	(-1.9e-4, 6.31)	(3.9e-7,8.9e-5)	(869,672)	259	7731.83
C6	(-2.6e-4, 9.43)	(3.9e-7,8.9e-5)	(1062,672)	259	9287.05
C7	(-4.4e-4, 14.8)	(3.9e-7,8.9e-5)	(1330,672)	259	11104.54
C8	(-5.6e-4, 20.5)	(3.9e-7,8.9e-5)	(1566,672)	259	13914.11

(a) Solid models

Exp	$(\lambda_{\min}, \lambda_{\max})$	(ξ, η)	(deg, #it)	#eigs	total (s)
E1	(-7.0e-7, 2.86)	(3.9e-7,8.9e-5)	(585,612)	257	5369.784
E2	(-2.6e-6, 7.27)	(3.9e-7,8.9e-5)	(933,632)	262	10083.951
E3	(-7.1e-6, 17.1)	(3.9e-7,8.9e-5)	(1430,652)	262	16570.223
E4	(-1.6e-5, 32.0)	(3.9e-7,8.9e-5)	(1957,652)	262	24614.738
E5	(-4.1e-5, 68.5)	(3.9e-7,8.9e-5)	(2863,652)	262	37793.401
E6	(-6.7e-5, 115.)	(3.9e-7,8.9e-5)	(3711,652)	262	44962.119
E7	(-1.4e-4, 202.)	(3.9e-7,8.9e-5)	(4922,652)	262	63054.411

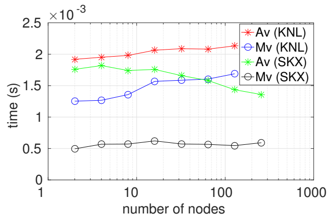
(b) Earth models

runtime. Note that there were a few cases where the parallel efficiency was greater than 1.0. This superlinearity effect is often seen and typically due to cache hierarchies [70].

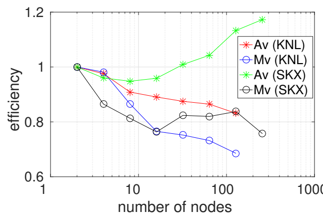
2) Computation of a fraction of the spectrum: In this section, we report the performance of using the polynomial filtered Lanczos algorithm to compute a fixed fraction of the spectra, that is we extracted all the eigenvalues located in interval (ξ, η) such that $(\xi - c)/d$ and $(\eta - c)/d$ are fixed to be nearly the same across the solid model problems C1–C8, and so across the Earth model problems E1–E8, where c and d are defined as before: $c = (\lambda_{\min} + \lambda_{\max})/2$ and $d = (\lambda_{\max} - \lambda_{\min})/2$. Therefore, the degree of the polynomial filter was roughly the same within the two sets of problems. Recall that the polynomial filter is determined by only two parameters, namely the center γ and the "bar" value τ that is fixed for all the runs in this paper. In Table V, for each problem, we provide a tight bound of the smallest and the largest eigenvalues $(\lambda_{\min}, \lambda_{\max})$, the portion of the entire spectrum, the degree of the polynomial filter 'deg', the number of the Lanczos iterations required '#it' and the number of the eigenvalues in the interval '#eigs'. The time for computing all the wanted eigenvalues and the average time for computing each eigenvalue are tabulated as well running on Stampede2 (Phase 2) with SKX CPUs. The main result we show here is that the CPU time for computing each eigenvalue stays roughly the same, so in this sense, the proposed polynomial filtered Lanczos algorithm is scalable with the problem sizes and the number of processes.

3) Computing normal modes in a fixed subinterval: Another common task in practice is to compute all eigenvalues in a prescribed interval included in the spectral interval. In this set of experiments, we present the performance for computing all the eigenvalues in a given interval, which contains about the same number of eigenvalues while increasing the sizes of the problems. In Table VI, we report the performance of computing eigenvalues in a fixed subinterval for problems C1–C8 and for problems E1–E7 respectively on Stampede2 (Phase 2) with SKX CPUs. A notable difference in this set of experiments from the previous one is that the required degree of the polynomial filter, in general, increases with the problem sizes, since the spectrum becomes wider as the problem sizes increase and thus higher degrees are required to construct the polynomial filter with the same τ value on a relatively narrower interval. On the other hand, the number of iterations required to compute all the eigenvalues stayed constant, which suggests constant average convergence rate for computing the eigenvalues of increasing size problems. Therefore, as shown in the table, the total time for computing the desired eigenvalues increases in proportion to the polynomial degree.

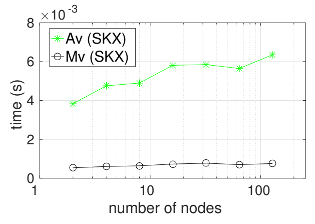
Fig. 7. Weak scalability study for parallel SpMV on TACC Stampede2 Phase 1 (KNL, 68 cores per node) and Phase 2 (SKX, 48 cores per node).



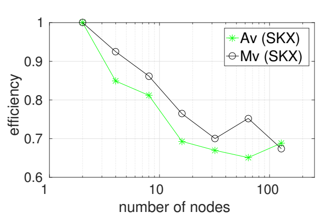
(a) Parallel SpMV time (solid)



(b) Parallel efficiency (solid)



(c) Parallel SpMV time (Earth)



(d) Parallel efficiency (Earth)

D. *Strong scalability*

The last set of experiments tested the strong scalability of the proposed algorithm to compute the 259 eigenvalues in the interval $(3.948e{-}7, 8.8826e{-}5)$ for problem C3 and the 262 eigenvalues in $(3.948e{-}7, 8.8826e{-}5)$ for problem E3. The degrees of the polynomial filter, the numbers of iterations required and the total computation times with 384 processes on 8 SKX CPUs are the same as those shown in Table VI. In Table VII, we present the performance of solving problem C3 with 192 to 1536 processes, including the times for performing a SpMV with A and M, the time for a solve with M with Chebyshev iterations, and the total time for computing the wanted eigenvalues, which all appear well parallelized with the increasing numbers of processes. The parallel efficiency in the strong scalability sense, that is $n_0 T_{n_0}/(n_p T_{np})$, provided in the last column of the table demonstrates a good strong scalability achieved by our eigensolver. Similar results for problem E3 are shown in Table VII as well.

Table VII. STRONG SCALABILITY TESTS FOR PROBLEMS C3 AND E3.

SKX nn/np	T-Av (s)	T-Mv (s)	T-$M^{-1}v$ (s)	total (s)	eff.
4/192	0.003398	0.001180	0.024681	6854.54	1.0
8/384	0.001741	0.000571	0.011874	3247.78	1.1
16/768	0.000687	0.000326	0.006695	1779.14	.96
32/1536	0.000357	0.000239	0.004828	1259.08	.68

(a) Solid model C3

SKX nn/np	T-Av (s)	T-Mv (s)	T-$M^{-1}v$ (s)	total (s)	eff.
4/192	0.007920	0.001385	0.029170	34319.28	1.0
8/384	0.004901	0.000639	0.013370	16570.22	1.0
16/768	0.003778	0.000381	0.007847	10071.56	.85

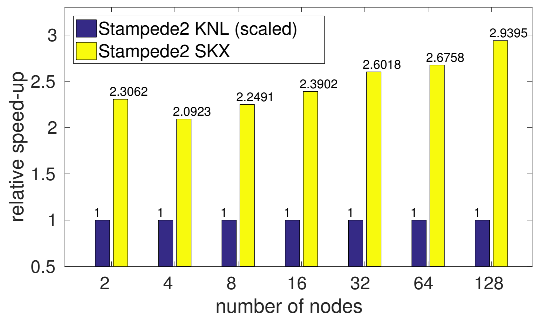
(b) Earth model E3

On Stampede2 (Phase 1) with KNL CPUs, the scalability and parallel efficiency of the proposed algorithm were similar as on Phase 2, but the overall performance was found consistently lower. The complete performance results are omitted in the paper, whereas a comparison of solving (C1–C7) in Table VI on the two machines is given in Figure 8, which shows a speedup of up to 2.9 on SKX CPUs.

V. CONCLUSION

We developed and exploited a highly parallel algorithm to compute the point spectrum of the elastic-gravitational system describing the normal modes of terrestrial planets. The system is discretized with a Continuous Galerkin method, more precisely, with the mixed FEM on unstructured tetrahedral meshes on the fluid and solid regions. The normal modes of the relevant GEP were extracted with a Lanczos approach combined with polynomial filtering, which can significantly enhance the memory and computational efficiency without loss of accuracy. The computational experiments were performed on Stampede2 at the TACC to demonstrate the high parallel efficiency and scalability of our proposed approach. We scaled our algorithm to several hundred million elements, which is still far from the limit of current supercomputers.

Fig. 8. Comparison of solving C1-C7 in Table VI on TACC Stampede2 Phase 1 (KNL) and Phase 2 (SKX).



Our future work includes further increasing the degrees of freedom to capture high-frequency normal modes, and incorporating self-gravitation leading to an additional dense matrix that requires special methods to perform matrix-vector multiplications in the GEP (5). It will also lead us to develop new methods to explore the inverse spectral problems to study internal planetary structures, including density, wave speeds, anisotropy, etc. This work can provide new opportunities and insights to study large earthquakes and simulate global seismic waves, and can be generalized to benefit many other applications as well.

ACKNOWLEDGMENT

J.S and M.dH were supported by the Simons foundation and XSEDE TG-EAR170019. The work of R.L was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344 (LLNL-CONF-753896). Y.X and Y.S were supported by NSF grants CCF-1318597, DMS-1521573, CCF-1505970.

References

[1] F. A. Dahlen and J. Tromp, *Theoretical global seismology*. Princeton University press, 1998.

[2] P. Lognonné, "Planetary seismology," *Annu. Rev. Earth Planet. Sci.*, vol. 33, pp. 571–604, 2005.

[3] J. Woodhouse and A. Deuss, "Theory and observations – Earth's free oscillations," *Seismology and Structure of the Earth: Treatise on Geophysics*, vol. 1, p. 31, 2010.

[4] J. Park, T.-R. A. Song, J. Tromp, E. Okal, S. Stein, G. Roult, E. Clevede, G. Laske, H. Kanamori, P. Davis *et al.*, "Earth's free oscillations excited by the 26 December 2004 Sumatra-Andaman earthquake," *Science*, vol. 308, no. 5725, pp. 1139–1144, 2005.

[5] A. Dziewonski and F. Gilbert, "Solidity of the inner core of the Earth inferred from normal mode observations," *Nature*, vol. 234, no. 5330, p. 465, 1971.

[6] J. H. Woodhouse, D. Giardini, and X.-D. Li, "Evidence for inner core anisotropy from free oscillations," *Geophysical Research Letters*, vol. 13, no. 13, pp. 1549–1552, 1986.

[7] J. Tromp, "Support for anisotropy of the Earth's inner core from free oscillations," *Nature*, vol. 366, no. 6456, p. 678, 1993.

[8] B. Romanowicz and L. Bréger, "Anomalous splitting of free oscillations: a reevaluation of possible interpretations," *Journal of Geophysical Research: Solid Earth*, vol. 105, no. B9, pp. 21 559–21 578, 2000.

[9] A. Deuss, J. C. Irving, and J. H. Woodhouse, "Regional variation of inner core anisotropy from seismic normal mode observations," *Science*, vol. 328, no. 5981, pp. 1018–1020, 2010.

[10] M. Ishii and J. Tromp, "Normal-mode and free-air gravity constraints on lateral variations in velocity and density of Earth's mantle," *Science*, vol. 285, no. 5431, pp. 1231–1236, 1999.

[11] P. Koelemeijer, A. Deuss, and J. Ritsema, "Density structure of Earth's lowermost mantle from Stoneley mode splitting observations," *Nature Communications*, vol. 8, 2017.

[12] J. Tromp, C. Tape, and Q. Liu, "Seismic tomography, adjoint methods, time reversal and banana-doughnut kernels," *Geophysical Journal International*, vol. 160, no. 1, pp. 195–216, 2005.

[13] V. Akcelik, G. Biros, and O. Ghattas, "Parallel multiscale gauss-newton-krylov methods for inverse wave propagation," in *Supercomputing, ACM/IEEE 2002 Conference*. IEEE, 2002.

[14] W. Banerdt, S. Smrekar, P. Lognonné, T. Spohn, S. Asmar, D. Banfield, L. Boschi, U. Christensen, V. Dehant, W. Folkner *et al.*, "InSight: a discovery mission to explore the interior of Mars," in *Lunar and Planetary Science Conference*, vol. 44, 2013, p. 1915.

[15] M. Golombek, D. Kipp, N. Warner, I. J. Daubar, R. Fergason, R. L. Kirk, R. Beyer, A. Huertas, S. Piqueux, N. Putzig *et al.*, "Selection of the InSight landing site," *Space Science Reviews*, vol. 211, no. 1-4, pp. 5–95, 2017.

[16] M. P. Panning, P. Lognonné, W. B. Banerdt, R. Garcia, M. Golombek, S. Kedar, B. Knapmeyer-Endrun, A. Mocquet, N. A. Teanby, J. Tromp *et al.*, "Planned products of the Mars structure service for the InSight mission to Mars," *Space Science Reviews*, vol. 211, no. 1-4, pp. 611–650, 2017.

[17] T. Lay, Q. Williams, and E. J. Garnero, "The core–mantle boundary layer and deep Earth dynamics," *Nature*, vol. 392, no. 6675, p. 461, 1998.

[18] R. Buland and F. Gilbert, "Computation of free oscillations of the Earth," *Journal of Computational Physics*, vol. 54, no. 1, pp. 95–114, 1984.

[19] J. Woodhouse and D. Doornbos, "The calculation of eigenfrequencies and eigenfunctions of the free oscillations of the Earth and the Sun," *Seismological algorithms*, pp. 321–370, 1988.

[20] G. Masters, M. Barmine, and S. Kientz, "Mineos: User Manual Version 1.0. 2," *Cal Inst of Tech*, 2011.

[21] A. Deuss and J. H. Woodhouse, "Theoretical free-oscillation spectra: the importance of wide band coupling," *Geophysical Journal International*, vol. 146, no. 3, pp. 833–842, 2001.

[22] A. Deuss and J. Woodhouse, "Iteration method to determine the eigenvalues and eigenvectors of a target multiplet including full mode coupling," *Geophysical Journal International*, vol. 159, no. 1, pp. 326–332, 2004.

[23] D. Al-Attar, J. H. Woodhouse, and A. Deuss, "Calculation of normal mode spectra in laterally heterogeneous earth models using an iterative direct solution method," *Geophysical Journal International*, vol. 189, no. 2, pp. 1038–1046, 2012.

[24] B. Valette, "Free oscillations spectrum of an elastic, self-gravitating, uniformly rotating body with a fluid inclusion," *Comptes Rendus de L acaemie Des Sciences Serie I-Mathematique*, vol. 309, no. 6, pp. 419–422, 1989.

[25] R. J. O'Connell and A. M. Dziewonski, "Excitation of the Chandler wobble by large earthquakes," *Nature*, vol. 262, no. 5566, p. 259, 1976.

[26] D. A. Yuen and W. Peltier, "Normal modes of the viscoelastic earth," *Geophysical Journal International*, vol. 69, no. 2, pp. 495–526, 1982.

[27] J. Tarpley, "The ionospheric wind dynamoI: Lunar tide," *Planetary and Space Science*, vol. 18, no. 7, pp. 1075–1090, 1970.

[28] D. Komatitsch, S. Tsuboi, C. Ji, and J. Tromp, "A 14.6 billion degrees of freedom, 5 teraflops, 2.5 terabyte earthquake simulation on the earth simulator," in *Supercomputing, 2003 ACM/IEEE Conference*. IEEE, 2003.

[29] M. V. de Hoop, S. Holman, and H. Pham, "On the system of elastic-gravitational equations describing the oscillations of the earth," *arXiv preprint arXiv:1511.03200*, 2015.

[30] P.-O. Persson and G. Strang, "A simple mesh generator in MATLAB," *SIAM review*, vol. 46, no. 2, pp. 329–345, 2004.

[31] H. Si, "TetGen, a Delaunay-based quality tetrahedral mesh generator," *ACM Transactions on Mathematical Software (TOMS)*, vol. 41, no. 2, p. 11, 2015.

[32] A. M. Dziewonski and D. L. Anderson, "Preliminary reference Earth model," *Physics of the earth and planetary interiors*, vol. 25, no. 4, pp. 297–356, 1981.

[33] J. Shi, M. V. de Hoop, R. Li, Y. Xi, and Y. Saad, "Fast eigensolver for computing Earth's normal modes," *Proceedings of the Project Review, Geo-Mathematical Imaging Group*, vol. 2, pp. 317–345, 2017.

[34] B. N. Parlett, *The Symmetric Eigenvalue Problem*, ser. Classics in Applied Mathematics. Philadelphia: SIAM, 1998, no. 20.

[35] E. Polizzi, "Density-matrix-based algorithm for solving eigenvalue problems," *Phys. Rev. B*, vol. 79, p. 115112, Mar 2009. [Online]. Available: http://link.aps.org/doi/10.1103/PhysRevB.79.115112

[36] T. Sakurai and H. Sugiura, "A projection method for generalized eigenvalue problems using numerical integration," *J. Comput. Appl. Math.*, vol. 159, no. 1, pp. 119 – 128, 2003, japan-China Joint Seminar on Numerical Mathematics; In Search for the Frontier of Computational and Applied Mathematics toward the 21st Century. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S037704270300565X

[37] T. Sakurai and H. Tadano, "CIRR: a Rayleigh-Ritz type method with contour integral for generalized eigenvalue problems," *Hokkaido Mathematical Journal*, vol. 36, pp. 745–757, 2007.

[38] J. Asakura, T. Sakurai, H. Tadano, T. Ikegami, and K. Kimura, "A numerical method for nonlinear eigenvalue problems using contour integrals," *JSIAM Letters*, vol. 1, pp. 52–55, 2009.

[39] Y. Xi and Y. Saad, "Computing Partial Spectra with Least-Squares Rational Filters," *SIAM J. Sci. Comput.*, vol. 38, no. 5, pp. A3020–A3045, 2016. [Online]. Available: http://dx.doi.org/10.1137/16M1061965

[40] K.-J. Bathe, *Finite element procedures*. Klaus-Jurgen Bathe, 2006.

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

[42] R. B. Lehoucq, D. C. Sorensen, and C. Yang, *ARPACK users' guide: solution of large-scale eigenvalue problems with implicitly restarted Arnoldi methods*. Siam, 1998, vol. 6.

[43] Y. Saad, "Filtered conjugate residualtype algorithms with applications," *SIAM Journal on Matrix Analysis and Applications*, vol. 28, no. 3, pp. 845–870, 2006. [Online]. Available: https://doi.org/10.1137/060648945

[44] H. Fang and Y. Saad, "A Filtered Lanczos Procedure for Extreme and Interior Eigenvalue Problems," *SIAM Journal on Scientific Computing*, vol. 34, no. 4, pp. A2220–A2246, 2012. [Online]. Available: https://doi.org/10.1137/110836535

[45] R. Li, Y. Xi, E. Vecharynski, C. Yang, and Y. Saad, "A Thick-Restart Lanczos algorithm with polynomial filtering for Hermitian eigenvalue problems," *SIAM J. Sci. Comput.*, vol. 38, no. 4, pp. A2512–A2534, 2016. [Online]. Available: http://dx.doi.org/10.1137/15M1054493

[46] R. Li, Y. Xi, L. Erlandson, and Y. Saad, "The Eigenvalues Slicing Library (EVSL): Algorithms, Implementation, and Software," *arXiv preprint arXiv:1802.05215*, 2018.

[47] Y. Xi, R. Li, and Y. Saad, "Fast computation of spectral densities for generalized eigenvalue problems," *submitted to SIAM J. Sci. Comput.*, 2017.

[48] C. Lanczos, *Applied Analysis*, ser. Dover Books on Mathematics. Dover Publications, 1988.

[49] L. Kamenski, W. Huang, and H. Xu, "Conditioning of finite element equations with arbitrary anisotropic meshes," *Math. Comput.*, vol. 83, no. 289, pp. 2187–2211, 2014. [Online]. Available: http://dx.doi.org/10.1090/S0025-5718-2014-02822-6

[50] A. Wathen, "Realistic eigenvalue bounds for the Galerkin mass matrix," *IMA J. Numer. Anal.*, vol. 7, no. 4, pp. 449–457, 1987.

[51] A. Wathen and T. Rees, "Chebyshev semi-iteration in preconditioning for problems including the mass matrix." *Electron. Trans. Numer. Anal.*, vol. 34, pp. 125–135, 2008. [Online]. Available: http://eudml.org/doc/225663

[52] E. G. Boman, K. Deweese, and J. R. Gilbert, *An Empirical Comparison of Graph Laplacian Solvers*, pp. 174–188. [Online]. Available: https://epubs.siam.org/doi/abs/10.1137/1.9781611974317.15

[53] Y. Saad, *Iterative Methods for Sparse Linear Systems, 2nd edition*. Philadelpha, PA: SIAM, 2003.

[54] G. Karypis, K. Schloegel, and V. Kumar, "ParMETIS–parallel graph partitioning and fill-reducing matrix ordering, version 4," *Department of Computer Science, University of Minnesota*, 2014.

[55] U. V. Catalyurek and C. Aykanat, "Hypergraph-partitioning-based decomposition for parallel sparse-matrix vector multiplication," *IEEE Transactions on Parallel and Distributed Systems*, vol. 10, no. 7, pp. 673–693, Jul 1999.

[56] A. Bienz, W. D. Gropp, and L. N. Olson, "TAPSpMV: Topology-Aware Parallel Sparse Matrix Vector Multiplication," *CoRR*, vol. abs/1612.08060, 2016. [Online]. Available: http://arxiv.org/abs/1612.08060

[57] R. H. Bisseling and W. Meesen, "Communication balancing in parallel sparse matrix-vector multiplication." *ETNA. Electronic Transactions on Numerical Analysis [electronic only]*, vol. 21, pp. 47–65, 2005. [Online]. Available: http://eudml.org/doc/128024

[58] B. Uçar and C. Aykanat, "Encapsulating multiple communication-cost metrics in partitioning sparse rectangular matrices for parallel matrix-vector multiplies," *SIAM Journal on Scientific Computing*, vol. 25, no. 6, pp. 1837–1859, 2004. [Online]. Available: https://doi.org/10.1137/S1064827502410463

[59] B. Vastenhouw and R. H. Bisseling, "A two-dimensional data distribution method for parallel sparse matrix-vector multiplication," *SIAM Review*, vol. 47, no. 1, pp. 67–95, 2005. [Online]. Available: https://doi.org/10.1137/S0036144502409019

[60] E. Kayaaslan, C. Aykanat, and B. Uçar, "1.5D Parallel Sparse Matrix-Vector Multiply," *SIAM Journal on Scientific Computing*, vol. 40, no. 1, pp. C25–C46, 2018. [Online]. Available: https://doi.org/10.1137/16M1105591

[61] A. Yoo, A. H. Baker, R. Pearce, and V. E. Henson, "A Scalable Eigensolver for Large Scale-free Graphs Using 2D Graph Partitioning," in *Proceedings of 2011 International Conference for High Performance Computing, Networking, Storage and Analysis*, ser. SC '11. New York, NY, USA: ACM, 2011, pp. 63:1–63:11. [Online]. Available: http://doi.acm.org/10.1145/2063384.2063469

[62] Z. Li, Y. Saad, and M. Sosonkina, "pARMS: a parallel version of the algebraic recursive multilevel solver," *Numerical linear algebra with applications*, vol. 10, no. 5-6, pp. 485–509, 2003.

[63] R. D. Falgout and U. M. Yang, "hypre: A library of high performance preconditioners," in *Computational Science — ICCS 2002*, P. M. A. Sloot, A. G. Hoekstra, C. J. K. Tan, and J. J. Dongarra, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2002, pp. 632–641.

[64] S. Balay, S. Abhyankar, M. Adams, J. Brown, P. Brune, K. Buschelman, L. Dalcin, V. Eijkhout, W. Gropp, D. Kaushik *et al.*, "PETSc Users Manual Revision 3.8," Argonne National Lab.(ANL), Argonne, IL (United States), Tech. Rep., 2017.

[65] S. Ceylan, M. van Driel, F. Euchner, A. Khan, J. Clinton, L. Krischer, M. Böse, S. Stähler, and D. Giardini, "From initial models of seismicity, structure and noise to synthetic seismograms for Mars," *Space Science Reviews*, pp. 1–16, 2017.

[66] J. F. Clinton, D. Giardini, P. Lognonné, B. Banerdt, M. van Driel, M. Drilleau, N. Murdoch, M. Panning, R. Garcia, D. Mimoun *et al.*, "Preparing for InSight: An Invitation to Participate in a Blind Test for Martian Seismicity," *Seismological Research Letters*, 2017.

[67] K. J. Maschho and D. Sorensen, "A portable implementation of ARPACK for distributed memory parallel architectures," in *Proceedings of the Copper Mountain Conference on Iterative Methods, April*, 1996, pp. 9–13.

[68] P. R. Amestoy, I. S. Duff, and J.-Y. L'excellent, "Multifrontal parallel distributed symmetric and unsymmetric solvers," *Computer methods in applied mechanics and engineering*, vol. 184, no. 2-4, pp. 501–520, 2000.

[69] A. George, "Nested dissection of a regular finite element mesh," *SIAM Journal on Numerical Analysis*, vol. 10, no. 2, pp. 345–363, 1973. [Online]. Available: https://doi.org/10.1137/0710032

[70] V. Kumar, A. Grama, A. Gupta, and G. Karypis, *Introduction to Parallel Computing: Design and Analysis of Algorithms*. Redwood City, CA, USA: Benjamin-Cummings Publishing Co., Inc., 1994.

[71] Z. Gimbutas and L. Greengard, "FMMLIB3D 1.2, FORTRAN libraries for fast multiple method in three dimensions," 2011.

[72] U. Ayachit, "The paraview guide: a parallel visualization application," 2015.

VI. Artifact Description Appendix

In this appendix, we provide general descriptions of the code, the software dependencies, compilation, and installation, and technical details of the experiments.

A. Core models and software dependencies

We provide technical details of the core modules of the code used in this work.

1) Mesh and model generator: We used several MATLAB scripts to generate the unstructured tetrahedral mesh as our input model.

- **Algorithm**: mesh and model building.
- **Program**: MATLAB.
- **Dependencies**: DistMesh [30] for discontinuities, TetGen [31] for mesh files and FMMLIB3D [71] (built by `make mwrap` and `make mex-matlab`) for the reference gravity.
- **Output**: It generated three mesh files that describe the elements-to-vertices relations, locations of the vertices and neighbor information of the elements. It also constructed the P wave speed (V_p), S wave speed (V_s) and density files, which are constructed by the user as well.
- **Hardwares**: We ran it on a workstation for small models. For generating large models, we used Pittsburgh Supercomputing Center Bridges Large Memory (HPE ProLiant DL580 servers, 3TB memory).

2) Eigensolver:

- **Algorithms**: We used `ChebLanNR`, a polynomial filtered Lanczos method, as the eigensolver, as well as `Chebiter`, the Chebyshev iterations for solving the linear systems with the mass matrix. Both are available in pEVSL.
- **Programming language**: C with MPI.
- **Compilation**: Intel MPI C compiler `mpiicc -DUNIX -O3 -g -Wall -MKL (lp64 & blacs)` with Intel MKL or MPI C compiler `mpicc -lopenblas -lpthread` with OpenBLAS.
- **Software dependencies**: pEVSL

3) Planetary normal modes: We developed a framework for the elastic-gravitational system for computing the normal modes.

- **Algorithms**: continuous Galerkin mixed finite element method.
- **Programming language**: Fortran90 with MPI.
- **Compilation**: Intel MPI Fortran compiler `mpiifort -c -O3 -qopenmp -MKL (lp64 & blacs) -lmetis lparmetis chebiter.o -lpevsl` with Intel MKL or MPI Fortran `mpif90` with Lapack and Blas.
- **Software dependencies**: ParMetis [54] to partition unstructured graphs with multiple constraints.
- **Output**: Eigenfrequencies in the prescribed interval as well as their corresponding eigenfunctions.

B. Comparisons with other methods

For the comparisons with shift-and-invert Lanczos methods in Section IV-A and IV-B, we used the following packages:

- **MUMPS**: We applied multifrontal parallel distributed direct solvers [68] with the nested dissection ordering from ParMetis to factorize the shifted matrices. The MUMPS library was built with `-lmkl_scalapack`, `-lmkl_lapack95` and `-lmkl_blacs`. Parameter `ICNTL(4) = 2` was set to output the memory usage in MUMPS.
- **PARPACK** We applied PARPACK [42], [67] to perform the shift-and-invert Lanczos method with the factorization obtained from MUMPS. To make PARPACK library, Lapack and Blacs are needed as well. We included `dmumps_struc.h` in the code to perform the factorization in double precision.

C. Experiment setup

Since our program does not depend on a special hardware, it should run on most machines. We precomputed all the experimental models C1–C8 and E1–E8 in Table IV. On Stampede2, we used modules intel/17.0.4, impi/17.0.3 to compile our codes. In the input text file, we set up the base name of the mesh files and the directories of the input and output data paths. We also set up the interval range of the desired eigenfrequencies and whether or not the reference gravity was needed to be included. We used the SBATCH system to submit jobs and usually set "export OMP_NUM_THREADS = 1" and "export MV2_ENABLE_AFFINITY=0". Once we set the number of nodes and the number of tasks per node, we used `ibrun ./plmvcg_Stampede2.out`, where `plmvcg_Stampede2.out` was our executable file.

D. Execution workflow

During the execution, each process read a piece of the input mesh and V_s model files. We then utilized ParMetis to partition the unstructured vertices-to-vertices graph. `MPI_ALLTOALL` and `MPI_ALLTOALLV` were used to redistribute the input data and load the other model files in parallel. The distributed matrices were generated on each process. After that, the required matrix-vector multiplications were set up, and the parallel Chebyshev iteration method was initialized for the mass matrices A_p and M. We then estimated the bounds of the eigenvalues of $M^{-1}A$ and constructed the polynomial filter on each process. The `ChebLanNR` solver was then performed to compute all the normal modes in the prescribed subinterval. At last, we evaluated the relative errors of the computed normal modes, which are typical of the order of 10^{-11}. As a post-processing step, we used ParaView [72] for the data visualization.