RESEARCH ARTICLE

WILEY

Weight-adjusted discontinuous Galerkin methods: Matrix-valued weights and elastic wave propagation in heterogeneous media

Jesse Chan

Rice University, 6100 Main Street, MS 134, Houston, Texas, USA

Correspondence
Jesse Chan, Rice University, 6100 Main Street, MS 134, Houston, Texas, USA.
Email: jesse.chan@rice.edu

Summary
Weight-adjusted inner products are easily invertible approximations to weighted L^2 inner products. These approximations can be paired with a discontinuous Galerkin (DG) discretization to produce a time-domain method for wave propagation which is low storage, energy stable, and high-order accurate for arbitrary heterogeneous media and curvilinear meshes. In this work, we extend weight-adjusted DG methods to the case of matrix-valued weights, with the linear elastic wave equation as an application. We present a DG formulation of the symmetric form of the linear elastic wave equation, with upwind-like dissipation incorporated through simple penalty fluxes. A semidiscrete convergence analysis is given, and numerical results confirm the stability and high-order accuracy of weight-adjusted DG for several problems in elastic wave propagation.

1 | INTRODUCTION

Efficient and accurate methods for elastic wave propagation form a foundation for a broad range of applications, from seismic and medical imaging to rupture and earthquake simulation. Finite differences are the most common choice of method[1]; however, finite element methods have garnered interest due to their low numerical dispersion and ability to accommodate geometrically flexible unstructured meshes.

Typical methods for time-domain wave propagation use explicit time stepping, since the hyperbolic partial differential equations which govern wave propagation admit a reasonable stable time-step restriction. However, unless special techniques (such as diagonal mass lumping) are applied, finite element methods require the inversion of a global mass matrix when paired with explicit time integrators. Spectral element methods sidestep this issue on hexahedral meshes by choosing nodal basis functions which are discretely orthogonal with respect to an underintegrated L^2 inner product, which produces a diagonal mass matrix.[2] The inversion of a globally coupled matrix can also be avoided through the use of discontinuous Galerkin (DG) methods, which result in a locally invertible block diagonal mass matrices. Due to difficulties in extending mass-lumping techniques from hexahedra to tetrahedra, high-order DG methods are often used for seismic simulations which require the use of simplicial meshes.[3-8] High-order DG methods also lend themselves well to efficient implementations using graphics processing units (GPUs).[9-12]

Most high-order DG methods on simplicial meshes assume that models of media and material coefficients are constant over each element, which allows them to deal with discontinuous wave speeds across element interfaces. However, if the media is such that material gradients are nonzero in the interior of an element, piecewise constant approximations can yield inaccurate simulations of wave propagation.[13-15] This limitation can be overcome by incorporating subelement heterogeneities into weighted mass matrices, resulting in a DG method which is both high-order accurate and energy stable.[14,15] On tetrahedral meshes, this approach necessitates the precomputation and storage of factorizations or inverses

for each local mass matrix, which greatly increases both storage costs and data transferred at high orders of approximation. These costs are especially problematic for accelerator architectures such as GPUs, which possess limited memory.

Storage costs associated with weighted mass matrices can be avoided by approximating weighted L^2 inner products using weight-adjusted inner products, which result in easily invertible approximations to weighted mass matrices.[16,17] For sufficiently regular weights, high-order accuracy is also retained. When paired with an energy-stable DG formulation, these approximations result in weight-adjusted DG (WADG) methods, which preserve energy stability and high-order accuracy while retaining a low asymptotic storage cost. Additionally, unlike mass-lumping techniques, WADG methods do not rely on the use of inexact quadrature rules and reduce to the exact inversion of mass matrices for constant weights.

Weight-adjusted DG methods have been applied to acoustic wave propagation in heterogeneous media and on curvilinear meshes.[16,17] Both of these previous applications have involved scalar weighting functions. In this work, we extend weight-adjusted inner products to matrix-valued weights. This provides a way to approximate the inverse of a block system of mass matrices which are coupled together by a spatially varying matrix-valued weighting function. This approximate inverse involves the application of scalar mass matrix inverses and the matrix-free application of a system of weighted block mass matrices using quadrature. We show that this approach reduces storage and computational costs compared to the storage of inverses or factorizations of the full block mass matrix system and apply this approximation to derive a stable and high-order accurate method for elastic wave propagation in arbitrary heterogeneous media. This method is based on an energy-stable DG formulation of the symmetric form of the elastic wave equations, with upwind-like numerical dissipation introduced through simple penalty fluxes.[18] In contrast to the fluxes proposed in Ye et al,[8] the penalty fluxes used here can be made to be independent of material coefficients.

This work proceeds as follows: Sections 2 and 3 present an energy-stable DG formulation with simple penalty fluxes for the symmetric hyperbolic form of the elastic wave equation and discuss issues related to storage and inversion of local mass matrices for material coefficients with subelement variations. Section 4 extends weight-adjusted approximations to weighted L^2 inner products and mass matrices to the case of matrix-valued weights and provides interpolation estimates which account for the regularity of the matrix weight. These results are incorporated into a WADG method for the linear elastic wave equations in Section 5. Finally, numerical results in Section 6 demonstrate the accuracy of this method for several problems in linear elasticity.

2 | SYMMETRIC FORM OF THE ELASTIC WAVE EQUATION

We begin with the linear elastic wave equation in a domain $\Omega \in \mathbb{R}^d$. These equations can be written as a first-order velocity-stress system for velocity $\boldsymbol{v}$ and symmetric stress tensor $\tilde{\boldsymbol{S}}$

$$\rho \frac{\partial \boldsymbol{v}}{\partial t} = \nabla \cdot \tilde{\boldsymbol{S}} + \boldsymbol{f},$$

$$\frac{\partial \tilde{\boldsymbol{S}}}{\partial t} = \frac{1}{2} \underline{\boldsymbol{C}} \left(\nabla \boldsymbol{v} + \nabla \boldsymbol{v}^T \right),$$

where $\boldsymbol{f}$ is the body force per unit volume, ρ is density, and $\underline{\boldsymbol{C}}$ is the symmetric constitutive stiffness tensor relating stress and strain. We rewrite these equations as a symmetric hyperbolic system of partial differential equations[19] using Voigt notation

$$\rho \frac{\partial \boldsymbol{v}}{\partial t} = \sum_{i=1}^{d} \boldsymbol{A}_i^T \frac{\partial \boldsymbol{\sigma}}{\partial \boldsymbol{x}_i} + \boldsymbol{f},$$

$$\boldsymbol{C}^{-1} \frac{\partial \boldsymbol{\sigma}}{\partial t} = \sum_{i=1}^{d} \boldsymbol{A}_i \frac{\partial \boldsymbol{v}}{\partial \boldsymbol{x}_i}, \tag{1}$$

where $\boldsymbol{C}$ is the symmetric matrix form of the constitutive tensor $\underline{\boldsymbol{C}}$ and $\boldsymbol{\sigma}$ is a vector of length $N_d = \frac{d(d+1)}{2}$, the number of unique entries of the stress tensor $\tilde{\boldsymbol{S}}$ in d dimensions. We note that the matrices $\boldsymbol{A}_i$ are spatially constant, while ρ, $\boldsymbol{C}$, and $\boldsymbol{C}^{-1}$ can vary spatially. Furthermore, we will assume that ρ and $\boldsymbol{C}$ are positive-definite and bounded pointwise such that

$$0 < \rho_{\min} \leq \rho(\boldsymbol{x}) \leq \rho_{\max} < \infty,$$

$$0 < c_{\min} \leq \boldsymbol{u}^T \boldsymbol{C}(\boldsymbol{x}) \boldsymbol{u} \leq c_{\max} < \infty,$$

$$0 < \tilde{c}_{\min} \leq \boldsymbol{u}^T \boldsymbol{C}^{-1}(\boldsymbol{x}) \boldsymbol{u} \leq \tilde{c}_{\max} < \infty$$

for all $\boldsymbol{x} \in \mathbb{R}^d$ and all $\boldsymbol{u} \in \mathbb{R}^{N_d}$.

In 2 dimensions, $\boldsymbol{v} = (\boldsymbol{v}_1, \boldsymbol{v}_2)^T$ and $\boldsymbol{\sigma} = (\sigma_{xx}, \sigma_{yy}, \sigma_{xy})^T$

$$\tilde{\boldsymbol{S}} = \begin{pmatrix} \sigma_{xx} & \sigma_{xy} \\ \sigma_{xy} & \sigma_{yy} \end{pmatrix},$$

while the matrices $\boldsymbol{A}_i$ are

$$\boldsymbol{A}_1 = \begin{pmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{pmatrix}, \qquad \boldsymbol{A}_2 = \begin{pmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 0 \end{pmatrix}.$$

In 3 dimensions, the velocity is $\boldsymbol{v} = (\boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{v}_3)^T$, while $\boldsymbol{\sigma} = (\sigma_{xx}, \sigma_{yy}, \sigma_{zz}, \sigma_{yz}, \sigma_{xz}, \sigma_{xy})^T$ denotes the unique entries of the stress tensor $\tilde{\boldsymbol{S}}$

$$\tilde{\boldsymbol{S}} = \begin{pmatrix} \sigma_{xx} & \sigma_{xy} & \sigma_{xz} \\ \sigma_{xy} & \sigma_{yy} & \sigma_{yz} \\ \sigma_{xz} & \sigma_{yz} & \sigma_{zz} \end{pmatrix}.$$

The matrices $\boldsymbol{A}_i$ are then

$$\boldsymbol{A}_1 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}, \qquad \boldsymbol{A}_2 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix}, \qquad \boldsymbol{A}_3 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}.$$

In general anisotropic media, $\boldsymbol{C}$ is symmetric and positive-definite. For 2-dimensional isotropic media, $\boldsymbol{C}$ and its inverse are given as

$$\boldsymbol{C} = \begin{pmatrix} 2\mu + \lambda & \lambda & 0 \\ \lambda & 2\mu + \lambda & 0 \\ 0 & 0 & \mu \end{pmatrix}, \qquad \boldsymbol{C}^{-1} = \frac{1}{4\mu^2 + \mu\lambda} \begin{pmatrix} 2\mu + \lambda & -\lambda & 0 \\ -\lambda & 2\mu + \lambda & 0 \\ 0 & 0 & \frac{4\mu^2 + \mu\lambda}{\mu} \end{pmatrix},$$

where λ, μ are Lame parameters. For 3-dimensional isotropic media, $\boldsymbol{C}$ and its inverse are given instead by

$$\boldsymbol{C} = \begin{pmatrix} 2\mu + \lambda & \lambda & \lambda & \\ \lambda & 2\mu + \lambda & \lambda & \\ \lambda & \lambda & 2\mu + \lambda & \\ & & & \mu\boldsymbol{I}^{3\times3} \end{pmatrix}, \qquad \boldsymbol{C}^{-1} = \frac{1}{\mu(2\mu + 3\lambda)} \begin{pmatrix} \mu + \lambda & -\lambda/2 & -\lambda/2 & \\ -\lambda/2 & \mu + \lambda & -\lambda/2 & \\ -\lambda/2 & -\lambda/2 & \mu + \lambda & \\ & & & \frac{1}{\mu(2\mu+3\lambda)}\boldsymbol{I}^{3\times3} \end{pmatrix}.$$

We will consider both spatially varying isotropic and anisotropic media in this work.

3 | AN ENERGY-STABLE DG FORMULATION FOR ELASTIC WAVE PROPAGATION

Energy-stable DG methods have been constructed based on nonsymmetric formulations of the elastodynamics equations.[20] However, it is also straightforward to derive an energy-stable DG formulation based on the symmetric first-order formulation of the elastic wave equations in (13). We assume that the domain Ω is Lipschitz and exactly triangulated by a mesh Ω_h, which consists of elements D^k. We further assume that each element D^k is the image of a reference element $\hat{D}$ under the local elemental mapping

$$\boldsymbol{x}^k = \boldsymbol{\Phi}^k \hat{\boldsymbol{x}},$$

where $\boldsymbol{x}^k = \{x^k, y^k\}$ for $d = 2$ and $\boldsymbol{x}^k = \{x^k, y^k, z^k\}$ for $d = 3$ denote physical coordinates on D^k; and $\hat{\boldsymbol{x}} = \{\hat{x}, \hat{y}\}$ for $d = 2$ and $\hat{\boldsymbol{x}} = \{\hat{x}, \hat{y}, \hat{z}\}$ for $d = 3$ denote coordinates on the reference element. We denote the determinant of the Jacobian of $\boldsymbol{\Phi}^k$ as J and refer to it as the Jacobian for the remainder of this work.

We will approximate solution components over each element D^k from an approximation space $V_h(D^k)$, which we define as the composition of the mapping $\boldsymbol{\Phi}^k$ and a reference approximation space $V_h(\hat{D})$

$$V_h(D^k) = \boldsymbol{\Phi}^k \circ V_h(\hat{D}).$$

The global approximation space $V_h(\Omega_h)$ is then defined as the direct sum of elemental approximation spaces

$$V_h(\Omega_h) = \bigoplus_{D^k} V_h(D^k).$$

For the remainder of this work, we will take $V_h(\hat{D}) = P^N(\hat{D})$, where $P^N(\hat{D})$ is the polynomial space of total degree N on the reference simplex. In 2 dimensions, P^N on a triangle is

$$P^N(\hat{D}) = \left\{ \hat{x}^i \hat{y}^j, \quad 0 \le i + j \le N \right\},$$

and in 3 dimensions, P^N on a tetrahedron is

$$P^N(\hat{D}) = \left\{ \hat{x}^i \hat{y}^j \hat{z}^k, \quad 0 \le i + j + k \le N \right\}.$$

We denote the L^2 inner product and norm over D^k by $(\cdot, \cdot)_{L^2(D^k)}$, such that

$$(\boldsymbol{g}, \boldsymbol{h})_{L^2(D^k)} = \int_{D^k} \boldsymbol{g} \cdot \boldsymbol{h} \, \mathrm{d}\boldsymbol{x} = \int_{\hat{D}} \boldsymbol{g} \cdot \boldsymbol{h} \, J \mathrm{d}\hat{\boldsymbol{x}}, \qquad \|\boldsymbol{g}\|^2_{L^2(D^k)} = (\boldsymbol{g}, \boldsymbol{g})_{L^2(D^k)},$$

where $\boldsymbol{g}, \boldsymbol{h}$ are real vector-valued functions. Global L^2 inner products and norms are using local L^2 inner products and norms

$$(\boldsymbol{g}, \boldsymbol{h})_{\cdot} = \sum_{D^k \in \Omega_h} (\boldsymbol{g}, \boldsymbol{h})_{L^2(D^k)}, \qquad \|\boldsymbol{g}\|^2 = \sum_{D^k \in \Omega_h} \|\boldsymbol{h}\|^2_{L^2(D^k)}.$$

We define also the L^2 inner product and norm over the boundary ∂D^k of an element

$$\langle \boldsymbol{u}, \boldsymbol{v} \rangle_{L^2(\partial D^k)} = \int_{\partial D^k} \boldsymbol{u} \cdot \boldsymbol{v} \, \mathrm{d}\boldsymbol{x} = \sum_{f \in \partial D^k} \int_{\hat{f}} \boldsymbol{u} \cdot \boldsymbol{v} J^f \mathrm{d}\hat{\boldsymbol{x}}, \qquad \|\boldsymbol{u}\|^2_{L^2(\partial D^k)} = \langle \boldsymbol{u}, \boldsymbol{u} \rangle_{L^2(\partial D^k)},$$

where J^f is the Jacobian of the mapping from a reference face $\hat{f}$ to a physical face f of an element.

Let f be a face of an element D^k with neighboring element $D^{k,+}$ and unit outward normal $\boldsymbol{n}$. Let u be a function which is discontinuous across element interfaces. We define the interior value u^- and exterior value u^+ on a face f of D^k such that

$$u^- = u|_{f \cap \partial D^k}, \qquad u^+ = u|_{f \cap \partial D^{k,+}}.$$

The jump and average of a scalar function $u \in V_h(\Omega_h)$ over f are then defined as

$$[\![u]\!] = u^+ - u^-, \qquad \{\!\{u\}\!\} = \frac{u^+ + u^-}{2}.$$

Jumps and averages of vector-valued functions $\boldsymbol{u} \in \mathbb{R}^m$ and $\tilde{\boldsymbol{S}} \in \mathbb{R}^{m \times n}$ are then defined component-wise

$$([\![\boldsymbol{u}]\!])_i = [\![\boldsymbol{u}_i]\!], \qquad 1 \le i \le m, \qquad \left([\![\tilde{\boldsymbol{S}}]\!]\right)_{ij} = [\![\tilde{\boldsymbol{S}}_{ij}]\!], \qquad 1 \le i \le m, \quad 1 \le j \le n.$$

We can now specify a DG formulation for the linear elastic wave equations in (13). Symmetric hyperbolic systems readily admit a DG formulation based on penalty fluxes.[21] For the linear elastic wave equation in symmetric first-order form, this formulation is given as

$$\sum_{D^k \in \Omega_h} \left(\rho \frac{\partial \boldsymbol{v}}{\partial t}, \boldsymbol{w} \right)_{L^2(D^k)} = \sum_{D^k \in \Omega_h} \left(\left(\sum_{i=1}^{d} \boldsymbol{A}_i^T \frac{\partial \boldsymbol{\sigma}}{\partial x_i} + \boldsymbol{f}, \boldsymbol{w} \right)_{L^2(D^k)} + \left\langle \frac{1}{2} \boldsymbol{A}_n^T [\![\boldsymbol{\sigma}]\!] + \frac{\tau_v}{2} \boldsymbol{A}_n^T \boldsymbol{A}_n [\![\boldsymbol{v}]\!], \boldsymbol{w} \right\rangle_{L^2(\partial D^k)} \right),$$

$$\sum_{D^k \in \Omega_h} \left(\boldsymbol{C}^{-1} \frac{\partial \boldsymbol{\sigma}}{\partial t}, \boldsymbol{q} \right)_{L^2(D^k)} = \sum_{D^k \in \Omega_h} \left(\left(\sum_{i=1}^{d} \boldsymbol{A}_i \frac{\partial \boldsymbol{v}}{\partial x_i}, \boldsymbol{q} \right)_{L^2(D^k)} + \left\langle \frac{1}{2} \boldsymbol{A}_n [\![\boldsymbol{v}]\!] + \frac{\tau_\sigma}{2} \boldsymbol{A}_n \boldsymbol{A}_n^T [\![\boldsymbol{\sigma}]\!], \boldsymbol{q} \right\rangle_{L^2(\partial D^k)} \right),$$

(2)

for all $\boldsymbol{w}, \boldsymbol{q} \in V_h(\Omega_h)$. Here, $\boldsymbol{A}_n$ is the normal matrix defined on a face f as $\boldsymbol{A}_n = \sum_{i=1}^{d} n_i \boldsymbol{A}_i$. In 2 dimensions, $\boldsymbol{A}_n$ is

$$\boldsymbol{A}_n = \begin{pmatrix} n_x & 0 \\ 0 & n_y \\ n_y & n_x \end{pmatrix}.$$

while in 3 dimensions, $\boldsymbol{A}_n$ is

$$
\boldsymbol{A}_n = \begin{pmatrix} n_x & 0 & 0 \\ 0 & n_y & 0 \\ 0 & 0 & n_z \\ 0 & n_z & n_y \\ n_z & 0 & n_x \\ n_y & n_x & 0 \end{pmatrix}.
$$

The terms τ_v, τ_σ are penalty parameters which are introduced on element interfaces. We assume that $\tau_v, \tau_\sigma \geq 0$ and that they are piecewise constant over each shared face between 2 elements. These penalty parameters can be taken to be 0, which corresponds to a DG method using a nondissipative central flux.[14,22] In Section 3.2, we show that when τ_v, τ_σ are positive, they introduce a dissipation of energy in a manner which is similar to the upwind flux.[20,23] We note that the stability of the DG formulation is independent of the magnitude of these parameters; however, as discussed in Section 6.1, naively choosing the values of these parameters can result in a stiffer semidiscrete system of ordinary differential equations (ODEs) and a smaller maximum stable time-step.

In many applications, $\boldsymbol{f}$ is a point source or Dirac delta, which is not L^2 integrable. Thus, $(\boldsymbol{f}, \boldsymbol{w})_{L^2(D^k)}$ may not be well-defined. In such cases when $\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{\beta}(\boldsymbol{x})\delta(\boldsymbol{x} - \boldsymbol{x}_0)$ (for some vector $\boldsymbol{\beta}(\boldsymbol{x}) \in \mathbb{R}^d$), we commit a variational crime and evaluate its contribution as

$$
\sum_k (\boldsymbol{\beta}(\boldsymbol{x})\delta(\boldsymbol{x} - \boldsymbol{x}_0), \boldsymbol{w})_{L^2(D^k)} = \int_\Omega \boldsymbol{w} \cdot \boldsymbol{\beta}\delta(\boldsymbol{x} - \boldsymbol{x}_0) = \boldsymbol{w}(\boldsymbol{x}_0) \cdot \boldsymbol{\beta}(\boldsymbol{x}_0).
$$

Finally, we note that, unlike the penalty DG formulation given in Ye et al,[8] material coefficients $\rho, \boldsymbol{C}$ appear only on the left-hand side of (2). Thus, efficient techniques for constant coefficient formulations[24] can be used to evaluate the right-hand side of the formulation, even in the presence of subelement variations in $\rho, \boldsymbol{C}$.

3.1 | Boundary conditions

In this work, we assume boundary conditions on velocity and traction of the form

$$
\boldsymbol{v} = \boldsymbol{v}_{\text{bc}}, \qquad \tilde{\boldsymbol{S}}\boldsymbol{n} = \boldsymbol{t}_{\text{bc}},
$$

where $\boldsymbol{v}_{\text{bc}}$ and $\boldsymbol{t}_{\text{bc}}$ are given values. Traction boundary conditions where $\boldsymbol{t}_{\text{bc}} = 0$ are referred to as free-surface boundary conditions. We follow[20,25] and impose boundary conditions on the DG formulation through exterior values and jumps of the solution. Boundary conditions on the normal component of the stress can be imposed by noting that the numerical flux contains the term $[\![\boldsymbol{A}_n^T\boldsymbol{\sigma}]\!] = [\![\tilde{\boldsymbol{S}}\boldsymbol{n}]\!]$.

For a face which lies on a boundary, velocity boundary conditions are imposed by setting

$$
[\![\boldsymbol{v}]\!] = 2(\boldsymbol{v}_{\text{bc}} - \boldsymbol{v}^-), \qquad [\![\boldsymbol{A}_n^T\boldsymbol{\sigma}]\!] = [\![\tilde{\boldsymbol{S}}\boldsymbol{n}]\!] = 0,
$$

while traction boundary conditions are enforced through

$$
[\![\boldsymbol{A}_n^T\boldsymbol{\sigma}]\!] = [\![\tilde{\boldsymbol{S}}\boldsymbol{n}]\!] = 2\left(\boldsymbol{t}_{\text{bc}} - \tilde{\boldsymbol{S}}^-\boldsymbol{n}\right) = 2\left(\boldsymbol{t}_{\text{bc}} - \boldsymbol{A}_n^T\boldsymbol{\sigma}^-\right), \qquad [\![\boldsymbol{v}]\!] = 0.
$$

For problems which involve the truncation of infinite or large domains, absorbing boundary conditions are required. For such cases, we impose simple extrapolation absorbing boundary conditions[25] through jumps

$$
[\![\boldsymbol{A}_n^T\boldsymbol{\sigma}]\!] = [\![\tilde{\boldsymbol{S}}\boldsymbol{n}]\!] = -\tilde{\boldsymbol{S}}^-\boldsymbol{n} = -\boldsymbol{A}_n^T\boldsymbol{\sigma}^-, \qquad [\![\boldsymbol{v}]\!] = -\boldsymbol{v}^-.
$$

We note that more accurate absorbing conditions can be imposed using, for example, perfectly matched layers[26] or high-order absorbing boundary conditions.[27,28]

In all cases, boundary conditions are imposed by computing numerical fluxes using these modified jumps. This imposition guarantees energy stability for free surface, nonreflective, and homogeneous velocity boundary conditions.

3.2 | Energy stability

One can show that the DG formulation is energy stable for zero body load, zero velocity, and traction boundary conditions, and nonreflective boundary conditions. We note that this stability holds for both the case when τ_v, τ_σ are 0 (which corresponds to a central flux) and when they are positive (which corresponds to a penalty flux). Integrating the velocity

equations in (2) by parts gives

$$
\sum_{D^k \in \Omega_h} \left(\rho \frac{\partial \boldsymbol{v}}{\partial t}, \boldsymbol{w} \right)_{L^2(D^k)} = \sum_{D^k \in \Omega_h} -\left(\sum_{i=1}^{d} \boldsymbol{\sigma}, \boldsymbol{A}_i \frac{\partial \boldsymbol{w}}{\partial \boldsymbol{x}_i} \right)_{L^2(D^k)} + \left\langle \boldsymbol{A}_n^T \{\{\boldsymbol{\sigma}\}\} + \frac{\tau_v}{2} \boldsymbol{A}_n^T \boldsymbol{A}_n [\![\boldsymbol{v}]\!], \boldsymbol{w} \right\rangle_{L^2(\partial D^k)},
$$

$$
\sum_{D^k \in \Omega_h} \left(\boldsymbol{C}^{-1} \frac{\partial \boldsymbol{\sigma}}{\partial t}, \boldsymbol{q} \right)_{L^2(D^k)} = \sum_{D^k \in \Omega_h} \left(\sum_{i=1}^{d} \boldsymbol{A}_i \frac{\partial \boldsymbol{v}}{\partial \boldsymbol{x}_i}, \boldsymbol{q} \right)_{L^2(D^k)} + \left\langle \frac{1}{2} \boldsymbol{A}_n [\![\boldsymbol{v}]\!] + \frac{\tau_\sigma}{2} \boldsymbol{A}_n \boldsymbol{A}_n^T [\![\boldsymbol{\sigma}]\!], \boldsymbol{q} \right\rangle_{L^2(\partial D^k)}.
$$

Taking $(\boldsymbol{w}, \boldsymbol{q}) = (\boldsymbol{v}, \boldsymbol{\sigma})$ and adding both equations together yields

$$
\sum_{D^k \in \Omega_h} \frac{1}{2} \frac{\partial}{\partial t} \left((\rho \boldsymbol{v}, \boldsymbol{v})_{L^2(D^k)} + (\boldsymbol{C}^{-1} \boldsymbol{\sigma}, \boldsymbol{\sigma})_{L^2(D^k)} \right)
$$

$$
= \sum_{D^k \in \Omega_h} \left\langle \boldsymbol{A}_n^T \{\{\boldsymbol{\sigma}\}\} + \frac{\tau_v}{2} \boldsymbol{A}_n^T \boldsymbol{A}_n [\![\boldsymbol{v}]\!], \boldsymbol{v} \right\rangle_{\partial D^k} + \left\langle \frac{1}{2} \boldsymbol{A}_n [\![\boldsymbol{v}]\!] + \frac{\tau_\sigma}{2} \boldsymbol{A}_n \boldsymbol{A}_n^T [\![\boldsymbol{\sigma}]\!], \boldsymbol{\sigma} \right\rangle_{\partial D^k}
$$

$$
= \sum_{D^k \in \Omega_h} \sum_{f \in \partial D^k} \int_f \left(\boldsymbol{v}^T \boldsymbol{A}_n^T \{\{\boldsymbol{\sigma}\}\} + \frac{\tau_v}{2} \boldsymbol{v}^T \boldsymbol{A}_n^T \boldsymbol{A}_n [\![\boldsymbol{v}]\!] + \frac{1}{2} \boldsymbol{\sigma}^T \boldsymbol{A}_n [\![\boldsymbol{v}]\!] + \frac{\tau_\sigma}{2} \boldsymbol{\sigma}^T \boldsymbol{A}_n \boldsymbol{A}_n^T [\![\boldsymbol{\sigma}]\!] \right) \, \mathrm{d}\boldsymbol{x},
$$

where the term

$$
\sum_{D^k \in \Omega_h} \frac{1}{2} \left((\rho \boldsymbol{v}, \boldsymbol{v})_{L^2(D^k)} + (\boldsymbol{C}^{-1} \boldsymbol{\sigma}, \boldsymbol{\sigma})_{L^2(D^k)} \right)
$$

is the total energy of the system. Let Γ_h denote the set of unique faces in Ω_h, and let $\Gamma_v, \Gamma_\sigma, \Gamma_{\text{abc}}$ denote the parts of the boundary where velocity, traction, and nonreflective boundary conditions are imposed, respectively. We separate surface terms into contributions from interior shared faces and from boundary faces. On an interior shared face, we sum contributions from the 2 adjacent elements to yield

$$
\sum_{f \in \Gamma_h \backslash \partial \Omega} \int_f \left(\boldsymbol{v}^T \boldsymbol{A}_n^T \{\{\boldsymbol{\sigma}\}\} + \frac{\tau_v}{2} \boldsymbol{v}^T \boldsymbol{A}_n^T \boldsymbol{A}_n [\![\boldsymbol{v}]\!] + \frac{1}{2} \boldsymbol{\sigma}^T \boldsymbol{A}_n [\![\boldsymbol{v}]\!] + \frac{\tau_\sigma}{2} \boldsymbol{\sigma}^T \boldsymbol{A}_n \boldsymbol{A}_n^T [\![\boldsymbol{\sigma}]\!] \right) \, \mathrm{d}\boldsymbol{x}
$$

$$
= - \sum_{f \in \Gamma_h \backslash \partial \Omega} \int_f \left(\frac{\tau_v}{2} |\boldsymbol{A}_n [\![\boldsymbol{v}]\!]|^2 + \frac{\tau_\sigma}{2} |\boldsymbol{A}_n^T [\![\boldsymbol{\sigma}]\!]|^2 \right) \, \mathrm{d}\boldsymbol{x}.
$$

For faces which lie on the boundary Γ_v where velocity boundary conditions are imposed, $[\![\boldsymbol{v}]\!] = -2\boldsymbol{v}^-$, $[\![\boldsymbol{A}_n^T \boldsymbol{\sigma}]\!] = 0$, and $\boldsymbol{A}_n^T \{\{\boldsymbol{\sigma}\}\} = \boldsymbol{S} \boldsymbol{n}^-$, implying that

$$
\sum_{f \in \Gamma_v} \int_f \left(\boldsymbol{v}^T \boldsymbol{A}_n^T \{\{\boldsymbol{\sigma}\}\} + \frac{\tau_v}{2} \boldsymbol{v}^T \boldsymbol{A}_n^T \boldsymbol{A}_n [\![\boldsymbol{v}]\!] + \frac{1}{2} \boldsymbol{\sigma}^T \boldsymbol{A}_n [\![\boldsymbol{v}]\!] + \frac{\tau_\sigma}{2} \boldsymbol{\sigma}^T \boldsymbol{A}_n \boldsymbol{A}_n^T [\![\boldsymbol{\sigma}]\!] \right) \, \mathrm{d}\boldsymbol{x}
$$

$$
= \sum_{f \in \Gamma_v} \int_f \left((\boldsymbol{v}^-)^T \boldsymbol{A}_n^T \boldsymbol{\sigma}^- - (\boldsymbol{\sigma}^-)^T \boldsymbol{A}_n \boldsymbol{v}^- - \tau_v |\boldsymbol{A}_n \boldsymbol{v}^-|^2 \right) \, \mathrm{d}\boldsymbol{x} = - \sum_{f \in \Gamma_v} \int_f \left(\tau_v |\boldsymbol{A}_n \boldsymbol{v}^-|^2 \right) \, \mathrm{d}\boldsymbol{x}.
$$

For faces which lie on Γ_σ, $\boldsymbol{A}_n^T [\![\boldsymbol{\sigma}]\!] = -2 \boldsymbol{A}_n^T \boldsymbol{\sigma}^-$, $\boldsymbol{A}_n^T \{\{\boldsymbol{\sigma}\}\} = 0$, and $[\![\boldsymbol{v}]\!] = 0$, yielding a similar contribution

$$
\sum_{f \in \Gamma_\sigma} \int_f \left(\boldsymbol{v}^T \boldsymbol{A}_n^T \{\{\boldsymbol{\sigma}\}\} + \frac{\tau_v}{2} \boldsymbol{v}^T \boldsymbol{A}_n^T \boldsymbol{A}_n [\![\boldsymbol{v}]\!] + \frac{1}{2} \boldsymbol{\sigma}^T \boldsymbol{A}_n [\![\boldsymbol{v}]\!] + \frac{\tau_\sigma}{2} \boldsymbol{\sigma}^T \boldsymbol{A}_n \boldsymbol{A}_n^T [\![\boldsymbol{\sigma}]\!] \right) \, \mathrm{d}\boldsymbol{x} = - \sum_{f \in \Gamma_\sigma} \int_f \left(\tau_\sigma |\boldsymbol{A}_n^T \boldsymbol{\sigma}^-|^2 \right) \, \mathrm{d}\boldsymbol{x}.
$$

Finally, for faces in Γ_{abc}, we have $\boldsymbol{A}_n^T \{\{\boldsymbol{\sigma}\}\} = \frac{1}{2} \boldsymbol{A}_n^T \boldsymbol{\sigma}^-$, $\boldsymbol{A}_n^T [\![\boldsymbol{\sigma}]\!] = -\boldsymbol{A}_n^T \boldsymbol{\sigma}^-$, and $[\![\boldsymbol{v}]\!] = -\boldsymbol{v}^-$, yielding

$$
\sum_{f \in \Gamma_{\text{abc}}} \int_f \left(\boldsymbol{v}^T \boldsymbol{A}_n^T \{\{\boldsymbol{\sigma}\}\} + \frac{\tau_v}{2} \boldsymbol{v}^T \boldsymbol{A}_n^T \boldsymbol{A}_n [\![\boldsymbol{v}]\!] + \frac{1}{2} \boldsymbol{\sigma}^T \boldsymbol{A}_n [\![\boldsymbol{v}]\!] + \frac{\tau_\sigma}{2} \boldsymbol{\sigma}^T \boldsymbol{A}_n \boldsymbol{A}_n^T [\![\boldsymbol{\sigma}]\!] \right) \, \mathrm{d}\boldsymbol{x} =
$$

$$
- \sum_{f \in \Gamma_{\text{abc}}} \int_f \left(\frac{\tau_v}{2} |\boldsymbol{A}_n \boldsymbol{v}^-|^2 + \frac{\tau_\sigma}{2} |\boldsymbol{A}_n^T \boldsymbol{\sigma}^-|^2 \right) \, \mathrm{d}\boldsymbol{x}.
$$

Combining all face contributions together gives the following result:

Theorem 3.1. *The DG formulation (2) is energy stable for $\tau_v, \tau_\sigma \geq 0$, in the sense that*

$$
\sum_{D^k \in \Omega_h} \frac{1}{2} \frac{\partial}{\partial t} \left((\rho v, v)_{L^2(D^k)} + \left(C^{-1} \sigma, \sigma \right)_{L^2(D^k)} \right) = - \sum_{f \in \Gamma_h \backslash \partial \Omega} \int_f \left(\frac{\tau_v}{2} |A_n [\![v]\!]|^2 + \frac{\tau_\sigma}{2} |A_n^T [\![\sigma]\!]|^2 \right) \, dx
$$

$$
- \sum_{f \in \Gamma_v} \int_f \left(\tau_v |A_n v^-|^2 \right) \, dx - \sum_{f \in \Gamma_\sigma} \int_f \left(\tau_\sigma |A_n^T \sigma^-|^2 \right) \, dx - \sum_{f \in \Gamma_{abc}} \int_f \left(\frac{\tau_v}{2} |A_n v^-|^2 + \frac{\tau_\sigma}{2} |A_n^T \sigma^-|^2 \right) \, dx \leq 0.
$$

(3)

Since ρ and C^{-1} are positive definite, the left-hand side of (3) is an L^2-equivalent norm on (v, σ), and Theorem 3.1 implies that the magnitude of the DG solution (v, σ) is nonincreasing in time. This also shows that dissipation present for positive penalization constants τ_v, τ_σ acts on nonconforming components with nonzero jumps $A_n^T [\![v]\!]$ and $A_n [\![\sigma]\!]$. In fact, it was shown in Chan and Warburton[21] that, in the limit as $\tau_v, \tau_\sigma \to \infty$, the eigenspaces of DG discretizations split into a conforming part consisting of u, σ which satisfy

$$
A_n [\![u]\!] = 0, \qquad A_n^T [\![\sigma]\!] = 0
$$

and a nonconforming part (defined through the L^2 orthogonal complement) corresponding to eigenvalues contain real parts which approach $-\infty$. For the linear elastic wave equations, these conditions are equivalent to requirements of C^0 continuity for u and normal continuity of the stress tensor $[\![\tilde{S}]\!] n = 0$.

3.3 | The semidiscrete matrix system for DG

The solution to (2) can be approximated by discretizing in space and using an explicit time integrator, which requires only evaluations of local contributions over D^k to the DG formulation

$$
\left(\rho \frac{\partial v}{\partial t}, w \right)_{L^2(D^k)} = \left(\sum_{i=1}^d A_i^T \frac{\partial \sigma}{\partial x_i}, w \right)_{L^2(D^k)} + \left\langle \frac{1}{2} A_n^T [\![\sigma]\!] + \frac{\tau_v}{2} A_n^T A_n [\![v]\!], w \right\rangle_{L^2(\partial D^k)},
$$

$$
\left(C^{-1} \frac{\partial \sigma}{\partial t}, q \right)_{L^2(D^k)} = \left(\sum_{i=1}^d A_i \frac{\partial v}{\partial x_i}, q \right)_{L^2(D^k)} + \left\langle \frac{1}{2} A_n [\![v]\!] + \frac{\tau_\sigma}{2} A_n A_n^T [\![\sigma]\!], q \right\rangle_{L^2(\partial D^k)}.
$$

(4)

Let $\{\phi_i\}_{i=1}^{N_p}$ be a basis for $P^N (\hat{D})$.* We define the reference mass matrix $\widehat{M}$ and the physical mass matrix M for an element D^k as

$$
\left(\widehat{M} \right)_{ij} = \int_{\hat{D}} \phi_j \phi_i d\hat{x}, \qquad (M)_{ij} = \int_{D^k} \phi_j \phi_i \, dx = \int_{\hat{D}} \phi_j \phi_i J d\hat{x}.
$$

For affine mappings, J is constant and $M = J\widehat{M}$. We also define weak differentiation matrices S_k and face mass matrix M_f such that

$$
(S_k)_{ij} = \int_{D^k} \frac{\partial \phi_j}{\partial x_k} \phi_i \, dx, \qquad \left(M_f \right)_{ij} = \int_f \phi_j \phi_i \, dx = \int_{\hat{f}} \phi_j \phi_i J^f d\hat{x},
$$

where J^f is the Jacobian of the mapping from a reference face $\hat{f}$ to f. For affinely mapped simplices, J^f is also constant and $M_f = J^f \widehat{M}_f$, where the definition of the reference face mass matrix $\widehat{M}_f$ is analogous to the definition of the reference mass matrix $\widehat{M}$.

Finally, we define weighted mass matrices. Let $w(x) \in \mathbb{R}$ and $W(x) \in \mathbb{R}^{m \times n}$. Then, scalar and matrix-weighted mass matrices M_w and M_W are defined through

$$
(M_w)_{ij} = \int_{D^k} w(x) \phi_j(x) \phi_i(x) \, dx, \qquad M_W = \begin{pmatrix} M_{W_{1,1}} & \cdots & M_{W_{1,n}} \\ \vdots & \ddots & \vdots \\ M_{W_{m,1}} & \cdots & M_{W_{m,n}} \end{pmatrix},
$$

where $M_{W_{i,j}}$ is the scalar weighted mass matrix weighted by the (i, j) entry of W.

---

*In our implementations, we use nodal basis functions at Warp and Blend interpolation points.[29] These basis functions are defined implicitly using an orthogonal polynomial basis on the reference simplex.[23] However, we note that the implementation and formulation are independent of the specific choice of polynomial basis.

Local contributions to the DG variational form may then be evaluated in a quadrature-free manner using these matrices. Let $\boldsymbol{\Sigma}_i, \boldsymbol{V}_i$ denote vectors containing degrees of freedom for solution components $\boldsymbol{\sigma}_i, \boldsymbol{v}_i$ such that

$$\boldsymbol{\sigma}_i(\boldsymbol{x}, t) = \sum_{j=1}^{N_p} (\boldsymbol{\Sigma}_i(t))_j \phi_j(\boldsymbol{x}), \qquad 1 \leq i \leq N_d,$$

$$\boldsymbol{v}_i(\boldsymbol{x}, t) = \sum_{j=1}^{N_p} (\boldsymbol{V}_i(t))_j \phi_j(\boldsymbol{x}), \qquad 1 \leq i \leq d.$$

Then, the local DG formulation can be written as a block system of ODEs by concatenating $\boldsymbol{\Sigma}_i, \boldsymbol{V}_i$ into single vectors $\boldsymbol{\Sigma}, \boldsymbol{V}$ and using the Kronecker product $\otimes$

$$\boldsymbol{M}_{\rho \boldsymbol{I}} \frac{\partial \boldsymbol{V}}{\partial t} = \sum_{i=1}^{d} \left(\boldsymbol{A}_i^T \otimes \boldsymbol{S}_i \right) \boldsymbol{\Sigma} + \sum_{f \in \partial D^k} \left(\boldsymbol{I} \otimes \boldsymbol{M}_f \right) \boldsymbol{F}_v,$$

$$\boldsymbol{M}_{\boldsymbol{C}^{-1}} \frac{\partial \boldsymbol{\Sigma}}{\partial t} = \sum_{i=1}^{d} \left(\boldsymbol{A}_i \otimes \boldsymbol{S}_i \right) \boldsymbol{V} + \sum_{f \in \partial D^k} \left(\boldsymbol{I} \otimes \boldsymbol{M}_f \right) \boldsymbol{F}_\sigma. \tag{5}$$

where $\boldsymbol{F}_v$ and $\boldsymbol{F}_\sigma$ denote degrees of freedom for velocity and stress numerical fluxes.

To apply standard time integration methods, we must invert $\boldsymbol{M}_{\rho \boldsymbol{I}}$ and $\boldsymbol{M}_{\boldsymbol{C}^{-1}}$ to isolate $\frac{\partial v}{\partial t}$ and $\frac{\partial \sigma}{\partial t}$ on the left-hand side. While the inversion of $\boldsymbol{M}_{\rho \boldsymbol{I}}$ and $\boldsymbol{M}_{\boldsymbol{C}^{-1}}$ can be parallelized from element to element, doing so typically requires either the precomputation and storage of large dense matrix inverses or the on-the-fly construction and solution of a large dense matrix system at every time-step. The former option requires a large amount of storage, while the latter option is computationally expensive and difficult to parallelize. This cost can be overcome for $\rho, \boldsymbol{C}$ which are constant over an element D^k, in which case $\boldsymbol{M}_{\rho \boldsymbol{I}}$ is block diagonal with identical blocks $\boldsymbol{M}_\rho = \rho \boldsymbol{M}$, while $\boldsymbol{M}_{\boldsymbol{C}^{-1}}$ reduces to

$$\boldsymbol{M}_{\boldsymbol{C}^{-1}} = \begin{pmatrix} C_{1,1}^{-1} \boldsymbol{M} & \cdots & C_{1,N_d}^{-1} \boldsymbol{M} \\ \vdots & \ddots & \vdots \\ C_{N_d,1}^{-1} \boldsymbol{M} & \cdots & C_{N_d,N_d}^{-1} \boldsymbol{M} \end{pmatrix} = \left(\boldsymbol{C}^{-1} \otimes \boldsymbol{M} \right).$$

Then, $\boldsymbol{M}_\rho^{-1} = \frac{1}{\rho} \boldsymbol{M}^{-1} = \frac{1}{J\rho} \widehat{\boldsymbol{M}}^{-1}$, and $\boldsymbol{M}_{\boldsymbol{C}^{-1}}^{-1} = \boldsymbol{C} \otimes \boldsymbol{M}^{-1} = \boldsymbol{C} \otimes \left(\frac{1}{J} \widehat{\boldsymbol{M}}^{-1} \right)$, and each matrix inverse can be applied using the inverse of the reference mass matrix $\widehat{\boldsymbol{M}}^{-1}$ and the values of $\rho, \boldsymbol{C}$, and J over each element. Applying this observation to (5) then yields the following local system of ODEs:

$$\frac{\partial \boldsymbol{V}}{\partial t} = \sum_{i=1}^{d} \left(\frac{1}{\rho} \boldsymbol{A}_i^T \otimes \boldsymbol{D}_i \right) \boldsymbol{\Sigma} + \sum_{f \in \partial D^k} \left(\frac{1}{\rho} \boldsymbol{I} \otimes \boldsymbol{L}_f \right) \boldsymbol{F}_v,$$

$$\frac{\partial \boldsymbol{\Sigma}}{\partial t} = \sum_{i=1}^{d} \left(\boldsymbol{C} \boldsymbol{A}_i \otimes \boldsymbol{D}_i \right) \boldsymbol{V} + \sum_{f \in \partial D^k} \left(\boldsymbol{C} \otimes \boldsymbol{L}_f \right) \boldsymbol{F}_\sigma,$$

where we have introduced the differentiation matrix $\boldsymbol{D}_i = \boldsymbol{M}^{-1} \boldsymbol{S}_i$ and lift matrix $\boldsymbol{L}_f = \boldsymbol{M}^{-1} \boldsymbol{M}_f$. For affine elements, both derivative and lift matrices may be applied using only geometric factors and reference derivative and lift matrices.

Unfortunately, if ρ and $\boldsymbol{C}$ vary spatially within an element, the above approach can no longer be used to invert $\boldsymbol{M}_\rho$ and $\boldsymbol{M}_{\boldsymbol{C}^{-1}}$ in an efficient and low-storage manner. For isotropic media, 1 way to address subelement variations in material parameters is to diagonalize the matrix $\boldsymbol{C}$ through a change of variables.[30] This results in a local system of ODEs with only scalar weighted mass matrices,[14] which can be treated using scalar weight-adjusted approximations. We take a different approach in addressing these issues and approximate the matrix-weighted L^2 inner product (and corresponding matrix-weighted mass matrix $\boldsymbol{M}_{\boldsymbol{C}^{-1}}$) using a weight-adjusted approximation which is low storage, simple to invert, energy stable, and provably high-order accurate for spatially varying weights $\rho, \boldsymbol{C}$ with sufficiently regularity.

4 | WEIGHT-ADJUSTED INNER PRODUCTS FOR MATRIX-VALUED WEIGHTS

Weight-adjusted inner products are high-order accurate approximations of weighted L^2 inner products. These can be interpreted as generalizations of mass lumping techniques, reducing to mass lumping when integrals are evaluated with

appropriate quadrature rules. These weight-adjusted inner products result in weight-adjusted mass matrices, whose inverses approximate the inverse of a weighted L^2 mass matrix.

We wish to apply weight-adjusted approximations to avoid the inversion of $\boldsymbol{M}_\rho$ and $\boldsymbol{M}_{C^{-1}}$. Approximating the inverse of $\boldsymbol{M}_\rho$ can be done using weight-adjusted approximations for scalar weights,[16,17] which we review in Section 4.1. We then extend scalar weight-adjusted approximations to matrix-valued weights in Section 4 to approximate the inverse of $\boldsymbol{M}_{C^{-1}}$.

4.1 | Scalar weight-adjusted inner products

We introduce standard Lebesgue L^p norms and their associated L^p spaces over a general domain Ω

$$\|u\|_{L^p(\Omega)} = \left(\int_\Omega u^p \, d\boldsymbol{x} \right)^{1/p}, \qquad L^p(\Omega) = \left\{ u : \Omega \to \mathbb{R}, \quad \|u\|_{L^p(\Omega)} < \infty \right\}$$

for $1 \leq p < \infty$. For $p = \infty$, these are defined as

$$\|u\|_{L^\infty(\Omega)} = \inf \left\{ C \geq 0 : |u(\boldsymbol{x})| \leq C \quad \forall \boldsymbol{x} \in \Omega \right\}, \qquad L^\infty(\Omega) = \left\{ u : \Omega \to \mathbb{R}, \quad \|u\|_{L^\infty(\Omega)} < \infty \right\}.$$

These induce L^p Sobolev seminorms and norms of degree s

$$|u|_{W^{s,p}(\Omega)} = \left(\sum_{|\alpha|=s} \|D^\alpha u\|_{L^p(\Omega)}^p \right)^{1/p}, \qquad |u|_{W^{s,\infty}(\Omega)} = \max_{|\alpha|=s} \|D^\alpha u\|_{L^\infty(\Omega)},$$

$$\|u\|_{W^{s,p}(\Omega)} = \left(\sum_{|\alpha|\leq s} \|D^\alpha u\|_{L^p(\Omega)}^p \right)^{1/p}, \qquad \|u\|_{W^{s,\infty}(\Omega)} = \max_{|\alpha|\leq s} \|D^\alpha u\|_{L^\infty(\Omega)},$$

where $\alpha = \{\alpha_1, \ldots, \alpha_d\}$ is a multi-index such that

$$D^\alpha u = \frac{\partial^{\alpha_1}}{\partial x^{\alpha_1}} \frac{\partial^{\alpha_2}}{\partial y^{\alpha_2}} u, \qquad d = 2,$$

$$D^\alpha u = \frac{\partial^{\alpha_1}}{\partial x^{\alpha_1}} \frac{\partial^{\alpha_2}}{\partial y^{\alpha_2}} \frac{\partial^{\alpha_3}}{\partial z^{\alpha_3}} u, \qquad d = 3.$$

Let Π_N denote the L^2 projection on the element D^k. For affine elements where J is constant such that

$$(wu, v)_{L^2(D^k)} = (wu, vJ)_{\hat{D}} = J(wu, v)_{\hat{D}},$$

Π_N is equivalent to the L^2 projection on the reference element $\hat{D}$. We define 2 operators $T_w : L^2(D^k) \to P^N(D^k)$ and $T_w^{-1} : L^2(D^k) \to P^N(D^k)$ such that

$$T_w u = \Pi_N(wu),$$
$$\left(w T_w^{-1} u, v \right)_{L^2(D^k)} = (u, v)_{L^2(D^k)}, \qquad \forall v \in P^N(D^k).$$

A weighted L^2 inner product $(wu, v)_{L^2(D^k)}$ can be approximated by a weight-adjusted inner product

$$(wu, v)_{L^2(D^k)} = (T_w u, v)_{L^2(D^k)} \approx \left(T_{1/w}^{-1} u, v \right)_{L^2(D^k)}$$

based on the observation that $T_{1/w}^{-1} u \approx uw$. The intuition behind this approximation is that, by the definition of $T_{1/w}^{-1}$,

$$\left(\frac{1}{w} T_{1/w}^{-1} u - u, v \right)_{L^2(D^k)} = 0, \qquad \forall v \in P^N(D^k).$$

This shows that $\frac{1}{w} T_{1/w}^{-1} u - u$ is orthogonal to all polynomials of degree N, implying that $\frac{1}{w} T_{1/w}^{-1} u - u \approx 0$ and $T_{1/w}^{-1} u \approx wu$ (for $w(x), u(x)$ which are smooth and well-represented by polynomials).

This approximation is made precise by the following estimates for approximations of the product uw and weighted moments on affinely mapped elements:

Theorem 4.1. *Let D^k be quasi-regular with representative size $h = diam\left(D^k\right)$. For $N \geq 0$, $w \in W^{N+1,\infty}\left(D^k\right)$, and $u \in W^{N+1,2}\left(D^k\right)$,*[16, (theorem 5)]

$$\|uw - T_w u\|_{L^2(D^k)} \leq C_w h^{N+1} \|u\|_{W^{N+1,2}(D^k)}, \tag{6}$$

$$\left\|uw - T_{1/w}^{-1} u\right\|_{L^2(D^k)} \leq C_w h^{N+1} \|u\|_{W^{N+1,2}(D^k)}, \tag{7}$$

where $C_w = C\|w\|_{L^\infty(D^k)} \left\|\frac{1}{w}\right\|_{L^\infty(D^k)} \|w\|_{W^{N+1,\infty}(D^k)}$.

These results rely on a scalar weighted interpolation estimate derived in Chan et al and Warburton[17,18] for a general nonaffine element D^k.

Theorem 4.2. *Let D^k be a quasi-regular element with representative size $h = diam\left(D^k\right)$. For $N \geq 0$, $w \in W^{N+1,\infty}\left(D^k\right)$, and $u \in W^{N+1,2}\left(D^k\right)$,*[17, (theorem 1)]

$$\left\|u - \frac{1}{w}\Pi_N(wu)\right\|_{L^2(D^k)} \leq Ch^{N+1}\left\|\frac{1}{\sqrt{J}}\right\|_{L^\infty(D^k)}\left\|\frac{\sqrt{J}}{w}\right\|_{L^\infty(D^k)}\|w\|_{W^{N+1,\infty}(D^k)}\|u\|_{W^{N+1,2}(D^k)}.$$

4.2 | Extension to matrix weights

We now generalize weight-adjusted inner products to the case of matrix-valued weights. We first define appropriate generalizations of norms used in Section 4.1 to vector-valued functions. Let $D^\alpha v$ denote component-wise differentiation of v with respect to a d-dimensional multi-index α. Then, vector L^p Sobolev norms for $v(x) \in \mathbb{R}^m$ can be defined as

$$|v|_{W^{k,p}}^p = \sum_{i=1}^m |v_i|_{W^{k,p}}^p, \qquad \|v\|_{W^{k,p}}^p = \sum_{i=1}^m \|v_i\|_{W^{k,p}}^p \qquad 1 \leq p < \infty,$$

$$|v|_{W^{k,\infty}} = \max_i |v_i|_{W^{k,\infty}}, \qquad \|v\|_{W^{k,\infty}} = \max_i \|v_i\|_{W^{k,\infty}}.$$

The corresponding Sobolev spaces $W^{k,p}$ and $W^{k,\infty}$ are defined similarly to the scalar case.

Let $W(x)$ be a matrix-valued weight function which is pointwise symmetric positive-definite

$$0 < w_{\min} \leq \|W(x)\|_2 \leq w_{\max} < \infty, \qquad 0 < \tilde{w}_{\min} \leq \|W^{-1}(x)\|_2 \leq \tilde{w}_{\max} < \infty, \qquad \forall x \in \Omega.$$

We define a kth-order Sobolev norm for $W(x)$ in terms of the induced p-norm

$$\|W(x)\|_{k,p,\infty}^p = \sum_{|\alpha| \leq k} \sup_x \|D^\alpha W(x)\|_p^p,$$

where $D^\alpha W(x)$ again denotes component-wise differentiation. While this norm is not submultiplicative, the following bound holds

$$\|Wv\|_{W^{k,p}}^p = \sum_{|\alpha| \leq k} \|D^\alpha (Wv)\|_{L^p}^p \leq C_N \int \sum_{|\alpha| \leq k} \sum_{|\beta| \leq |\alpha|} \left\|\left(D^\beta W\right)\left(D^{\alpha-\beta} v\right)\right\|_p^p \, dx$$

$$\leq C_N \int \left(\sum_{|\alpha| \leq k} \|(D^\alpha W)\|_p\right)^p \left(\sum_{|\alpha| \leq k} \|D^\alpha v\|_p\right)^p \, dx$$

$$\leq C_N \|W\|_{k,p,\infty}^p \|v\|_{k,p}^p,$$

where we have used Leibniz rule, Cauchy-Schwarz, and the arithmetic-geometric mean inequality.

The following theorem extends Theorem 4.2 to matrix weights by computing weighted interpolation estimates for the quantity $W^{-1}\Pi_N(Wv)$.

Theorem 4.3. *Let D^k be a quasi-regular element with representative size $h = diam\left(D^k\right)$. For $N \geq 0$, $W \in \left(W^{N+1,\infty}\left(D^k\right)\right)^{d \times d}$, and $v \in \left(W^{N+1,2}\left(D^k\right)\right)^d$,*

$$\left\|v - W^{-1}\Pi_N(Wv)\right\|_{L^2(D^k)} \leq Ch^{N+1}\left\|\sqrt{J}\right\|_{L^\infty(D^k)}\left\|\frac{1}{\sqrt{J}}\right\|_{L^\infty(D^k)} \tilde{w}_{\max} \|W\|_{N+1,2,\infty} \|v\|_{W^{N+1,2}(D^k)}.$$

Proof. The proof is similar to the scalar case. Using vector-valued versions of Bramble-Hilbert and a scaling argument for quasi-regular elements yields

$$
\begin{aligned}
\left\| \boldsymbol{v} - \boldsymbol{W}^{-1}\Pi_N\left(\boldsymbol{W}\boldsymbol{v}\right)\right\|_{L^2(D^k)} &\leq C_1 \left\|\sqrt{J}\right\|_{L^\infty(D^k)} \sup_{\boldsymbol{x}} \left\|\boldsymbol{W}^{-1}\right\|_2 \left\|\boldsymbol{W}\boldsymbol{v} - \Pi_N\left(\boldsymbol{W}\boldsymbol{v}\right)\right\|_{L^2(\hat{D})} \\
&\leq C_1 \left\|\sqrt{J}\right\|_{L^\infty(D^k)} \sup_{\boldsymbol{x}} \left\|\boldsymbol{W}^{-1}\right\|_2 \left|\boldsymbol{W}\boldsymbol{v}\right|_{W^{N+1,2}(\hat{D})} \\
&\leq C_2 h^{N+1} \left\|\sqrt{J}\right\|_{L^\infty(D^k)} \left\|\frac{1}{\sqrt{J}}\right\|_{L^\infty(D^k)} \sup_{\boldsymbol{x}} \left\|\boldsymbol{W}^{-1}\right\|_2 \left\|\boldsymbol{W}\boldsymbol{v}\right\|_{W^{N+1,2}(D^k)} \\
&\leq C_3 h^{N+1} \left\|\sqrt{J}\right\|_{L^\infty(D^k)} \left\|\frac{1}{\sqrt{J}}\right\|_{L^\infty(D^k)} \tilde{w}_{\max} \left\|\boldsymbol{W}\right\|_{N+1,2,\infty} \left\|\boldsymbol{v}\right\|_{W^{N+1,2}(D^k)}.
\end{aligned}
$$

$\square$

4.2.1 | Weight-adjusted approximations with matrix weights

Let $\Pi_N \boldsymbol{u}$ be defined as the L^2 projection applied to each component of the vector-valued function $\boldsymbol{u}$. We then define the operator $T_{\boldsymbol{W}}$ analogously to the scalar case

$$
T_{\boldsymbol{W}}\boldsymbol{v} = \Pi_N\left(\boldsymbol{W}\boldsymbol{v}\right).
$$

The inverse operator $T_{\boldsymbol{W}}^{-1}$ is defined implicitly via

$$
\left(\boldsymbol{W}T_{\boldsymbol{W}}^{-1}\boldsymbol{v}, \delta\boldsymbol{v}\right)_{L^2(D^k)} = (\boldsymbol{v}, \delta\boldsymbol{v})_{L^2(D^k)}, \qquad \forall \delta\boldsymbol{v} \in \left(P^N\left(D^k\right)\right)^m.
$$

This definition is a straightforward generalization of T_w^{-1} to matrix-valued weights $\boldsymbol{W}$. Conveniently, all properties of T_w, T_w^{-1} for scalar $w(\boldsymbol{x})$[16] carry over to matrix weights $\boldsymbol{W}(\boldsymbol{x})$ as well.

Lemma 4.4. *Let Π_N denote the component-wise L^2 projection, and let $\boldsymbol{W} \in (L^\infty)^{m \times m}$. Then, $T_{\boldsymbol{W}}$ satisfies the following properties:*

1. *$T_{\boldsymbol{W}}^{-1} T_{\boldsymbol{W}} = \Pi_N$.*
2. *$\Pi_N T_{\boldsymbol{W}}^{-1} = T_{\boldsymbol{W}}^{-1}\Pi_N = T_{\boldsymbol{W}}^{-1}$.*
3. *$\left\|T_{\boldsymbol{W}}^{-1}\right\|_{L^2(D^k)} \leq \tilde{w}_{\max}$.*
4. *$\left(T_{\boldsymbol{W}}^{-1}\boldsymbol{v}, \boldsymbol{w}\right)_{L^2(D^k)}$ forms an inner product on $\left(P^N\right)^m \times \left(P^N\right)^m$, which is equivalent to the L^2 inner product with equivalence constants $C_1 = \tilde{w}_{\min}, C_2 = \tilde{w}_{\max}$.*

Proof. The proofs of properties 1 and 2 are consequences of the definition of $T_{\boldsymbol{W}}$ and are identical to proofs for the scalar case. Property 3 is a straightforward extension from the scalar case. Let $\boldsymbol{v} \in \left(P^N\right)^m$. Then,

$$
\left(T_{\boldsymbol{W}}^{-1}\boldsymbol{v}, \boldsymbol{v}\right)_{L^2(D^k)} = \left(\boldsymbol{W}^{-1}\boldsymbol{W}T_{\boldsymbol{W}}^{-1}\boldsymbol{v}, \boldsymbol{v}\right)_{L^2(D^k)} \leq \sup_{\boldsymbol{x}} \left\|\boldsymbol{W}^{-1}(\boldsymbol{x})\right\| \left(\boldsymbol{W}T_{\boldsymbol{W}}^{-1}\boldsymbol{v}, \boldsymbol{v}\right)_{L^2(D^k)} = \tilde{w}_{\max} \left\|\boldsymbol{v}\right\|_{L^2(D^k)}^2.
$$

Property 4 then simply requires the lower bound

$$
\left(T_{\boldsymbol{W}}^{-1}\boldsymbol{v}, \boldsymbol{v}\right)_{L^2(D^k)} = \left(\boldsymbol{W}^{-1}\boldsymbol{W}T_{\boldsymbol{W}}^{-1}\boldsymbol{v}, \boldsymbol{v}\right)_{L^2(D^k)} \geq \inf_{\boldsymbol{x}} \left\|\boldsymbol{W}^{-1}(\boldsymbol{x})\right\| \left\|\boldsymbol{v}\right\|_{L^2(D^k)}^2 = \tilde{w}_{\min} \left\|\boldsymbol{v}\right\|_{L^2(D^k)}^2.
$$

$\square$

Using Theorem 4.3, we may also show that the matrix-valued weight-adjusted inner product is also high-order accurate for sufficiently regular $\boldsymbol{W}$.

Theorem 4.5. *Let D^k be a quasi-regular element with representative size $h = diam\left(D^k\right)$. For $N > 0$,*

$$
\left\|\boldsymbol{W}\boldsymbol{v} - T_{\boldsymbol{W}^{-1}}\boldsymbol{v}\right\|_{L^2(D^k)} \leq C_{\boldsymbol{W}} h^{N+1} \left\|\boldsymbol{v}\right\|_{W^{N+1,2}(D^k)}
$$

with constant $C_{\boldsymbol{W}}$ depending on $\boldsymbol{W}$ and N

$$
C_{\boldsymbol{W}} = C_N \left\|\sqrt{J}\right\|_{L^\infty(D^k)} \left\|\frac{1}{\sqrt{J}}\right\|_{L^\infty(D^k)} \tilde{w}_{\max} w_{\max} \left\|\boldsymbol{W}\right\|_{N+1,2,\infty}.
$$

Proof. The proof follows the scalar case. The triangle inequality gives

$$\left\| \boldsymbol{W}\boldsymbol{v} - T_{\boldsymbol{W}^{-1}}^{-1}\boldsymbol{v} \right\|_{L^2(D^k)} \leq \left\| \boldsymbol{W}\boldsymbol{v} - \Pi_N(\boldsymbol{W}\boldsymbol{v}) \right\|_{L^2(D^k)} + \left\| \Pi_N(\boldsymbol{W}\boldsymbol{v}) - T_{\boldsymbol{W}^{-1}}^{-1}\boldsymbol{v} \right\|_{L^2(D^k)}.$$

The former term is bounded by interpolation estimates and by arguments used in the proof of Theorem 4.3. The latter term is bounded

$$
\begin{aligned}
\left\| \Pi_N(\boldsymbol{W}\boldsymbol{v}) - T_{\boldsymbol{W}^{-1}}^{-1}\boldsymbol{v} \right\|_{L^2(D^k)} &= \left\| T_{\boldsymbol{W}^{-1}}^{-1}\Pi_N\left(T_{\boldsymbol{W}^{-1}}\Pi_N(\boldsymbol{W}\boldsymbol{v})\right) - T_{\boldsymbol{W}^{-1}}^{-1}\Pi_N\boldsymbol{v} \right\|_{L^2(D^k)} \\
&\leq C_N \left\| T_{\boldsymbol{W}^{-1}}^{-1} \right\| \left\| T_{\boldsymbol{W}^{-1}}\Pi_N(\boldsymbol{W}\boldsymbol{v}) - \Pi_N\boldsymbol{v} \right\|_{L^2(D^k)} \\
&\leq C_N \left\| T_{\boldsymbol{W}^{-1}}^{-1} \right\| \left\| \Pi_N\left(\boldsymbol{W}^{-1}\Pi_N(\boldsymbol{W}\boldsymbol{v})\right) - \Pi_N\boldsymbol{v} \right\|_{L^2(D^k)} \\
&\leq C_N w_{\max} \left\| \boldsymbol{W}^{-1}\Pi_N(\boldsymbol{W}\boldsymbol{v}) - \boldsymbol{v} \right\|_{L^2(D^k)},
\end{aligned}
$$

where we have used $\left\| T_{\boldsymbol{W}^{-1}}^{-1} \right\| \leq w_{\max}$ (Lemma 4.4) and $\|\Pi_N\|_{L^2(D^k)} = 1$. An application of Theorem 4.3 to bound $\left\| \boldsymbol{W}^{-1}\Pi_N(\boldsymbol{W}\boldsymbol{v}) - \boldsymbol{v} \right\|_{L^2(D^k)}$ completes the proof. $\qquad\square$

We note that these estimates are tight, in the sense that estimates for scalar weight-adjusted inner products are recovered when the matrix weight is taken to be $\boldsymbol{W} = w\boldsymbol{I}$.

4.2.2 | Approximation of weighted mass matrix inverses

The advantage of using weight-adjusted inner products is that the corresponding weight-adjusted mass matrices are straightforward to invert. For scalar weights, the weight-adjusted mass matrix approximates the weighted L^2 mass matrix and its inverse by

$$\boldsymbol{M}_w \approx \boldsymbol{M}\boldsymbol{M}_{1/w}^{-1}\boldsymbol{M}, \qquad \boldsymbol{M}_w^{-1} \approx \boldsymbol{M}^{-1}\boldsymbol{M}_{1/w}\boldsymbol{M}^{-1}.$$

By evaluating $\boldsymbol{M}_{1/w}$ in a matrix-free fashion, the inverse of the weight-adjusted mass matrix $\boldsymbol{M}^{-1}\boldsymbol{M}_{1/w}\boldsymbol{M}^{-1}$ yields a low storage implementation using a sufficiently accurate quadrature rule. Let $\widehat{\boldsymbol{x}}_i, \widehat{w}_i$ denote quadrature points and weights on the reference element, and let $\boldsymbol{V}_q$ denote the matrix

$$\left(\boldsymbol{V}_q\right)_{ij} = \phi_j(\widehat{\boldsymbol{x}}_i).$$

whose columns correspond to evaluations of basis functions at quadrature points. Then, for affine elements, $\boldsymbol{M} = J\widehat{\boldsymbol{M}} = J\boldsymbol{V}_q^T\text{diag}(\widehat{w}_i)\boldsymbol{V}_q$, where $\widehat{\boldsymbol{M}}$ is the reference mass matrix and J is the determinant of the Jacobian of the reference-to-physical mapping, which is constant for affine mappings. Additionally,

$$\boldsymbol{M}_{1/w} = J\boldsymbol{V}_q^T\text{diag}(\widehat{w}_i/w(\widehat{\boldsymbol{x}}_i))\boldsymbol{V}_q,$$

where $w(\widehat{\boldsymbol{x}}_i)$ denotes the evaluation of the weight function $w(\boldsymbol{x})$ at quadrature points. Thus, for a vector $\boldsymbol{u}$, the inverse of the weight-adjusted mass matrix can be applied as follows

$$\boldsymbol{M}^{-1}\boldsymbol{M}_{1/w}\boldsymbol{M}^{-1}\boldsymbol{u} = \boldsymbol{M}^{-1}\boldsymbol{V}_q^T\text{diag}(\widehat{w}_i)\text{diag}\left(\frac{1}{w(\boldsymbol{x}_i)}\right)\boldsymbol{V}_q\boldsymbol{M}^{-1}\boldsymbol{u} = \boldsymbol{P}_q\text{diag}\left(\frac{1}{w(\widehat{\boldsymbol{x}}_i)}\right)\boldsymbol{V}_q\frac{1}{J}\widehat{\boldsymbol{M}}^{-1}\boldsymbol{u},$$

where we have introduced the quadrature-based L^2 projection operator on the reference element $\boldsymbol{P}_q = \widehat{\boldsymbol{M}}^{-1}\boldsymbol{V}_q^T\text{diag}(\widehat{w}_i)$.

In the context of DG using explicit time-stepping, the factor of $\widehat{\boldsymbol{M}}^{-1}$ can be premultiplied into the right-hand side (ie, the evaluation of the spatial discretization). Then, applying the weight-adjusted mass matrix requires only storage of 2 reference matrices $\boldsymbol{V}_q$ and $\boldsymbol{P}_q$ and the values of the weight function at quadrature points $w(\widehat{\boldsymbol{x}}_i)$. We assume that the number of quadrature points is $O(N^3)$, which is true for most simplicial quadratures.[31,32]

The overall storage cost of applying the weight-adjusted mass matrix using the above implementation is $O(N^3)$ per element, while the precomputation and storage of DG operators involving inverses of weighted mass matrices requires $O(N^6)$ storage per element. However, we note that unlike the aforementioned quadrature-based implementation of WADG, the strategy of precomputation and storage used by Mercerat and Glinsky[14] can accomodate arbitrarily high-accuracy quadrature rules without any increase in computational cost.

Finally, we note that the computational impact of storage costs vary from architecture to architecture. As pointed out in Chan et al and Warburton,[16-18] the limited storage of accelerator architectures such as GPUs limits the maximum feasible problem size and decreasing storage costs with respect to the degree N allows one to run higher-order simulations on larger meshes before running out of memory. However, for distributed parallelism implementations of DG on large supercomputing clusters, storage limitations may be less of an issue. We limit our focus to GPU computations in this work and present results comparing the computational cost of WADG to several alternatives in Section 6.4.

For weight-adjusted inner products with matrix-valued weights, the corresponding weight-adjusted mass matrices approximate weighted L^2 mass matrices and inverses in a similar fashion

$$
\begin{aligned}
\boldsymbol{M_W} &\approx (\boldsymbol{I} \otimes \boldsymbol{M}) \, \boldsymbol{M}_{\boldsymbol{W}^{-1}}^{-1} \, (\boldsymbol{I} \otimes \boldsymbol{M}) , \\
\boldsymbol{M_W}^{-1} &\approx \left(\boldsymbol{I} \otimes \boldsymbol{M}^{-1} \right) \boldsymbol{M}_{\boldsymbol{W}^{-1}} \left(\boldsymbol{I} \otimes \boldsymbol{M}^{-1} \right) .
\end{aligned}
\tag{8}
$$

We note that, when $\boldsymbol{W}$ is constant over D^k, $\boldsymbol{M_W}$ reduces to the Kronecker product of the inverse stiffness tensor and the local mass matrix

$$
\boldsymbol{M_W} = \boldsymbol{W} \otimes \boldsymbol{M}, \qquad \boldsymbol{M_W}^{-1} = \boldsymbol{W}^{-1} \otimes \boldsymbol{M}^{-1}.
$$

In this case, we also have $\boldsymbol{M}_{\boldsymbol{W}^{-1}} = \boldsymbol{W}^{-1} \otimes \boldsymbol{M}$, and substituting this explicit inverse into the weight-adjusted mass matrix inverse $\left(\boldsymbol{I} \otimes \boldsymbol{M}^{-1} \right) \boldsymbol{M}_{\boldsymbol{W}^{-1}} \left(\boldsymbol{I} \otimes \boldsymbol{M}^{-1} \right)$ in (8) recovers the exact inversion of $\boldsymbol{M_W}$.

5 ⎸ AN ENERGY-STABLE WADG FORMULATION FOR ELASTIC WAVE PROPAGATION

We construct a WADG method by simply replacing the weighted L^2 inner products appearing in the left-hand side of the local DG formulation (4) with weight-adjusted approximations

$$
\begin{aligned}
\left(\boldsymbol{T}_{1/\rho}^{-1} \frac{\partial \boldsymbol{v}}{\partial t}, \boldsymbol{w} \right)_{L^2(D^k)} &= \left(\sum_{i=1}^{d} \boldsymbol{A}_i^T \frac{\partial \boldsymbol{\sigma}}{\partial \boldsymbol{x}_i}, \boldsymbol{w} \right)_{L^2(D^k)} + \left\langle \frac{1}{2} \boldsymbol{A}_n^T [\![\boldsymbol{\sigma}]\!] + \frac{\tau_v}{2} \boldsymbol{A}_n \boldsymbol{A}_n^T [\![\boldsymbol{v}]\!], \boldsymbol{w} \right\rangle_{L^2(\partial D^k)}, \\
\left(\boldsymbol{T}_C^{-1} \frac{\partial \boldsymbol{\sigma}}{\partial t}, \boldsymbol{q} \right)_{L^2(D^k)} &= \left(\sum_{i=1}^{d} \boldsymbol{A}_i \frac{\partial \boldsymbol{v}}{\partial \boldsymbol{x}_i}, \boldsymbol{q} \right)_{L^2(D^k)} + \left\langle \frac{1}{2} \boldsymbol{A}_n [\![\boldsymbol{v}]\!] + \frac{\tau_\sigma}{2} \boldsymbol{A}_n^T \boldsymbol{A}_n [\![\boldsymbol{\sigma}]\!], \boldsymbol{q} \right\rangle_{L^2(\partial D^k)}.
\end{aligned}
$$

Since the right-hand side of the WADG formulation is identical to the right-hand side of the DG formulation (2), WADG preserves a variant of the energy stability in Theorem 3.1.

$$
\begin{aligned}
\sum_{D^k \in \Omega_h} \frac{1}{2} \frac{\partial}{\partial t} &\left(\left(\boldsymbol{T}_{1/\rho}^{-1} \boldsymbol{v}, \boldsymbol{v} \right)_{L^2(D^k)} + \left(\boldsymbol{T}_C^{-1} \boldsymbol{\sigma}, \boldsymbol{\sigma} \right)_{L^2(D^k)} \right) \leq - \sum_{f \in \Gamma_h \backslash \partial \Omega} \int_f \left(\frac{\tau_v}{2} |\boldsymbol{A}_n [\![\boldsymbol{v}]\!]|^2 + \frac{\tau_\sigma}{2} |\boldsymbol{A}_n^T [\![\boldsymbol{\sigma}]\!]|^2 \right) \, \mathrm{d}\boldsymbol{x} \\
&- \sum_{f \in \Gamma_v} \int_f \left(\tau_v |\boldsymbol{A}_n \boldsymbol{v}^-|^2 \right) \mathrm{d}\boldsymbol{x} - \sum_{f \in \Gamma_\sigma} \int_f \left(\tau_\sigma |\boldsymbol{A}_n^T \boldsymbol{\sigma}^-|^2 \right) \mathrm{d}\boldsymbol{x} - \sum_{f \in \Gamma_{abc}} \int_f \left(\frac{\tau_v}{2} |\boldsymbol{A}_n \boldsymbol{v}^-|^2 + \frac{\tau_\sigma}{2} |\boldsymbol{A}_n^T \boldsymbol{\sigma}^-|^2 \right) \mathrm{d}\boldsymbol{x} \leq 0.
\end{aligned}
$$

The use of weight-adjusted inner products replaces the weighted L^2 mass matrices in (5) by their weight-adjusted approximations. Inverting these weight-adjusted mass matrices yields the following local system of ODEs for $\boldsymbol{V}, \boldsymbol{\Sigma}$

$$
\begin{aligned}
\frac{\partial \boldsymbol{V}}{\partial t} &= \left(\boldsymbol{I} \otimes \boldsymbol{M}^{-1} \right) \boldsymbol{M}_{(\rho^{-1} \boldsymbol{I})} \left(\sum_{i=1}^{d} \left(\boldsymbol{A}_i^T \otimes \boldsymbol{D}_i \right) \boldsymbol{\Sigma} + \sum_{f \in \partial D^k} \left(\boldsymbol{I} \otimes \boldsymbol{L}_f \right) \boldsymbol{F}_v \right), \\
\frac{\partial \boldsymbol{\Sigma}}{\partial t} &= \left(\boldsymbol{I} \otimes \boldsymbol{M}^{-1} \right) \boldsymbol{M}_C \left(\sum_{i=1}^{d} \left(\boldsymbol{A}_i \otimes \boldsymbol{D}_i \right) \boldsymbol{V} + \sum_{f \in \partial D^k} \left(\boldsymbol{I} \otimes \boldsymbol{L}_f \right) \boldsymbol{F}_\sigma \right).
\end{aligned}
$$

In practice, the matrices $\left(\boldsymbol{I} \otimes \boldsymbol{M}^{-1} \right) \boldsymbol{M}_{(\rho^{-1} \boldsymbol{I})}$ and $\left(\boldsymbol{I} \otimes \boldsymbol{M}^{-1} \right) \boldsymbol{M}_C$ are applied in a matrix-free fashion using reference element matrices and values of $\rho, \boldsymbol{C}$ at quadrature points. After fusing operations together, this procedure can be boiled down to multiplication by 2 rectangular matrices.[17]

One drawback of the analysis presented in this work is that the accuracy of the weight-adjusted approximation is not guaranteed in the incompressible limit $\mu/\lambda \to 0$. In this case, the stiffness matrix $\boldsymbol{C}$ becomes singular, and the constant

$\tilde{w}_{\max}$ in the upper bound on $\|C^{-1}\|$ blows up. However, numerical experiments in Section 6.2.3 suggest that, while taking $\mu/\lambda \approx 0$ (or C nearly singular) results in larger relative errors for σ, the accuracy of the WADG solution for v does not degrade as significantly for near-incompressible materials.

5.1 | Energy stability on curvilinear meshes

We have shown the energy stability of the DG formulation (2) on meshes of affine elements by assuming exact integration of all terms. However, energy stability can still be guaranteed if integrals are evaluated inexactly using quadrature. Instead of discretizing the "strong" DG formulation (2), we discretize the "skew-symmetric" formulation, where the right-hand side of the velocity equations is integrated by parts,[†] resulting in the (local) formulation

$$\int_{\hat{D}} T^{-1}_{(\rho J)^{-1}} \frac{\partial v}{\partial t} w \, \mathrm{d}\hat{x} = \int_{\hat{D}} \left(\sum_{i=1}^{d} A_i^T \frac{\partial \sigma}{\partial x_i}, w \right) J \mathrm{d}\hat{x} + \sum_{f \in \partial D^k} \int_{\hat{f}} \left(\frac{1}{2} A_n^T [\![\sigma]\!] + \frac{\tau_v}{2} A_n^T A_n [\![v]\!] \right) w J^f \mathrm{d}\hat{x},$$

$$\int_{\hat{D}} T^{-1}_{(J^{-1}C)} \frac{\partial \sigma}{\partial t} q \, \mathrm{d}\hat{x} = -\int_{\hat{D}} \left(\sum_{i=1}^{d} v, A_i^T \frac{\partial q}{\partial x_i} \right) J \mathrm{d}\hat{x} + \sum_{f \in \partial D^k} \int_{\hat{f}} \left(\frac{1}{2} A_n \{\!\{v\}\!\} + \frac{\tau_\sigma}{2} A_n A_n^T [\![\sigma]\!] \right) q J^f \mathrm{d}\hat{x},$$

where we have incorporated spatial variations of J into the definition of the weights on the left-hand side.

The proof of energy stability in Theorem 3.1 follows,[17] requiring only algebraic manipulations of this formulation. The proof does not require that integration-by-parts holds discretely. This implies that a discrete version of energy stability is still guaranteed in the presence of inexact quadrature, where integrals in Theorem 3.1 are replaced with quadrature approximations. This is especially important for discretizations on curvilinear meshes, where the exact integration of spatially varying geometric factors and Jacobians can be either prohibitively expensive for high-order curvilinear mappings or impossible for rational mappings.[33,34]

We note that, to ensure energy stability on curved and nonaffine elements, the "skew-symmetric" formulation must be evaluated explicitly using quadrature, which is typically more expensive than quadrature-free evaluations used to evaluate the "strong" DG formulation (2). These costs can be slightly reduced for most curvilinear meshes by evaluating the DG formulation using the "skew-symmetric" formulation on curvilinear elements and the more efficient "strong" DG formulation on affine elements.[17]

5.2 | Convergence analysis

Using estimates from Section 4.2.1, we can extend the semidiscrete convergence analysis in Chan et al, Warburton, Houston et al[16,18,35] to linear elastic wave propagation on meshes of affine elements. Techniques in Warburton[18] can be used to extend this analysis to curvilinear elements.

Let U, U_h denote the exact and discrete WADG solutions, respectively. We will assume that $U, \frac{\partial U}{\partial t}$ are sufficiently regular such that

$$\|U\|_{W^{N+1,2}(\Omega_h)} < \infty, \qquad \left\|\frac{\partial U}{\partial t}\right\|_{W^{N+1,2}(\Omega_h)} < \infty,$$

where we define $\|U\|^2_{W^{N+1,2}(\Omega_h)} = \sum_k \|U\|^2_{W^{N+1,2}(D^k)}$.

In terms of group variables $U = (v, \sigma)$ and $V = (w, q) \in (V_h(\Omega_h))^d \times (V_h(\Omega_h))^{N_d}$, the WADG formulation can be written as

$$\left(T^{-1}_{A_0^{-1}} \frac{\partial U}{\partial t}, V \right) + a(U, V) + b(U, V) = (f, V),$$

$$a(U, V) = \sum_{D^k \in \Omega_h} \left(-\left(\sum_{i=1}^{d} \sigma, A_i \frac{\partial w}{\partial x_i} \right)_{L^2(D^k)} + \left(\sum_{i=1}^{d} A_i \frac{\partial v}{\partial x_i}, q \right)_{L^2(D^k)} \right),$$

$$b(U, V) = \sum_{D^k \in \Omega_h} \left(\left\langle A_n^T \{\!\{\sigma\}\!\} + \frac{\tau_v}{2} A_n^T A_n [\![v]\!], w \right\rangle_{L^2(\partial D^k)} + \left\langle \frac{1}{2} A_n [\![v]\!] + \frac{\tau_\sigma}{2} A_n A_n^T [\![\sigma]\!], q \right\rangle_{L^2(\partial D^k)} \right),$$

[†]The choice of which equation to integrate by parts is arbitrary, since integrating the stress equations by parts also results in an energy stable formulation.

where $A_0(x) = \text{diag}\left(\rho(x)I^{d\times d}, C^{-1}(x)\right)$. Energy stability implies that

$$b(U, U) = \sum_{D^k \in \Omega_h} \left(\frac{\tau_v}{2} \| [\![A_n v]\!] \|^2_{L^2(\partial D^k)} + \frac{\tau_\sigma}{2} \| [\![A_n^T \sigma]\!] \|^2_{L^2(\partial D^k)} \right)$$

$$\left(T_{A_0^{-1}}^{-1} \frac{\partial U}{\partial t}, U \right) + b(U, U) = (f, U).$$

Since the DG formulation (2) is consistent, these solutions satisfy

$$\left(A_0 \frac{\partial U}{\partial t}, V \right) + a(U, V) + b(U, V) = (f, V),$$

$$\left(T_{A_0^{-1}}^{-1} \frac{\partial U_h}{\partial t}, V \right) + a(U_h, V) + b(U_h, V) = (f, V),$$

(9)

for all $V \in (V_h(\Omega_h))^d \times (V_h(\Omega_h))^{N_d}$. We decompose the error $U - U_h$ into a projection error ϵ and discretization error η.

$$U - U_h = (\Pi_N U - U_h) - (\Pi_N U - U) = \eta - \epsilon.$$

We assume that $U_h(x, 0)$ is the L^2 projection of the exact initial condition, such that $\eta|_{t=0} = 0$. We also introduce a consistency error $\delta = A_0 U - T_{A_0^{-1}}^{-1} U$ resulting from the approximation of $A_0 U$ by a weight-adjusted inner product

$$A_0 \frac{\partial U}{\partial t} - T_{A_0^{-1}}^{-1} \frac{\partial U_h}{\partial t} = \frac{\partial}{\partial t} \left(A_0 U - T_{A_0^{-1}}^{-1} U \right) + \frac{\partial}{\partial t} T_{A_0^{-1}}^{-1} (\Pi_N U - U_h) = \frac{\partial \delta}{\partial t} + \frac{\partial}{\partial t} \left(T_{A_0^{-1}}^{-1} \eta \right),$$

where we have used that $T_{A_0^{-1}}^{-1} = T_{A_0^{-1}}^{-1} \Pi_N$. Subtracting the DG and WADG formulations in (9) and setting $V = \eta$ then yields

$$\frac{1}{2} \frac{\partial}{\partial t} \left(T_{A_0^{-1}}^{-1} \eta, \eta \right) + b(\eta, \eta) = \left(-\frac{\partial \delta}{\partial t}, \eta \right) + a(\epsilon, \eta) + b(\epsilon, \eta),$$

(10)

where we have used that $a(\eta, \eta) = 0$ by skew symmetry.

We bound $a(\epsilon, \eta) + b(\epsilon, \eta)$ on the right-hand side by integrating by parts the stress equation and using the component-wise L^2 orthogonality of ϵ to derivatives of η. This reduces the term $a(\epsilon, \eta) + b(v, \eta)$ to surface contributions over each element, which can be combined with contributions from neighboring elements to yield

$$\sum_{D^k \in \Omega_h} \left(\left\langle A_n^T \{\!\{\epsilon_\sigma\}\!\} + \frac{\tau_v}{2} A_n^T A_n [\![\epsilon_v]\!], \eta_v \right\rangle_{L^2(\partial D^k)} + \left\langle A_n \{\!\{\epsilon_v\}\!\} + \frac{\tau_\sigma}{2} A_n A_n^T [\![\epsilon_\sigma]\!], \eta_\sigma \right\rangle_{L^2(\partial D^k)} \right)$$

$$= \frac{1}{2} \sum_{D^k \in \Omega_h} \left(\left\langle \{\!\{\epsilon_\sigma\}\!\} - \frac{\tau_v}{2} A_n [\![\epsilon_v]\!], A_n [\![\eta_v]\!] \right\rangle_{L^2(\partial D^k)} + \left\langle \{\!\{\epsilon_v\}\!\} - \frac{\tau_\sigma}{2} A_n^T [\![\epsilon_\sigma]\!], A_n^T [\![\eta_\sigma]\!] \right\rangle_{L^2(\partial D^k)} \right)$$

$$\leq \frac{1}{2} \sum_{D^k \in \Omega_h} \left\| \{\!\{\epsilon_\sigma\}\!\} - \frac{\tau_v}{2} A_n [\![\epsilon_v]\!] \right\|_{L^2(\partial D^k)} \left\| A_n [\![\eta_v]\!] \right\|_{L^2(\partial D^k)} + \left\| \{\!\{\epsilon_v\}\!\} - \frac{\tau_\sigma}{2} A_n^T [\![\epsilon_\sigma]\!] \right\|_{L^2(\partial D^k)} \left\| A_n^T [\![\eta_\sigma]\!] \right\|_{L^2(\partial D^k)}$$

$$\leq C_\tau \sum_{D^k \in \Omega_h} \| \epsilon \|_{L^2(\partial D^k)} \left(\frac{\tau_v}{2} \| A_n [\![\eta_v]\!] \|^2_{L^2(\partial D^k)} + \frac{\tau_\sigma}{2} \| A_n^T [\![\eta_\sigma]\!] \|^2_{L^2(\partial D^k)} \right)^{1/2},$$

where C_τ is proportional to $\max(\tau_v, \tau_\sigma)$. Using Young inequality with $\alpha = C_\tau/2$ yields the following bound

$$|a(\epsilon, \eta) + b(\epsilon, \eta)| \leq b(\eta, \eta) + \frac{C_\tau^2}{4} \sum_{D^k \in \Omega_h} \| \epsilon \|^2_{L^2(\partial D^k)}.$$

Applying this to (10) and using Cauchy-Schwarz on $\left(\frac{\partial \delta}{\partial t}, \eta \right)$ then yields

$$\frac{1}{2} \frac{\partial}{\partial t} \left(T_{A_0^{-1}}^{-1} \eta, \eta \right) + b(\eta, \eta) \leq \left\| \frac{\partial \delta}{\partial t} \right\| \| \eta \| + b(\eta, \eta) + \sum_{D^k \in \Omega_h} \frac{C_\tau^2}{4} \| \epsilon \|^2_{L^2(\partial D^k)}.$$

We eliminate factors of $b(\boldsymbol{\eta}, \boldsymbol{\eta})$ on both sides and bound right-hand side terms. The trace term is bounded using a standard hp trace inequality[36] and an interpolation estimate

$$\sum_{D^k \in \Omega_h} \|\epsilon\|^2_{L^2(\partial D^k)} \leq \sum_{D^k \in \Omega_h} Ch^{-1} \|\epsilon\|^2_{L^2(D^k)} = Ch^{-1} \|\epsilon\|^2 \leq Ch^{2N+1} \|\boldsymbol{U}\|^2_{W^{N+1,2}}.$$

Since $\boldsymbol{A}_0$ is independent of t, Theorem 4.5 gives

$$\left\| \frac{\partial \boldsymbol{\delta}}{\partial t} \right\| = \left\| \boldsymbol{A}_0 \frac{\partial \boldsymbol{U}}{\partial t} - T_{\boldsymbol{A}_0^{-1}}^{-1} \frac{\partial \boldsymbol{U}}{\partial t} \right\| \leq Ch^{N+1} A_{\max} \|\boldsymbol{A}_0\|_{W^{N+1,\infty}(\Omega_h)} \left\| \frac{\partial \boldsymbol{U}}{\partial t} \right\|,$$

where $A_{\max} = \max \left(\frac{\rho_{\max}}{\rho_{\min}}, c_{\max} \tilde{c}_{\max} \right)$. Then, integrating from $[0, T]$ and using Lemma 4.4 yields

$$A_{\min} \|\boldsymbol{\eta}\|^2 \leq C \int_0^T h^{N+1} A_{\max} \|\boldsymbol{A}_0\|_{W^{N+1,\infty}(\Omega_h)} \left\| \frac{\partial \boldsymbol{U}}{\partial t} \right\| \|\boldsymbol{\eta}\| + h^{2N+1} \|\boldsymbol{U}\|^2_{W^{N+1,2}(\Omega)} \, dt, \qquad (11)$$

where $A_{\min} = \min(\rho_{\min}, \tilde{c}_{\min})$. Applying the modified Gronwall inequality (Dolejší and Feistauer[37, (lemma 1.10)]; see also Chan et al[16]) to (11) then yields

$$\|\boldsymbol{\eta}\| \leq \frac{1}{A_{\min}} \int_0^T CA_{\max} h^{N+1} \|\boldsymbol{A}_0\|_{W^{N+1,\infty}(\Omega_h)} \left\| \frac{\partial \boldsymbol{U}}{\partial t} \right\| dt + \sup_{t \in [0,T]} \sqrt{\int_0^T Ch^{2N+1} \|\boldsymbol{U}\|^2_{W^{N+1,2}(\Omega)} \, dt}$$

$$\leq \frac{CTh^{N+1/2}}{A_{\min}} \sup_{t \in [0,T]} \left(h^{1/2} A_{\max} \|\boldsymbol{A}_0\|_{W^{N+1,\infty}(\Omega_h)} \left\| \frac{\partial \boldsymbol{U}}{\partial t} \right\| + \|\boldsymbol{U}\|_{W^{N+1,2}(\Omega_h)} \right).$$

The triangle inequality gives the final estimate

$$\|\boldsymbol{U} - \boldsymbol{U}_h\| \leq (C_1 + C_2 T) h^{N+1/2} \sup_{t \in [0,T]} \left(h^{1/2} \|\boldsymbol{A}_0\|_{W^{N+1,\infty}(\Omega_h)} \left\| \frac{\partial \boldsymbol{U}}{\partial t} \right\|_{W^{N+1,2}(\Omega_h)} + \|\boldsymbol{U}\|_{W^{N+1,2}(\Omega_h)} \right),$$

where C_2 depends on $A_{\min}, A_{\max}$. From this estimate, we expect L^2 errors to decrease proportionally to $O(h^{N+1/2})$ under mesh refinement, which mirrors theoretical results given in Chan et al and Warburton.[16,18] Optimal rates of $O(h^{N+1})$ are often observed in practice. However, we do also observe $O(h^{N+1/2})$ rates of convergence for $N = 1, \ldots, 5$ for certain problems, which suggests that the theoretical estimate is tight.

6 | NUMERICAL EXPERIMENTS

The following sections present several numerical experiments validating the stability and accuracy of the proposed method in 2 and 3 dimensions. The energy stability of the method is also confirmed for examples with subcell variations in heterogeneous media and curvilinear meshes. The convergence of the new DG formulation in piecewise constant isotropic media is confirmed using analytic solutions, while the convergence of the method for high-order approximations of heterogeneous media is confirmed using a fine-grid reference solution. Finally, the method is applied to problems with anisotropy and stiffness matrices $\boldsymbol{C}$ with subelement variations.

In all experiments, we follow Chan et al[16] and compute application of weight-adjusted mass matrices[31] using a quadrature exact for polynomials of degree $(2N + 1)$. Time integration is performed using the low-storage fourth-order 5-stage Runge-Kutta scheme of Carpenter and Kennedy,[38] and the time-step is chosen based on the global estimate

$$dt = \min_k \frac{C_{\text{CFL}}}{\sup_{\boldsymbol{x} \in \Omega} \|\boldsymbol{C}(\boldsymbol{x})\|_2 C_N \|J^f\|_{L^\infty(\partial D^k)} \|J^{-1}\|_{L^\infty(D^k)}}, \qquad (12)$$

where $C_N = O(N^2)$ is the order-dependent constant in the surface polynomial trace inequality[12] and C_{CFL} is a tunable global CFL constant. This estimate is derived by bounding the eigenvalues of the spatial DG discretization matrix appearing in the semidiscrete system of ODEs. We note that the usual factor of h arises through the term

$$\frac{1}{\|J^f\|_{L^\infty(\partial D^k)} \|J^{-1}\|_{L^\infty(D^k)}} = O(h)$$

due to the fact that $\|J^{-1}\|_{L^\infty(D^k)} = O(h^{-d})$ and $\|J^f\|_{L^\infty(D^k)} = O(h^{d-1})$ in d dimensions.

Finally, in all following experiments, we use $\tau_v = \tau_\sigma = 1$ and $C_{CFL} = 1$ unless specified otherwise. We have arbitrarily chosen these parameters for simplicity, though a more nuanced choice of penalty parameters and CFL constant may improve numerical and computational performance for certain problems. We note that because dt is derived through an upper bound on the spectral radius of the discretization matrix, this estimate of the time-step is rather conservative, and we have observed that it is possible to take $C_{CFL} > 1$ without losing stability under our choice of time-stepping scheme.

6.1 | Spectra and choice of penalty parameter

We first verify the energy stability of the WADG method for arbitrary heterogeneous media. We use a stiffness matrix constructed using similarity transforms, such that at every quadrature point, $C(x) = UDU^T$, where D is a diagonal matrix with random positive entries $d_{min} \leq D_{ii} \leq d_{max}$ and U is a random unitary matrix. Let A_h denote the matrix induced by the global semidiscrete DG formulation, such that the time evolution of the solution v, σ is governed by

$$\frac{\partial Q}{\partial t} = A_h Q,$$

with Q denotes a vector of degrees of freedom for (v, σ). Figure 1 shows computed eigenvalues of A_h for $[d_{min}, d_{max}] = [1/10, 1]$ and $[10^{-5}, 1]$, $\tau_v = \tau_\sigma = 0$ and $\tau_v = \tau_\sigma = 1$ under discretization parameters $N = 4$ and $h = 1/4$. In both cases, the largest real part of any eigenvalue is $O(10^{-14})$, verifying the energy stability of the WADG discretization for arbitrary media.

For practical simulations, the choice of τ_v, τ_σ remains to be specified. Taking $\tau_v, \tau_\sigma > 0$ results in damping of under-resolved spurious components of the solution[21]; however, a naive selection of these penalty parameters can result in an overly restrictive time-step restriction for stability. We wish to choose τ_v, τ_σ as large as possible without increasing the value of $\|A_h\|$ when using a central flux (ie, $\tau_v = \tau_\sigma = 0$). For example, in Figure 1, we observe that the value of $\|A_h\|$ for a central flux is roughly half as large as the value of $\|A_h\|$ when taking $\tau_v = \tau_\sigma = 1$. We note that the growth in $\|A_h\|$ when $\tau_v = \tau_\sigma = 1$ is due to the large negative real part of the extremal eigenvalues of A_h, which mirrors observations in Chan and Warburton[21] that a subset of the eigenvalues of A_h approach $-\infty$ as the penalty parameter increases.

Initial numerical experiments suggest that varying the penalty parameters spatially and scaling τ_v and τ_σ independently of each other can offset the artificial stiffness induced by a naive choice of penalty parameters. For example, material coefficients can be taken into account by scaling the penalty parameters such that flux terms are dimensionally consistent. One example of such a scaling is

$$\tau_v = \gamma_v \sup_{x \in f} \sqrt{\|\{\{C(x)\}\}\| \{\{\rho(x)\}\}}, \qquad \tau_\sigma = \gamma_\sigma \sup_{x \in f} \frac{1}{\sqrt{\|\{\{C(x)\}\}\| \{\{\rho(x)\}\}}},$$

where the supremum is taken locally over each face and γ_v, γ_σ are dimensionless constants. We note that the optimal choice of scaling is outside of the scope of this current paper and will be explored in future work.
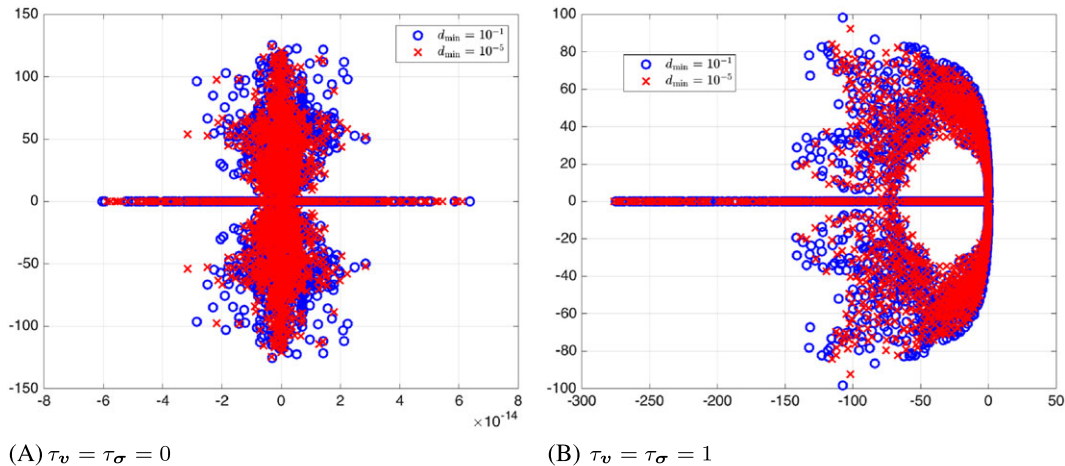

(A) $\tau_v = \tau_\sigma = 0$ (B) $\tau_v = \tau_\sigma = 1$

FIGURE 1 Spectra for $N = 4$ and $h = 1/4$, using randomly chosen $C(x)$ with eigenvalues between $[d_{min}, 1]$ at each quadrature point. For both $\tau_v = \tau_\sigma = 0$ and $\tau_v = \tau_\sigma = 1$, the largest real part of the spectra is $O(10^{-14})$

6.2 | Analytic solutions

Next, we study the accuracy and convergence of WADG method for several analytical solutions in linear elasticity. In all cases, the solution is expressed in terms of the displacement vector $\boldsymbol{u}(\boldsymbol{x}, t) = (u_1, \ldots, u_d)$. Initial conditions for velocity and stress are computed through

$$\boldsymbol{v}(\boldsymbol{x}, t) = \frac{\partial \boldsymbol{u}}{\partial t}, \qquad \boldsymbol{\sigma} = \boldsymbol{C} \frac{1}{2} \left(\nabla \boldsymbol{u} + \nabla \boldsymbol{u}^T \right).$$

Unless otherwise stated, we report relative L^2 errors for all components of the solution $\boldsymbol{U} = (\boldsymbol{v}, \boldsymbol{\sigma})$,

$$\frac{\|\boldsymbol{U} - \boldsymbol{U}_h\|_{L^2(\Omega)}}{\|\boldsymbol{U}\|_{L^2(\Omega)}} = \frac{\left(\sum_{i=1}^m \|\boldsymbol{U}_i - \boldsymbol{U}_{i,h}\|_{L^2(\Omega)}^2 \right)^{1/2}}{\left(\sum_{i=1}^m \|\boldsymbol{U}_i\|_{L^2(\Omega)}^2 \right)^{1/2}}.$$

6.2.1 | Harmonic oscillation of a square

We first examine convergence on a unit square domain with $\lambda = \mu = \rho = 1$. The components of the displacement vector are given by

$$u_1(x, y, t) = \cos(\omega \pi t) \cos(\pi x) \sin(\pi y),$$
$$u_2(x, y, t) = -\cos(\omega \pi t) \sin(\pi x) \cos(\pi y),$$

where $\omega = \sqrt{2\mu}$. Zero traction boundary conditions are imposed. Figure 2 shows L^2 errors computed at time $T = 5$, using uniform triangular meshes constructed by bisecting a uniform mesh of quadrilaterals along the diagonal.

For $N = 1, \ldots, 5$, $O(h^{N+1})$ rates of convergence are observed when using the penalty flux with $\tau_v = \tau_\sigma = 1$. When using a central flux (with $\tau_v = \tau_\sigma = 0$), we observe a so-called "even-odd" pattern,[14,23] where the convergence rate is $O(h^N)$ for N odd and between $O(h^{N+1/2})$ and $O(h^{N+1})$ for N even. This behavior improves upon the theoretical estimate derived in Delcourte and Glinsky,[7] which is $O(h^N)$ for a sufficiently small time-step size.

For quasi-uniform meshes, the optimal rate of convergence of spatial L^2 errors under uniform mesh refinement is $O(h^{N+1})$,[23] which is greater than the $O(h^{N+1/2})$ rate of convergence which can be proven for dissipative DG discretizations on general meshes[39] (though optimal rates of convergence are often observed in numerical experiments). We note that for $N = 4$ and $N = 5$, we observe results for both fluxes which are better than the fourth-order accuracy of our time-stepping scheme. This is most likely due to the benign nature of the solution and the choice of time-step (12), which scales as $O(h/N^2)$. For $N = 4, 5$, the results of Figure 2 suggest that the resulting time-step is small enough such that temporal errors of $O(dt^4)$ are small relative to spatial discretization errors of $O(h^{N+1})$.

6.2.2 | Rayleigh and Lamb waves

Next, we examine the convergence of WADG for Rayleigh and Lamb waves, both of which test the imposition of traction-free boundary conditions.
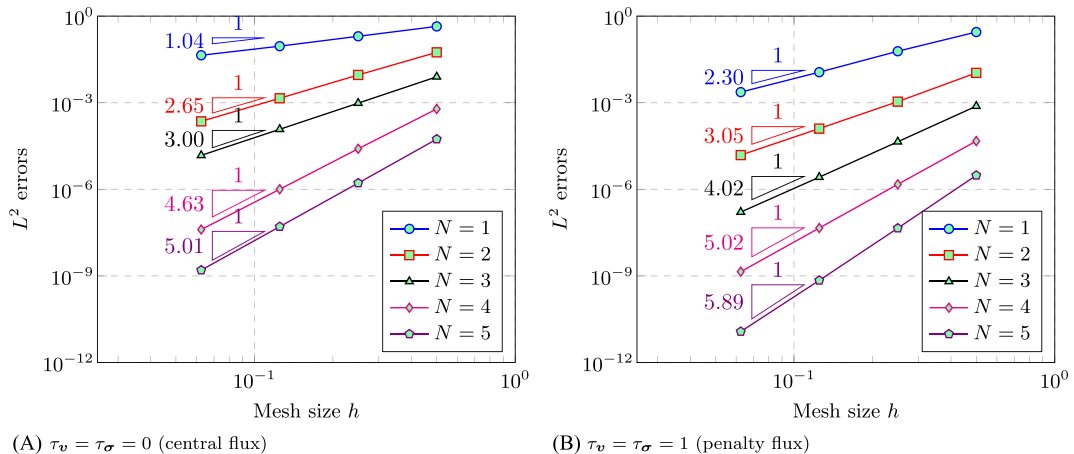

(A) $\tau_v = \tau_\sigma = 0$ (central flux) (B) $\tau_v = \tau_\sigma = 1$ (penalty flux)

FIGURE 2 Convergence of L^2 errors for harmonic oscillation

Rayleigh waves are elastic surface waves which decay exponentially away from the surface. These waves are given by the displacement vector

$$\boldsymbol{u}(x,y,t) = e^{-\omega x\sqrt{1-\xi^2}} \begin{pmatrix} \cos(\omega(y+c_r t)) \\ \sqrt{1-\xi^2}\sin(\omega(y+c_r t)) \end{pmatrix}$$
$$+ \left(\frac{\xi^2}{2}-1\right) e^{-\omega x\sqrt{1-\frac{\xi^2\mu}{2\mu+\lambda}}} \begin{pmatrix} \cos(\omega(y+c_r t)) \\ \sin(\omega(y+c_r t))/\sqrt{1-\frac{\xi^2\mu}{2\mu+\lambda}} \end{pmatrix},$$

where ω is the wavespeed, c_r is the Rayleigh phase velocity $c_r = \xi\sqrt{\mu}$, and ξ satisfies

$$\sqrt{1-\xi^2}\sqrt{1-\frac{\xi^2\mu}{2\mu+\lambda}} - \left(\frac{\xi^2}{2}-1\right)^2 = 0.$$

In our computations, we use $\rho = \mu = \lambda = 1$, $\xi = 0.949554083888034$, and $\omega = 2\pi$.[20] We solve on the domain $[0,2]\times[0,1]$ using a sequence of uniform triangular meshes and enforce traction-free boundary conditions at $x=0$ and exact Dirichlet boundary conditions at $x=2$. Periodic boundary conditions are applied at $y=0$ and $y=1$.

Lamb waves are supported by elastic waveguides with traction-free (free surface) boundary conditions at the top and bottom of the domain. The displacement of these waves is given by

$$u_1(x,y,t) = (-kB_1\cos(py) - qB_2\cos(qy))\sin(kx-\omega t),$$
$$u_2(x,y,t) = (-pB_1\sin(py) + kB_2\sin(qy))\cos(kx-\omega t),$$

where k is the wavenumber and ω is the frequency, and the constants p and q are defined as

$$p^2 = \frac{\omega^2}{2\mu+\lambda} - k^2, \qquad q^2 = \frac{\omega^2}{\mu} - k^2.$$

The wavenumber k and frequency ω are related through a dispersion relation. The ratio of the amplitudes B_1/B_2 can be determined using other parameters, implying that B_1 and B_2 are unique up to a scaling constant. In our experiments, we use $\rho = \mu = 1$, $\lambda = 2$, $k = 2\pi$. For these values,[20] $\omega = 13.137063197233$, $B_1 = 126.1992721468$, and $B_2 = 53.88807700007$. We solve on the domain $[-1,1]\times[-1/2,1/2]$, with traction-free boundary conditions at $y=\pm1/2$ and periodic boundary conditions at $x=\pm1$.

Figures 3 and 4 show L^2 errors for Rayleigh and Lamb waves at time $T=5$, respectively. As with the harmonic oscillation solution, both central and penalty fluxes are considered. For penalty fluxes, the computed convergence rates fall between the optimal rate of $O(h^{N+1})$ and theoretical rate of $O(h^{N+1/2})$.[40] For central fluxes, we observe the odd-even pattern for Lamb waves. However, for Rayleigh waves, we observe the theoretical $O(h^N)$ rate of convergence.

As with the harmonic oscillation problem, the accuracy of the numerical method is theoretically limited by the fourth-order accuracy of the time-stepping scheme. The observed higher-order accuracy for $N=4$ and $N=5$ again suggests that the solution is smooth in time and the time-step is small enough to render temporal discretization errors small relative to spatial discretization errors.
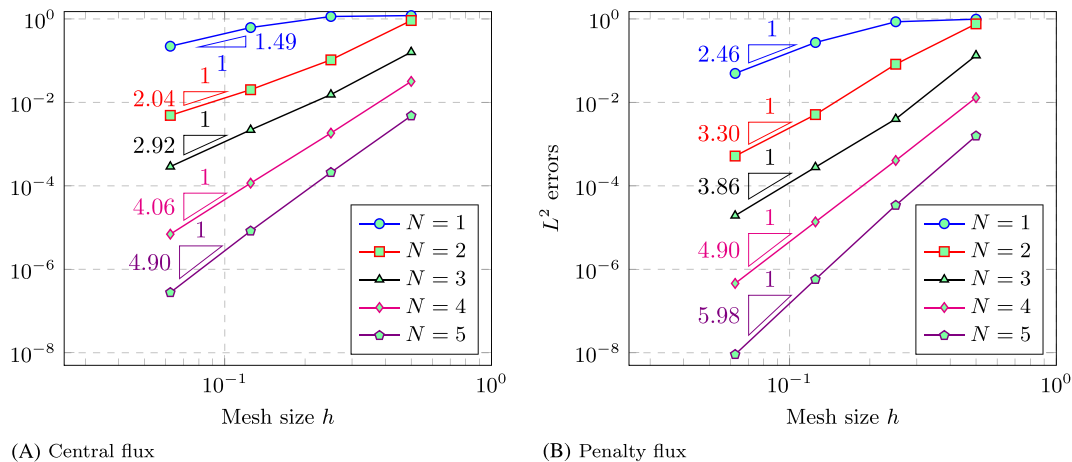

(A) Central flux (B) Penalty flux

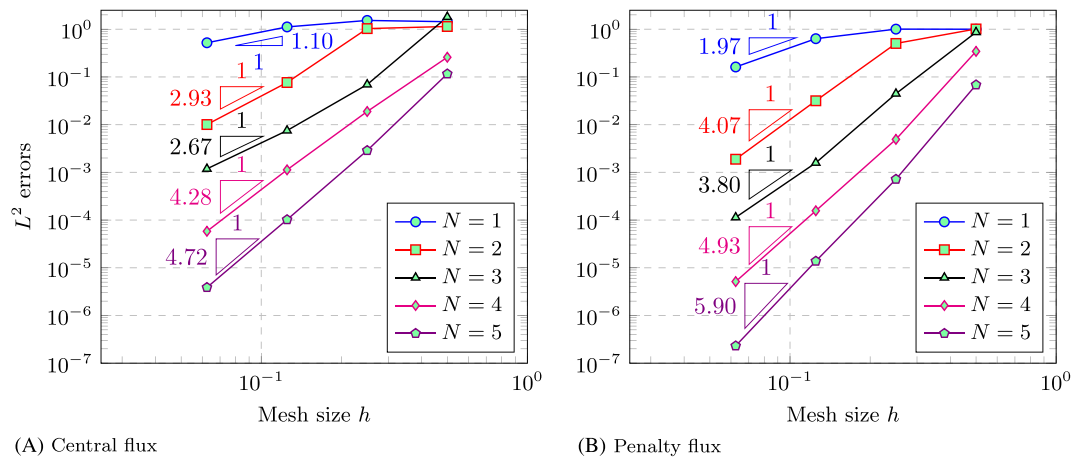**FIGURE 3** Convergence of L^2 errors for the Rayleigh wave solution

FIGURE 4 Convergence of L^2 errors for the Lamb wave solution

TABLE 1 Behavior of WADG for linear elastic wave propagation in the incompressible limit as $\mu/\lambda \to 0$. Errors are shown for various orders and mesh resolutions using the penalty flux.

	$\mu = 1$	$\mu = 0.1$	$\mu = 0.01$	$\mu = 0.001$	$\mu = 0.0001$
		(a) Relative error in v			
$N = 2, h = 1/4$	3.1063e-02	2.6972e-02	3.0858e-02	4.0103e-02	5.6835e-02
$N = 3, h = 1/4$	3.1677e-03	2.6848e-03	2.9854e-03	3.6960e-03	4.9362e-03
$N = 4, h = 1/4$	2.8726e-04	2.4990e-04	2.9142e-04	3.9774e-04	5.2469e-04
$N = 2, h = 1/8$	3.1819e-03	2.5476e-03	2.8877e-03	3.5608e-03	4.7010e-03
$N = 3, h = 1/8$	1.8867e-04	1.6925e-04	1.8509e-04	2.2520e-04	2.8301e-04
$N = 4, h = 1/8$	8.4999e-06	7.5750e-06	8.1094e-06	1.0760e-05	1.4886e-05
		(b) Relative error in σ			
$N = 2, h = 1/4$	6.8250e-02	7.4130e-02	1.2104e-01	2.1333e-01	4.1451e-01
$N = 3, h = 1/4$	9.2980e-03	1.0685e-02	1.8109e-02	3.2046e-02	5.7612e-02
$N = 4, h = 1/4$	9.7251e-04	1.1687e-03	2.1941e-03	4.1378e-03	6.8404e-03
$N = 2, h = 1/8$	9.8138e-03	1.1429e-02	2.0331e-02	3.7889e-02	7.6516e-02
$N = 3, h = 1/8$	6.6596e-04	7.8157e-04	1.5353e-03	3.1239e-03	5.6341e-03
$N = 4, h = 1/8$	3.4272e-05	4.2639e-05	8.8393e-05	1.9764e-04	3.6052e-04

6.2.3 | Rayleigh waves in near-incompressible materials

As noted in Section 5, error estimates for isotropic elasticity no longer hold in the incompressible limit $\mu/\lambda \to 0$ due to the fact that C becomes singular. We use the propagation of Rayleigh waves to examine the behavior of WADG for near-incompressible materials. We follow Sjögreen and Petersson[41] and Appelö and Hagstrom[42] and fix $\lambda = 1$ and set $\mu = 1, 0.1, 0.01, 0.001, 0.0001$. Since the Rayleigh wave propagates with speed proportional to $\sqrt{\mu}$, we compute L^2 errors at the final time $1/(4\sqrt{\mu})$ to ensure a fair comparison between solutions at different values of μ.

Table 1 shows relative errors for v, σ at different orders and mesh sizes using the penalty flux. The relative errors for σ grow as $\mu/\lambda \to 0$. This is not surprising, as the constant in the error estimates of Section 4.2.1 depends on $\|C\|$, which blows up in the incompressible limit. However, because the magnitude of σ also decreases as $\mu/\lambda \to 0$, relative errors for v remain roughly the same magnitude for near-incompressible materials. Decreasing μ by 4 orders of magnitude results in up to a 10-fold increase in relative error for σ but less than a 2-fold increase in error for v.

6.2.4 | Stoneley waves

A Stoneley wave is supported along the interface between 2 solids.[43] Like Rayleigh waves, Stoneley waves decay exponentially away from the interface and test the effectiveness of numerical fluxes across interfaces. We follow Wilcox et al[20] and

Appelö and Hagstrom[42] and use discontinuous media defined by

$$(\rho, \lambda, \mu) = \begin{cases} (10, 3, 3) & y > 0, \\ (1, 1, 1) & y < 0. \end{cases}$$

The displacement vector for a Stoneley wave is then given by

$$u_1(x, y, t) = \begin{cases} \operatorname{Re}\left(\left(ikB_1 e^{-kb_{1p}y} + kb_{1s}B_2 e^{-kb_{1s}y}\right) e^{i(ky-\omega t)}\right), & y > 0, \\ \operatorname{Re}\left(\left(-kb_{1p}B_1 e^{-kb_{1p}y} + ikB_2 e^{-kb_{1s}y}\right) e^{i(kx-\omega t)}\right), & y < 0, \end{cases}$$

$$u_2(x, y, t) = \begin{cases} \operatorname{Re}\left(\left(ikB_3 e^{kb_{2p}y} - kb_{2s}B_4 e^{kb_{2s}y}\right) e^{i(kx-\omega t)}\right), & y > 0, \\ \operatorname{Re}\left(\left(kb_{2p}B_3 e^{kb_{2p}y} + ikB_4 e^{kb_{2s}y}\right) e^{i(kx-\omega t)}\right), & y < 0, \end{cases}$$

where c_{st} is the Stoneley wave speed and

$$k = \omega/c_{st}, \qquad b_{jp} = \sqrt{1 - \frac{c_{st}^2}{(2\mu_j + \lambda_j)/\rho_j}}, \qquad b_{js} = \sqrt{1 - \frac{c_{st}^2}{(\mu_j)/\rho_j}}, \qquad j = 1, 2.$$

The Stoneley wave speed c_{st} can be determined based material parameters and interface conditions, and the amplitudes B_1, B_2, B_3, and B_4 are determined from c_{st} up to scaling by a constant. For the parameters used in this study,[20] we take $c_{st} = 0.546981324213884, B_1 = i0.2952173626624, B_2 = -0.6798795208473, B_3 = i0.5220044931212$, and $B_4 = -0.9339639688697$. We assume $k = 1$, which gives $\omega = c_{st}$.
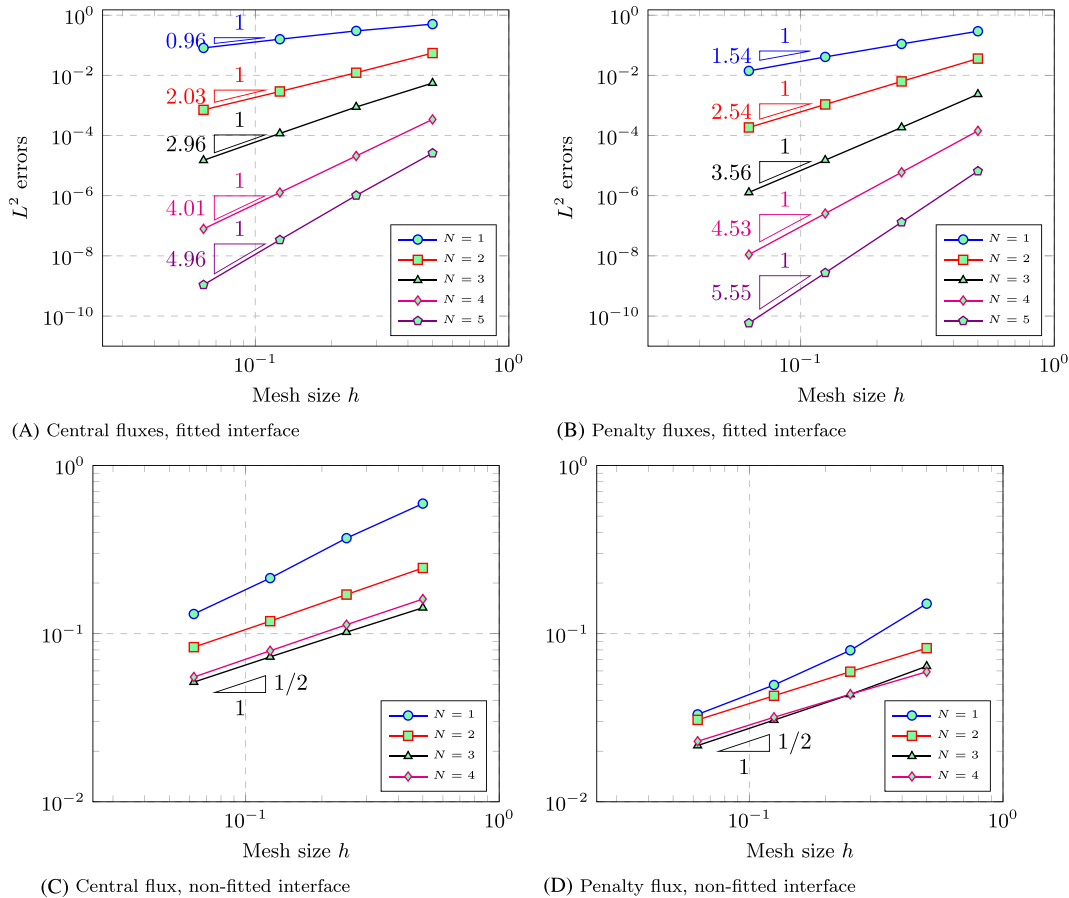

FIGURE 5 Convergence of weight-adjusted discontinuous Galerkin for a Stoneley wave using A,B, a fitted mesh aligned with the interface and C,D, a nonfitted mesh where the interface does not lie exactly on an element boundary

We solve on the domain $[-1, 1] \times [-5, 5]$, and enforce Dirichlet boundary conditions at all boundaries using the exact solution. Figure 5 shows L^2 errors for 2 uniform meshes of triangles constructed by bisecting a quadrilateral mesh of $K_{1D} \times 5K_{1D}$ elements. Figure 5B shows errors at time $T = 5$ when K_{1D} is even and the mesh is fitted to the interface at $y = 0$, while Figure 5D shows errors when K_{1D} is odd and the interface cuts through element interiors.

When the mesh is fitted to the interface, computed convergence rates using penalty fluxes match the theoretical $O(h^{N+1/2})$ rate. When using central fluxes, we observe $O(h^N)$ rates of convergence. This matches the behavior observed when using central fluxes for the Rayleigh wave problem, instead of the odd-even pattern of convergence observed when using central fluxes for the harmonic oscillation and Lamb wave solutions.

When the mesh is not fitted to the interface exactly, we compute the application of the weight-adjusted mass matrix inverse using a quadrature rule from Xiao and Gimbutas[31] which is exact for degree $2N + 1$ polynomials. Since the values of ρ, μ, and λ are positive at all quadrature points, the method is energy stable. However, since the exact solution is discontinuous, the error in elements cut by the interface is $O(1)$, resulting in L^2 errors which converge at rate $O(h^{1/2})$ for both penalty and central fluxes. We have also computed errors on a sequence of unfitted unstructured uniform meshes, as well as on a sequence of uniform meshes with randomly perturbed vertex positions. In both cases, similar $O(h^{1/2})$ rates of convergence were observed. We note that, when using piecewise constant approximations of μ and λ, we observe the same $O(h^{1/2})$ convergence rate, though errors are roughly twice as large in magnitude.

6.2.5 | Convergence to manufactured and reference solutions

To check the accuracy of the method for problems with smoothly varying heterogeneous media, we follow Mercerat and Glinsky[14] and consider a manufactured solution. We assume isotropic media and incorporate variations into the stiffness matrix $\boldsymbol{C}$ by taking λ such that

$$\lambda(x, y) = \lambda_0 + \tilde{\lambda}(x, y),$$

where λ_0 is a constant. We assume the displacement solution $\boldsymbol{u}$ is given as a plane wave

$$u_1(x, y, t) = \cos(k(x - c_P t)), \qquad u_2(x, y, t) = \cos(k(x - c_S t)),$$

where $c_P = \sqrt{(2\mu + \lambda_0)/\rho}$ and $c_S = \sqrt{\mu/\rho}$ are the P- and S-wave velocities corresponding to Lame parameters μ, λ_0.

This plane wave is the solution of the homogeneous elastic wave equations with $\tilde{\lambda}(x, y) = 0$. However, this is not true if $\tilde{\lambda}(x, y) \neq 0$ varies spatially. To test the convergence of our method when $\boldsymbol{C}$ contains smoothly varying coefficients, we modify our equations by adding source terms $\boldsymbol{f}_\sigma$ such that the plane wave solution satisfies

$$\begin{aligned} \rho \frac{\partial \boldsymbol{v}}{\partial t} &= \sum_{i=1}^{d} \boldsymbol{A}_i^T \frac{\partial \boldsymbol{\sigma}}{\partial \boldsymbol{x}_i}, \\ \boldsymbol{C}^{-1} \frac{\partial \boldsymbol{\sigma}}{\partial t} &= \sum_{i=1}^{d} \boldsymbol{A}_i \frac{\partial \boldsymbol{v}}{\partial \boldsymbol{x}_i} + \boldsymbol{f}_\sigma. \end{aligned} \tag{13}$$

Using the fact that the plane wave is the solution to the homogeneous equations, it is straightforward to show that the source terms are

$$\boldsymbol{f}_\sigma = -\tilde{\lambda} \nabla \cdot \boldsymbol{v} \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix},$$

where $\boldsymbol{v} = \frac{\partial \boldsymbol{u}}{\partial t}$ is the velocity of the exact plane wave solution. These source terms are computed using the same quadrature rule used for WADG.

Figure 6 shows the convergence of L^2 errors for a plane wave manufactured solution with $k = \pi$. We set $\rho = 1, \mu = 1, \lambda_0 = 2$, and $\tilde{\lambda}(x, y) = \frac{1}{2} \sin(2\pi x) \sin(2\pi y)$ and compute errors at final time $T = 5$ for $N = 1, \ldots, 5$. We observe that the L^2 errors convergence at a rate between the theoretical $O(h^{N+1/2})$ and optimal $O(h^{N+1})$ rates for the penalty flux with $\tau_v = \tau_\sigma = 1$. For central fluxes, we observe an even-odd pattern of convergence, with rates near $O(h^N)$ for N odd and $O(h^{N+1})$ for N even.

We also examine the accuracy of the WADG method for smoothly varying heterogeneous media by comparing against a reference spectral element method solution of degree $N = 50$ on a unit square $[-1, 1]^2$. Zero traction boundary conditions are enforced weakly through numerical fluxes.[20] We use a heterogeneous isotropic medium with ρ, λ, μ set to

$$\rho(\boldsymbol{x}) = 1, \qquad \lambda(\boldsymbol{x}) = 1 + .25 \sin(\pi x) \sin(\pi y), \qquad \mu(\boldsymbol{x}) = 1 + .25 \cos(\pi x) \cos(\pi y).$$
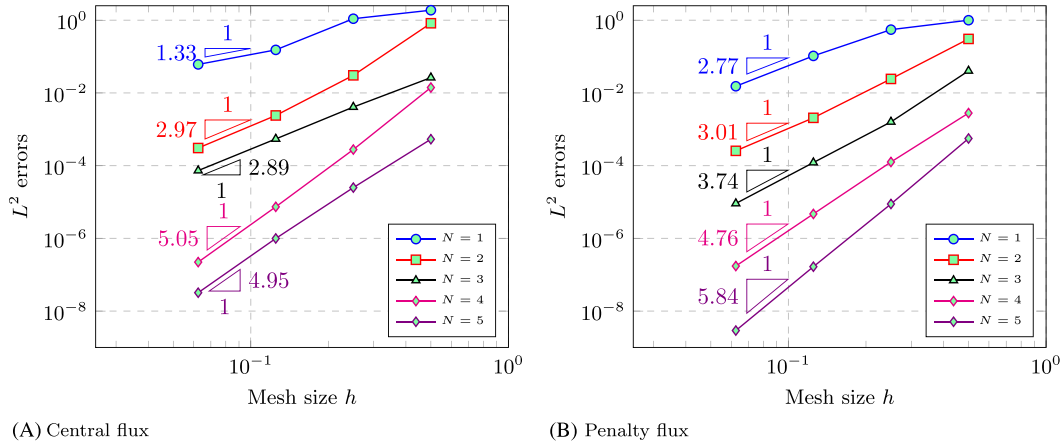
FIGURE 6 Convergence of weight-adjusted discontinuous Galerkin under mesh refinement to a manufactured solution with smoothly varying heterogeneous media for $N = 1, \ldots, 5$
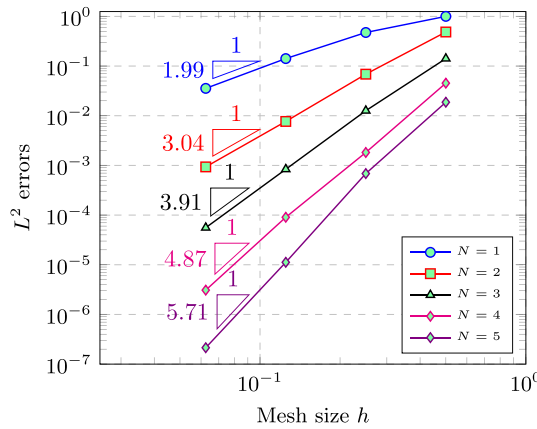

FIGURE 7 Convergence of weight-adjusted discontinuous Galerkin under mesh refinement to a reference $N = 50$ spectral method solution with smoothly varying heterogeneous media for $N = 1, \ldots, 5$ using a penalty flux

Initial stresses are set to 0, while the initial velocity is set to

$$v_1(\boldsymbol{x}, 0) = \cos(\pi x) \sin(\pi y), \qquad v_2(\boldsymbol{x}, 0) = -\sin(\pi x) \cos(\pi y).$$

Figure 7 shows L^2 errors with respect to the reference solution at time $T = 1/2$ for different mesh sizes and orders of approximation. Computed convergence rates fall between the optimal $O(h^{N+1})$ and predicted $O(h^{N+1/2})$ when using the penalty flux with penalty parameters set to 1.

6.2.6 | Curvilinear meshes

We now present numerical experiments verifying the stability and accuracy of the formulation presented in Section 5.1 for curvilinear meshes. We use isoparametric mappings in the following experiments, where the mapping from the reference element to each physical element is a polynomial of degree N. We construct these mappings by following Hesthaven and Warburton.[23] Starting from a uniform triangular mesh on the Lamb wave problem domain $\Omega = [0, 2] \times [0, 1]$, we place high-order Warp and Blend interpolation nodes on each element.[29] The physical locations (x_i, y_i) (for $i = 1, \ldots, N_p K$) of these nodes are then perturbed to produce new nodal positions $(\tilde{x}_i, \tilde{y}_i)$, where

$$\tilde{x}_i = x_i + \frac{1}{10} \cos\left(\frac{\pi}{2} x\right) \cos(3\pi y), \qquad \tilde{y}_i = y_i + \frac{1}{20} \sin(\pi x) \cos(3\pi y).$$

These new nodal positions $(\tilde{x}_i, \tilde{y}_i)$ now define a coordinate mapping from the reference element to a curved physical element, producing the warped mesh in Figure 8. This mesh warping is constructed such that the x and y deformations of each element are of roughly the same magnitude, while leaving the positions of nodes on the boundary unchanged.

(A) $\tau_v = \tau_\sigma = 0$ (central flux)   (B) $\tau_v = \tau_\sigma = 1$ (penalty flux)
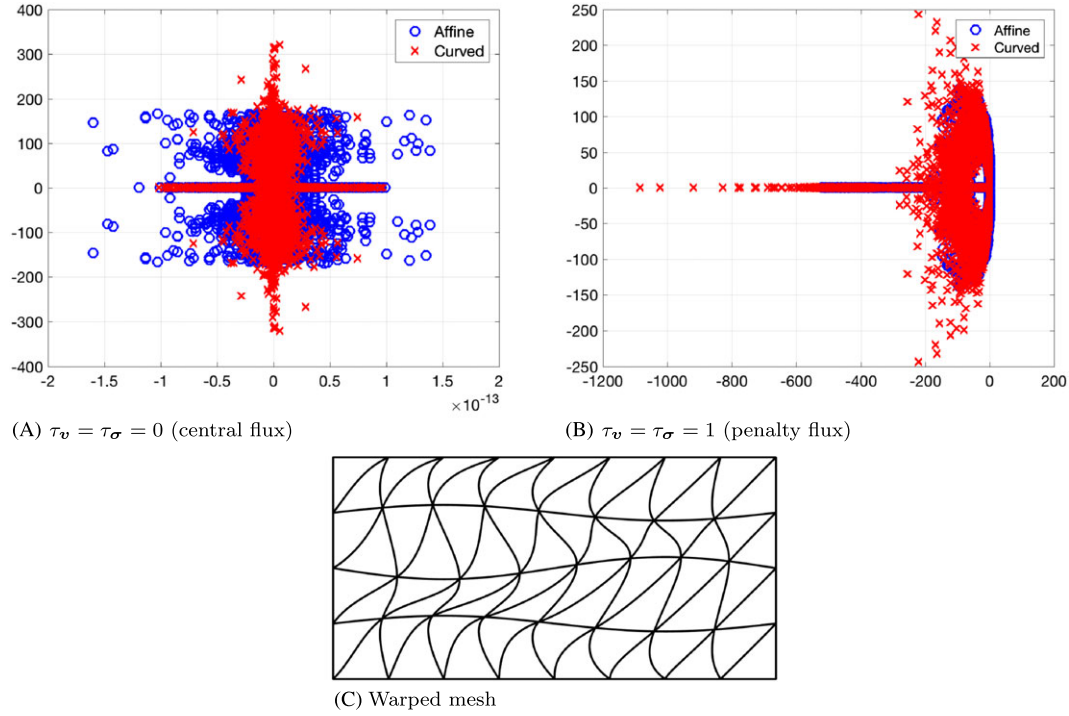

(C) Warped mesh

FIGURE 8 Spectra of the discontinuous Galerkin discretization matrix for central and penalty fluxes on a warped curvilinear mesh of degree $N = 3$
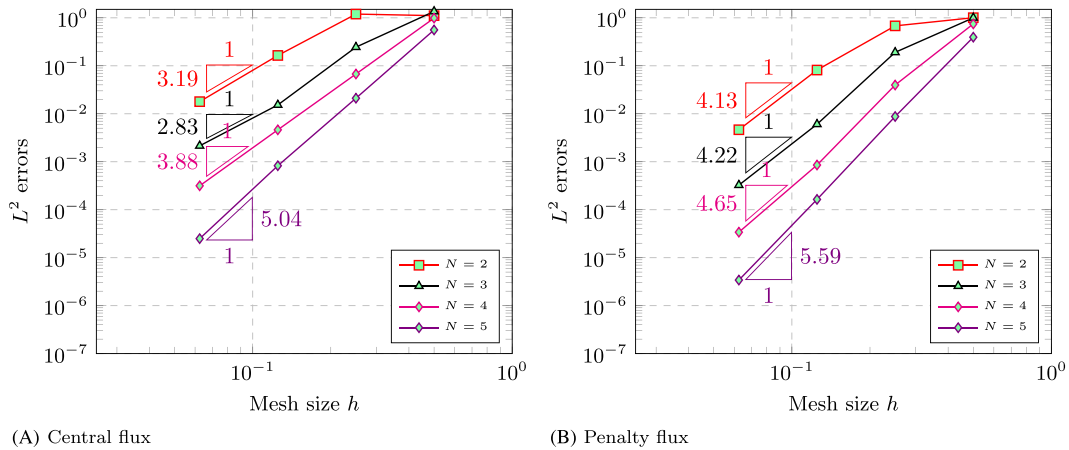

(A) Central flux   (B) Penalty flux

FIGURE 9 Convergence of weight-adjusted discontinuous Galerkin for the Lamb wave problem on curvilinear meshes

Figure 8 shows eigenvalues of the DG discretization matrix for $N = 3$ for both a uniform (affine) mesh and a warped curvilinear mesh. We use the quadrature-based skew-symmetric formulation introduced in Section 5.1 and consider both central and penalty fluxes (with penalty parameters set uniformly to 1). We observe that for both the central and penalty fluxes, all eigenvalues contain nonpositive real parts (up to machine precision), indicating that the semidiscrete system is energy stable. The introduction of the curvilinear warping appears to result in a magnification of the real and imaginary parts of larger magnitude eigenvalues.

We also compute L^2 errors on a sequence of refined curvilinear meshes for $N = 2, \ldots, 5$, skipping $N = 1$ as it reduces to the affine case. These curvilinear meshes are constructed using the warping procedure described previously. Errors for both central and penalty fluxes are shown in Figure 9. We observe rates of convergence of L^2 errors which are consistent with the rates observed for affine meshes in Section 6.2.2.

6.3 | Application examples

We next demonstrate the accuracy and flexibility of WADG for several application-based problems in linear elasticity with heterogeneity and anisotropy. All computations are done using penalty parameters $\tau_v = \tau_\sigma = 1$ unless specified otherwise.

6.3.1 | Stiff inclusion

The stiff inclusion problem is a common test of methods for linear elastic wave propagation,[3,25,42] where an inclusion with higher wavespeed is embedded within a nonstiff region. Waves which reach this region of high wavespeed are transmitted through the inclusion, bouncing back and forth within the region. This vibration then produces waves which propagate outward from the inclusion.

We solve on a domain $[-1, 1] \times [-0.5, 0.5]$ with a rectangular inclusion located at $[-0.5, 0.5] \times [-0.1, 0.1]$. Outside of the inclusion, material parameters are taken to be

$$\rho = 1, \qquad \mu = 1, \qquad \lambda = 2.$$

Within the inclusion, material parameters are taken to be

$$\rho = 1, \qquad \mu = 100, \qquad \lambda = 200,$$

such that the wave speed in the rectangular inclusion is ten times that of the wave speed outside. A pulse is generated through velocity boundary conditions at $x = -1$

$$v_1(x, y, t) = \begin{cases} \sin(\pi t / t_0), & t < t_0 \\ 0, & t \geq t_0 \end{cases}, \qquad v_2(x, y, t) = 0.$$

In our experiments, we take $t_0 = 0.025$. Traction free boundary conditions are enforced at all other domain boundaries.

We construct a uniform triangular mesh by dividing each element of a quadrilateral meshes along the diagonal to produce triangular meshes, using 100×50 quadrilateral elements in the x and y coordinates, respectively. This is the same mesh resolution and polynomial degree used by Käser and Dumbser[3] and provides roughly the same number of degrees of freedom in the x and y directions as the numerical setup used by Appelö and Hagstrom.[42] Figure 10 shows values of $|\sigma_{xx} + \sigma_{yy}|$ at final time $T = 0.4$. Following the approach taken in the literature,[3,25,42] a nonlinear color scale is used to distinguish small-amplitude waves and produce a *schlieren*-style image. The results show qualitatively good agreement with results in the literature.

6.3.2 | Heterogeneous anisotropic material

We next examine a model wave propagation problem in heterogeneous anisotropy media.[5,44,45] The density $\rho = 7100$ is constant over the domain, while the entries of the stiffness matrix $\boldsymbol{C}$ are taken to be

$$\boldsymbol{C}_{11} = 0.165, \quad \boldsymbol{C}_{12} = 0.05, \quad \boldsymbol{C}_{22} = 0.062, \quad \boldsymbol{C}_{33} = 0.0396, \qquad x < 0,$$
$$\boldsymbol{C}_{11} = 0.165, \quad \boldsymbol{C}_{12} = 0.0858, \quad \boldsymbol{C}_{22} = 0.165, \quad \boldsymbol{C}_{33} = 0.0396, \qquad x > 0,$$

with the remaining entries determined by symmetry or set to 0 if unspecified. For $x < 0$, this corresponds to an anisotropic material, while for $x > 0$, this corresponds to an isotropic material with $\mu = 0.0396$, $\lambda = 0.0858$.
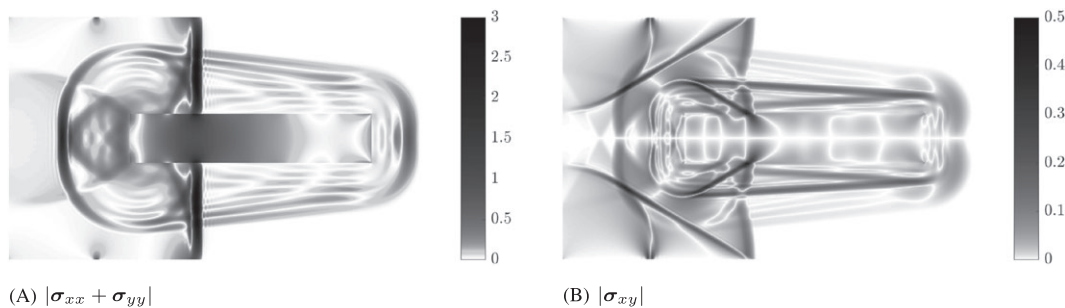

(A) $|\sigma_{xx} + \sigma_{yy}|$ (B) $|\sigma_{xy}|$

FIGURE 10 Values of different solution fields at $T = 0.4$ for the stiff inclusion problem of Leveque[25] with order of approximation $N = 5$

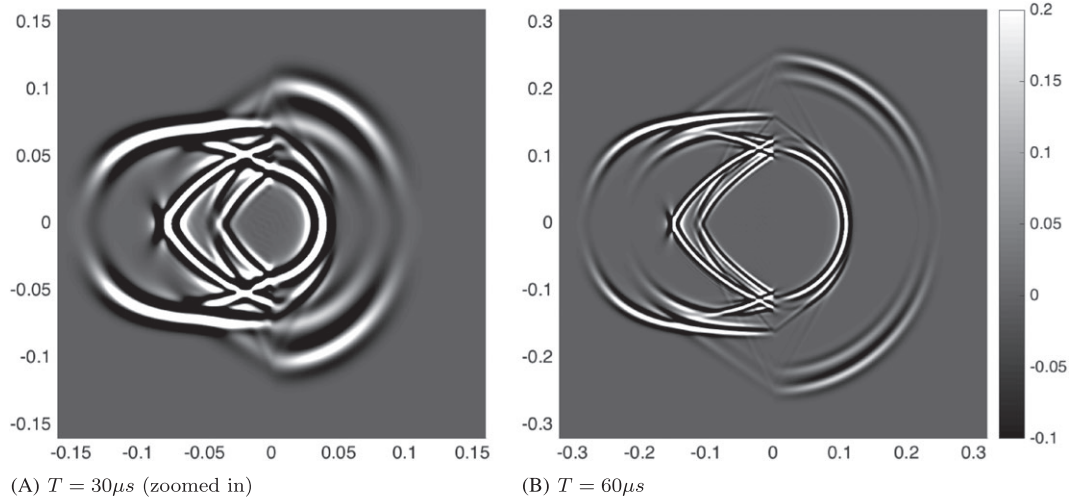**FIGURE 11** An example of wave propagation in heterogeneous anisotropic media. The vertical component of the velocity v_2 is shown at $T = 30$ and $T = 60 \mu s$

The computational domain is taken to be $[-0.32, 0.32]^2$, and we use $N = 5$ and a triangular mesh of 32 768 elements constructed by subdividing a grid of 128×128 uniform quadrilaterals. To provide a fair comparison to results in the literature, the degree N and the mesh size are chosen based on the numerical setups used in de la Puente et al and Komatitsch et al.[5,45] Komatitsch et al[45] use a 130×130 grid of uniform quadrilateral elements of degree $N = 5$. De la Puente et al[5] use an unstructured mesh of 37 944 elements of degree $N = 5$, where the average triangle edge length matches the edge length of the triangles in our mesh. Forcing is applied to the y component of the velocity by a Ricker wavelet point source

$$f(\boldsymbol{x}, t) = \left(1 - 2(\pi f_0(t - t_0))^2\right) e^{-(\pi f_0(t-t_0))^2} \delta(x - x_0),$$

where $x_0 = -.02, f_0 = .17$, and $t_0 = 1/f_0$.[‡]

We take the penalty parameters to be $\tau_v = \tau_\sigma = 1/2$. While there is little visual difference between taking the penalty parameters to be 1 instead of 1/2, we observe that taking a smaller penalty parameter makes it possible to use a larger time-step ($C_{\text{CFL}} = 10$) without blowing up. Reducing the penalty parameter further does not appear to allow a significant increase in the maximum stable time-step. This suggests that the naive choice of $\tau_v = \tau_\sigma = 1$ is not optimal with respect to the maximum stable time-step and stiffness of the semidiscrete system, as discussed in Section 6.1.

Figure 11 shows the y component of velocity v_2 at times $T = 30 \mu s$ (zoomed in) and $T = 60 \mu s$. Both results show qualitative agreement with reference results from de la Puente et al, Carcione et al, and Komatitsch et al.[5,44,45]

6.4 | A 3-dimensional example and computational results

We now present a 3-dimensional example of elastic wave propagation in heterogeneous media with subelement variations and a discontinuity across an interface. We consider isotropic elastic wave propagation on the cube $[-0.5, 0.5]^3$ with a discontinuity in material coefficients across $z = 0$

$$\rho = 1, \qquad \mu(\boldsymbol{x}) = \begin{cases} 2 + w(\boldsymbol{x}), & z < 0, \\ 1 + w(\boldsymbol{x}), & z > 0, \end{cases} \qquad \lambda(\boldsymbol{x}) = \begin{cases} 2, & z < 0, \\ 1, & z > 0, \end{cases}$$

where $w(\boldsymbol{x}) = 0.5 \cos(3\pi x) \cos(3\pi y) \cos(3\pi z)$. Forcing is applied to the x component of velocity through a smoothed point source and Ricker wavelet

$$f(\boldsymbol{x}, t) = \left(1 - 2(\pi f_0(t - t_0))^2\right) e^{-(\pi f_0(t-t_0))^2} e^{-(a\|\boldsymbol{x} - \boldsymbol{x}_0\|)^2},$$

where $\boldsymbol{x}_0 = (0, 0, 0.1)^T, a = 100, f_0 = 10$, and $t_0 = 1/f_0$.

[‡]All values and units are adapted from de la Puente et al and Komatitsch et al[5,45] and correspond to units of meters, kilograms, and microseconds.
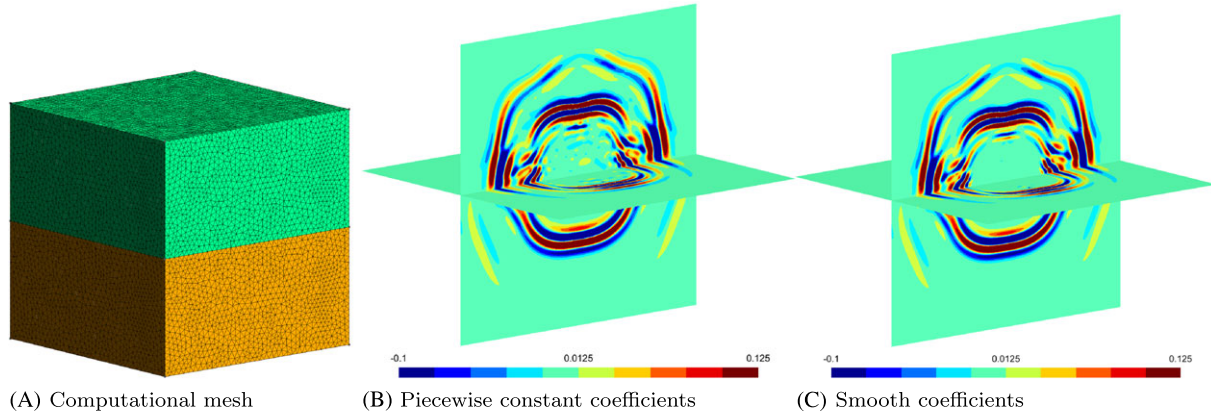
FIGURE 12 Mesh and xz, xy slices of $\boldsymbol{v}_1$ at $T = 0.5$. The order of approximation is taken to be $N = 5$

Figure 12 shows the x velocity of the computed solution at $T = 0.5$, and Figure 12A shows the unstructured mesh of 222 824 tetrahedral elements of degree $N = 5$ used to compute both solutions. To capture the discontinuity in material parameters, the elements of this mesh are made to conform to the $z = 0$ plane. The mesh resolution and degree are chosen to resolve the spatial variation of the smoothed point source present in the forcing function. For piecewise constant coefficients (using the average of each coefficient over an element), spurious reflections are observed in the solution. When smoothly varying coefficients are resolved within an element using WADG, these spurious reflections disappear.

These computations are performed on an Nvidia GTX 980 GPU, following the implementation of GPU-accelerated DG methods outlined in Klöckner et al.[9] This approach breaks the computational work for each time-step into volume and surface kernels (for the evaluation of the DG formulation) and an update kernel (for the application of a time integration method). In this implementation, we apply the weight-adjusted mass matrix inverse within the update kernel as well. Strategies for volume and surface kernels follow Klöckner et al,[9] while computational approaches for WADG are outlined in Chan et al.[17]

Noninvasive GPU-accelerated implementations of WADG are described in Chan et al,[17] where kernels for the acoustic wave equation in isotropic media are reused. We rewrite high-order DG methods based on explicit inversion of weighted mass matrices (as done in Mercerat and Glinsky[14] and Bencomo[15]) into a similar noninvasive form with equivalent storage. The semidiscrete form of standard DG yields a system of ODEs over each element

$$\boldsymbol{M}_w \frac{\mathrm{d}\boldsymbol{U}}{\mathrm{dt}} = \boldsymbol{A}_h \boldsymbol{U},$$

where $\boldsymbol{A}_h \boldsymbol{U}$ denotes the evaluation of the DG right-hand side for some local vector $\boldsymbol{U}$. Multiplying by an unweighted mass matrix on both sides gives

$$\boldsymbol{M}^{-1} \boldsymbol{M}_w \frac{\mathrm{d}\boldsymbol{U}}{\mathrm{dt}} = \boldsymbol{M}^{-1} \boldsymbol{A}_h \boldsymbol{U}.$$

The right-hand side is the same as the right-hand side for the case when the weighting function is $w(x) = 1$ and can reuse DG kernels for isotropic wave propagation. The influence of the spatially varying coefficient is incorporated by inverting the weighted projection matrix $\boldsymbol{P}_w = \left(\boldsymbol{M}^{-1}\boldsymbol{M}_w\right)^{-1} = \boldsymbol{M}_w^{-1}\boldsymbol{M}$ and applying it to the right-hand side. Since the weight w is spatially varying and distinct from element-to-element, we precompute and store $\boldsymbol{P}_w$ explicitly over each element prior to time-stepping. The WADG method is equivalent to replacing the matrices $\boldsymbol{M}_w^{-1}\boldsymbol{M}$ with the weight-adjusted projection matrix $\boldsymbol{M}^{-1}\boldsymbol{M}_{1/w}$, which can be applied in a matrix-free fashion as described in Section 4.2.2.

We now examine computational costs associated with the use of the WADG method. Computational statistics are computed using the Nvidia profiler `nvprof`. We consider first the costs associated with the use of weight-adjusted approximations to scalar weighted mass matrices. While WADG clearly reduces storage costs associated with high-order DG methods, it is less clear how WADG affects computational runtime on accelerator and many-core architectures.

We compare the application of precomputed and stored weighted projection matrices $\boldsymbol{P}_w$ with a matrix-free application of the weight-adjusted matrix $\boldsymbol{M}^{-1}\boldsymbol{M}_{1/w}$ using a GPU-accelerated implementation. As described in Klöckner et al and

TABLE 2 Runtimes (nanoseconds) per element for weighted and weight-adjusted projections as a function of batch size N_{batch}. The lowest runtimes are highlighted in bold

Batch Size N_{batch}	1	2	3	4	5	6
	(a) Weight-Adjusted Projection $\boldsymbol{P}_w$					
N = 1	2.264	1.224	0.8927	0.7514	0.6976	**0.6571**
N = 2	3.185	2.83	**2.791**	2.823	2.837	2.86
N = 3	**9.907**	10.14	10.25	10.16	10.19	10.21
N = 4	**29.41**	29.74	30.01	30.03	30.23	30.47
N = 5	74.48	74.39	74.16	74.01	**73.88**	74.32
N = 6	173.4	173.9	175.2	171	**170.5**	172.4
N = 7	**329.4**	330.4	329.6	331		
	(b) Weight-Adjusted Projection $\boldsymbol{M}^{-1}\boldsymbol{M}_{1/w}$					
N = 1	2.382	1.26	0.8833	0.6986	**0.5832**	0.6534
N = 2	3.864	2.101	2.495	1.968	2.28	**1.964**
N = 3	7.092	**6.788**	6.9	6.89	6.888	6.836
N = 4	24.79	**22.15**	26.63	24.02	24.16	24
N = 5	70.62	76.6	61.55	**56.35**	56.38	58.98
N = 6	179.8	145.7	**129.9**	145.7	131.9	144.1
N = 7	411.6	412.7	**393**	510.3		

Chan et al,[9,17] we batch process $N_{\text{batch}} \geq 1$ elements within a single-kernel workgroup. Table 2 displays the average runtime per element for the weighted and weight-adjusted projection kernels as a function of batch size N_{batch} when using a mesh of 50 000 elements and a quadrature which is exact for polynomials of degree $2N + 1$.

When processing only a single element per batch, WADG is less efficient than weighted projection at all orders. However, the cost of WADG goes down rapidly with the number of elements per batch. After optimizing over the batch size, weight-adjusted projection is faster than weighted projection up to $N = 6$. At $N = 7$, WADG is slower than weighted projection, as the batch size is limited by the maximum number of active threads.[§] We note that, if the strength of quadrature is reduced from $2N + 1$ to $2N$, the resulting WADG runtimes are faster than weighted projection at all tested orders, achieving between a 1.5× and 2.3× speedup for $N = 1, \ldots, 7$ while maintaining virtually identical numerical results.[16]

We can take a closer look at these results using the Nvidia profiler `nvprof`, looking in particular at the metrics `gld_load_throughput` and `dram_read_throughput`. Both metrics track data throughput; however, the former includes data fetched from cache, while the latter does not. We fix $N = 2$, increase the batch size, and record the output given by `nvprof` for the weighted projection and weight-adjusted kernels. Table 3 shows that the value of `dram_read_throughput` for the weighted projection kernel is higher than that of the weight-adjusted kernel, implying that more data are streamed through the kernel. However, the value of `gld_load_throughput` for the weight-adjusted kernel is higher than that of the weighted projection kernel. This indicates that, while the loading of precomputed and stored weighted projection matrices exploits the high bandwidth available to GPUs, it does not take advantage of cache locality due to the fact that the projection matrices must be loaded separately over each element. In contrast, the matrix-free implementation of WADG allows the matrices $\boldsymbol{V}_q, \boldsymbol{P}_q$ to be reused over multiple elements once loaded into cache.

For $N_{\text{batch}} = 1$, the cache efficiency of the weight-adjusted kernel is offset by the increased computational cost of quadrature-based interpolation and projection. However, increasing N_{batch} to 3 elements increases both the values of `gld_load_throughput` and `dram_read_throughput`, resulting in a roughly 2× speedup in runtime for the weight-adjusted kernel. These results show that the low-storage nature of the weight-adjusted kernel frees up bandwidth in exchange for increased computational work, while taking advantage of data locality.

Finally, we compute the GFLOPS per second and bandwidth (GB/s) achieved by each of the kernels for elastic wave propagation in our implementation. The results are shown in Figure 13 and are qualitatively similar to the results reported

[§]In our implementation, the number of active threads per workgroup is the number of quadrature points multiplied by the number of elements per batch. The batch size is limited by the maximum number of threads in a workgroup (1024 for Nvidia GPUs). For $N = 7$, the quadrature rule of degree $2N + 1$ contains 214 points; the largest batch size we can run is then 4 elements, as processing 5 elements per batch requires 1070 threads.

TABLE 3 Reported global load throughput (`gld_throughput`) and device memory read throughput (`dram_read_throughput`) in GB/s for $N = 2$ and various N_{batch}

	$N_{\text{batch}} = 1$	$N_{\text{batch}} = 3$	$N_{\text{batch}} = 5$
(a) `dram_read_throughput`, (GB/s)			
Weighted projection $\boldsymbol{P}_w$	135.81	156.23	153.13
Weight adjusted projection $\boldsymbol{M}^{-1}\boldsymbol{M}_{1/w}$	25.041	38.811	42.736
(b) `gld_throughput`, (GB/s)			
Weighted projection $\boldsymbol{P}_w$	353.10	411.87	410.95
Weight adjusted projection $\boldsymbol{M}^{-1}\boldsymbol{M}_{1/w}$	804.30	1e+03	1e+03

(A) GFLOPS per second)
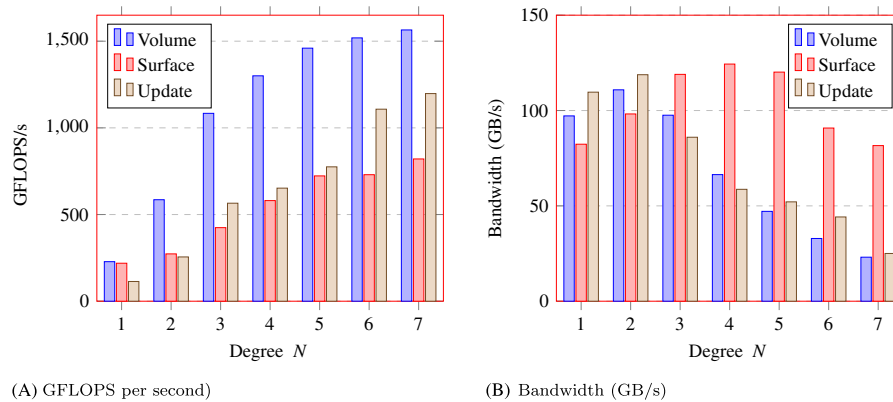

(B) Bandwidth (GB/s)

FIGURE 13 Profiled GFLOPS/s and bandwidth (GB/s) for volume, surface, and update kernels. Results are presented for an Nvidia GTX 980 GPU, on a mesh of 9918 elements

in Modave et al[11] for the volume and surface kernels for elasticity. The GFLOPS/s and bandwidth for the update kernel fall between the reported values for the volume and surface kernel. The runtime of the update kernel for elastic wave propagation (in which the weight-adjusted projection matrix is applied) constitutes between 40% and 50% of the total runtime for $N = 1, \ldots, 7$. In comparison, the update kernel for piecewise constant material properties takes roughly 35% of the runtime at $N = 1$ and 10% of the total runtime at $N = 7$, due to the fact that no additional matrix multiplications are necessary in the update kernel if material properties are assumed to be constant within an element.

7 | CONCLUSIONS

This work presents a WADG method for the linear elastic wave equations with arbitrary heterogeneous media. The method is energy stable and high-order accurate for arbitrary stiffness matrices, and a slight modification results in an energy-stable method for curvilinear meshes as well. The penalty numerical fluxes for this formulation are simple to derive and implement, and their lack of dependence on the stiffness matrix allows for a unified treatment of isotropic and anisotropic media. Numerical examples confirm the accuracy of this method for analytic solutions of the elastic wave equations, as well as its high-order accuracy with respect to a reference solution for smoothly varying heterogeneous media. Results obtained using this method also show good agreement with existing results in the literature for both problems involving both isotropic and anisotropic heterogeneous media. Finally, we provide computational results demonstrating the performance of the proposed methods on a single GPU.

We note that the implementation of this method reduces to the application of the weight-adjusted mass matrix inverse and the evaluation of constant-coefficient terms in the DG formulation. The cost of the latter step can be reduced (especially at high orders of approximation) by using fast methods based on Bernstein-Bezier bases for the application of derivative and lift matrices for constant-coefficient terms.[24] Future work will also involve a more careful study of discretization parameters (such as the penalty parameters and the points per wavelength required for accuracy), as well as the application of the proposed method to more realistic geophysical settings.

ORCID

Jesse Chan http://orcid.org/0000-0003-2077-3636

REFERENCES

1. Virieux J. P-SV wave propagation in heterogeneous media: velocity-stress finite-difference method. *Geophysics*. 1986;51(4):889-901.

2. Komatitsch D, Vilotte JP. The spectral element method: an efficient tool to simulate the seismic response of 2D and 3D geological structures. *Bull Seismol Soc Am*. 1998;88(2):368-392.

3. Käser M, Dumbser M. An arbitrary high-order discontinuous Galerkin method for elastic waves on unstructured meshes - I. The two-dimensional isotropic case with external source terms. *Geophys J Int*. 2006;166(2):855-877.

4. Dumbser M, Käser M. An arbitrary high-order discontinuous Galerkin method for elastic waves on unstructured meshes - II. The three-dimensional isotropic case. *Geophys J Int*. 2006;167(1):319-336.

5. de la Puente J, Käser M, Dumbser M, Igel H. An arbitrary high-order discontinuous Galerkin method for elastic waves on unstructured meshes-IV. Anisotropy. *Geophys J Int*. 2007;169(3):1210-1228.

6. Delcourte S, Fezoui L, Glinsky-Olivier N. A high-order discontinuous Galerkin method for the seismic wave propagation. In: ESAIM: Proceedings, Vol. 27. EDP Sciences; 2009:70-89.

7. Delcourte S, Glinsky N. Analysis of a high-order space and time discontinuous Galerkin method for elastodynamic equations. Application to 3D wave propagation. *ESAIM: Math Modell Numer Anal*. 2015;49(4):1085-1126.

8. Ye R, de Hoop MV, Petrovitch CL, Pyrak-Nolte LJ, Wilcox LC. A discontinuous Galerkin method with a modified penalty flux for the propagation and scattering of acousto-elastic waves. *Geophys J Int*. 2016;205(2):1267-1289.

9. Klöckner A, Warburton T, Bridge J, Hesthaven JS. Nodal discontinuous Galerkin methods on graphics processors. *J Comput Phys*. 2009;228(21):7863-7882.

10. Modave A, St-Cyr A, Mulder WA, Warburton T. A nodal discontinuous Galerkin method for reverse-time migration on GPU clusters. *Geophys J Int*. 2015;203(2):1419-1435.

11. Modave A, St-Cyr A, Warburton T. GPU performance analysis of a nodal discontinuous Galerkin method for acoustic and elastic models. *Computers & Geosciences*. 2016;91:64-76.

12. Chan J, Wang Z, Modave A, Remacle JF, Warburton T. GPU-accelerated discontinuous Galerkin methods on hybrid meshes. *J Comput Phys*. 2016;318:142-168.

13. Castro CE, Käser M, Brietzke GB. Seismic waves in heterogeneous material: subcell resolution of the discontinuous Galerkin method. *Geophys J Int*. 2010;182(1):250-264.

14. Mercerat ED, Glinsky N. A nodal high-order discontinuous Galerkin method for elastic wave propagation in arbitrary heterogeneous media. *Geophys J Int*. 2015;201(2):1101-1118.

15. Bencomo MJ. Discontinuous Galerkin and finite difference methods for the acoustic equations with smooth coefficients. *Master's Thesis*; 2015.

16. Chan J, Hewett RJ, Warburton T. Weight-adjusted discontinuous Galerkin methods: wave propagation in heterogeneous media. *arXiv preprint arXiv:1608.01944*. 2016. Accepted to SISC.

17. Chan J, Hewett RJ, Warburton T. Weight-adjusted discontinuous Galerkin methods: curvilinear meshes. *SIAM J Sci Comput*. 2016;39(6):A2395-A2421. (27 pages).

18. Warburton T. A low-storage curvilinear discontinuous Galerkin method for wave problems. *SIAM Journal on Scientific Computing*. 2013;35(4):A1987-A2012.

19. Hughes TJR, Marsden JE. Classical elastodynamics as a linear symmetric hyperbolic system. *Journal of Elasticity*. 1978;8(1):97-110.

20. Wilcox LC, Stadler G, Burstedde C, Ghattas O. A high-order discontinuous Galerkin method for wave propagation through coupled elastic-acoustic media. *J Comput Phys*. 2010;229(24):9373-9396.

21. Chan J, Warburton T. On the penalty stabilization mechanism for upwind discontinuous Galerkin formulations of first order hyperbolic systems. *Comput Math Appl*. 2017. http://www.sciencedirect.com/science/article/pii/S0898122117304819

22. Fezoui L, Lanteri S, Lohrengel S, Piperno S. Convergence and stability of a discontinuous Galerkin time-domain method for the 3D heterogeneous Maxwell equations on unstructured meshes. *ESAIM: Math Modell Numer Anal*. 2005;39(6):1149-1176.

23. Hesthaven JS, Warburton T. *Nodal Discontinuous Galerkin Methods: Algorithms, Analysis, and Applications*, Vol. 54: Springer; 2007.

24. Chan J, Warburton T. GPU-accelerated Bernstein-Bézier discontinuous Galerkin methods for wave problems. *SIAM J Sci Comput*. 2017;39(2):A628-A654.

25. LeVeque RJ. *Finite Volume Methods for Hyperbolic Problems*, Vol. 31: Cambridge University Press; 2002.

26. Berenger JP. A perfectly matched layer for the absorption of electromagnetic waves. *J Comput Phys*. 1994;114(2):185-200.

27. Hagstrom T, Warburton T. A new auxiliary variable formulation of high-order local radiation boundary conditions: corner compatibility conditions and extensions to first-order systems. *Wave Motion*. 2004;39(4):327-338.

28. Modave A, Atle A, Chan J, Warburton T. A GPU-accelerated nodal discontinuous Galerkin method with high-order absorbing boundary conditions and corner/edge compatibility. *Int J Numer Methods Eng*. 2017. https://doi.org/10.1002/nme.5576

29. Warburton T. An explicit construction of interpolation nodes on the simplex. *J Eng Math*. 2006;56(3):247-262.

30. Etienne V, Chaljub E, Virieux J, Glinsky N. An hp-adaptive discontinuous Galerkin finite-element method for 3-D elastic wave modelling. *Geophys J Int*. 2010;183(2):941-962.

31. Xiao H, Gimbutas Z. A numerical algorithm for the construction of efficient quadrature rules in two and higher dimensions. *Comput Math Appl*. 2010;59:663-676.

32. Karniadakis G, Sherwin S. *Spectral/hp Element Methods for Computational Fluid Dynamics*: Oxford University Press; 2013.

33. Engvall L, Evans JA. Isogeometric triangular Bernstein-Bézier discretizations: automatic mesh generation and geometrically exact finite element analysis. *Comput Meth Appl Mech Eng*. 2016;304:378-407.

34. Michoski C, Chan J, Engvall L, Evans JA. Foundations of the blended isogeometric discontinuous Galerkin (BIDG) method. *Comput Meth Appl Mech Eng*. 2016;305:658-681. http://www.sciencedirect.com/science/article/pii/S0045782516300457

35. Houston P, Schwab C, Süli E. Discontinuous *hp*-finite element methods for advection-diffusion-reaction problems. *SIAM J Numer Anal*. 2002;39(6):2133-2163.

36. Warburton T, Hesthaven JS. On the constants in *hp*-finite element trace inverse inequalities. *Comput Meth Appl Mech Eng*. 2003;192(25):2765-2773.

37. Dolejší V, Feistauer M. *Discontinuous Galerkin Method: Analysis and Applications to Compressible Flow*, Vol. 48: Springer; 2015.

38. Carpenter MH, Kennedy CA. Fourth-order 2*n*-storage Runge-Kutta schemes. Technical Report NASA-TM-109112, NAS 1.15:109112, NASA Langley Research Center; 1994.

39. Cockburn B, Dong B, Guzmán J. Optimal convergence of the original DG method for the transport-reaction equation on special meshes. *SIAM J Numer Anal*. 2008;46(3):1250-1265.

40. Johnson C, Pitkäranta J. An analysis of the discontinuous Galerkin method for a scalar hyperbolic equation. *Math Comput*. 1986;46(173):1-26.

41. Sjögreen B, Petersson NA. A fourth order accurate finite difference scheme for the elastic wave equation in second order formulation. *SIAM J Sci Comput*. 2012;52(1):17-48.

42. Appelö D, Hagstrom T. An energy-based discontinuous Galerkin discretization of the elastic wave equation in second order form. 2015. Submitted to CMAME.

43. Stoneley R. Elastic waves at the surface of separation of two solids. *Proc. R. Soc. London, Ser. A, Containing Papers of a Mathematical and Physical Character*. 1924;106(738):416-428.

44. Carcione JM, Kosloff D, Kosloff R. Wave-propagation simulation in an elastic anisotropic (transversely isotropic) solid. *Q J Mech Appl Math*. 1988;41(3):319-346.

45. Komatitsch D, Barnes C, Tromp J. Simulation of anisotropic wave propagation based upon a spectral element method. *Geophysics*. 2000;65(4):1251-1260.