Proceedings of the Project Review, Geo-Mathematical Imaging Group (Purdue University, West Lafayette IN), Vol. 1 (2013) pp. 103-130.

A DISCONTINUOUS GALERKIN METHOD FOR THE ANISOTROPIC ELASTIC WAVE EQUATION

RUICHAO YE[*], CHRISTOPHER L. PETROVITCH[†], MAARTEN V. DE HOOP[‡], LAURA J. PYRAK-NOLTE[§], LUCAS WILCOX[¶], AND YINGCHONG SITU[||]

Abstract. In this paper, the anisotropic elastic wave equation is solved computationally using the discontinuous Galerkin (DG) method for problems that range in scale from the laboratory to the planetary scale. Any numerical method used to propagate waves efficiently in complex media with geometric features spanning several length scales, must accommodate tetrahedral unstructured meshes to align the element boundaries with all of the sharp velocity contrasts that exist throughout the computational domain. Tetrahedrals are an ideal element type to use because numerous automated meshing packages are available. The DG method is well suited to meet these requirements because the approximation order can be adjusted to match the needs of any computational domain and it can be used for unstructured meshes with any element type. In this study, the DG method is used with the semi implicit-explicit (IMEX) Runge-Kutta method to discretize the spatial derivatives of the anisotropic elastic wave equation and to integrate forward in time. Several real world applications are given to demonstrate the power of the DG method. Most notably, a 3D transversely isotropic medium with rough surfaces based on laboratory measurements of fractures in Austin Chalk was used to show the ability of the DG method to simulate the complex wave fields associated with rough surfaces in carbonate rocks. At the Earth scale, the 3D SEAM model was adaptively meshed to align the tetrahedral facets with the shape velocity contrasts, and then a seismic reflection calculation was performed. From the applications discussed in this paper, the efficiency of using unstructured tetrahedral elements with the DG method is demonstrated through accurate computations of the anisotropic elastic wave equation for computational domains with extremely complex geometric features.

1. Introduction.

Even with modern high performance computing technologies, the ability to simulate wave propagation through the Earth's subsurface remains challenging because of the range of length and time scales required to model seismic wave interactions in complex media. For example, the Earth's subsurface contains complex physical structures that include micro-cracks, fractures, joints, velocity anomalies, folds, faults, etc. within the crust, in addition to the structure of the Earth (crust-mantle-core boundaries). In simulations, seismic waves are required to travel accurately over many wavelengths while maintaining a controllable amount of dispersion. Most of these structures require a Body-Fitted Coordinate (BFC) method to capture the complexity of the structure. While a regular grid can capture some of these features with curvilinear coordinates [19], unstructured meshes are more flexible and can be locally refined to match any complex geometry. The ideal computational method for modeling wave interaction with the subsurface would use an unstructured mesh equipped with high-order accuracy.

Many different methods exist to simulate the elastic wave equation such as Finite Difference Finite Time (FDFT) method and the Spectral Element (SE) method. Virieux (1986) modeled the linearly elastic and isotropic elastic wave equation with FDFT. In his approach, he placed the velocity and stress fields on a staggered grid that enabled the waves to propagate over large distances. However, the dispersion grew too large to provide accurate results [34]. A major disadvantage of FDFT methods is the layout of the grid points, i.e., domain features are required to line up with the grid points, otherwise dispersive errors grow considerably. The FE method was proposed to solve this issue by using unstructured meshes to account for complex geometries. While effective, inverting the mass matrix at each time step is computationally expensive. Inverting the mass matrix has been approached in two ways, mass lumping [15] and the SE method [18]. The SE method uses high order hexahedral elements, combined with the Gauss-Lobatto-Legendre quadrature points, to

[*] Department of Earth and Atmospheric Sciences, Purdue University, IN (rye@purdue.edu)

[†] Physics Department, Purdue University, IN (cpetrovi@purdue.edu)

[‡] Department of Mathematics, Purdue University, 150 N. University Street, West Lafayette IN 47907, USA., mdehoop@math.purdue.edu

[§] Physics Department, School of Civil Engineering, and the Department of Earth and Atmospheric Sciences, Purdue University, IN (ljpn@purdue.edu)

[¶] Navel Postgraduate School, CA (lwilcox@nps.edu)

[||] Department of Computer Science, Purdue University, IN (ysitu@cs.purdue.edu)

diagonalize the mass matrix. The SE method has been successfully used to simulate global earth models; however, it requires the use of hexahedral elements because quadrature points do not exist for a tetrahedral. Hexahedral elements are a steep requirement, as tetrahedral mesh generators are far more common and are able to model complex boundaries more efficiently. The DG method is an improvement to this computational toolbox with regard to the elastic wave equation. It can incorporate any type of mesh element and has, by construction, a block diagonal mass matrix, thereby accounting for the past methods shortcomings.

The DG method was first used by Reed and Hill to model the neutron transport equation [29], but until recently was not used much. Several DG implementations exist now, most notably the ADER-DG approach by Käser et al. which uses upwinding Godouv fluxes and provides equal accuracy in both space and time [8]. Another noteworthy approach is the implementation by Etienne et al. that focuses on low-order solutions with convolutional perfectly matching layers (CPML) suitable for large scale 3D simulations [10]. Both of these approaches have accounted for elastic materials, but do not correctly handle acoustic-elastic interfaces. Wilcox et al. solved the Riemann problem exactly for the linearly elastic and isotropic wave equation, such that all complex geometries are accurately modeled by aligning the mesh elements along internal elastic-elastic, acoustic-acoustic, and acoustic-elastic interfaces [35]. All of these implementations have formulated the DG method in the non-conservative form that requires the material properties within any given element to be constant. Smith *et al.* have removed this restriction by writing the elastic wave equation in its conservative form [33].

In this paper, we focus on domains with anisotropic media and apply it to realistic problems in global seismology and laboratory scale rock physics. In the next section, the strong formulation for the fully the anisotropic elastic wave equation is derived. Then, a stable numerical flux that is suitable for the general anisotropic elastic wave equation is presented. In Section 4, we summarize the implementation of the external boundary conditions and present the derivation CPML regions. In Section 5, suitable explicit/explicit-implicit time integration algorithms are described. Then, a detailed overview of the parallelization of the DG method is described. Our implementation was verified by generating Rayleigh and Stoneley waves (Section 7). Our DG codes were then extensively tested by computing the solutions to laboratory rock physics problems, and global seismology problems, both of which included embedding anisotropic materials. Lastly, the various advantages and disadvantages are discussed in Section 9.

2. Discontinuous Galerkin Method. The generalized Hooke's law in an elasto-dynamic system is expressed by relating stress, S_{ij}, and strain, E_{kl}. Assuming small deformations gives a linear relationship, i.e. $S_{ij} = c_{ijkl}E_{kl}$, where c_{ijkl} is the stiffness tensor. The 81 component stiffness tensor can be reduced, in general, to 21 independent components through symmetry arguments and is written in matrix form by defining the stress and strain vectors as $\boldsymbol{S} = (S_{11}, S_{22}, S_{33}, S_{23}, S_{12}, S_{13})^T$ and $\boldsymbol{E} = (E_{11}, E_{22}, E_{33}, E_{23}, E_{12}, E_{13})^T$. The stiffness tensor is then rewritten as a 6 by 6 matrix, $\boldsymbol{C}$, defined by,

$$\boldsymbol{S} = \begin{bmatrix} C_{11} & C_{12} & C_{13} & 2C_{14} & 2C_{15} & 2C_{16} \\ C_{12} & C_{22} & C_{13} & 2C_{24} & 2C_{25} & 2C_{26} \\ C_{13} & C_{23} & C_{33} & 2C_{34} & 2C_{35} & 2C_{36} \\ C_{14} & C_{24} & C_{34} & 2C_{44} & 2C_{45} & 2C_{46} \\ C_{15} & C_{25} & C_{35} & 2C_{45} & 2C_{55} & 2C_{56} \\ C_{16} & C_{26} & C_{36} & 2C_{46} & 2C_{56} & 2C_{66} \end{bmatrix} \boldsymbol{E} \equiv \boldsymbol{CE} \tag{2.1}$$

Written in this form, all of the symmetry classes are only specific cases of the more general class, i.e. isotropic, transversely isotropic, orthorhombic, etc. For example, the isotropic case can be used simply by setting all of the C_{ij} components to zero except for $C_{11} = \lambda + 2\mu$, $C_{12} = C_{13} = C_{23} = \lambda$, $C_{44} = \mu$, $C_{55} = \mu$, and $C_{66} = \mu$. (λ, μ) are the *Lamé* constants and ρ is the density. By combining Equation 2.1 with the equation of motion, the anisotropic elasto-dynamic equations is written in terms of the strain, $\boldsymbol{E}$, and the particle velocity, $\boldsymbol{v}$,

$$\begin{aligned} \frac{\partial \boldsymbol{E}}{\partial t} &= \frac{1}{2}(\nabla \boldsymbol{v} + \nabla \boldsymbol{v}^T), \\ \rho \frac{\partial \boldsymbol{v}}{\partial t} &= \nabla \cdot (\boldsymbol{C}\boldsymbol{E}) + \boldsymbol{f}, \end{aligned} \tag{2.2}$$

This can be simplified for the acoustic wave equation. The first-order pressure-velocity expression is,

$$\begin{aligned} \frac{\partial \boldsymbol{P}}{\partial t} &= \lambda \nabla \cdot \boldsymbol{v} \\ \frac{\partial \boldsymbol{v}}{\partial t} &= \frac{1}{\rho} \nabla \boldsymbol{P} \end{aligned} \tag{2.3}$$

Both Equation 2.2 and Equation 2.3 are rewritten in a unified form,

$$\partial_t \boldsymbol{q} + \nabla \cdot (\tilde{A}\boldsymbol{q}) = \boldsymbol{g}, \tag{2.4}$$

in which $\tilde{A} \equiv \tilde{A}(\boldsymbol{x})$ contains spatial dependent media parameters. For the three dimensional anisotropic elastic equation, we define the array of unknowns as,

$$\boldsymbol{q} = (E_{11}, E_{22}, E_{33}, E_{23}, E_{13}, E_{12}, v_1, v_2, v_3)^T, \tag{2.5}$$

Then, the operator, $\tilde{A}$, is defined as,

$$\tilde{A} \equiv A^x n_x + A^y n_y + A^z n_z \tag{2.6}$$

where A^i are the Jacobi matrices for the system and are defined as follows:

$$\begin{aligned}
A^x &= \left(\begin{array}{cccccc|ccc}
 & & & & & & 1 & 0 & 0 \\
 & & & & & & 0 & 0 & 0 \\
 & & & \mathbf{0} & & & 0 & 0 & 0 \\
 & & & & & & 0 & 0 & 0 \\
 & & & & & & 0 & 0 & \frac{1}{2} \\
 & & & & & & 0 & \frac{1}{2} & 0 \\
\hline
C_{11} & C_{12} & C_{13} & 2C_{14} & 2C_{15} & 2C_{16} & & & \\
C_{16} & C_{26} & C_{36} & 2C_{46} & 2C_{56} & 2C_{66} & & \mathbf{0} & \\
C_{15} & C_{25} & C_{35} & 2C_{45} & 2C_{55} & 2C_{56} & & &
\end{array}\right), \\
A^y &= \left(\begin{array}{cccccc|ccc}
 & & & & & & 0 & 0 & 0 \\
 & & & & & & 0 & 1 & 0 \\
 & & & \mathbf{0} & & & 0 & 0 & 0 \\
 & & & & & & 0 & 0 & \frac{1}{2} \\
 & & & & & & 0 & 0 & 0 \\
 & & & & & & \frac{1}{2} & 0 & 0 \\
\hline
C_{16} & C_{26} & C_{36} & 2C_{46} & 2C_{56} & 2C_{66} & & & \\
C_{12} & C_{22} & C_{23} & 2C_{24} & 2C_{25} & 2C_{26} & & \mathbf{0} & \\
C_{14} & C_{24} & C_{34} & 2C_{44} & 2C_{45} & 2C_{46} & & &
\end{array}\right), \\
A^z &= \left(\begin{array}{cccccc|ccc}
 & & & & & & 0 & 0 & 0 \\
 & & & & & & 0 & 0 & 0 \\
 & & & \mathbf{0} & & & 0 & 0 & 1 \\
 & & & & & & 0 & \frac{1}{2} & 0 \\
 & & & & & & \frac{1}{2} & 0 & 0 \\
 & & & & & & 0 & 0 & 0 \\
\hline
C_{15} & C_{25} & C_{35} & 2C_{45} & 2C_{55} & 2C_{56} & & & \\
C_{14} & C_{24} & C_{34} & 2C_{44} & 2C_{45} & 2C_{46} & & \mathbf{0} & \\
C_{13} & C_{23} & C_{33} & 2C_{34} & 2C_{35} & 2C_{36} & & &
\end{array}\right).
\end{aligned} \tag{2.7}$$

In this paper, the three-dimensional system described by Equation 2.4 is solved numerically with the DG method. The spatial domain is partitioned into tetrahedral elements, D^e, and the solution, $\boldsymbol{q}$, is discretized, $\boldsymbol{q}_h$, over the mesh. The grid size, h, is defined as the average radius of each tetrahedral's inscribed sphere. Over the entire domain, the solution of each element is required to be in L_2 and since this is a Galerkin method, both the trial functions, $\boldsymbol{q}_h$, and the test functions, $\boldsymbol{p}$, are also in L_2. The inner product of the trial and test functions is defined as,

$$\langle \boldsymbol{q}_h\,,\,\boldsymbol{p}_h\rangle = \int_{D^e} \boldsymbol{q}_h \boldsymbol{p}_h \mathrm{d}V. \tag{2.8}$$

Applying this inner product to the discrete form of Equation yields 2.4,

$$\int_{D^e} \frac{\partial \boldsymbol{q}_h}{\partial t}\boldsymbol{p}_h \mathrm{d}V + \int_{D^e} \nabla\cdot(\tilde{A}\boldsymbol{q}_h)\boldsymbol{p}_h\mathrm{d}V = \int_{D^e} \boldsymbol{g}_h\boldsymbol{p}_h\mathrm{d}V. \tag{2.9}$$

Then, integrating the above equation by parts, the weak formula of wave equation is obtained,

$$\int_{D^e} \frac{\partial \boldsymbol{q}_h}{\partial t}\boldsymbol{p}_h \mathrm{d}V - \int_{D^e} (\tilde{A}\boldsymbol{q}_h)\cdot(\nabla\boldsymbol{p}_h)\mathrm{d}V + \int_{\partial D^e} \tilde{F}(\boldsymbol{q}_h)\boldsymbol{p}_h\mathrm{d}S = \int_{D^e} \boldsymbol{g}_h\boldsymbol{p}_h\mathrm{d}V, \tag{2.10}$$

where $\tilde{F}(\boldsymbol{q}_h)$ is the numerical flux that couples each element to its neighbor. Without this coupling, the solution over each element would be completely local. Therefore, it cannot be zero (as in FEM) and will be discussed in the next section. We note that because of this, each element may be treated completely individually, decoupling all volume integrations. For this reason, the global mass matrix is block diagonal.

At this point, a classical FE or even many DG implementations stop at the weak formulation, however in this study, we found it to be more convenient to write the final equations in the strong form [13]. Equation 2.10 is integrated by parts again,

$$\int_{D^e} \frac{\partial \boldsymbol{q}_h}{\partial t}\boldsymbol{p}_h \mathrm{d}V + \int_{D^e} \nabla\cdot(\tilde{A}\boldsymbol{q}_h)\boldsymbol{p}_h\mathrm{d}V - \int_{\partial D^e} \left(\boldsymbol{n}\cdot(\tilde{A}\boldsymbol{q}_h) - \tilde{F}(\boldsymbol{q}_h)\right)\boldsymbol{p}_h\mathrm{d}S = \int_{D^e} \boldsymbol{g}_h\boldsymbol{p}_h\mathrm{d}V, \tag{2.11}$$

Lastly, to define the matrices required to deploy this DG implementation, the solution is expanded into N_p nodal trial basis functions,

$$\boldsymbol{q}_h(\boldsymbol{x},t) = \sum_{n=1}^{N_p} \hat{\boldsymbol{q}}_h(\boldsymbol{x}_n,t)\phi_n(\boldsymbol{x}), \boldsymbol{x}\in\mathbb{R}^3, \tag{2.12}$$

In general, the material properties are allowed to vary within each element. To write this explicitly, A^x, A^y and A^z are also expanded into the same basis, i.e.

$$A^m(\boldsymbol{x},t) = \sum_{n=1}^{N_p} \hat{A}^m(\boldsymbol{x}_n)\phi_n(\boldsymbol{x}),\ m = x,y,z. \tag{2.13}$$

In Equations 2.12 and 2.13, $\boldsymbol{x}_n$ are the nodal points. The Jacobi polynomials $\{J_n^{\alpha,\beta}(x)\}$ were used as the basis functions, where the nodal points were constructed by minimizing the Vandermonde matrix [13]. The basis functions are constructed by taking the product of three Jacobi polynomials, i.e.,

$$\phi_n(\xi,\eta,\zeta) = \sqrt{8}J_i^{0,0}(\xi)J_j^{2i+1,0}(\eta)(1-\eta)^i J_k^{2i+2j+2,0}(\zeta)(1-\zeta)^{i+j}, \tag{2.14}$$

where $0 \le i+j+k \le N$ and (ξ,η,ζ) are the coordinates of the reference tetrahedral. This will construct an Nth order solution over each element, which will require $N_p = \frac{1}{6}(N+1)(N+2)(N+3)$

degrees of freedom. The references coordinates are defined in terms of the standard tetrahedron, $\boldsymbol{I}$, defined by [13],

$$\boldsymbol{I} = \{(r,s,t)|(r,s,t) \geq -1; r+s+t \leq -1\}. \tag{2.15}$$

So [13],

$$\xi = -2\frac{1+r}{s+t} - 1,\ \eta = 2\frac{1+s}{1-t} - 1,\ \zeta = t. \tag{2.16}$$

By substituting Equations 2.12 and 2.13, $\boldsymbol{x}_n$ into Equation 2.11, along with assuming $\boldsymbol{p}_h$ is equal to $\phi_i(\boldsymbol{x})$ (since this is a Galerkin approximation), the first term is written,

$$\int_{D^e} \frac{\partial \boldsymbol{q}_h}{\partial t}\boldsymbol{p}_h \mathrm{d}V = \sum_{i,j=1}^{N_p} \frac{\partial}{\partial t}(\hat{\boldsymbol{q}}_h)_j \int_{D^e} \phi_j(\boldsymbol{x})\phi_i(\boldsymbol{x})\mathrm{d}V, \tag{2.17}$$

and the second term yields, after algebraic manipulation,

$$\begin{aligned}
&\int_{D^e} \nabla \cdot (\tilde{A}\boldsymbol{q}_h)\boldsymbol{p}_h \mathrm{d}V \\
&= \sum_{m=x,y,z}\sum_{i,j,k=1}^{N_p} (\hat{A}^m)_k(\hat{\boldsymbol{q}}_h)_j \int_{D^e} \left(\partial_m\phi_j(\boldsymbol{x})\phi_k(\boldsymbol{x}) + \phi_j(\boldsymbol{x})\partial_m\phi_k(\boldsymbol{x})\right)\phi_i(\boldsymbol{x})\mathrm{d}V \\
&= \sum_{i,j=1}^{N_p}(\hat{\boldsymbol{q}}_h)_j \sum_{k=1}^{N_p}\sum_{m=x,y,z}(\hat{A}^m)_k \int_{D^e} \left(\partial_m\phi_j(\boldsymbol{x})\phi_k(\boldsymbol{x}) + \phi_j(\boldsymbol{x})\partial_m\phi_k(\boldsymbol{x})\right)\phi_i(\boldsymbol{x})\mathrm{d}V.
\end{aligned} \tag{2.18}$$

Now, the final semi-discrete form of the elastic wave equation can be written,

$$\frac{\partial}{\partial t}\mathcal{M}\hat{\boldsymbol{q}}_h + \mathcal{S}\hat{\boldsymbol{q}}_h - \mathcal{F}\hat{\boldsymbol{q}}_h = \mathcal{G}, \tag{2.19}$$

The *local* mass matrix , $\mathcal{M}_{ij}^{D_e}$, and stiffness matrix, $\mathcal{S}_{ij}^{D_e}$ are defined as,

$$\begin{aligned}
\mathcal{M}_{ij}^{D_e} &= \int_{D^e} \phi_i(\boldsymbol{x})\phi_j(\boldsymbol{x})\mathrm{d}V, \\
\mathcal{S}_{ij}^{D_e} &= \sum_{k=1}\sum_{m=x,y,z}(\hat{A}^m)_k \int_{D^e} \left(\partial_m\phi_j(\boldsymbol{x})\phi_k(\boldsymbol{x}) + \phi_j(\boldsymbol{x})\partial_m\phi_k(\boldsymbol{x})\right)\phi_i(\boldsymbol{x})\mathrm{d}V,
\end{aligned} \tag{2.20}$$

where $\mathcal{F}$ is the numerical flux and the source term is defined by,

$$\mathcal{G} = \int_{D^e} (g)\phi(\boldsymbol{x})_i \tag{2.21}$$

In the next section, the numerical flux is defined to complete DG method for the anisotropic elastic wave equation.

3. Numerical Flux. The selection of numerical flux approach is of utmost importance when formulating a DG method. The numerical flux must accurately couple the localized elements, while remaining stable. We begin by defining some notation: the normal jump $[\![\boldsymbol{q}]\!] = \boldsymbol{n}(\boldsymbol{q}^- - \boldsymbol{q}^+)$, the total jump $[\boldsymbol{q}] = \boldsymbol{q}^- - \boldsymbol{q}^+$ and the average $\{\{q\}\} = \frac{1}{2}(\boldsymbol{q}^- + \boldsymbol{q}^+)$ of the solution, $\boldsymbol{q}$. The superscripts "$-$" and "$+$" indicate the solution inside the element in question (along the element boundary) and the solution inside its neighboring element, respectively. The most common flux used is the central flux, which is merely the average of the solution of both sides along the normal direction of the

Flux type	Penalty term $\mathcal{P}_F(\boldsymbol{q}^-, \boldsymbol{q}^+)$
Central	0
Upwind	$\frac{1}{2}[\![\boldsymbol{n}\cdot(\tilde{A}\boldsymbol{q})]\!]$
Lax-Friedrich	$\frac{1}{2}\lvert\alpha\rvert[\boldsymbol{q}]$
Hybrid	$\frac{1}{2}\lvert\alpha\rvert\left([\boldsymbol{q}] - (\boldsymbol{n}\cdot[\boldsymbol{q}])\boldsymbol{n}\right)$

TABLE 3.1
Several different numerical fluxes

interface, i.e., $\tilde{F}^{\text{central}}(\boldsymbol{q}) = \{\{\boldsymbol{n}\cdot(\tilde{A}\boldsymbol{q})\}\}$. Many other types of fluxes can be regarded as the central flux plus a penalty term (noted by $\mathcal{P}_F$). In general, numerical flux is expressed as [13],

$$\tilde{F}(\boldsymbol{q}) = \{\{\boldsymbol{n}\cdot(\tilde{A}\boldsymbol{q})\}\} + \mathcal{P}_F(\boldsymbol{q}^-, \boldsymbol{q}^+), \tag{3.1}$$

where the choice of $\mathcal{P}_F(\cdot,\cdot)$ defines the type of numerical flux (Table 3.1). If, for example, we chose $\mathcal{P}_F(\boldsymbol{q}^-, \boldsymbol{q}^+) = \frac{1}{2}[\![\boldsymbol{n}\cdot(\tilde{A}\boldsymbol{q})]\!]$, the upwinding flux is obtained. Upwinding fluxes are usually less dissipative than a central flux, however they require the system to be diagonalizable. For the problem at hand, the system is of the non-principal type because the S-waves, in general, cannot be decoupled. Because of this, a central-flux is used in our implementation. Often times, a central flux is applied by setting $\mathcal{P}_F(\boldsymbol{q}^-, \boldsymbol{q}^+)$ to zero, however doing so leads to instabilities. The Lax-Friedrich flux, also called the Rusanov flux [7], is given by $\mathcal{P}_F(\boldsymbol{q}^-, \boldsymbol{q}^+) = \frac{1}{2}\lvert\alpha\rvert[\boldsymbol{q}]$ where α is the largest eigenvalue, i.e. the (quasi-)P-wave velocity of the elastic system.

In our formulation, a hybrid Lax-Friedrich flux is used, whose penalty term is given by

$$\mathcal{P}_F(\boldsymbol{q}^-, \boldsymbol{q}^+) = \frac{1}{2}\lvert\alpha\rvert\left(\boldsymbol{n}\times\boldsymbol{n}\times[\boldsymbol{q}]\right) = \frac{1}{2}\lvert\alpha\rvert\left([\boldsymbol{q}] - (\boldsymbol{n}\cdot[\boldsymbol{q}])\boldsymbol{n}\right), \tag{3.2}$$

This produces a central flux along the normal direction of the interface and a Lax-Fridrichs flux along the tangential direction. More explicitly, Equation 3.2 is expressed in terms of the strain and velocity as,

$$\mathcal{P}_F\left(\begin{pmatrix}\boldsymbol{E}\\ \boldsymbol{v}\end{pmatrix}^-, \begin{pmatrix}\boldsymbol{E}\\ \boldsymbol{v}\end{pmatrix}^+\right) = \frac{1}{2}\lvert\alpha\rvert\begin{pmatrix}\boldsymbol{E}^- - \boldsymbol{E}^+ - ([\boldsymbol{E}]\boldsymbol{n})\otimes\boldsymbol{n}\\ \boldsymbol{v}^- - \boldsymbol{v}^+ - ([\boldsymbol{v}]\cdot\boldsymbol{n})\boldsymbol{n}\end{pmatrix}, \tag{3.3}$$

where $\otimes$ is the vector outer product.

The boundary conditions of an elastic system give different restrictions along the normal and tangential directions. For the normal direction, both velocity and stress are continuous across the interface, while along the tangential direction, only the velocity is required to be continuous. No restriction of continuity is required for the shear stress. The hybrid flux provides a stable flux scheme for the elasto-dynamic system by treating the normal and tangential boundary conditions separately. Substituting Equation 3.2 and 3.1 into Equation 2.11, and noting that $\boldsymbol{n}\cdot\tilde{A}\boldsymbol{q}_h - \{\{\boldsymbol{n}\cdot\tilde{A}\boldsymbol{q}_h\}\} = [\boldsymbol{n}\cdot\tilde{A}\boldsymbol{q}_h\,]$ we are left with a stable DG scheme for elasto-dynamic system, i.e.

$$\begin{aligned}\int_{D^e}\frac{\partial\boldsymbol{q}_h}{\partial t}\boldsymbol{p}_h\mathrm{d}V &+ \int_{D^e}\nabla\cdot(\tilde{A}\boldsymbol{q}_h)\boldsymbol{p}_h\mathrm{d}V\\ &- \int_{\partial D^e}\frac{1}{2}\Big([\boldsymbol{n}\cdot(\tilde{A}\boldsymbol{q}_h)] - \lvert\alpha\rvert\left([\boldsymbol{q}_h\,] - (\boldsymbol{n}\cdot[\boldsymbol{q}_h\,])\boldsymbol{n}\right)\Big)\boldsymbol{p}_h\mathrm{d}S = \int_{D^e}\boldsymbol{g}_h\boldsymbol{p}_h\mathrm{d}V.\end{aligned} \tag{3.4}$$

Next, a projection matrix T is defined, such that $T\boldsymbol{q}_h = (\boldsymbol{n} \cdot \boldsymbol{q}_h)\boldsymbol{n}$,

$$T = \left(\begin{array}{cccccc|ccc} n_x^2 & 0 & 0 & 0 & n_xn_z & n_xn_y & & & \\ 0 & n_y^2 & 0 & n_yn_z & 0 & n_xn_y & & & \\ 0 & 0 & n_z^2 & n_yn_z & n_xn_z & 0 & & & \\ 0 & \frac{n_yn_z}{2} & \frac{n_yn_z}{2} & \frac{n_y^2+n_z^2}{2} & \frac{n_xn_y}{2} & \frac{n_xn_z}{2} & & \mathbf{0} & \\ \frac{n_xn_z}{2} & 0 & \frac{n_xn_z}{2} & \frac{n_xn_y}{2} & \frac{n_x^2+n_z^2}{2} & \frac{n_yn_z}{2} & & & \\ \frac{n_xn_y}{2} & \frac{n_xn_y}{2} & 0 & \frac{n_xn_z}{2} & \frac{n_yn_z}{2} & \frac{n_x^2+n_y^2}{2} & & & \\ \hline & & & & & & n_x^2 & n_xn_y & n_xn_z \\ & & & \mathbf{0} & & & n_xn_y & n_y^2 & n_yn_z \\ & & & & & & n_xn_z & n_yn_z & n_z^2 \end{array}\right) \tag{3.5}$$

where (n_x, n_y, n_z) is the unit outer normal vector of ∂D^e. By expanding $\boldsymbol{q}_h$ and $\tilde{A}$ (Equation 2.12 and 2.13) in the flux term of Equation 3.4 yields,

$$\begin{aligned} &\int_{\partial D^e} \frac{1}{2}\Big([\boldsymbol{n}\cdot(\tilde{A}\boldsymbol{q}_h)] - |\alpha|(\mathcal{I}_9 - T)[\boldsymbol{q}_h]\Big)\boldsymbol{p}_h \mathrm{d}S \\ &\qquad = \sum_{m=x,y,z}\sum_{i,j,k=1}^{N_p}(\hat{A}^m)_k(\hat{\boldsymbol{q}}_h)_j\int_{\partial D^e} n_m(\boldsymbol{x})\phi_j(\boldsymbol{x})\phi_k(\boldsymbol{x})\phi_i(\boldsymbol{x})\mathrm{d}S \\ &\qquad = \sum_{i,j=1}^{N_p}(\hat{\boldsymbol{q}}_h)_j\sum_{k=1}^{N_p}\sum_{m=x,y,z}(\hat{A}^m)_k\int_{\partial D^e} n_m(\boldsymbol{x})\phi_j(\boldsymbol{x})\phi_k(\boldsymbol{x})\phi_i(\boldsymbol{x})\mathrm{d}S. \end{aligned} \tag{3.6}$$

where $\mathcal{I}_9$ is the 9×9 identity matrix. Now, the flux matrix can be defined explicitly,

$$\mathcal{F}_{ij}^{D_e} = \sum_{k=1}^{N_p}\sum_{m=x,y,z}(\hat{A}^m)_k\int_{\partial D^e} n_m(\boldsymbol{x})\phi_j(\boldsymbol{x})\phi_k(\boldsymbol{x})\phi_i(\boldsymbol{x})\mathrm{d}S. \tag{3.7}$$

Substituting Equation 2.20 and 3.7 into 3.4, and defining $\mathcal{G}^{D^e} = \int_{D^e}\boldsymbol{g}_h\boldsymbol{p}_h\mathrm{d}V$ as the forcing term, the semi-discretized Discontinuous Galerkin scheme is obtained:

$$\frac{\partial}{\partial t}\mathcal{M}^{D_e}\hat{\boldsymbol{q}}_h + \mathcal{S}^{D_e}\hat{\boldsymbol{q}}_h - \mathcal{F}^{D_e}\hat{\boldsymbol{q}}_h = \mathcal{G}^{D_e}. \tag{3.8}$$

Now that the numerical flux has been derived for all of the internal boundaries between elements, the external boundary conditions will be considered.

4. External Boundary Conditions and PML.

4. External Boundary Conditions and PML. Another important aspect of DG implementation is the external boundary conditions for the computational domain. The free-surface boundary condition and the absorbing boundary condition are among the most frequently used boundary conditions in seismology. For free-surface conditions, waves are completely reflected from the boundary. This type of boundary condition is often used for rock surfaces, slits, the surface of the Earth and fault facets. Absorbing boundary conditions are used to compute the solution of an infinite domain. The computational domain was encased in a perfect matching layer (PML) to enhance the effect of the absorbing boundary condition. PML's work by modifying the PDE in PML region such that the waves are mapped to the complex plane and decay exponentially in the real plane.

4.1. Free-surface boundary condition. In the DG framework, we cannot directly impose Dirichlet boundary conditions as is done for continuous FEM. We only have control over the flux conditions of each element. Therefore, we impose Dirichlet conditions by applying ghost fluxes on the boundary of the domain that replicate the desired values. Free-surface boundary conditions require that $(\boldsymbol{CE})\boldsymbol{n} = 0$ on $\partial\Omega$, i.e., the external normal stress at the boundary reduces to zero. There is, however, no direct constraint over the velocity or the tangential stress. By placing an equal and opposite traction on the boundary at all times, i.e. $\boldsymbol{S}^- = -\boldsymbol{S}^+$, the free-surface boundary conditions are weakly imposed. To implement these conditions, the jump across any external boundary is redefined as,

$$\boldsymbol{n} \cdot (\boldsymbol{v}^- - \boldsymbol{v}^+) = 0, \tag{4.1}$$

$$(\boldsymbol{CE}^- - \boldsymbol{CE}^+)\boldsymbol{n} = 2(\boldsymbol{CE}^-)\boldsymbol{n}. \tag{4.2}$$

This is accomplished by modifying the corresponding term in $\mathcal{F}\hat{\boldsymbol{q}}_h$. A convergence test and numerical application of this implementation will be shown in Section 7.2.

4.2. Absorbing boundary condition. To simulate an infinite computation domain, the jump across can be redefined to absorb the wave. The absorbing boundary is applied with the DG framework by setting $\boldsymbol{v}^+$ and $\boldsymbol{E}^+$ to zero and in effect acting as if there is no incoming wave, thereby allowing the wave to exit the domain. However, absorbing boundary conditions for two or greater dimensions are notoriously not well suited when waves graze the boundary. For this reason, we also provide perfectly matched boundary layers to reduce the wave amplitude while minimizing artificial reflections.

4.3. Convolutional perfect matching layer. As mentioned earlier, the absorbing boundary flux conditions reduce the reflections of waves that exit the domain perpendicular to the boundary. However, as the angle between the wave front and the boundary decreases, the approximation begins to fail. A convolutional perfectly matching layer (CPML) was used to simulate an infinite computational domain [17] to mitigate this effect. CPML's absorb any incoming waves by adding a damping function, $d(\boldsymbol{x})$, to the wave equation in the frequency-domain form, i.e.

$$-\mathrm{i}\omega\tilde{\boldsymbol{q}} + \nabla(\tilde{A}\tilde{\boldsymbol{q}}) = \tilde{\boldsymbol{f}} \tag{4.3}$$

becomes,

$$-\mathrm{i}\omega\hat{\boldsymbol{q}} + \nabla(\tilde{A}\hat{\boldsymbol{q}}) + d(\boldsymbol{x})\hat{\boldsymbol{q}} = \hat{\boldsymbol{f}}, \tag{4.4}$$

where $\tilde{\boldsymbol{q}}$ and $\tilde{A}$ is the solution to the wave equation (in frequency space) and the Jacobi matrices, respectively. Note that the solution changed from $\tilde{\boldsymbol{q}}$ to $\hat{\boldsymbol{q}}$ denoting the equation with and without CPML's. The advantage of adding the damping function in this fashion is that the solution to Equation 4.4 exponentially decays while the CPML is present, i.e.

$$\hat{\boldsymbol{q}} = \tilde{\boldsymbol{q}}\, \mathrm{e}^{-\int d(\boldsymbol{x})dx} \tag{4.5}$$

The form of $d(\boldsymbol{x})$ is restricted in such a way that it equals zero within the computational domain and at the boundary, and increases inside the CPML. In this study, $d(\boldsymbol{r}) = -\log\gamma\ (1 - \cos\frac{2\pi\boldsymbol{r}}{h})$ was chosen, where h is the thickness of the layer, $\boldsymbol{r}$ is the distance from the interface between the CPML and the computational domain, and γ is called the reflection coefficient. In our study, setting $\gamma = 1/1000$ yields satisfactory results.

To implement the CPML, Equation 4.4 is multiplied by $-\mathrm{i}\omega$ and divided by $-\mathrm{i}\omega + d$, where d's dependence on $\boldsymbol{x}$ is now assumed. A function, $\alpha(\boldsymbol{x})$ is added to the numerator and denominator of the coefficient of the spatial derivative terms to force stability [17],

$$-\mathrm{i}\omega\hat{\boldsymbol{q}} + \frac{\mathrm{i}\omega + \alpha(\boldsymbol{x})}{-\mathrm{i}\omega + d + \alpha(\boldsymbol{x})}\nabla(\tilde{A}\hat{\boldsymbol{q}}) = \frac{\mathrm{i}\omega}{-\mathrm{i}\omega + d}\,\hat{\boldsymbol{f}} \tag{4.6}$$

In the authors experience, setting $\alpha(\boldsymbol{x})$ to unity gave satisfactory results. The right hand side will be zero because we assume that there are no source terms in the CPML. By Fourier transforming to the time domain,

$$\partial_t\boldsymbol{q} + \nabla(\tilde{A}\boldsymbol{q}) - \int_0^t d(\boldsymbol{x})H(t-\tau)\mathrm{e}^{-(d+\alpha)(t-\tau)}\nabla(\tilde{A}\boldsymbol{q})d\tau = 0 \tag{4.7}$$

is written. $H(t)$ is the Heaviside function and it is always unity because the time integration is from 0 to t. The integral in the third term is defined as $\Upsilon(t, \boldsymbol{x})$. Next, the integral in $\Upsilon(t, \boldsymbol{x})$ is removed by taking a partial derivative with respect to t. Finally, a coupled PDE between the solution to the wave equation, $\boldsymbol{q}$, and the new CPML variable is, Υ, is written:

$$\begin{aligned} \partial_t \boldsymbol{q} &= -\nabla(\tilde{A}\boldsymbol{q}) - \Upsilon \\ \partial_t \Upsilon &= -(d+\alpha)\Upsilon + d\,\nabla(\tilde{A}\boldsymbol{q}) \end{aligned} \tag{4.8}$$

The last step in the derivation of a DG method for the anisotropic wave equation, since it has been written in a semi-discrete form that allows for CPML regions, is to consider the discretization of the time derivative.

5. Time Discretization. In this section, we discuss a time discretization that is computationally efficient for the complex domains used in this study. Often, the computational meshes used to model the subsurface must contain regions where the characteristic lengths of the elements drop far below that of a wavelength because the subsurface contains very complex geometries and discontinuities. As a result, many simulations are not possible because the time steps must be equally reduced to produce a stable solution. We propose two different time discretization schemes: (1) for non-complex domains, it is advantageous to use a traditional Runga-Kutta (RK) method and (2) for complex domains, a semi implicit-explicit (IMEX) method is used. The IMEX method enables the solver to perform implicit time integration in areas of over sampling, while keeping the computational efficiency of RK in regions of proper sampling.

5.1. Explicit Runge-Kutta. After discretizing the spatial derivatives of the elastic wave equation using the DG method, the temporal derivative is integrated. An explicit time integration method is used when the variation in element size is small. There are a variety of time-stepping methods available, however we chose to use the five stage low-storage explicit Runge-Kutta (LSERK) method [13]. LSERK is an explicit method whose time-step is dictated by the Courant-Friedrichs-Lewy (CFL) condition and allows for the largest possible stable time step. The time step was chosen so that,

$$c_p \frac{\Delta t}{\Delta x} = \frac{K_{CFL}}{p^2}, \tag{5.1}$$

where c_p is the maximum wave-speed, i.e., the (quasi-)P wave velocity, and K_{CFL} is the CFL constant. Recent efforts to define, quantitatively, a stable CFL condition can be found in [6, 3]. However in this study, Equation 5.1 is sufficient.

The LSERK method is preferred over other methods because it saves memory at the cost of computation time. As the models grow larger, memory becomes the bottleneck rather than computation time. Rewriting Equation 3.8 with all of the spatial derivatives on the right hand side gives,

$$\frac{d\hat{\boldsymbol{q}}_h}{dt} = -(\mathcal{M})^{-1}\left((-\mathcal{S}+\mathcal{F})\hat{\boldsymbol{q}}_h + \mathcal{G}\right) = \mathcal{R}_h(\hat{\boldsymbol{q}}_h, t), \tag{5.2}$$

the LSERK algorithm is as follows:

$$\begin{aligned} &\mathbf{u}^{(0)} = \hat{\boldsymbol{q}}_h^n, \\ &i \in [1, \dots, 5] : \begin{cases} \mathbf{k}^{(i)} = a_i \mathbf{k}^{(i-1)} + \Delta t \mathcal{R}_h\left(\mathbf{u}^{(i-1)}, t^n + c_i \Delta t\right), \\ \mathbf{u}^{(i)} = \mathbf{u}^{(i-1)} = b_i \mathbf{k}^{(i)}, \end{cases} \\ &\hat{\boldsymbol{q}}_h^{n+1} = \mathbf{u}^{(5)}. \end{aligned}$$

where the constants a_i and b_i are given in [13] on page 64.

5.2. Explicit-Implicit Runge-Kutta. When the domain in question contains complex geometries within large domains, such as rough surfaces or discontinuities in the subsurface, the resulting mesh will contain regions of over sampling with respect to the desired wavelengths. This hinders the use of an ERK method because the stability depends heavily on the size of the time step, which in turn is dependent on the region of highest sampling. Here we present the IMEX method [2, 24] which allows the regions of over sampling to be integrated in time with an L-stable third order and 3-stage Diagonally Implicit Runge-Kutta (DIRK) method, while using a fast and simple 4-stage third order ERK method in the regions of more reasonable sampling (8-10 nodes per wavelength).

To begin, the right hand side of Equation 5.2 is split into two vectors.

$$\frac{d\hat{\boldsymbol{q}}_h}{dt} = \mathcal{R}_h(\hat{\boldsymbol{q}}_h,t) = \begin{pmatrix} 0 \\ \hat{\mathcal{R}}_h^{exp}(\hat{\boldsymbol{q}}_h,t) \end{pmatrix} + \begin{pmatrix} \hat{\mathcal{R}}_h^{imp}(\hat{\boldsymbol{q}}_h,t) \\ 0 \end{pmatrix} \tag{5.3}$$

This allows the two regions to interact on the boundaries alone, allowing the time steps to be set, ideally, by the correctly sampled region. The beauty of this method is that the DIRK and ERK methods where created so that each of the stages is at the same point in time. This means that the system can be solved without requiring an interpolation at the boundary of the implicit-explicit regions. The IMEX method's Butcher table for the 4-stage explicit method is,

$$\begin{array}{c|c} c & a_{i,j} \\ \hline & b_i \end{array} = \begin{array}{c|ccc} 0 & & & \\ 0.4358665215 & 0.4358665215 & & \\ 0.7179332608 & 0.2820667392 & 0.4358665215 & \\ 1.0 & 1.208496649 & -0.644363171 & 0.4358665215 \\ \hline & 1.208496649 & -0.644363171 & 0.4358665215 \end{array} \tag{5.4}$$

and for the 3-stage implicit method,

$$\begin{array}{c|c} \hat{c} & \hat{a}_{i,j} \\ \hline & \hat{b}_i \end{array} = \begin{array}{c|cccc} 0 & & & & \\ 0.4358665215 & 0.4358665215 & & & \\ 0.7179332608 & 0.3212788860 & 0.3966543747 & & \\ 1.0 & 0.3212788860 & 0.5529291479 & 0.5529291479 & \\ \hline & 0 & 1.208496649 & -0.644363171 & 0.4358665215 \end{array} \tag{5.5}$$

With the RK constants in place, the over all IMEX integration from t_{i-1} to t_i is simple. Note that stages denoted with a hat, i.e. $\hat{k}_i$, correspond to stages in the explicit method while those without correspond to stages in the implicit method. Initially, the first stage of the explicit method is computed,

$$\hat{k}_1 = \hat{\mathcal{R}}_h^{exp}(\hat{\boldsymbol{q}}_h^{(i-1)},t) \tag{5.6}$$

then, for $i = 1, 2, 3$, all of the implicit stages and the rest of the explicit stages can be solved for,

$$k_i = \hat{\mathcal{R}}_h^{imp}(\hat{\boldsymbol{q}}_h^{(i)},t) \quad \text{where,} \quad \mathbf{q}_n^i = u_n^{i-1} + \Delta t \sum_{j=1}^{i} a_{i,j} k_j + \Delta t \sum_{j=1}^{i} \hat{a}_{i,j} \hat{k}_j \tag{5.7}$$

$$\hat{k}_{i+1} = \hat{\mathcal{R}}_h^{exp}(\hat{\boldsymbol{q}}_h^{(i)},t). \tag{5.8}$$

At this point, the solution at t_{i+1} is,

$$\hat{\boldsymbol{q}}_h^i = \hat{\boldsymbol{q}}_h^i + \Delta t \sum_{i=1}^{3} b_i k_i + \sum_{i=1}^{4} \hat{b}_i \hat{k}_i \tag{5.9}$$

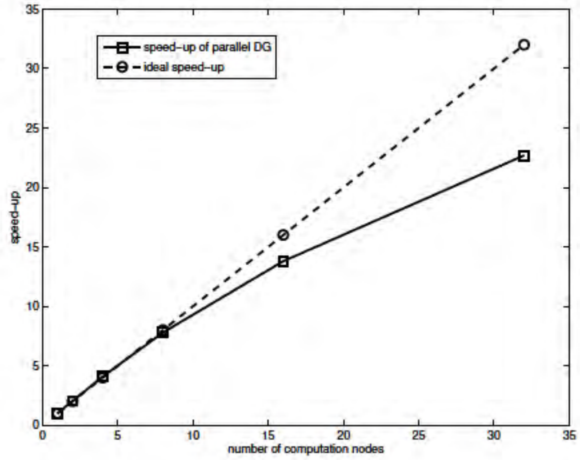

FIG. 1. *Speed up of parallelized DG implementation on the Coates cluster at Purdue University, ranging from 2 to 32 nodes with 8 cores per node.*

6. Parallelization. The DG framework is well suited for parallelization because the volume integrations are localized over each element. The only communication that is required is the numerical fluxes between adjacent processor nodes. Because of this, the DG method can be treated as an embarrassingly parallel code, like the FDTD method, after a domain decomposition is performed. A given connected computational domain is divided into an arbitrary number of subdomains using an external library, e.g. METIS [16]. METIS uses a partitioning scheme that is based on graph partitioning and is optimized to reduce the amount of communication between processor nodes. Once the domain is decomposed each computational node calculates its own subdomain from structured arrays that store the local information of the computational domain. Elements that are not in, but connected to, the current domain are labeled "ghost elements", and stored in a separate structure array. The elements that belong to a computational node's domain are called "host elements."

The method was constructed by using non-blocking communications via the message passing interface (MPI) library. MPI provides subroutines to communicate efficiently between computational nodes in a point-to-point fashion. Inside each subdomain, a derived data type is defined to send the fluxes from the array of host elements to the nodes neighbors, and a second derived data type to receive the fluxes from the array of ghost elements, using the MPI subroutine MPI_TYPE_INDEXED. During each time step, the fluxes that belong to the interface between computational partitions (ghost elements) are updated over the network using non-blocking communications. In this way, the solution for all of interior elements is computed and the data are sent over the network concurrently. Once this computation is complete, the algorithm waits for the communication of the ghost elements to complete before updating them. This algorithm ensures that all computations are based on the updated solution state.

We conducted a performance test of our parallel code on the Coates computer cluster at Purdue University, by using a maximum of 32 nodes. Each node consists of a dual 2.5 GHz Quad-Core AMD 2380 CPU with 32 GB memory. A cubic model was used with constant density and wave speed, containing 1,310,720 tetrahedral elements and using 3^{rd} order polynomials. The code was run for 1,000 time steps. The increase in speed is shown in Figure 1.

7. Convergence Tests.

7.1. Plane wave. The homogeneous isotropic elastic wave equation for a plane wave was the first convergence test considered. The material properties of the domain were set such that the P-wave velocity, c_p, was equal to 2 m/s and S-wave velocity, c_s, was equal to 1 m/s throughout the computational domain. The analytic solution is derived by assuming that the P and S waves are

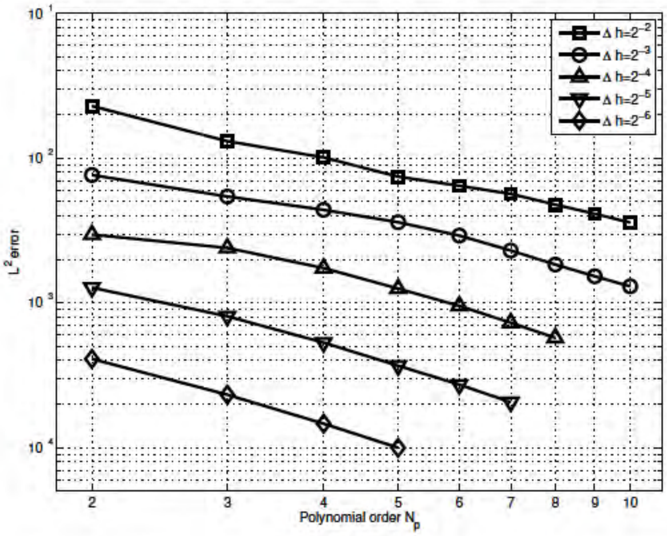

FIG. 2. *L_2 error of the plane wave solution as a function of polynomial order.*

propagating along an arbitrary direction, $\boldsymbol{r}$. Then, the potential of the plane wave is given by

$$\begin{aligned} \Phi(\boldsymbol{x},t) =& A\mathrm{e}^{\mathrm{i}\omega(c_p t-\boldsymbol{r}\cdot\boldsymbol{x})} \\ \Psi(\boldsymbol{x},t) =& B\mathrm{e}^{\mathrm{i}\omega(c_s t-\boldsymbol{r}\cdot\boldsymbol{x})}, \end{aligned} \tag{7.1}$$

where the displacement is given by $\boldsymbol{u} = \nabla\Phi + \nabla\times\Psi$. For simplicity, we set both amplitudes, (A, B), to one. The analytic solution of the velocity field $\boldsymbol{v} = \frac{\partial}{\partial t}\boldsymbol{u}$ is,

$$\boldsymbol{v} = c_p\boldsymbol{r}\sin(\boldsymbol{x}\cdot\boldsymbol{r} - c_p t) + c_s\boldsymbol{t}\sin(\boldsymbol{x}\cdot\boldsymbol{r} - c_s t), \tag{7.2}$$

where $\boldsymbol{t}$ is the direction of motion for the shear wave and is perpendicular to $\boldsymbol{r}$. The computational domain used for this study was a periodic cube, spanning $\Omega = [0,1]\times[0,1]\times[0,1]$. The L_2 error between the numerical solution obtained with DG method and the exact solution with element size and polynomial order over time is shown in Figure 2.

7.2. Free-surface: Rayleigh wave. A numerical test was also performed to test the accuracy on homogeneous isotropic elastic solid model with free surface boundary condition, which refers to Rayleigh and Love waves that both decay exponentially in the z direction. The analytic solution for a Rayleigh wave in a homogeneous isotropic half-space is written in terms of the complex amplitudes of the potentials [1],

$$\begin{aligned} \phi =& A\mathrm{e}^{\mathrm{i}krz}\mathrm{e}^{\mathrm{i}\omega(t-x/c)}, \\ \psi =& B\mathrm{e}^{\mathrm{i}ksz}\mathrm{e}^{\mathrm{i}\omega(t-x/c)}, \end{aligned} \tag{7.3}$$

where $k = \omega/c$ is the wave number of surface wave, $\nabla\phi = \mathbf{u}$, $\nabla\times\psi = \mathbf{u}$ ($\mathbf{u}$ is the displacement), and r and s are coefficients related to the P wave speed c_p and S wave speed c_s,

$$r = \sqrt{\frac{c^2}{c_p^2} - 1}, s = \sqrt{\frac{c^2}{c_s^2} - 1}. \tag{7.4}$$

Applying the free-surface boundary condition at the domain boundary, we obtain the equation for the relationship between amplitudes and Rayleigh wave speed,

$$\begin{aligned} \left(2-\frac{c^2}{c_s^2}\right)A-2\mathrm{i}\left(1-\frac{c^2}{c_s^2}\right)^{\frac{1}{2}}B &=0 \\ 2\mathrm{i}\left(1-\frac{c^2}{c_p^2}\right)^{\frac{1}{2}}A+\left(2-\frac{c^2}{c_s^2}\right)B &=0. \end{aligned} \tag{7.5}$$

For a nontrivial solution, c must satisfy the determinant of the coefficient matrix, which yields the Rayleigh equation

$$\left(2-\left(\frac{c}{c_s}\right)^2\right)^2=4\sqrt{1-\left(\frac{c}{c_p}\right)^2}\sqrt{1-\left(\frac{c}{c_s}\right)^2}. \tag{7.6}$$

Then system is solved as

$$B=\frac{2\mathrm{i}c_s^2}{2c_s^2-c^2}\sqrt{1-(\frac{c}{c_p})^2}A. \tag{7.7}$$

From the above, we derive the velocity-strain component as

$$\begin{aligned} v_x &=\Re\left(\frac{\omega^2}{c}\left[A\mathrm{e}^{\mathrm{i}krz}+sB\mathrm{e}^{\mathrm{i}ksz}\right]\mathrm{e}^{\mathrm{i}\omega(t-x/c)}\right), \\ v_z &=\Re\left(-\frac{\omega^2}{c}\left[rA\mathrm{e}^{\mathrm{i}krz}-B\mathrm{e}^{\mathrm{i}ksz}\right]\mathrm{e}^{\mathrm{i}\omega(t-x/c)}\right), \\ \varepsilon_{xx} &=\Re\left(-\mathrm{i}\frac{\omega^3}{c^2}\left[A\mathrm{e}^{\mathrm{i}krz}+sB\mathrm{e}^{\mathrm{i}ksz}\right]\mathrm{e}^{\mathrm{i}\omega(t-x/c)}\right), \\ \varepsilon_{zz} &=\Re\left(-\mathrm{i}\frac{\omega^3}{c^2}\left[r^2A\mathrm{e}^{\mathrm{i}krz}-sB\mathrm{e}^{\mathrm{i}ksz}\right]\mathrm{e}^{\mathrm{i}\omega(t-x/c)}\right), \\ \varepsilon_{xz} &=\Re\left(\mathrm{i}\frac{\omega^3}{c^2}\left[rA\mathrm{e}^{\mathrm{i}krz}+\frac{s^2-1}{2}B\mathrm{e}^{\mathrm{i}ksz}\right]\mathrm{e}^{\mathrm{i}\omega(t-x/c)}\right), \\ v_y &=0,\ \varepsilon_{xy}=0,\ \varepsilon_{yy}=0,\ \varepsilon_{yz}=0. \end{aligned} \tag{7.8}$$

To perform a convergence analysis, a computational domain was chosen to be $[-1,1]\times[-1,1]\times[0,10]$ meters with periodic boundary condition along x and y direction. The wave amplitude is negligible at $z=10$ so absorbing boundary conditions were used on the $z=10$ plane without hindering the solution. The material properties were chosen such that the P-wave velocity was 2 m/s and the S-wave velocity was 1 m/s with a density of 1 kg^3/m. The coefficients were set to $(A,B)=(1.0,1.5652)$. The p convergence rates are shown in Figure 3.

7.3. Elastic-Elastic Interface: Stoneley wave. The proposed DG method is used to compute the boundary wave that travels along an elastic-elastic interface, i.e. a Stoneley wave. Stoneley waves decay exponentially away from the interface as they propagate. This simulation demonstrates the ability of this DG method to compute, correctly, a domain with an internal elastic-elastic interface. The domain is split into two half-spaces, each with different material properties. The analytic solution is written in terms of the complex amplitudes of the potentials [1], where subscripts of 1 and 2 denote the two different materials.

$$\begin{aligned} \phi_1 &= A_1\mathrm{e}^{-kb_{p,1}z}\mathrm{e}^{\mathrm{i}kx},\ \text{for } z>0 \\ \psi_1 &= B_1\mathrm{e}^{-kb_{s,1}z}\mathrm{e}^{\mathrm{i}kx},\ \text{for } z>0 \\ \phi_2 &= A_2\mathrm{e}^{-kb_{p,2}z}\mathrm{e}^{\mathrm{i}kx},\ \text{for } z<0 \\ \psi_2 &= B_2\mathrm{e}^{-kb_{s,2}z}\mathrm{e}^{\mathrm{i}kx},\ \text{for } z<0 \end{aligned} \tag{7.9}$$

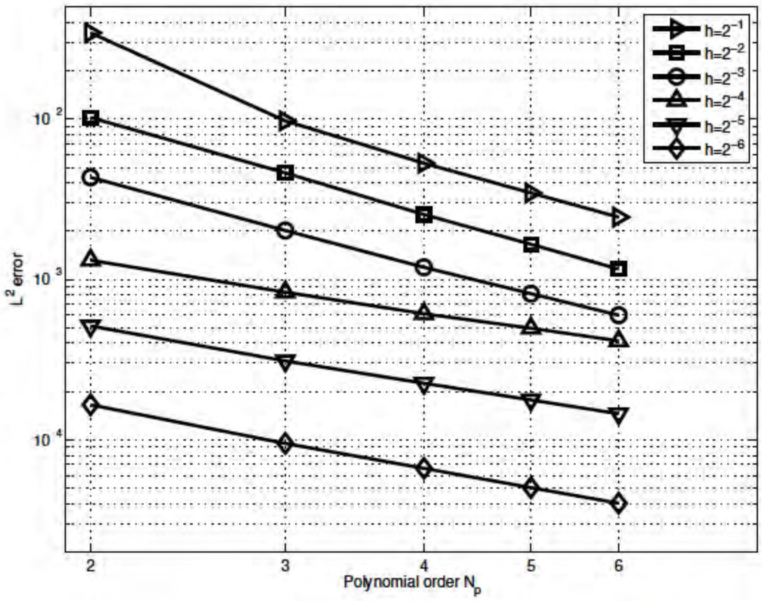

FIG. 3. *L_2 convergence of a Rayleigh wave as a function of polynomial order.*

where $\nabla \phi_i = \mathbf{u}_i$ and $\nabla \times \psi_i = \mathbf{u}_i$ and $\mathbf{u}_i$ is the displacement in medium i. The constants b_j are

$$
\begin{aligned}
b_{p,1} &= \left(1 - \frac{c^2}{c_{p,1}^2}\right)^{1/2} \\
b_{s,1} &= \left(1 - \frac{c^2}{c_{s,1}^2}\right)^{1/2} \\
b_{p,2} &= \left(1 - \frac{c^2}{c_{p,2}^2}\right)^{1/2} \\
b_{s,2} &= \left(1 - \frac{c^2}{c_{s,2}^2}\right)^{1/2}
\end{aligned}
\tag{7.10}
$$

where c is the propagation speed of the Stoneley wave and $c_{p,i}$ and $c_{s,i}$ are the P and S wave velocities of each medium, respectively. By applying the elastic-elastic boundary conditions at the interface, the wave amplitudes are found. To obtain a non-trivial solution, the following determinant must equal zero.

$$
\begin{vmatrix}
2\rho_1 c_{s,1}^2 b_{p,1} & 2\rho_2 c_{s,2}^2 b_{p,2} & \rho_1 c_{s,1}^2 \left(2 - \frac{c^2}{c_{s,1}^2}\right) & \rho_2 c_{s,2}^2 \left(2 - \frac{c^2}{c_{s,2}^2}\right) \\
\rho_1(c^2 - 2c_{s,1}^2) & -\rho_2(c^2 - 2c_{s,2}^2) & -2\rho_1 c_{s,1}^2 b_{s,1} & 2\rho_2 c_{s,2}^2 b_{s,2} \\
1 & \text{-1} & b_{s,1} & -b_{s,2} \\
b_{p,1} & b_{p,2} & 1 & 1
\end{vmatrix} = 0
\tag{7.11}
$$

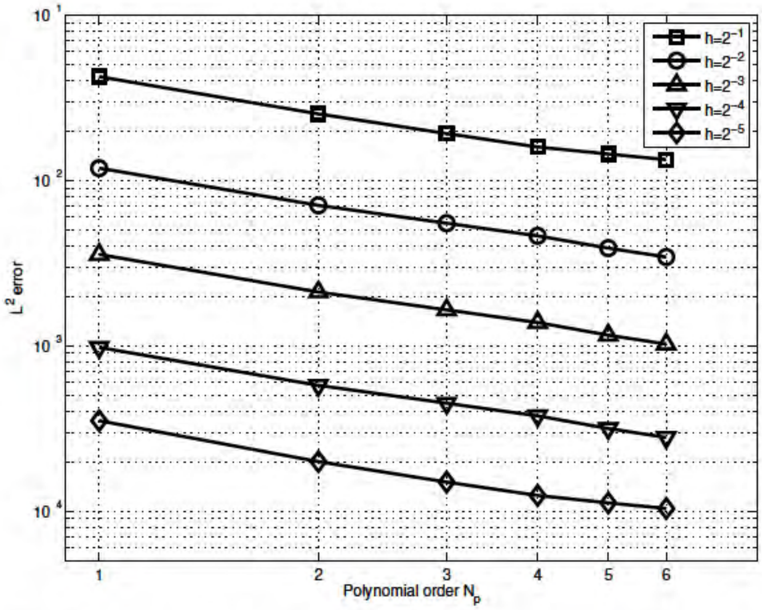

FIG. 4. *L^2 convergence of a Stoneley wave as a function of polynomial order.*

which simplifies to,

$$
\begin{aligned}
& r^4\left[\left(\frac{\rho_1}{\rho_2}-1\right)^2-\left(\frac{\rho_1}{\rho_2}b_{p,2}+b_{p,1}\right)\left(\frac{\rho_1}{\rho_2}b_{s,2}+b_{s,1}\right)\right] \\
& +4r^2\left(\frac{\rho_1}{\rho_2}\frac{c_{s,1}^2}{c_{s,2}^2}-1\right)\left(\frac{\rho_1}{\rho_2}b_{p,2}b_{s,2}-b_{p,1}b_{s,1}-\frac{\rho_1}{\rho_2}+1\right) \\
& +4\left(\frac{\rho_1}{\rho_2}\frac{c_{s,1}^2}{c_{s,2}^2}-1\right)^2(b_{p,1}b_{s,1}-1)(b_{p,2}b_{s,2}-1)=0
\end{aligned}
\tag{7.12}
$$

where $r=\frac{c}{c_{s,2}}$.

In our convergence study, a domain of [-1,-1] by [-1,-1] by [-1,-1] meters was used with periodicity in the x and y directions and free surface boundary conditions in the z direction. Because the waves decay exponentially in the z direction ,free surface boundary conditions are justified. The material properties were selected such that $\lambda_1=\mu_1=3, \rho_1=10$ and $\lambda_1=\mu_1=\rho_1=1.1$. Therefore, the constants above are defined as $A_1=-0.3113$, $B_1=-0.7577$, $A_2=0.5225$, $B_2=-0.9967$. The p convergence rates are shown in Figure 4.

8. Applications.

8.1. Laboratory Rough Surface. The potential of this DG method to model complex domains is demonstrated by computing the scattered wave-fields from rough fractures. A laser profilometer was used to measure the surface roughness of an induced fracture in a carbonate rock (Austin Chalk). From these measurements, a profile of the surface was extracted to provide a rough boundary in an otherwise rectangular shaped domain (Figure 5). Receivers were placed along the left boundary to measure the signals in time. Acrylic cast of the surface were made to provide a homogeneous bulk material with constant wave speeds ($c_p=2730$ m/s, $c_s=1430$ m/s, and $\rho=1185$ kg/m^3). With these values, the material properties of the mesh, (λ,μ,ρ), were calculated. It should be noted that to replicate this domain using the more traditional FDTD methods, the rough surface would have to be oversampled with respect to the wavelength to reduce numerical dispersion. In

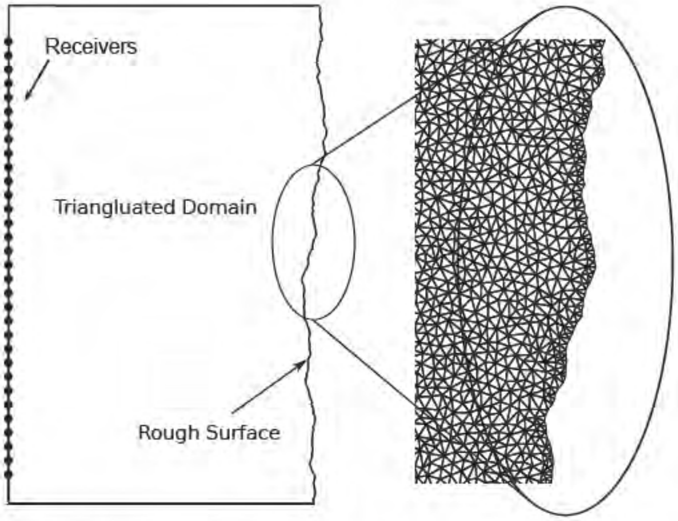

FIG. 5. *The rectangular domain where the laboratory surface roughness measurements are used on the right boundary.*

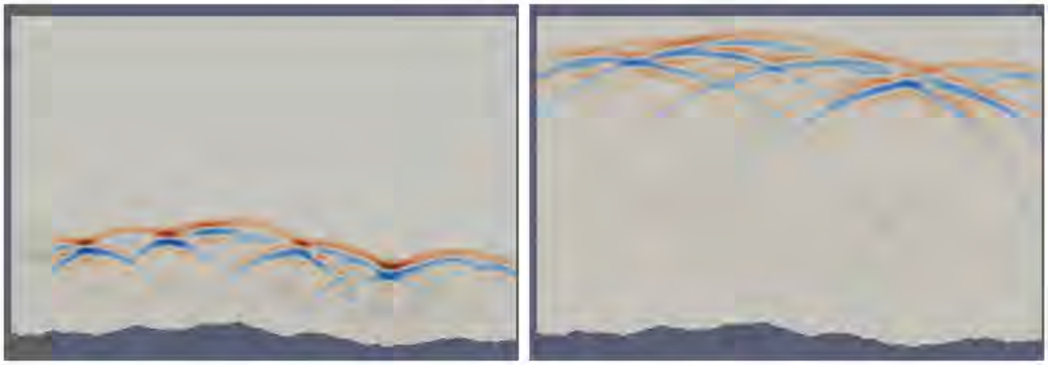

FIG. 6. *Snapshots of the (top) near-field and (bottom) far-field solutions. (Right) Laboratory reflection measurements of similarly scattered waves using water coupled transducers.*

the case of DG, the triangles near the rough surface used low-order approximations to account for their decrease in size. With the triangular mesh defined, a plane wave P-wave was scattered off of the rough surface (with an incident angle of zero). Two snapshots in time were taken from the wave field; the first one shows the near-field solution and the other image shows the far-field solution (Figure 6).

Overall, applying the DG method to realistic fractured surfaces shows promise. However, the complication in comparing numerical simulations to laboratory measurements is the ability to match source/receiver responses.

8.2. Laboratory Fracture: Interface Waves. In this section, the DG implementation was used to generate fracture interface waves [26, 12, 21, 28, 11, 4, 5, 31, 30]. Interface waves are generalized Rayleigh waves. When a shear wave propagates along the surface of a medium, the components polarized perpendicular to the surface generate a Rayleigh wave with a velocity $\approx$0.92 v_s, where v_s is the bulk shear wave velocity. A Rayleigh wave contains both longitudinal and transverse particle motion. v_s is the bulk shear wave (S-wave) velocity. However, if two rough surfaces are placed in contact, they create a mechanical discontinuity that contains regions of contact and regions of void spaces, i.e. a non-welded contact. When the same wave that generated a Rayleigh

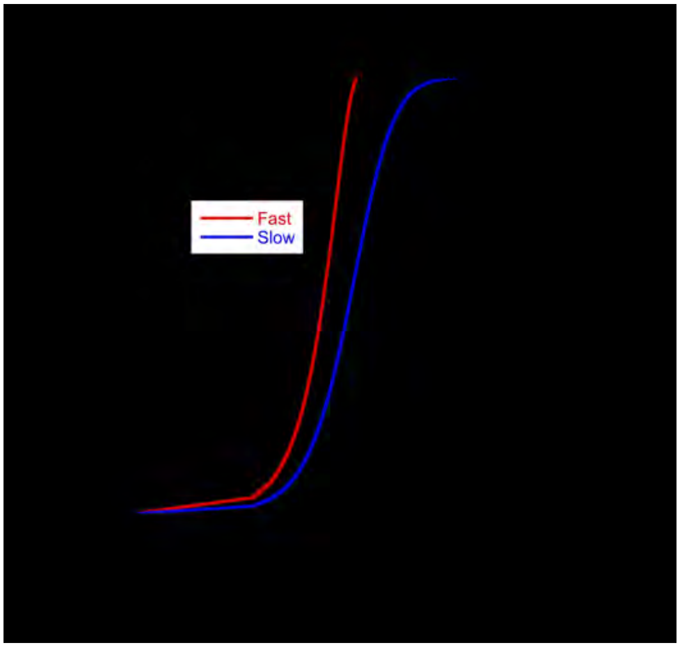

FIG. 7. *Theoretical prediction of fracture specific stiffness from the fracture interface wave velocity. Computed for the material properties of Lucite: v_p = 2730 m/s, v_c = 1430 m/s, ρ = 1185 kg/m^3 and a central frequency 1 MHz.*

wave propagates along this non-welded interface, Rayleigh waves will be generated at the void spaces and will couple through the areas of contact. The arriving fracture interface wave has a velocity that ranges between the classic Rayleigh wave velocity (all void spaces or free surface) and the bulk shear wave velocity (all welded area). The behavior of fracture interface waves is controlled by the mechanical properties of the bulk rock (matrix) and the geometry of the fracture (i.e., aperture and contact area distributions). In other words, fracture interface waves can be used to probe the mechanical properties of fractures in rock. Namely, by measuring the velocities of the propagating fracture interface waves, the fracture specific stiffness can be estimated. Fracture specific stiffness is defined as ratio of increment of stress to the resulting increment in displacement. Pyrak-Nolte & Morris [27] & Petrovitch *et. al.* [25] show that fracture specific stiffness is implicitly linked to the hydraulic properties of a fracture and provides a link between the hydraulic and seismic response of a fracture. Fracture interface waves have been theoretically derived by using displacement-discontinuity theory [21] and observed experimentally along both natural and synthetic fractures at the laboratory scale [28, 22].

We use the DG method to simulate fracture interface waves along fractures with different geometry, i.e. for a range of contact area. An aperture map of a fracture was created to define the domain of the non-welded interface. In the laboratory, an intact sample of Austin Chalk was fractured and an acrylic cast was made of each side. Then surface roughness measurements were performed using laser profilometry and analyzed for each cast to create a map of the void spaces and contact areas formed by the two surfaces. Once the aperture map was obtained, a two dimensional profile was taken from the data set and meshed. The wave velocities of the acrylic casts were measured in the laboratory (at 1 MHz) and found to have a P-wave velocity of $v_p = 2730$ m/s and a S-wave velocity $v_s = 1430$ m/s. The density of the acrylic was $\rho = 1185$ kg/m^3. With these values, the material properties of the mesh, (λ, μ, ρ), were calculated.

From the displacement-discontinuity approach to interface waves [21], the velocity-stiffness relation was computed. The relationship is shown in Figure 7. For a 1 MHz signal, a fracture interface wave travels at the Rayleigh wave velocity for fracture specific stiffnesses below 10^9 Pa/m. For

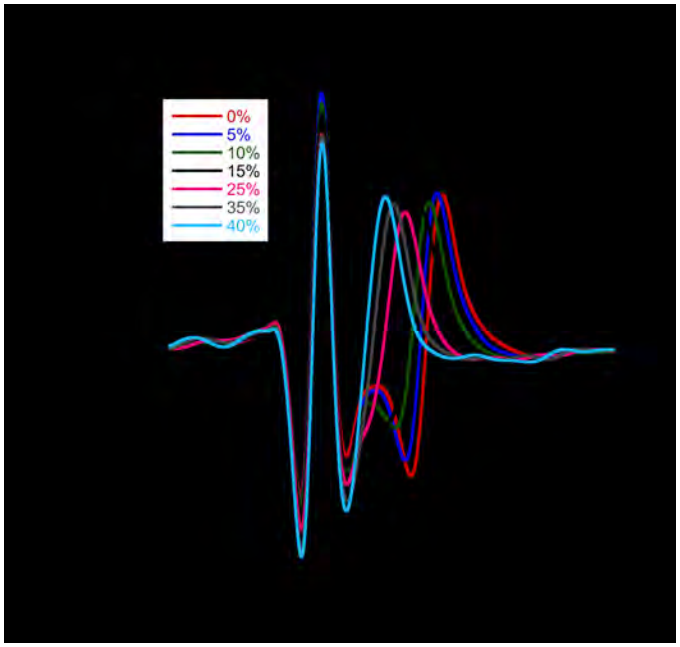

FIG. 8. *Averaged signal across the fracture plane for a range of contact area.*

fracture specific stiffnesses above 10^{14} Pa/m, the fracture interface wave will merge with the initial shear wave front (all welded) and travel at the bulk (or matrix) shear wave velocity. However, between these limits, two fracture interface waves exist called the fast and slow interface wave and travel with speed, v_I. The fast wave depends on the normal fracture stiffness while the slow depends on the shear fracture stiffness. While analytically there exists two distinct interface waves, in this study the difference in wave speeds was to small to resolve for small sample lengths.

To see the shift in the interface velocity, the fracture stiffness was increased by numerically applying a normal load that increased the contact area between the two surfaces. This was accomplished by adjusting the percentage of contact area during the meshing phase. While this is not technically a correct loading method (see Hopkins [14] & Pyrak-Nolte & Morris [27] for another approach), it will illuminate the essential wave physics in the problem. The simulated contact area percentages were: 0%, 5%, 10%, 15%, 25%, 35%, and 40%.

Signals were averaged over 1cm to simulate the data received by a finite size laboratory transducer. The waveforms are shown in Figure 8. We note that the bulk shear wave arrivals only differ in amplitude while the trailing wave systematically decreases in arrival time when the contact area increases. To calculate the interface wave velocity, arrival times were picked after decomposing the signals into a wavelet basis, Figure 9. From the wavelet transforms, the interface wave is clearly observed. However, as the contact area increases, the two waves begin to interfere. Therefore using the central frequency of the interface wave becomes rather difficult. The arrival times of the fracture interface waves (used to calculate the velocities in Table 8.1) were picked by determining the time of arrival time of the maximum amplitude for a frequency of 1 MHz. Table 8.1 lists the interface wave velocities along with the estimated specific stiffnesses. The DG formulation presented here is able to predict the increase in stiffness from the change in the velocity and simulate the observed changes in spectral content [22].

8.3. Homogeneous anisotropic media. In this section, a simple example of the DG methods ability to accurately compute the anisotropic elastic wave equation is given. The same material properties were given to every finite element in a 5 × 5 × 5 meter cubic domain to produce a homogeneous orthorhombic anisotropic medium. Orthorhombic materials have different P-wave and

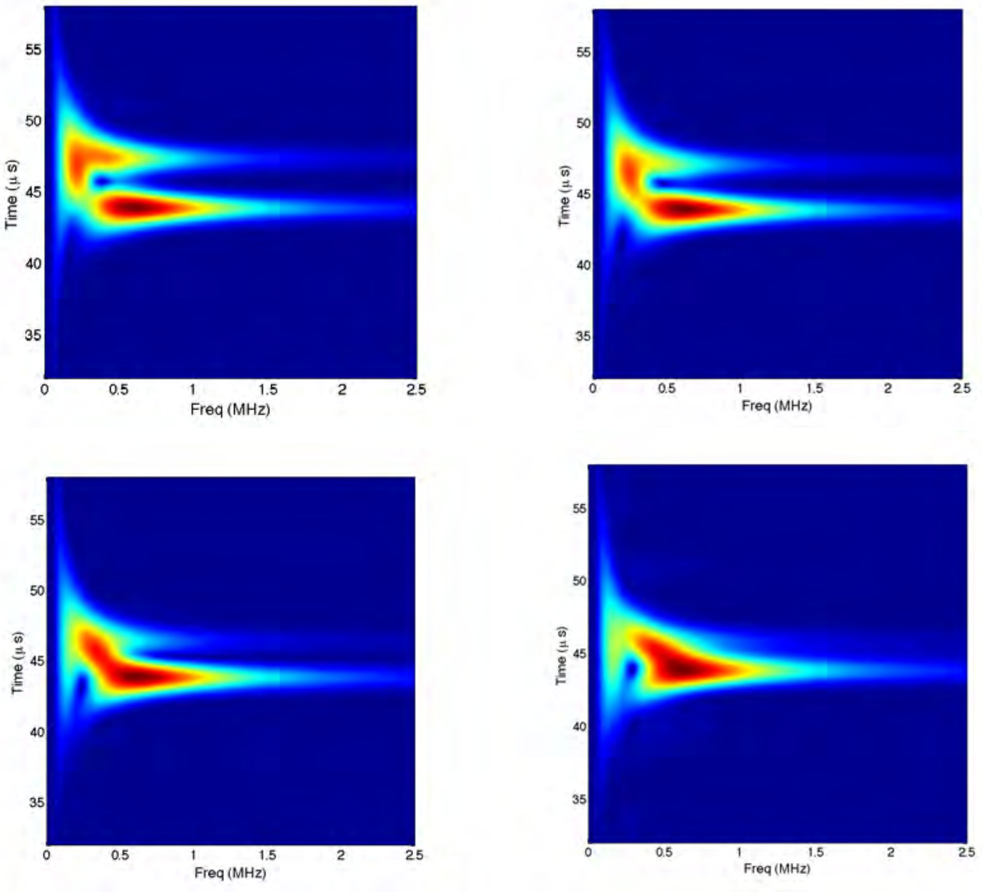

Fig. 9. *Wavelet decompositions of the 0% (top left), 10% (top right), 25% (bottom left), and 40% (bottom right).*

% Contact Area	$v_{interface}$ (m/s)	Stiffness (Pa/m)
0%	1311.7	n/a
5%	1314.9	6.68×10^9
10%	1318.2	1.00×10^{11}
15%	1324.8	2.51×10^{11}
25%	1338.2	6.31×10^{11}
35%	1348.4	7.94×10^{11}
40%	1357.6	1.06×10^{12}

Table 8.1

Calculated stiffnesses via the slow interface wave velocities, $v_{interface}$, and displacement-discontinuity theory.

S-wave velocities in three orthogonal directions, that is, they have three mutually orthogonal planes of symmetry. Each plane of symmetry has its own P-wave and S-wave velocity. For convenience, the symmetry planes were aligned with the Cartesian reference coordinate system, i.e. $\hat{x}$, $\hat{y}$, and $\hat{z}$. The material properties were selected to produce a medium whose P-wave velocity is 5.51 m/s, 4.38 m/s, and 4.00 m/s and S-wave velocity is 2.16 m/s, 3.26 m/s, and 3.58 m/s in the $\hat{x}$, $\hat{y}$, and $\hat{z}$ directions, respectively (Table 8.2). Images of the velocity field are shown in Figure 10. Note that the direct waves are fastest in the x direction and slowest is in the z direction.

Orthorhombic systems have a wide range of real world applications. Theoretically, an or-

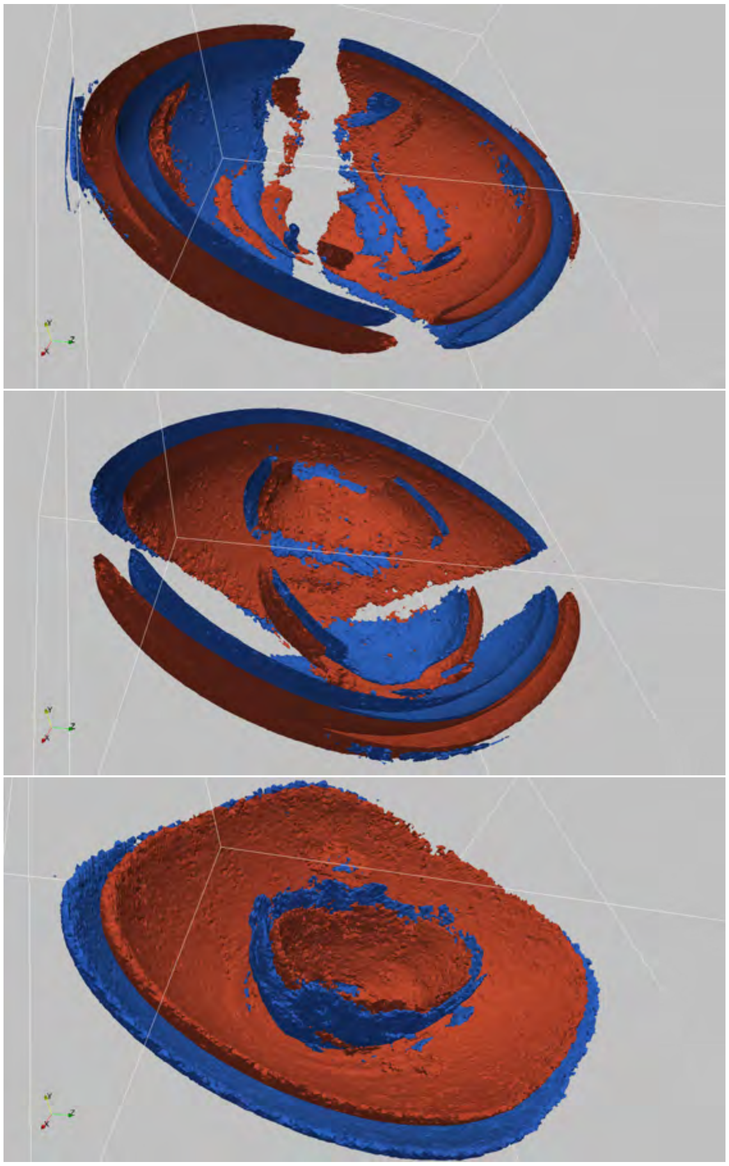

Fig. 10. *Snapshots contour of the velocity fields (top) V_x, (middle) V_y, and (bottom) V_Z at $t = 0.45s$.*

ρ	C_{11}	C_{22}	C_{33}	C_{44}	C_{55}	C_{66}	C_{23}	C_{13}	C_{12}
1.0	30.40	19.20	16.00	4.67	10.86	12.82	4.80	4.00	6.24

Table 8.2
Homogeneous orthorhombic parameters

thorhombic media is most commonly created when a vertically transversely isotropic (layered) background is embedded with parallel vertical fractures or when two or more mutually orthogonal fracture sets intersect at an arbitrary angle [36]. Sedimentary basins often exhibit orthorhombic symmetry, where fracture sets are commonly found in thick sandstone beds and granites. Therefore, while we are only simulating a point source in a simple cubic domain, we are demonstrating the DG methods ability to model a relevant background medium that many field scientists use to describe

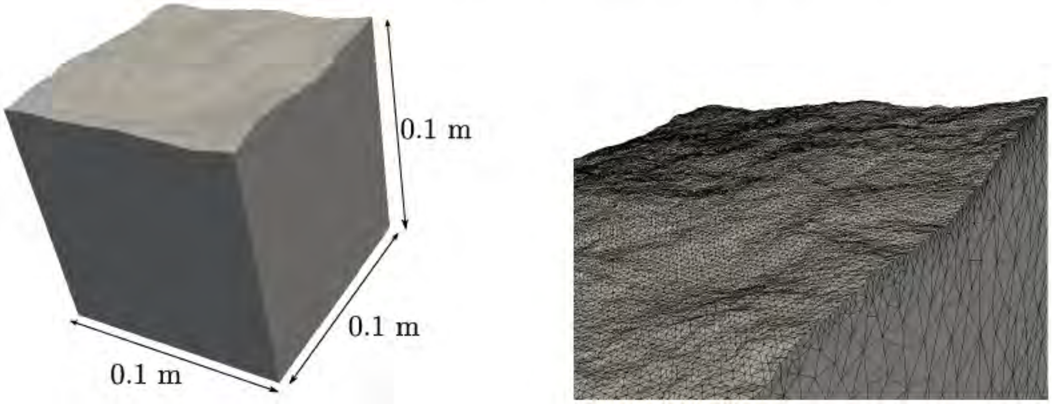

FIG. 11. *(Left) Domain of the digitized rough surface. (Right) Zoomed in of the resulting mesh*

ρ	C_{11}	C_{22}	C_{33}	C_{44}	C_{55}	C_{66}	C_{23}	C_{13}	C_{12}
1.0	24.00	16.00	24.00	4.00	5.20	4.00	8.00	13.60	8.00

TABLE 8.3
Transversely isotropic parameters used in carbonate simulations

the subsurface.

8.4. Fractured Carbonate. Finally, the proposed DG method is used to model the reflection of an explosive point source from a three-dimensional rough surface embedded in a transversely isotropic medium. This type of media closely resembles fractured samples of carbonate rocks [20]. Carbonates are abundantly found in nature and are often found when performing subsurface constructions, for example the work done for the tunnel and reservoir in Chicago, Illinois. They pose many complications when working with them in the field because the physical properties vary from site to site and are heterogeneous within the bulk rock. A homogeneous transversely isotropic medium can be used to model a carbonate because a layering of wave velocities is the most common form of heterogeneity found carbonates [23].

The surface roughness scan (described earlier in the interface wave section) was used to digitize the entire three-dimensional surface of an induced fracture in Austin Chalk. The computational domain was composed of a box with an edge length of 0.1 m (i.e., the same dimension as the sample). The surface was placed so that the displacement of the surface was on the top plane of the box, i.e. $z = 0.1$m (Figure 11). The resulting mesh contained 445,120 tetrahedral elements. The material properties were chosen such that the symmetry axis was in the $(\hat{x}, \hat{y}, \hat{z}) = (0, 1, 0)$ direction, P- and S-wave velocities along the axis of symmetry of 4000 m/s and 2280 m/s respectively, and of 4900 m/s and 2000 m/s, respectively, along the other two directions (note, these are not the velocity of Austin Chalk). Table 8.3 provides a list of the specific elastic constants used. A Ricker wavelet with a central frequency of 4 MHz was placed as an explosive source at $(x, y, z) = (.05, .05, 0)$. Two snapshots of the wave front were taken of the V_z component of the velocity (Figure 12) that display the near and far-field solutions at $t = 32\mu s$ and $t = 48\mu s$, respectively.

8.5. Geological Structure: 3-D Seismic Modeling. In this application, the DG method's ability to model the propagation of seismic waves in a field-scale domain with complex geological structures is demonstrated. The 3D SEAM (SEG Advanced Modeling) acoustic model was used that has heterogeneous structures and represents the sea-bed of the Gulf of Mexico. It spans a 21.9 km by 25.0 km region of the earth's surface and has a depth of 18.75 km, and is sampled over a 876 × 1001 × 751 regular grid with a step size of 25m. The model has several geological features that will test the robustness of the DG method. It contains a high-velocity salt body that extends through the

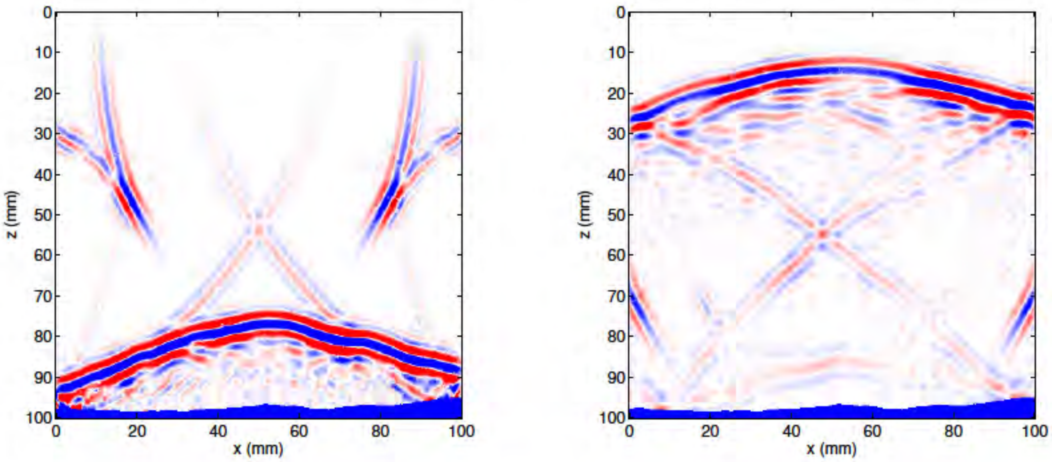

FIG. 12. *Slices of the V_z scattered wave field of after (left) 32 μs and (right) 48 μs from a 3D rough surface. Each slice was rasterized for display purposes.*

center of the model (Figure 13). The rapid contrast in velocity makes the model, in the language of PDE's, a stiff domain, which are notoriously difficult to compute with stability. Another geometric feature are layers of sediment located approximately 16 km under the surface. These layers will cause multiple scattered waves that will lead to constructive and destructive interference.

A tetrahedral mesh with 1,178,875 elements was generated adaptively for the rough contour of salt body as well as the smooth interface of sediments. Compared to a regular mesh, the tetrahedral mesh has the potential to properly represent the complex interfaces using less nodal points. We adaptively added nodes to the mode by first tracking the interfaces between high and low velocity regions inside the model using a common-place edge tracking technique. A uniform sample of nodes was placed along each of the detected surfaces. Using these key nodes, a tetrahedral mesh was created using Tetgen [32]. The use of an unstructured mesh is absolutely essential to model, correctly, a strongly heterogeneous media like SEAM.

A point source was placed at sea level $(x, y, z) = (10.5, 10.5, 0.0)$ and the source function was a Ricker wavelet with a central frequency of 10 Hz. The heterogeneity of the model generates scattered P and S waves. The primary energy coming from source reflects back to surface, also called a "diving wave," because of the gradually increasing velocity with depth. In addition, single and multiple scattering waves at the rough interior boundaries occur. Multiples scatterings are quite common because of the layered geometry. 12,034 receivers were placed at sea level in a rectangular grid that was 200 by 200 meters. The receivers measured all three components of the particle velocity as well as the acoustic pressure. Because the model simulates a half-space, absorbing boundaries and a PML region was applied on all sides of the model. A snapshot of the solution is shown in Figure 14, 15, and 16.

9. Discussion. In this paper, we have shown that the DG method is an ideal method to solve the anisotropic elastic wave equation because of its ability produce highly accurate solutions for domains with heterogeneous complex geometries. Complex wave types, Rayleigh and Stoneley surface waves, were used to demonstrate that the DG formulation has suitable convergence rates. These are an absolute necessity and a very robust test to use since the models in question are attempting to simulate realistic fractures and subsurface models. The resulting mesh when generating from a fracture geometry measured in the laboratory has literally hundreds of void spaces along a fracture plane; each of which will generate a Rayleigh wave. If any flaws exist in the implementation of the free-surface boundary conditions, the error in Rayleigh waves generated by each individual crack would render the final solution worthless. At the global scale, the complexity enters the computational domain when layering in the subsurface "ripples" (Figure 13) where heterogeneities appear.

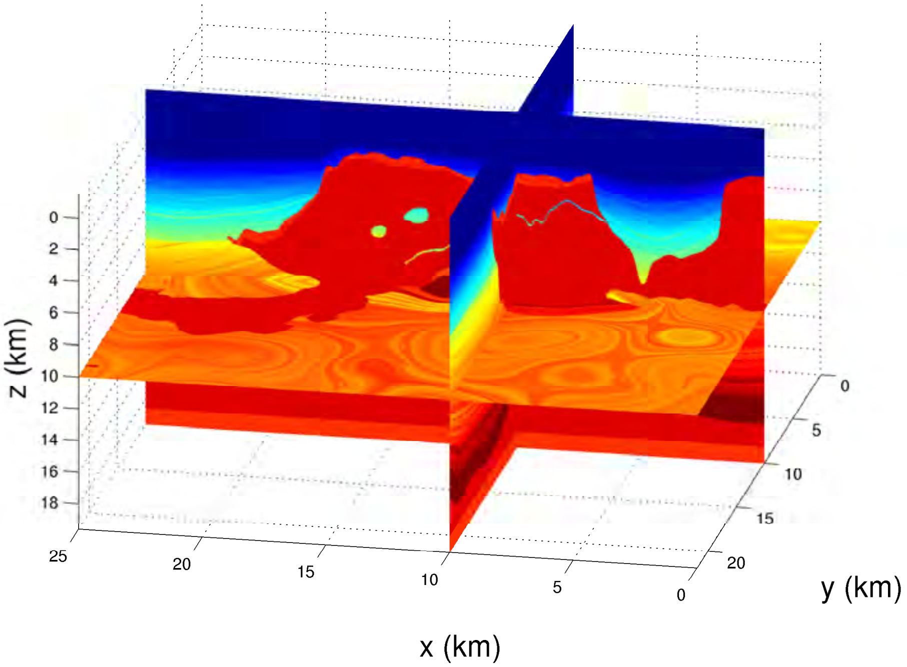

FIG. 13. *3D SEAM acoustic velocity model.*

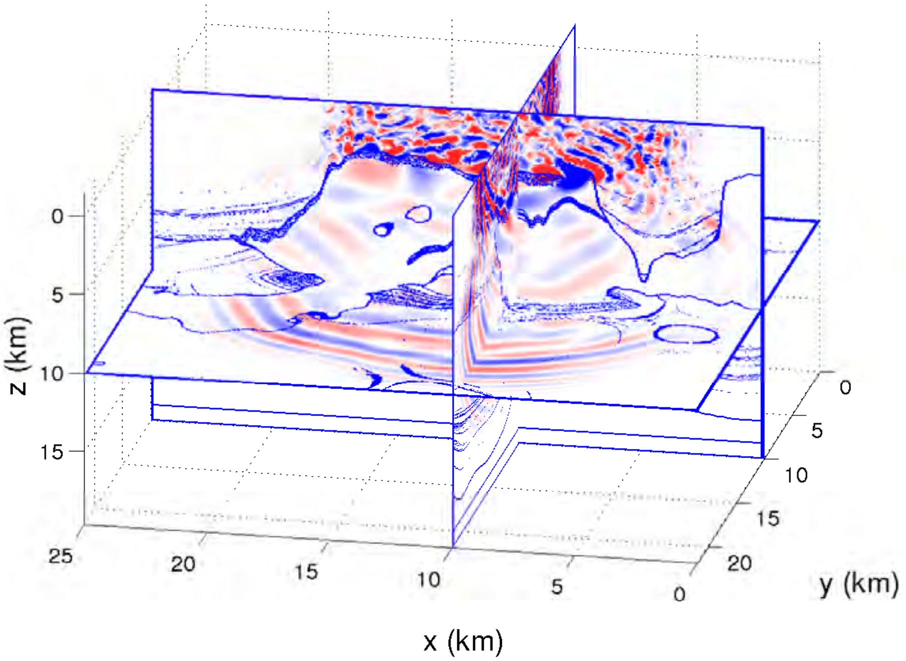

FIG. 14. *Snapshot of V_x taken at $t = 4.2s$ of the 3D SEAM velocity model.*

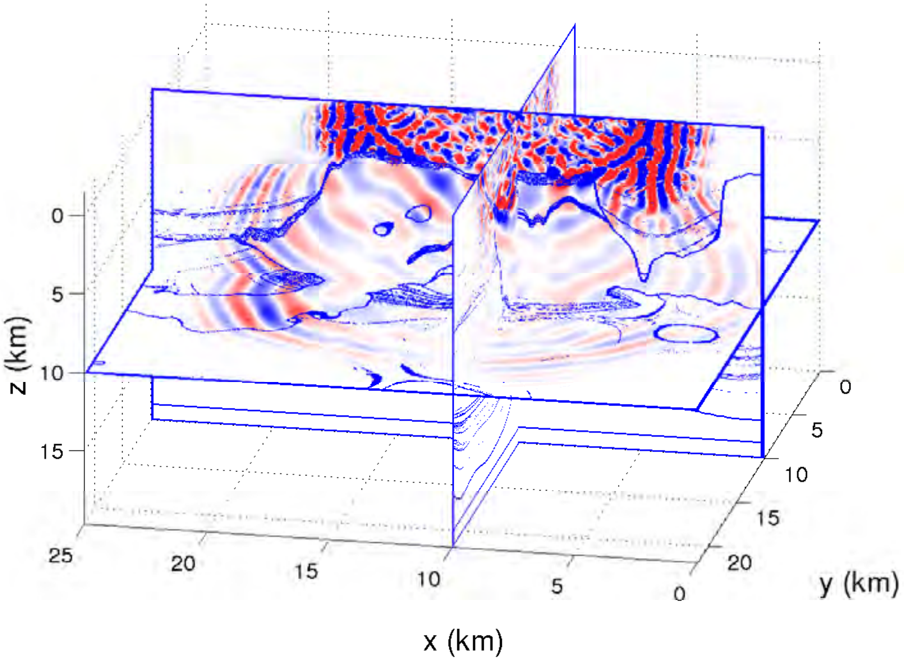

FIG. 15. *Snapshot of V_y taken at $t = 4.2s$ of the 3D SEAM velocity model.*

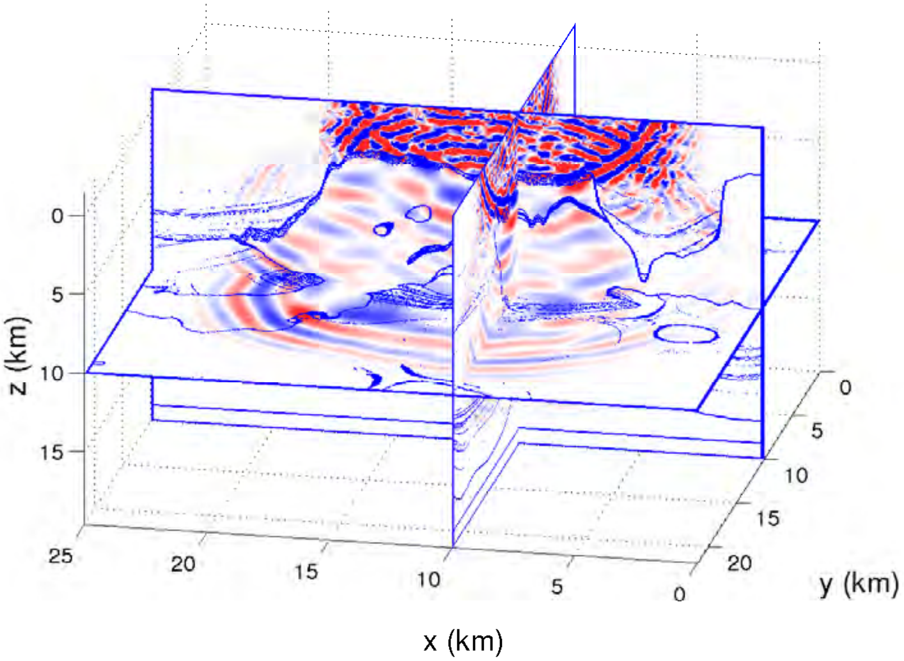

FIG. 16. *Snapshot of V_z taken at $t = 4.2s$ of the 3D SEAM velocity model.*

Such geometries require the DG method to accurately compute the Stoneley waves. Again, if the DG method failed, the accumulation of the errors would render the solution useless.

The success of any computational method hinges on its ability to represent, accurately, the domain in question. Unstructured tetrahedral meshes were paramount to the successful modeling of the complex geometries used in this paper because of the plethora of automated tetrahedral mesh generators. In the authors opinion, the ability to mesh a domain using tetrahedrals rather than hexahedrals when computing solutions for the subsurface is a tremendous advantage of the DG method over other high-order methods, namely the SE method. In most cases, the domains in question are far too complicated to allow scientists or engineers to create a systematic method to mesh the domain themselves and they must turn to automated meshing software. Because tetrahedral mesh generators are far more abundant and can handle more complex geometries than those that use hexahedrals, selecting a computational method that is not bound to a specific finite element type is preferred. In the authors' opinion, this is one of the major drawbacks to SEM.

Another aspect of developing a computational method to solve the anisotropic elastic wave equation is the inversion of the mass matrix. In the classic continuous finite element method, the semi-discrete form contains a mass matrix on the left hand side that couples each element together rendering it implicit. Since inverting the matrix is far to expensive, solving the system iteratively for each time step or factorizing the mass matrix has been the only acceptable solution. Currently, there are two schools of thought to create an explicit semi-discrete form of the elastic wave equation. The SE method evolved by, as mentioned, restricting the mesh to hexahedral elements so that the nodal points could be, cleverly, chosen at the Gauss-Lobatto-Legendre points. By using Gauss-Lobatto-Legendre quadrature over each element, the mass matrix will be diagonal, thereby allowing the semi-discrete form to be explicitly solved. The other method is to use a "broken" finite element mesh, where each element is discontinuous, i.e. the DG method. By doing so, each element will decouple by construction and the mass matrix will be block diagonal. Then, each block of the mass matrix will be equivalent (when the same type of element is used) and the inverse can be tabulated beforehand. The advantage of the latter method over the former is that it isn't dependent on a specific set of nodal points and works for any type of finite element while the Gauss-Lobatto-Legendre only exist for hexahedrals. However, the drawback to using the DG method is the number of unknowns. There will always be a larger amount of unknowns (for a given mesh) with the DG method rather than a continuous method because the discontinuous method will have two degrees of freedom over each facet node. This is required so that a flux condition is imposed to couple each local element into a global solution.

Mesh adaptively is another area where the DG method is superior to continuous methods. There are two types of adaptively. The first is h-adaptively, where the characteristic size, h, of the finite elements are allowed to vary. By doing so, one can specify areas of the computational domain that require a large density of nodes by reducing the element size and the mesh can be fit to all complex geometric topographies or discontinuities. However, then the use of automated meshing algorithms becomes the key to success. All FEM (classic FEM, SEM and DG) methods account for this h-adaptively and the true burden lies in creating the computational mesh. The second is p-adaptively, where the number of degrees of freedom per finite element is allowed to change from element to element. This becomes very important when the computational mesh is strongly h-adaptive. To produce a stable solution, the computational mesh must be created such that there are at least 8 to 10 degrees of freedom per wavelength. However, it is not always possible to account for the complex geometric structures in the subsurface without exceeding this number. p-Adaptively is used to combat this realization. Reducing the degrees of freedom in smaller elements (and increasing it for larger elements) has two advantages: the maximum allowable time step can be increased in regions of oversampling and the computational efficiency can be increased. Traditional continuous FEM methods account for p-adaptively, but require the use of rather complicated basis sets because the number of nodes on any given facet of an element differs. To the authors knowledge, the SEM method cannot handle p-adaptive meshes because it is tightly restricted by its quadrature requirements. p-adaptively poses no problem for the DG method because the volume of each element is completely localized. In fact, the weak formulation (Equation 2.10) is inherently p-adaptive by construction because of this localization. The only part of the implementation that is effected by a p-adaptive mesh is the flux, but this is simply handled by recognizing that lower order elements must be projected (L^2 projection) to the higher order elements such that the jumps and averages can be computed. In this light, we believe that when considering computational domains from the laboratory scale to the global scale, the DG method is a proper framework to provide high-order accurate results.

The last difficulty that must be discussed is the discretization of the time derivative. The mesh elements can vary by large amounts when meshing domains involve realistic geometries. This occurs when the geometric features vary over many wavelengths and the mesh generation software forces high quality elements. At which point, the mesh has two regions: one where elements have a reasonable number of degrees of freedom per wavelength and one where the elements are orders of magnitude smaller. Because of this, the time step, determined by the CFL conditions is orders of magnitude smaller than is required by the majority of the domain, thereby wasting a tremendous amount of computational effort. There have been two lines of thought to combat this problem: (1) allowing each element to have its own *local* time step and (2) employing semi implicit-explicit (IMEX) methods. Local time-stepping has been used successfully by Kaiser and Dumbser [9]. However, properly balancing the load in a parallel environment is difficult. The domain cannot be decomposed directly using more traditional methods, for example METIS [16], because each element requires a different amount of computational effort per simulation time. The other option is to break the residual vector into two parts; those with large sizes and those with small sizes, and then integrating the small elements implicitly and the larger explicitly. This allows the small element's time step to increase without risking instability at the cost of solving a system of equations. By mixing the two methods, the time step is set to a value that is closer to the traditional value, dictated by the CFL conditions for the large elements.

10. Conclusion. In this paper, we have proposed and shown that the DG method is an ideal framework to use when the computational domain in question contains anisotropic materials and complex geometric features over a range of length scales. The anisotropic elastic wave equation was written in its strong form so that the solution could be approximated with the DG method. A numerical flux was derived such that the localized finite elements were coupled in an accurate and stable manner. The external boundaries of the domain were also accounted for by providing numerical fluxes. Both free-surface and absorbing boundary conditions were provided. CPML regions were also derived to improve upon the absorbing boundary conditions. The CPML's enable this method to simulate an infinitely sized domain by reducing, for all practical purposes, the reflections from the boundary. Two time discretization schemes, the classic LSERK method and the more complex IMEX method were also presented. Convergence rates were found by utilizing the analytic solutions of two different surface waves: Rayleigh and Stoneley waves. By combining all of these elements, the method was tested with several complex rock physics and global seismology applications. Generalized Rayleigh waves, or interface waves, were shown to agree with Displacement Discontinuity theory by displaying a fracture specific stiffness dependence without the use of special "jump conditions." The DG method was also used to generate complex wave fields from an explosive point source in an orthorhombic medium as well as a transversely isotropic medium that was embedded into a domain with a 3D rough surface. These applications illustrated the DG method's ability to compute high-order solutions when complex computational domains with complex materials are used. Lastly, we moved to the global scale by considering the 3D SEAM model. This model displayed the DG methods ability to correctly handle a domain with extremely complex geometric features within its interior. After using the DG method to compute solutions in all of the applications listed, we demonstrated that the DG method is an ideal method to solve the worlds more complex wave propagation problems.

11. Acknowledgements. This work is supported by the Geosciences Research Program, Office of Basic Energy Sciences US Department of Energy (DEFG02-97ER14785 08, DE-FG02-

09ER16022), by the Geo-mathematical Imaging Group at Purdue University, the Purdue Research Foundation, and from the Computer Research Institute At Purdue University

REFERENCES

[1] Jan D Achenbach and Jan Drewes Achenbach. *Wave propagation in elastic solids*, volume 16. North-Holland Amsterdam, 1973.

[2] Uri M Ascher, Steven J Ruuth, and Raymond J Spiteri. Implicit-explicit runge-kutta methods for time-dependent partial differential equations. *Applied Numerical Mathematics*, 25(2):151–167, 1997.

[3] Caroline Baldassari, Héléne Barucq, Henri Calandra, and Julien Diaz. Numerical performances of a hybrid local-time stepping strategy applied to the reverse time migration. *Geophysical Prospecting*, 59(5):907–919, 2011.

[4] Gu BL. *Interface Waves on a Fracture in Rock.* PhD thesis, University of California, 1994.

[5] Gu BL, Nihei KT, Myer LR, and Pyrak-Nolte LJ. Fracture interface waves. *Journal of Geophysical Research-Solid Earth*, 1996.

[6] B. Cockburn. Discontinuous galerkin methods. *ZAMM - Journal of Applied Mathematics and Mechanics / Zeitschrift f ur Angewandte Mathematik und Mechanik*, 83(11):731–754, 2003.

[7] J De la Puente, M. Käser, M. Dumbser, and H. Igel. An arbitrary high order discontinuous galerkin method for elastic waves on unstructured meshes iv: Anisotropy. *Geophys. J. Int.*, 169(3):1210–1228, 2007.

[8] Michael Dumbser and Martin Käser. An arbitrary high-order discontinuous galerkin method for elastic waves on unstructured meshes–ii. the three-dimensional isotropic case. *Geophysical Journal International*, 167(1):319–336, 2006.

[9] Michael Dumbser, Martin Käser, and Eleuterio F Toro. An arbitrary high-order discontinuous galerkin method for elastic waves on unstructured meshes–v. local time stepping and p-adaptivity. *Geophysical Journal International*, 171(2):695–717, 2007.

[10] V Etienne, E Chaljub, J Virieux, and N Glinsky. An hp-adaptive discontinuous galerkin finite-element method for 3-d elastic wave modelling. *Geophysical Journal International*, 183(2):941–962, 2010.

[11] Murty GS and Kumar V. Elastic wave prpagation with kinematic discontinuity along anon-ideal interface between two isotropic elastic half-spaces. *Journal of Nondestructive Evaluation*, 1991.

[12] Boliang Gu, Kurt T Nihei, Larry R Myer, and Laura J Pyrak-Nolte. Fracture interface waves. *Journal of geophysical research*, 101(B1):827–835, 1996.

[13] J.S. Hesthaven and T. Warburton. *Nodal Discontinuous Galerkin Methods: Algorithms, Analysis, and Applications.* Springer Verlag, New York, 2008.

[14] Hopkins. *The Effect of Surface Roughness on Joint Stiffness, Aperture, and Acoustic Wave Propagation.* PhD thesis, 1990.

[15] Thomas JR Hughes. *The finite element method: linear static and dynamic finite element analysis.* Dover Publications, 2000.

[16] George Karypis and Vipin Kumar. Metis-unstructured graph partitioning and sparse matrix ordering system, version 2.0. 1995.

[17] D. Komatitsch and R. Martin. An unsplit convolutional perfectly matched layer improved at grazing incidence for the seismic wave equation. *Geophysics*, 72(5):SM155–SM167, 2007.

[18] Dimitri Komatitsch and Jean-Pierre Vilotte. The spectral element method: An efficient tool to simulate the seismic response of 2d and 3d geological structures. *Bulletin of the Seismological Society of America*, 88(2):368–392, 1998.

[19] Min-Ho Koo and DarrellI. Leap. Modeling three-dimensional groundwater flows by the body-fitted coordinate (bfc) method: I. stationary boundary problems. *Transport in Porous Media*, 30(2):217–239, 1998.

[20] W. Li, C.L. Petrovitch, and L.J. Pyrak-Nolte. The effect of fabric-controlled layering on compressional and shear wave propagation in carbonate rock. *International Journal of the JCRM*, 4(2):79–85, 2009.

[21] Pyrak-Nolte LJ and Cook NGW. Elastic interface waves along a fracture. *Geophysical Research Letters*, 14(11):1107–1110, 1987.

[22] DD Nolte, LJ Pyrak-Nolte, J Beachy, and C Ziegler. Transition from the displacement discontinuity limit to the resonant scattering regime for fracture interface waves. *International Journal of Rock Mechanics and Mining Sciences*, 37(1):219–230, 2000.

[23] R Nurmi, M Charara, M Waterhouse, and R Park. Heterogeneities in carbonate reservoirs: detection and analysis using borehole electrical imagery. *Geological Society, London, Special Publications*, 48(1):95–111, 1990.

[24] P.-O. Persson. High-order les simulations using implicit-explicit runge-kutta schemes. In *Proc. of the 49th AIAA Aerospace Sciences Meeting and Exhibit*, 2011.

[25] C.L. Petrovitch, L.J. Pyrak-Nolte, and D.D. Nolte. Scaling of fluid flow verus fracture stiffness, accepted for publication. *Geophysical Research Letters*, 2013.

[26] Laura J Pyrak-Nolte, Larry R Myer, and Neville GW Cook. Anisotropy in seismic velocities and amplitudes from multiple parallel fractures. *Journal of Geophysical Research*, 95(B7):11345–11, 1990.

[27] LJ Pyrak-Nolte and JP Morris. Single fractures under normal stress: The relation between fracture specific stiffness and fluid flow. *International Journal of Rock Mechanics and Mining Sciences*, 37(1):245–262,