An optimized 2D/3D Finite-difference Seismic Wave Propagator Using Rotated Staggered Grid for Complex Elastic Anisotropic Structures

Oumeng Zhang[1] and Douglas R. Schmitt[1]

[1]Purdue University, West Lafayette, IN, USA

Corresponding author: Oumeng Zhang (zhan3386@purdue.edu)

Key Points:

- We implement a rotated staggered grid (RSG) elastic wave solver based on open-source finite-difference Python package.
- This wave solver is optimized and can simulate the 2D/3D elastic wave propagation in complex anisotropy media and structure.
- The stability and performance of this RSG wave solver is tested and compared to traditional standard staggered grid (SSG).

Abstract

The progression of computational technologies has catalyzed a better understanding into the authentic properties of geological materials. Seismic anisotropy, a pivotal elastic characteristic of Earth's subsurface, profoundly affects seismic wave propagation, underscoring the need for its precise modeling to interpret geophysical data and to unravel subsurface structures. In response, we have engineered and implemented a Rotated Staggered Grid (RSG) finite-difference solver, integrated with the open-source finite-difference framework Devito, for modeling of wave propagation in intricate anisotropic media across two-dimensional (2D) and three-dimensional (3D) domains. We explore the foundational principles of finite-difference modeling and stress-velocity relationship and compare the RSG with the Standard Staggered Grid (SSG). Our implementation details optimization using fewer derivative operations and source excitation techniques to addresses the checkerboard artifacts. Our analysis highlight that the RSG scheme outperforms SSG in complex anisotropic media, in maintaining the stability of the wavefield. Performance evaluations demonstrate that the RSG approach is a robust alternative for 2D modeling, providing enhanced stability and precision, thereby enabling its regular application in seismic studies without the burden of excessive computational costs. This wave propagator is intended to deepen the geophysical understanding of complex anisotropy and its geological implications, thereby facilitating more comprehensive and frequent investigations into the challenges posed by complex anisotropic media and structures.

Plain Language Summary

Anisotropy, the variation in material properties with direction, is a fundamental aspect of the Earth's subsurface that should be accurately characterized to enhance our understanding of seismic wave dynamics. The complexity of elastic anisotropy necessitates advanced study to improve interpretations of seismic data, which are critical for geophysical exploration and seismic risk assessment for fractured media. Although conventional finite-difference method for seismic wave modeling is computationally efficient, it has been limited in their ability to capture these complex anisotropic behaviors, leading to computational instability. Recognizing the need for a more capable modeling strategy, we implement a Rotated Staggered Grid (RSG) wave propagator based on an open-source finite-difference package. Despite the theoretical proposal of this method two decades ago, a practical and openly accessible application has been notably absent in the field.

This RSG wave propagator effectively addresses the checkerboard problem and supports arbitrary even-order spatial finite-differencing, thereby enhancing the accuracy of seismic wavefield simulations. We evaluate the RSG wavesolver's stability and performance and offer a comparative analysis with traditional Standard Staggered Grid (SSG). The findings demonstrate that our RSG implementation not only resolves the challenges associated with complex anisotropy but also provides a stable and precise alternative for the seismic modeling.

1 Introduction

Modeling seismic wave propagation is an important tool for understanding the interior structure of the earth (Tromp et al., 2004), for evaluating geological hazards (Chaillat et al., 2009), for mapping the elastic physical properties of unreachable rocks (Sidler et al., 2013), and for detailed subsurface imaging in the pursuit of resources (Virieux & Operto, 2009). While analytical solutions provide rapid answers in simplified geometries, numerical modeling are required to understand more complex wave propagation in intricate geological and anisotropic structures.

Essentially, numerical models solve the wave equation within a defined structure. A variety of numerical methods have been developed that include the spectral element method (SEM) (Komatitsch & Tromp, 1999), the pseudo-spectrum method (PSM) (Kosloff & Baysal, 1982), finite-element method (FEM) (Serón et al., 1990), finite-difference method (FDM) (Virieux, 1986), and finite-volume method (FVM) (Dormy & Tarantola, 1995). All of these perform to varying degrees of success depending on the final application. Of these, the finite difference methods are most straightforwardly implemented and remain widely used (Graves, 1996; Levander, 1988; Operto et al., 2007). One disadvantage is that FDM models are constructed of rectangular cells that unavoidably result in stair-step approximations at curved interfaces resulting in some error. On the other hand, the simplified model construction and associated reduced computational cost provide an advantage.

That said, wave solver performance remains an impediment in modeling of complex structures with finite-difference method (FDM). Emerging full waveform inversion (Virieux & Operto, 2009) or adjoint methods (Tape et al., 2009) require updating and recalculating large forward seismic models thousands of times; calculations that can consume a supercomputer for days or weeks both taking time and contributing to greenhouse gas emissions. The tradeoff between accuracy and computing affordability remains a major concern for all seismic wave

propagation modeling. Meanwhile, with the advances of computing power future seismic models should allow for the investigation of more complex geological structures comprised of more realistic anisotropic material. Our motivation for this work stems from the need to understand wave propagation in a 3D fractured anisotropic structure.

As is well known, seismic anisotropy simply means to many that the wave speeds vary with the propagation direction through an elastic material or structure, although additional complications arise with wave polarizations, birefringence (shear wave splitting), and differences between plane wave and ray propagation (Crampin, 1984). The existence of anisotropy has been known for decades (Helbig & Thomsen, 2005), but it remains largely overlooked or deemed an unnecessary complication in seismic studies. All crystalline minerals are elastically anisotropic, as are foliated, fractured, and layered geological structures (Almqvist & Mainprice, 2017). The critical examination and quantification of this phenomenon, particularly in weakly anisotropic media, were significantly advanced through the groundbreaking work of Thomsen (1986). Nowadays, the utilization of Thomsen's parameter is common in industry and this approximation has led better quality seismic imaging (Tsvankin et al., 2010). However, as computational capabilities progress there is a growing realization (Assad, 2005; Cheadle et al., 1991; Malehmir & Schmitt, 2017; Schoenberg & Sayers, 1995; Tsvankin, 1997) that Thomsen's approximations are not sufficient to describe more complicated situations of tilted and fractured formations necessitating that elastic wave solvers capable of handling more general, lower symmetry cases of anisotropy.

The standard staggered grid method (SSG) introduced by Virieux (1986) is the most prevalent finite-difference scheme employed for elastic wave modeling. This scheme leverages the relationship between particle velocity and stress, strategically placing these components on a staggered grid. This arrangement effectively suppresses numerical dispersion, compared with the previous non-staggered grid, where all the components are at the same node. However, as demonstrated later, the SSG encounters limitations when dealing with intrinsic anisotropy or in scenarios where the symmetric axis does not align with the coordinate axis, such as for tilted transverse isotropy (TTI).

To address these challenges, both the rotated staggered grid (RSG) (Saenger, Gold, and Shapiro, 2000) and Lebedev schemes (Lisitsa & Vishnevskiy, 2010) were developed to offer

robust solutions for modeling seismic wave propagation in anisotropic media. While the Lebedev scheme offers the advantage of reduced memory usage, the RSG is especially notable for its decreased computational demand requiring fewer float floating point operations per second (FLOPs). This efficiency combined with RSG's precision in modeling complex anisotropies makes it a particularly appealing choice for extensive simulations. Given the distinct advantages and applications of both methods, the choice between RSG and Lebedev often depends on the specific objectives and constraints of individual studies. In this paper, while acknowledging the effectiveness of both approaches, we will primarily focus on the Rotated Staggered Grid (RSG) method.

Over the years, RSG methodologies have significantly evolved. Bohlen and Saenger (2006) applied RSG to model Rayleigh waves in 2D and noted that the accuracy is improved by larger numbers of nodes, suggesting potential difficulties for full 3D modelling. Krüger, Saenger, and Shapiro (2005) illustrated the efficacy of RSG in handling high elastic moduli contrast and simulate the seismic wave from a single crack. While limited in 2D with 2nd-spatial order, the results aligned closely with analytical solutions. Chen, Wang, and Zhao (2006) introduced a perfectly-matched layer (PML) for 2D RSG to achieve better absorption at the boundary; however, while effective, the additional computational effort required was not addressed. Bansal and Sen (2008) brought the study into the 3D realm, leveraging fourth-spatial ordered RSG to model S-wave splitting. Their insights emphasized the significance of S-wave splitting in fracture studies. W. Wang et al. (2017); Yang, Yan, and Liu (2015) optimize 2D RSG with variable-order and choice of finite-difference coefficient, respectively. Gao and Huang (2017) use fourth temporal order for better precision in 2D RSG. Whereas, this can be difficult to realize in 3D modeling due to the substantial memory demands. K. Wang et al. (2020) delved into fractured media modeling with 4th-ordered 2D RSG and suggest consistency with theoretical result. Zheng et al. (2023) implement RSG on GPUs using CUDA and emphasized the performance comparison between GPU and CPU. However, the implications of varying spatial orders with these platforms were left unexplored.

The trajectory of studies highlights the necessity for an integrated RSG approach encompassing both 2D and 3D geometries and media with lower symmetry than transversely isotropic with arbitrary spatial orders and potential arithmetic optimization. Here, we first briefly review the theory of staggered grid finite-difference modeling. We then detail our implementation

highlighting aspects such as arbitrary spatial order derivatives, optimization techniques, and source excitation. Finally, we evaluate and benchmark the computational performance of our RSG solver and compare to SSG. To the best of our knowledge, despite the long-standing proposal of RSG, no open-source implementations are readily accessible to the public. The code developed in this study within the existing Devito package is also described and made readily accessible.

2 Background Theory

2.1 Stress-velocity relation

Modeling of seismic wave propagation using finite-difference methods has already been extensively explored in the literature; and here simply review the fundamental principles. For a comprehensive understanding, readers are referred to Moczo, Robertsson, and Eisner (2007) and Iturrarán-Viveros and Sánchez-Sesma (2020). By revisiting these essential equations and their numerical representations, we aim to set a solid groundwork for our subsequent discussions and analyses.

The fundamental equation governing the motion of seismic waves is given by the equation of motion:

$$\rho \frac{\partial^2 u_i}{\partial t^2} = \frac{\partial \tau_{ij}}{\partial x_j} + f_i \qquad (1)$$

where ρ represents the density and u_i denotes the components of the displacement vector in a x-, y-, and z- Cartesian coordinates, respectively (with $i = 1,2,3$), τ_{ij} denotes the Cauchy stress tensor and f_i denotes the external body force applied to the system. The source is activated at specific points and introduced by the moment tensor, which describes the spatial orientation and characteristics of the seismic source (Vavryčuk, 2015) within the system or area of interest. The body force term f_i is frequently omitted in many seismic simulations, unless external forces become dominant, e.g., gravity in low-frequency seismology simulation.

The equation can alternatively be expressed in terms of a stress-velocity relation by substituting particle velocity for displacement omitting f_i:

$$\rho \frac{\partial v_i}{\partial t} = \frac{\partial \tau_{ij}}{\partial x_j} \qquad (2)$$

Where v_i is denotes particle velocity vector. The relationship between stress and strain in the context of seismic waves is governed by the elastic Hooke's Law. For small amplitude vibrations leading to microscopic deformations, the stress-strain relation remains linear:

$$\tau_{ij} = C_{ijkl}\varepsilon_{kl} \quad (3)$$

Where C_{ijkl} is the 4th order elastic stiffness tensor with 81 components and ε_{kl} is the strain tensor. The rate of change of strain with respect to time can be mathematically expressed in terms of particle velocities v_i, and spatial coordinates x_i, as:

$$\frac{\partial \varepsilon_{ij}}{\partial t} = \frac{1}{2}\left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i}\right) \quad (4)$$

To numerically model the seismic propagation, discretization in space and time of the equations is necessary. In a second-order temporal scheme, the discretization stencils can be formulated as:

$$v_i^+ = v_i + \Delta v_i = v_i + \frac{\partial \tau_{ij}}{\partial x_j}\frac{1}{\rho}\Delta t \quad (5)$$

and

$$\tau_{ij}^+ = \tau_{ij} + \Delta\tau_{ij} = \tau_{ij} + \frac{1}{2}C_{ijkl}\varepsilon_{kl}^+\Delta t = \tau_{ij} + \frac{1}{2}C_{ijkl}\left(\frac{\partial v_k^+}{\partial x_l} + \frac{\partial v_l^+}{\partial x_k}\right)\Delta t \quad (6)$$

Here two memory-intensive wavefield components of τ_{ij} and v_i represent stress and velocity components, respectively. In the notation, the absence of an upper script indicates the current field value, while the superscript '+' designates the field for the subsequent time step. The symbol Δt denotes the time spacing. Equations 5 and 6 are elastodynamic equations for elastic anisotropic media. The finite-difference modeling proceeds through updating these two wavefields v_i and τ_{ij} with the two Equations 5 and 6 alternatively.

2.2 Standard Staggered Grid (SSG) and Rotated Staggered Grid (RSG)

This section elucidates the distinct evaluation methods employed different schemes, aiming to provide insights on the process of the modeling and the required finite-difference operation.

The Standard Staggered Grid (SSG) assigns components varied locations in a staggered manner, as illustrated in Figure 1. Within a SSG unit cell, there are three primary nodal types: edge midpoints, that one of three index is shifted by half of the grid spacing, e.g., shear stress τ_{ij}, $i \neq j$; face centers, that two of three indices are shifted by half of the grid spacing, e.g., v_i; cell centers, that all three indices are shifted by half of the grid spacing, e.g., elastic stiffness C_{ijkl}, density ρ and normal stress τ_{ij}, $i=j$. Notably, no components are positioned at the vertices.

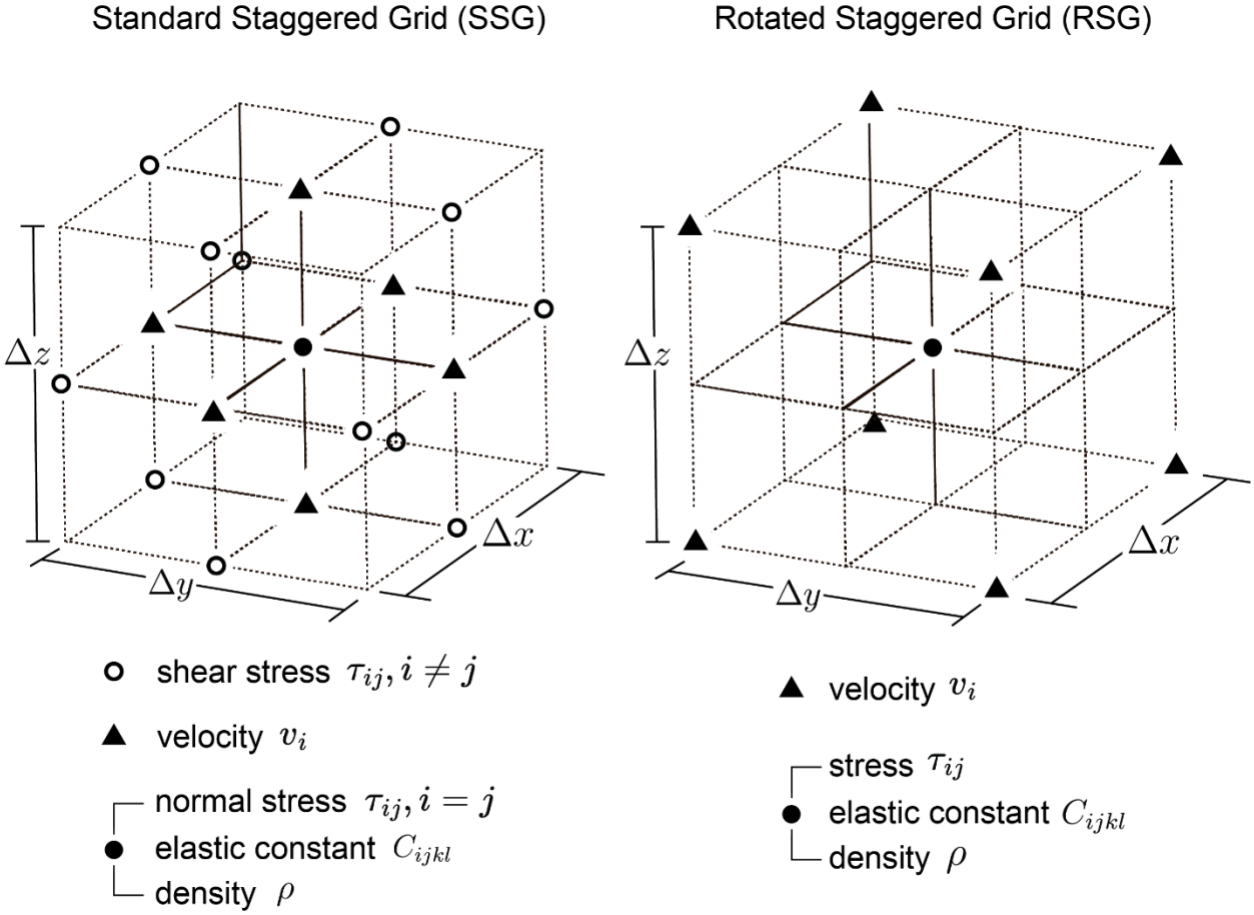

Figure 1. Sketches to compare the Standard Staggered Grid (SSG) and Rotated Staggered Grid (RSG). Symbols denotes variation of components and their distributed location within the grid.

With the consideration of varying location of difference components, a full expression of Equation 5 in SSG can be written as:

$$\Delta v_i = \begin{bmatrix} \Delta v_x(i_x+\frac{1}{2}, i_y, i_z) \\ \Delta v_y(i_x, i_y+\frac{1}{2}, i_z) \\ \Delta v_z(i_x, i_y, i_z+\frac{1}{2}) \end{bmatrix} = \frac{1}{\rho}\Delta t \begin{bmatrix} \frac{\partial \tau_{xx}(i_x,i_y,i_z)}{\partial x} + \frac{\partial \tau_{xy}\left(i_x+\frac{1}{2},i_y+\frac{1}{2},i_z\right)}{\partial y} + \frac{\partial \tau_{xz}\left(i_x+\frac{1}{2},i_y,i_z+\frac{1}{2}\right)}{\partial z} \\ \frac{\partial \tau_{yx}\left(i_x+\frac{1}{2},i_y+\frac{1}{2},i_z\right)}{\partial x} + \frac{\partial \tau_{yy}(i_x,i_y,i_z)}{\partial y} + \frac{\partial \tau_{yz}\left(i_x,i_y+\frac{1}{2},i_z+\frac{1}{2}\right)}{\partial z} \\ \frac{\partial \tau_{zx}\left(i_x+\frac{1}{2},i_y,i_z+\frac{1}{2}\right)}{\partial x} + \frac{\partial \tau_{zy}\left(i_x,i_y+\frac{1}{2},i_z+\frac{1}{2}\right)}{\partial y} + \frac{\partial \tau_{zz}(i_x,i_y,i_z)}{\partial z} \end{bmatrix} \quad (7)$$

Where i_x, i_y and i_z, are the indices of the grid cell. '+½' denotes it is staggered by shifting half grid spacing ($\Delta x/2$, $\Delta y/2$, $\Delta z/2$, respectively). A total of nine derivatives of τ_{ij} are evaluated to calculate the increment Δv_i and used to update the velocity wavefield v_i. Taking the calculation of $\Delta v_x(i_x+½, i_y, i_z)$ as example, assuming a second-order spatial derivative, the increment $\Delta v_x(i_x+½, i_y, i_z)$ equals to the sum of three derivatives:

$$\Delta v_x\left(i_x+\frac{1}{2}, i_y, i_z\right) = \frac{\partial \tau_{ij}}{\partial x_j} = \frac{\tau_{xx}(i_x+1,i_y,i_z)-\tau_{xx}(i_x,i_y,i_z)}{\Delta x} + \frac{\tau_{xy}\left(i_x+\frac{1}{2},i_y+\frac{1}{2},i_z\right)-\tau_{xy}\left(i_x+\frac{1}{2},i_y-\frac{1}{2},i_z\right)}{\Delta y} + \frac{\partial \tau_{xz}\left(i_x+\frac{1}{2},i_y,i_z+\frac{1}{2}\right)-\tau_{xz}\left(i_x+\frac{1}{2},i_y,i_z-\frac{1}{2}\right)}{\Delta z} \quad (8)$$

By taking advantage of the centered finite-difference, although the stress components τ_{ij} are staggered, the derivative with respect to three different orthogonal directions elegantly locate at the same point as Δv_x (e.g., $i_x+½, i_y, i_z$)

A higher even (2N) order spatial derivative can be expressed as:

$$\Delta v_x\left(i_x+\frac{1}{2}, i_y, i_z\right) = \frac{\partial \tau_{ij}}{\partial x_j} = \frac{\sum_{k=-N}^{N-1} B_k \tau_{xx}(i_x+1+k,i_y,i_z)}{\Delta x} + \frac{\sum_{k=-N}^{N-1} B_k \tau_{xy}\left(i_x+\frac{1}{2},i_y+\frac{1}{2}+k,i_z\right)}{\Delta y} + \frac{\sum_{k=-N}^{N-1} B_k \tau_{xz}\left(i_x+\frac{1}{2},i_y,i_z+\frac{1}{2}+k\right)}{\Delta z} \quad (9)$$

Where B_k is the finite-difference coefficient, if the derivative at point $x_i=0$ is desired, a distribution of neighboring points $\{(-N+0.5)\Delta x_i, \ldots, -1.5\Delta x_i, -0.5\Delta x_i, 0.5\Delta x_i, 1.5\Delta x_i, \ldots, (N-1+0.5)\Delta x_i\}$ are used, where $\Delta x_i=\Delta x, \Delta y, \Delta z$ respectively. The coefficient for orders 2 to 24 are computed and provided in Appendix A.

To update the wavefield τ_{ij} and obtain its increment $\Delta\tau_{ij}$ with Equation 6, the strain matrix is computed by firstly calculating $\partial v_i/\partial x_j$ then multiplied with stiffness C_{ijkl}. Similarly, an expression considering the component location in SSG to calculate the increment $\partial v_i/\partial x_j$ can be written as:

$$\frac{\partial v_i}{\partial x_j} = \begin{bmatrix} \frac{\partial v_x(i_x+\frac{1}{2},i_y,i_z)}{\partial x} & \frac{\partial v_x(i_x+\frac{1}{2},i_y,i_z)}{\partial y} & \frac{\partial v_x(i_x+\frac{1}{2},i_y,i_z)}{\partial z} \\ \frac{\partial v_y(i_x,i_y+\frac{1}{2},i_z)}{\partial x} & \frac{\partial v_y(i_x,i_y+\frac{1}{2},i_z)}{\partial y} & \frac{\partial v_y(i_x,i_y+\frac{1}{2},i_z)}{\partial z} \\ \frac{\partial v_z(i_x,i_y,i_z+\frac{1}{2})}{\partial x} & \frac{\partial v_z(i_x,i_y,i_z z+\frac{1}{2})}{\partial y} & \frac{\partial v_z(i_x,i_y,i_z+\frac{1}{2})}{\partial z} \end{bmatrix} \quad (10)$$

Taking the first component $\partial v_x/\partial x$ as example:

$$\frac{\partial v_x}{\partial x}\left(i_x, i_y, i_z\right) = \frac{v_x\left(i_x+\frac{1}{2},i_y,i_z\right)-v_x\left(i_x-\frac{1}{2},i_y,i_z\right)}{\Delta x} \quad (11)$$

By second-order centered finite-differencing, the resulted $\partial v_x/\partial x$ is at (i_x, i_y, i_z), so that normal strain at (i_x, i_y, i_z) is calculated and then multiplied by stiffness C_{ijkl} according to Equation 6 to obtain normal stress. Note (i_x, i_y, i_z) is where the normal stress is defined.

When calculating the shear strain, e.g. $\partial v_x/\partial y + \partial v_y/\partial x$ are evaluated respectively. Taking second-order centered finite-differencing as illustration:

$$\frac{\partial v_x}{\partial y}(i_x+\frac{1}{2}, i_y+\frac{1}{2}, i_z) = \frac{v_x\left(i_x+\frac{1}{2}, i_y+1, i_z\right) - v_x(i_x+\frac{1}{2}, i_y, i_z)}{\Delta y}$$

$$\frac{\partial v_y}{\partial x}\left(i_x+\frac{1}{2}, i_y+\frac{1}{2}, i_z\right) = \frac{v_y\left(i_x+1, i_y+\frac{1}{2}, i_z\right) - v_y\left(i_x, i_y+\frac{1}{2}, i_z\right)}{\Delta x} \quad (12)$$

Two derivatives $\partial v_x/\partial y$ and $\partial v_y/\partial x$ collocate at the point (i_x +½, i_y +½, i_z), where the shear strain is further evaluated at also. This can be extended to higher even spatial order too with the same method as Equation 9. Combining Equations 7 and 10, a total of 18 derivative operations are required to update the wavefield at each time step.

So far, the complete expression of Equation 6 in SSG with strain matrix ε_{ij} and τ_{ij} with their corresponding locations are:

$$\Delta\tau_{ij}\begin{pmatrix} i_x, i_y, i_z & i_x+\frac{1}{2}, i_y+\frac{1}{2}, i_z & i_x+\frac{1}{2}, i_y, i_z+\frac{1}{2} \\ i_x+\frac{1}{2}, i_y+\frac{1}{2}, i_z & i_x, i_y, i_z & i_x, i_y+\frac{1}{2}, i_z+\frac{1}{2} \\ i_x+\frac{1}{2}, iy, i_z+\frac{1}{2} & i_x, i_y+\frac{1}{2}, i_z+\frac{1}{2} & i_x, i_y, i_z \end{pmatrix} \cong$$

$$C_{ijkl}(i_x, i_y, i_z)\,\varepsilon_{kl}\begin{pmatrix} i_x, i_y, i_z & i_x+\frac{1}{2}, i_y+\frac{1}{2}, i_z & i_x+\frac{1}{2}, i_y, i_z+\frac{1}{2} \\ i_x+\frac{1}{2}, i_y+\frac{1}{2}, i_z & i_x, i_y, i_z & i_x, i_y+\frac{1}{2}, i_z+\frac{1}{2} \\ i_x+\frac{1}{2}, iy, i_z+\frac{1}{2} & i_x, i_y+\frac{1}{2}, i_z+\frac{1}{2} & i_x, i_y, i_z \end{pmatrix} \quad (13)$$

Theoretically, the components of wavefield variable e.g. τ_{ij}, ε_{ij} and v_j are continuous functions. Even within an unit cell and square root of half spacing apart, the wavefield $\varepsilon_{ij}(i_x+½, i_y+½, i_z)$ is different than $\varepsilon_{ij}(i_x, i_y, i_z)$. Although the SSG method's inherent design assures improved accuracy relative to non-staggered gird without necessitating adjustments in spatial spacing or the order of the derivative, a notable limitation arises due to this half spacing error, when it is performed on anisotropic media. Igel et al., (1995) overcome that issue by interpolation operation in Hooke's law (Equation 3) using Fourier Transform. For the off-diagonal elements, either the elasticity tensor C_{ijkl} or the wavefield ε_{ij} and τ_{ij} need to be interpolated to avoid error introduced by half spacing shift. However, this comes with increased computational costs.

The rotated staggered grid (RSG) can effectively cure this problem. In RSG scheme, the particle velocity components v_i are placed at vertices and stiffnesses C_{ijkl}, stresses t_{ij}, and density ρ components are at the center of the cell (Figure 1). The RSG derivative is calculated from four rotated directions diagonally across the center of the unit cell for 3D or two diagonal directions for 2D. For generality, all the following expressions are three-dimensional. The positive directions in the rotated coordinates are:

$$\widetilde{d_1} = \frac{\Delta x}{\Delta l}\mathbf{x} + \frac{\Delta y}{\Delta l}\mathbf{y} + \frac{\Delta z}{\Delta l}\mathbf{z}$$

$$\widetilde{d_2} = \frac{\Delta x}{\Delta l}\mathbf{x} + \frac{\Delta y}{\Delta l}\mathbf{y} - \frac{\Delta z}{\Delta l}\mathbf{z} \quad (14)$$

$$\widetilde{d_3} = \frac{\Delta x}{\Delta l}\mathbf{x} - \frac{\Delta y}{\Delta l}\mathbf{y} + \frac{\Delta z}{\Delta l}\mathbf{z}$$

$$\widetilde{d_4} = \frac{\Delta x}{\Delta l}\mathbf{x} - \frac{\Delta y}{\Delta l}\mathbf{y} - \frac{\Delta z}{\Delta l}\mathbf{z}$$

in case of non-equilateral grid, where Δx, Δy, and Δz are the grid spacings in three coordinate directions **x** , **y** , **z** respectively. $\Delta l = \sqrt[2]{\Delta x^2 + \Delta y^2 + \Delta z^2}$

The derivative operator in original coordinates can be replaced with a linear combination of the four rotated coordinates:

$$\frac{\partial}{\partial x} \approx \frac{\Delta l}{4\Delta x}\left(\frac{\partial}{\partial \widetilde{d_1}} + \frac{\partial}{\partial \widetilde{d_2}} + \frac{\partial}{\partial \widetilde{d_3}} + \frac{\partial}{\partial \widetilde{d_4}}\right)$$

$$\frac{\partial}{\partial y} \approx \frac{\Delta l}{4\Delta y}\left(\frac{\partial}{\partial \widetilde{d_1}} + \frac{\partial}{\partial \widetilde{d_2}} - \frac{\partial}{\partial \widetilde{d_3}} - \frac{\partial}{\partial \widetilde{d_4}}\right) \qquad (15)$$

$$\frac{\partial}{\partial z} \approx \frac{\Delta l}{4\Delta z}\left(\frac{\partial}{\partial \widetilde{d_1}} - \frac{\partial}{\partial \widetilde{d_2}} + \frac{\partial}{\partial \widetilde{d_3}} - \frac{\partial}{\partial \widetilde{d_4}}\right)$$

After calculating under this rotated staggered scheme, both normal strain and shear strain components will always locate at the center of the unit cell, then multiplied by the collocated elasticity tensor C_{ijkl} to calculate the increment of stress $\Delta\tau_{ij}$.

To update the v_i, a full expression of Equation 5 in RSG with the corresponding location of each component written out is:

$$\Delta v_i = \begin{bmatrix}\Delta v_x \\ \Delta v_y \\ \Delta v_z\end{bmatrix}\left(i_x + \frac{1}{2}, i_y + \frac{1}{2}, i_z + \frac{1}{2}\right) = \begin{bmatrix}\frac{\partial \tau_{xx}(i_x,i_y,i_z)}{\partial x} + \frac{\partial \tau_{xy}(i_x,i_y,i_z)}{\partial y} + \frac{\partial \tau_{xz}(i_x,i_y,i_z)}{\partial z} \\ \frac{\partial \tau_{yx}(i_x,i_y,i_z)}{\partial x} + \frac{\partial \tau_{yy}(i_x,i_y,i_z)}{\partial y} + \frac{\partial \tau_{yz}(i_x,i_y,i_z)}{\partial z} \\ \frac{\partial \tau_{zx}(i_x,i_y,i_z)}{\partial x} + \frac{\partial \tau_{zy}(i_x,i_y,i_z)}{\partial y} + \frac{\partial \tau_{zz}(i_x,i_y,i_z)}{\partial z}\end{bmatrix} \qquad (16)$$

Because there is no distinction in terms of location between normal stress and shear stress in RSG scheme. The evaluations on all nine derivatives are similar. To illustrate, we take the second spatial derivative $\partial \tau_{xx}/\partial x$ as example, and it can be expanded easily for all other evaluations:

$$\frac{\partial \tau_{xx}}{\partial x}\left(i_x+\frac{1}{2}, i_y+\frac{1}{2}, i_z+\frac{1}{2}\right) =$$

$$\frac{\Delta l}{4\Delta x}\left\{\begin{array}{l}\frac{\tau_{xx}(i_x+1,i_y+1,i_z+1)-\tau_{xx}(i_x,i_y,i_z)}{\Delta l}+\frac{\tau_{xx}(i_x+1,i_y+1,i_z)-\tau_{xx}(i_x,i_y,i_z+1)}{\Delta l}\\ +\frac{\tau_{xx}(i_x+1,i_y,i_z+1)-\tau_{xx}(i_x,i_y+1,i_z)}{\Delta l}+\frac{\tau_{xx}(i_x+1,i_y,i_z)-\tau_{xx}(i_x,i_y+1,i_z+1)}{\Delta l}\end{array}\right\} \quad (17)$$

As it shown by centered finite-difference, the derivative from four direction ends at vertices (i_x+½, i_y+½, i_z+½). Analogous to Equation 9, the centered finite-difference can be induced to higher even spatial order.

Similarly, to update τ_{ij} with Equation 6, the strain component is obtained by firstly calculating $\partial v_i/\partial x_j$. A complete expression for $\partial v_i/\partial x_j$ with the corresponding location of each component in RSG is:

$$\frac{\partial v_i}{\partial x_j}\left(i_x, i_y, i_z\right) = \begin{bmatrix} \frac{\partial v_x\left(i_x+\frac{1}{2},i_y+\frac{1}{2},i_z+\frac{1}{2}\right)}{\partial x} & \frac{\partial v_x\left(i_x+\frac{1}{2},i_y+\frac{1}{2},i_z+\frac{1}{2}\right)}{\partial y} & \frac{\partial v_x\left(i_x+\frac{1}{2},i_y+\frac{1}{2},i_z+\frac{1}{2}\right)}{\partial z} \\ \frac{\partial v_y\left(i_x+\frac{1}{2},i_y+\frac{1}{2},i_z+\frac{1}{2}\right)}{\partial x} & \frac{\partial v_y\left(i_x+\frac{1}{2},i_y+\frac{1}{2},i_z+\frac{1}{2}\right)}{\partial y} & \frac{\partial v_y\left(i_x+\frac{1}{2},i_y+\frac{1}{2},i_z+\frac{1}{2}\right)}{\partial z} \\ \frac{\partial v_z\left(i_x+\frac{1}{2},i_y+\frac{1}{2},i_z+\frac{1}{2}\right)}{\partial x} & \frac{\partial v_z\left(i_x+\frac{1}{2},i_y+\frac{1}{2},i_z+\frac{1}{2}\right)}{\partial y} & \frac{\partial v_z\left(i_x+\frac{1}{2},i_y+\frac{1}{2},i_z+\frac{1}{2}\right)}{\partial z} \end{bmatrix} \quad (18)$$

Therefore, the strain components ε_{ij} are also located at (i_x, i_y, i_z) with elasticity tensor C_{ijkl} to guarantee the stability of wave propagation.

3 Implementation

3.1 Symbolic computation with Devito

In the constantly evolving landscape of hardware technology, the challenge for physicists is not just to understand and model physical phenomena but also to continually adapt and optimize their code to leverage the latest computational platforms. This dual demand can divert significant time and effort away from primary research objectives.

Devito addresses this challenge by offering an interface that bridges high-level physics modeling with the intricacies of modern computation (Kukreja et al., 2016; Louboutin et al., 2019). Through its symbolic interface in Python, physicists can focus on the physics and mathematics, while Devito ensures compatibility and optimization for various hardware platforms, translating the high-level code to low-level C code suitable for CPUs or CUDA code for GPUs. More information about Devito may be found in Luporini et al., (2020).

Symbolic computing's inherent strength is its ability to manipulate and optimize equations symbolically before numerical values are even introduced. This pre-emptive optimization allows for more efficient computational strategies and minimizing redundancy. For instance, by analyzing the symbolic form of equations, redundancies can be identified and eliminated, and common sub-expressions can be factored out. This not only streamlines the actual computation but also paves the way for more sophisticated optimizations, such as adaptive mesh refinement or domain decomposition. Thus, the overarching advantage of Devito and symbolic computing is that it allows scientists to remain centered on their primary research, confident in the knowledge that the computational aspects are being managed efficiently and effectively.

3.2 Arithmetic optimization

The wave propagation is based on computing and update two wavefields τ_{ij} and v_i, (Equations 5 and 6) at each the progressing time. Nine derivative operations are necessary to update each of v_i (Equation 7). Another nine derivative operations of t_{ij} with respect to x_i are also required, (Equation 10), resulting in a total of 18 derivative operation in 3D. In contrast, RSG takes four times of derivative operation (Equation 15), because each derivative in the original coordinate consists of four derivative operations in rotated coordinates, see Equations 14 and 15, ending up with a total of 72 operations. To reduce this number, a minor but useful modification can be made. When computing the derivative of v_i:

$$\frac{\partial v_i}{\partial x_j} = \begin{bmatrix} \frac{dl}{4dx}\left(\frac{\partial}{\partial \widetilde{d_1}} + \frac{\partial}{\partial \widetilde{d_2}} + \frac{\partial}{\partial \widetilde{d_3}} + \frac{\partial}{\partial \widetilde{d_4}}\right) v_x & \frac{dl}{4dy}\left(\frac{\partial}{\partial \widetilde{d_1}} + \frac{\partial}{\partial \widetilde{d_2}} - \frac{\partial}{\partial \widetilde{d_3}} - \frac{\partial}{\partial \widetilde{d_4}}\right) v_x & \frac{dl}{4dz}\left(\frac{\partial}{\partial \widetilde{d_1}} - \frac{\partial}{\partial \widetilde{d_2}} + \frac{\partial}{\partial \widetilde{d_3}} - \frac{\partial}{\partial \widetilde{d_4}}\right) v_x \\ \frac{dl}{4dx}\left(\frac{\partial}{\partial \widetilde{d_1}} + \frac{\partial}{\partial \widetilde{d_2}} + \frac{\partial}{\partial \widetilde{d_3}} + \frac{\partial}{\partial \widetilde{d_4}}\right) v_y & \frac{dl}{4dy}\left(\frac{\partial}{\partial \widetilde{d_1}} + \frac{\partial}{\partial \widetilde{d_2}} - \frac{\partial}{\partial \widetilde{d_3}} - \frac{\partial}{\partial \widetilde{d_4}}\right) v_y & \frac{dl}{4dz}\left(\frac{\partial}{\partial \widetilde{d_1}} - \frac{\partial}{\partial \widetilde{d_2}} + \frac{\partial}{\partial \widetilde{d_3}} - \frac{\partial}{\partial \widetilde{d_4}}\right) v_y \\ \frac{dl}{4dx}\left(\frac{\partial}{\partial \widetilde{d_1}} + \frac{\partial}{\partial \widetilde{d_2}} + \frac{\partial}{\partial \widetilde{d_3}} + \frac{\partial}{\partial \widetilde{d_4}}\right) v_z & \frac{dl}{4dy}\left(\frac{\partial}{\partial \widetilde{d_1}} + \frac{\partial}{\partial \widetilde{d_2}} - \frac{\partial}{\partial \widetilde{d_3}} - \frac{\partial}{\partial \widetilde{d_4}}\right) v_z & \frac{dl}{4dz}\left(\frac{\partial}{\partial \widetilde{d_1}} - \frac{\partial}{\partial \widetilde{d_2}} + \frac{\partial}{\partial \widetilde{d_3}} - \frac{\partial}{\partial \widetilde{d_4}}\right) v_z \end{bmatrix} \quad (19)$$

Apparently, there are 12 derivative operations to calculate each row. But only four derivatives $\frac{\partial v_i}{\partial \widetilde{d_1}}, \frac{\partial v_i}{\partial \widetilde{d_2}}, \frac{\partial v_i}{\partial \widetilde{d_3}}, \frac{\partial v_i}{\partial \widetilde{d_4}}$ are needed. The numerical values of these four derivatives can be cached and reused their combinations to calculate the entire row. By employing this modification, the derivative operation times can be reduced to 12 (4 derivative/row times 3 rows) instead of 36 (12 derivative/row times 3 rows) originally. This modification trims the total operations from 72 to 48.

In practical tests, a 3D model of dimensions (251, 251, 251) with a 20-point boundary zone, runs on a computer cluster utilizing a single node with 24 cores, showed a reduction in computation time from approximately 343-356 seconds to between 303-316 seconds, representing an improvement of around 11%.

Another optimization is to leverage the inherent symmetries in the stress and strain tensors to optimize computations. Specifically, the conservation of angular momentum ensures that $\tau_{ij} = \tau_{ji}$, and the principle of rigid body rotations confirms $\varepsilon_{ij} = \varepsilon_{ji}$. With these symmetries in place, Hooke's Law can be compactly represented using a 6×6 Voigt matrix form, rather than a full 4th order elasticity tensor. For clarity, Hooke's Law in the Voigt notation is given by:

$$\begin{bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{zz} \\ \sigma_{yz} \\ \sigma_{zx} \\ \sigma_{xy} \end{bmatrix} = \begin{bmatrix} C_{11} & C_{12} & C_{13} & C_{14} & C_{15} & C_{16} \\ C_{12} & C_{22} & C_{23} & C_{24} & C_{25} & C_{26} \\ C_{13} & C_{23} & C_{33} & C_{34} & C_{35} & C_{36} \\ C_{14} & C_{24} & C_{34} & C_{44} & C_{45} & C_{46} \\ C_{15} & C_{25} & C_{35} & C_{45} & C_{55} & C_{56} \\ C_{16} & C_{26} & C_{36} & C_{46} & C_{56} & C_{66} \end{bmatrix} \begin{bmatrix} \epsilon_{xx} \\ \epsilon_{yy} \\ \epsilon_{zz} \\ \epsilon_{yz} + \epsilon_{zy} \\ \epsilon_{zx} + \epsilon_{xz} \\ \epsilon_{xy} + \epsilon_{yx} \end{bmatrix} \quad (20)$$

When the anisotropy is tilted, it's crucial to appropriately rotate the Voigt matrix to ensure the material's coordinate is correctly represented within the global coordinate system (Auld, 1973). According to our test, adopting the Voigt notation for Hooke's law, rather than the 4th order tensor form, can significantly diminishes repetitive floating-point operations in modeling, offering computational efficiency. Through our testing, we found that implementations utilizing matrix multiplication exhibit enhanced computational speed compared to those employing tensor multiplication. However, the degree of this speed advantage is contingent upon varying computational conditions and the dimensions of the model under consideration.

3.3 Source excitation

Although the computing domain is discretized into grids, the source/receiver points may not always align perfectly with a grid node. In such cases, coefficients weighted by the distance from the source location to its nearest adjacent nodes within the cell (4 nodes for 2D, 8 nodes for 3D) are used to position the source. (Virieux et al., 2012)

Another problem appears when the source is injected at the node point, one sub-grids is incomplete and uncoupled with another, thus, causes alternating amplitudes between neighboring grid points and appears as checkerboard pattern (cite figure). To avoid this, a source can be distributed over several adjacent nodes. (Hustedt et al., 2004; Virieux et al., 2012). However, smearing out the source point inject to more grids may downgrade the accuracy. In our implementation to overcome this and meanwhile, we minimize the source point dislocation and distortion of the wavelet. Instead of the single source point itself (i_{sx}, i_{sz}), where. i_s is the index of the source node, eight neighboring points around the source point for 2D $(i_{sx}+1, i_{sz}+1)$, $(i_{sx}, i_{sz}+1)$, $(i_{sx}-1, i_{sz}+1)$, $(i_{sx}+1, i_{sz})$, $(i_{sx}-1, i_{sz})$, $(i_{sx}-1, i_{sz}-1)$, $(i_{sx}, i_{sz}-1)$, $(i_{sx}+1, i_{sz}-1)$ are excited. For 3D, we also inject eight neighboring points (including source point itself) to fulfill all the stress component in the grid cell defined by the one grid positive shift in three dimensions. (i_{sx}, i_{sy}, i_{sz}), $(i_{sx}+1, i_{sy}, i_{sz})$, $(i_{sx}, i_{sy}+1, i_{sz})$, $(i_{sx}, i_{sy}, i_{sz}+1)$, $(i_{sx}+1, i_{sy}+1, i_{sz})$, $(i_{sx}+1, i_{sy}, i_{sz}+1)$, $(i_{sx}, i_{sy}+1, i_{sz}+1)$, $(i_{sx}+1, i_{sy}+1, i_{sz}+1)$. This source excitation will lead to dislocation less than single spatial spacing.

Figure 2 illustrates how increasing the spatial resolution can somewhat but not entirely ameliorate the checkerboard effect. Distributing the source injection effectively addresses this issue.

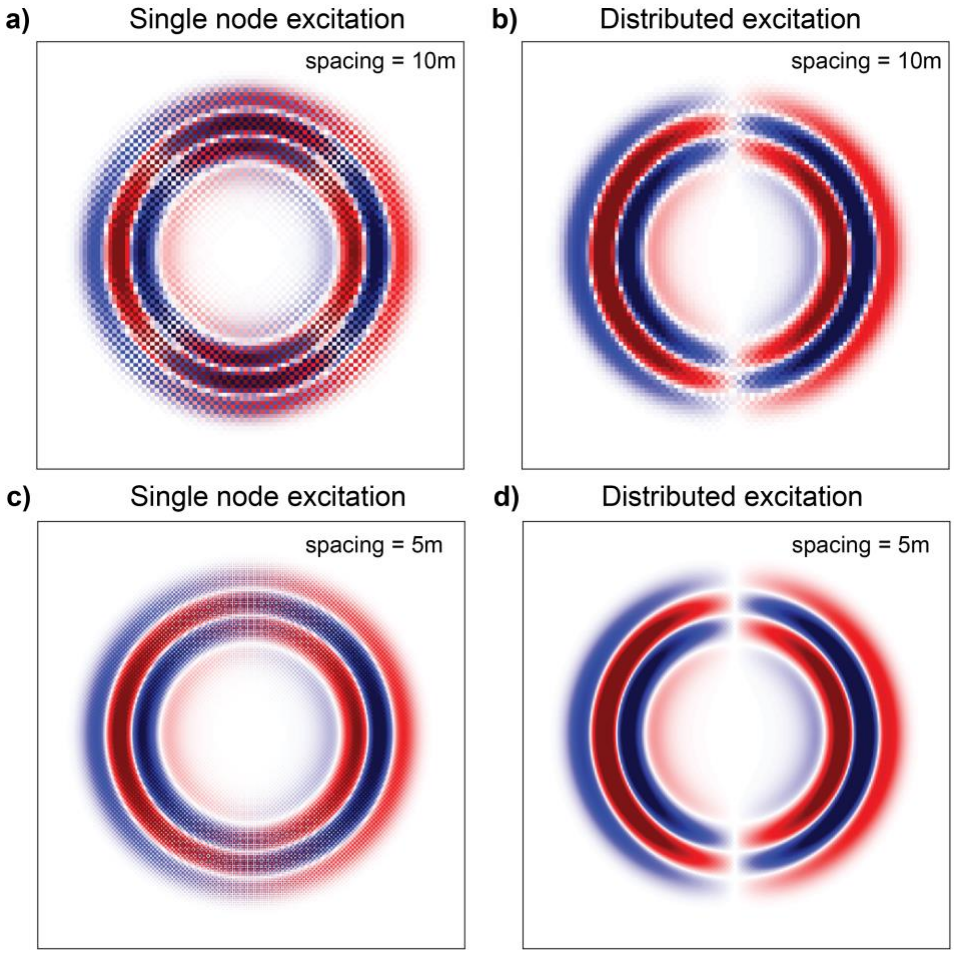

Figure 2. Snapshot of the vertical velocity (v_z) wavefield in a 1000m-by-1000m extent with alternative source excitation methods. (a) single node excitation with spatial spacing = 10m. (b) distributed source excitation with spatial spacing = 10m. (c) single node excitation with spatial spacing = 5m. (d) distributed source excitation with spatial spacing = 5m.

4 Results

4.1 Wavefield stability in anisotropic media

To evaluate the stability of RSG in modeling intricate anisotropic media in comparison to SSG, we analyzed wavefield snapshots from both approaches across varying media complexities. Four distinct configurations, each with increasing anisotropic intricacy, were examined. The first configuration uses the simplest isotropic media. The second configuration uses a vertical transverse isotropic media (VTI) and the third configuration rotates the same VTI media to be

363 tilted transverse isotropic media (TTI). The fourth configuration uses an alpha quartz, whose a
364 trigonal symmetry, from resonance ultrasound spectroscopy (Ogi et al., 2006). The elasticity
365 parameters, density and geometry are listed in Table 1. In all configurations, model size is 251 ×
366 251 × 251 with 4-meter spacing with eighth spatial order. The sources were 40Hz central frequency
367 Ricker wavelet, injected via the M_{zz} component exclusively to highlight shear wave behavior
368 (Figure 3a). The snapshot of the wavefields at 400ms for SSG and RSG, respectively, are shows
369 in Figure 3b.

370 **Table 1.** Elasticity parameters, density and geometry of the four configurations

Name	ISO	VTI	TTI	α-quartz
Density (g/cm^3) ρ	2.6	2.6	2.6	2.6
Dip angle θ (°)	---	---	60	---
Strike angle ψ (°)	----	---	30	---
C11 (GPa)	98.2	98.2	98.2	86.7
C22 (GPa)	98.2	98.2	98.2	86.7
C33 (GPa)	98.2	80.6	80.6	105.5
C12 (GPa)	36.9	24.0	24.0	6.9
C13 (GPa)	36.9	10.33	10.33	11.9
C23 (GPa)	36.9	10.33	10.33	11.9
C44 (GPa)	30.65	30.65	30.65	58.1
C55 (GPa)	30.65	30.65	30.65	58.1
C66 (GPa)	30.65	37.1	37.1	39.9
C14 (GPa)	0.	0.	0.	-18.0
C15 (GPa)	0.	0.	0.	0.
C16 (GPa)	0.	0.	0.	0.
C24 (GPa)	0.	0.	0.	18.0
C25 (GPa)	0.	0.	0.	0.
C26 (GPa)	0.	0.	0.	0.
C34 (GPa)	0.	0.	0.	0.
C35 (GPa)	0.	0.	0.	0.
C36 (GPa)	0.	0.	0.	0.
C45 (GPa)	0.	0.	0.	0.
C46 (GPa)	0.	0.	0.	0.
C56 (GPa)	0.	0.	0.	-18.0

371

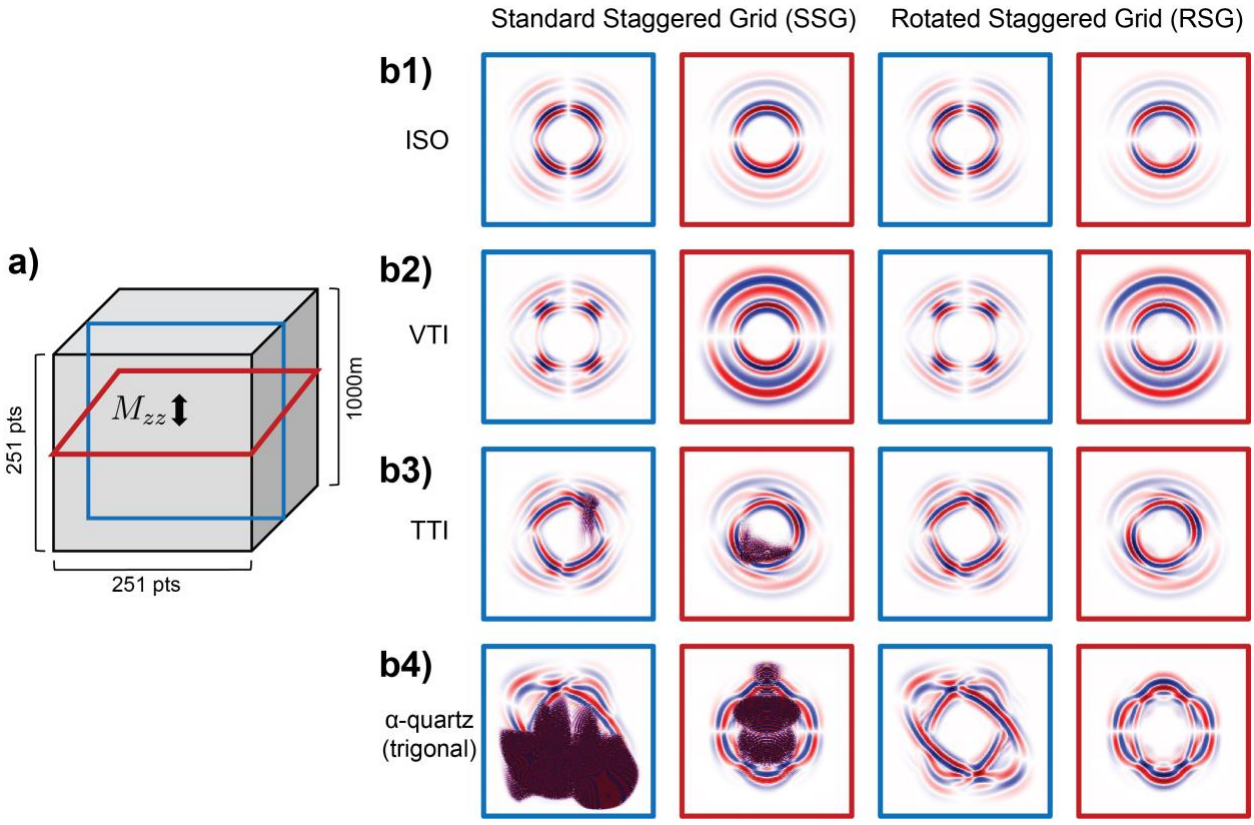

Figure 3. (a) Sketch shows the cubic modeling domain for the wavefield stability test. Spatial spacing is 4m. The red frame outlines the satellite view on the middle slice in z-direction. The blue frame outlines the cross-section view on the middle slice in x-direction. (b) Wavefield snapshots (vertical velocity component) of the 3D modeling at 400ms using the four configurations using SSG and RSG, respectively. (b1-b4) corresponds to the four configurations ISO, VTI, TTI and α-quartz, respectively, with parameters listed in Table 1.

4.2 Numerical dispersion

Numerical dispersion is an artifact that arises in the process of approximating continuous wave equations using discrete numerical methods. When finite-difference method discretizes continuous equation for computational purposes, if the spatial sampling is not inadequately small, the phase velocity of high frequency component with shorter wavelength can be misrepresented. This will lead to the phenomenon where waves travel at different velocities for different

frequencies which is not expected for body wave propagation in a pure elastic medium. Numerical dispersion is different than physical dispersion caused by material properties.

Numerical dispersion can be mitigated by increasing the finite-difference order. Many studies have derived the numerical dispersion error with various spatial order using dispersion relation of RSG analytically (Bohlen & Saenger, 2006, p. 33; Hustedt et al., 2004; Krüger et al., 2005; O'Brien, 2010; Saenger et al., 2000), and the solution shows the original direction x_i (where $i=1,2,3$ corresponds to **x-, y-** and **z-** axis) of SSG and rotated axis d_i (where $i=1,2,3,4$) of RSG are more inclined to disperse numerically. While these studies provide quantitative measures of dispersion errors, often in the form of percentages, these numerical values don't always capture the practical impact of the discrepancy. Direct comparison of waveforms can offer a more intuitive and representative understanding of the actual differences arising from numerical dispersion.

In the numerical analysis, we used a model configured for isotropic media with parameters: vp = 4km/s, vs = 2km/, and density = 2.6 g/cm^3. A Ricker wavelet with 50 Hz central frequency, sufficiently encompass the typical range for most seismic surveys. The spatial increments Δx and Δz are both set to 8 meters, covering a domain of 1000 meters in width and depth. The comparisons are between two directions, whose the most inclination to be numerically dispersive: $\boldsymbol{x_i}$, e.g. (1,0) in 2D or (1,0,0) in 3D direction and $\boldsymbol{d_i}$, e.g. (1,1) in 2D and (1,1,1) in 3D. The receiver is 300 m away from the source in the corresponding direction. Various even number spatial orders are tested. To obtain the same waveform for better comparison, instead of velocity in specific direction, we plot the divergence of the velocity, which is the volumetric strain rate, in Figures 4 and 5 for 2D and 3D respectively.

In the 2D analysis, the 4th order SSG displays minor numerical dispersion. It's invisible at 45 ° (rotated axis direction) in Figure 4c and slightly discernible at 90 ° (x-direction) in Figure 4a. However, the 4th order RSG demonstrates a contrasting behavior: the dispersion is mild at 90° (Figure 4b) but significantly pronounced at 45° (Figure 4d). To adequately counter this dispersion in RSG, an upgrade to the 8th order spatial domain is necessary, which reduces the dispersion to a barely noticeable level. For the dispersion to be imperceptible, a 16th order configuration is required.

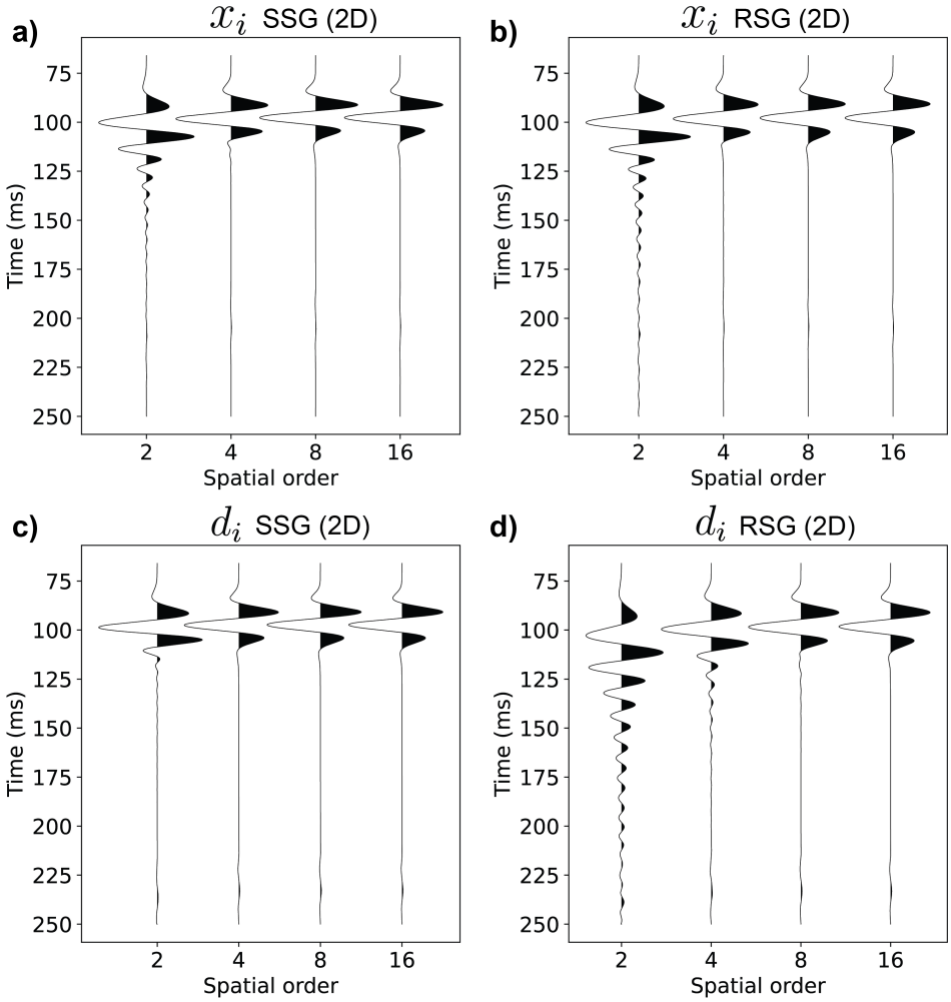

Figure 4. Result waveform from 2D SSG and RSG modeling at two selected directions x_i and d_i with increasing spatial order, where $i=1, 2$. (a) SSG waveform at the receiver location along the x-, z-axis. (b) RSG waveform at the receiver location along the x-, z-axis. (c) SSG waveform at the receiver location along the 45° rotated axis (±1, ±1). (d) RSG waveform at the receiver location along the 45° rotated axis (±1, ±1).

In the 3D analysis, the 2nd order RSG exhibits significant numerical dispersion, attributed to the reduction in nodes per wavelength divided by a factor of √3. The 2nd order 3D SSG performs adequately in the rotated direction (Figure 5c) but displays noticeable numerical dispersion (Figure 5a). At the 4th order, the 3D SSG becomes satisfactory in both directions, while RSG's performance remains severely dispersive. Even with spatial orders as high as 24, RSG still struggles in the rotated direction in Figure 5d.

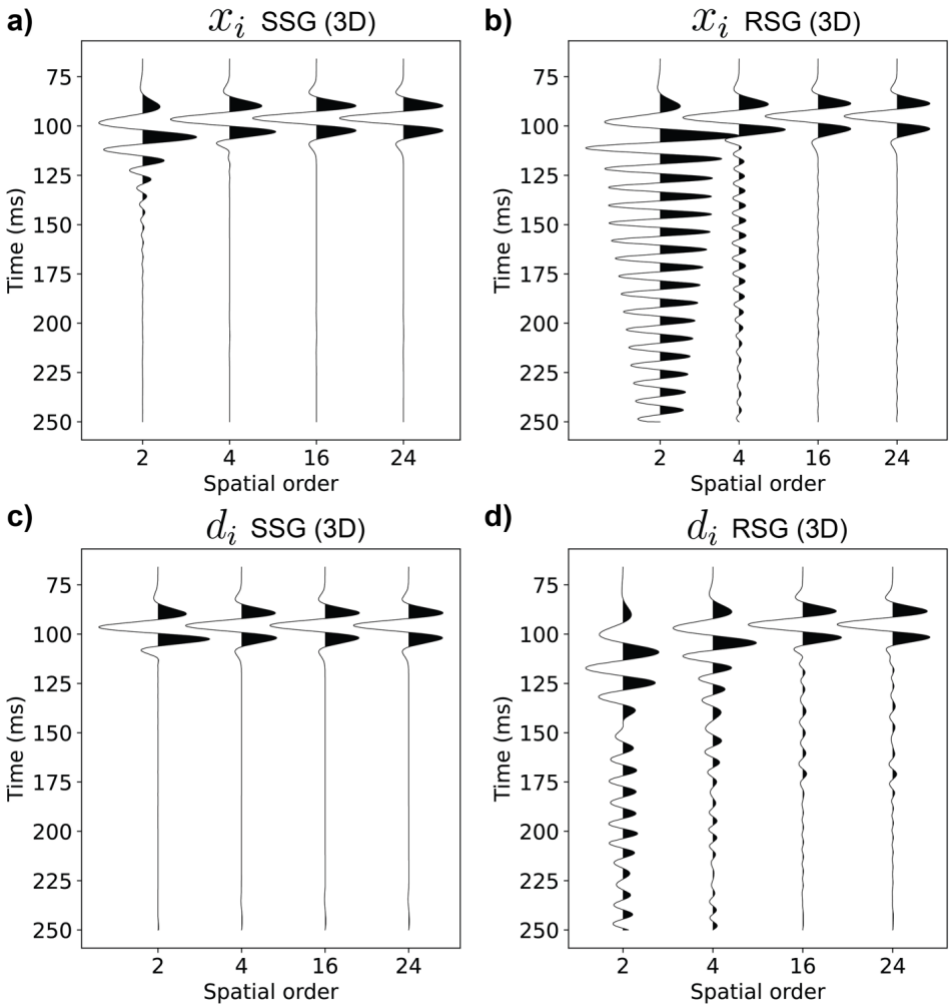

Figure 5. Resulted waveform from 3D SSG and RSG modeling at two selected directions x_i *(i=1,2,3)* and d_i *(i=1,2,3,4)* with increasing spatial order. (a) SSG waveform at the receiver location along the x-, y-, z-axis. (b) RSG waveform at the receiver location along the x-, y-, z-axis. (c) SSG waveform at the receiver location along the rotated axis (±1, ±1, ±1). (d) RSG waveform at the receiver location along the rotated axis (±1, ±1, ±1).

4.3 Performance comparison with SSG

The total computational time for seismic modeling can generally be divided into two main components: the stencil setup time, which pertains to the preparation of the computational stencil, and the actual computation time, which accounts for the primary simulation. To assess the

performance of our optimized RSG solver, we compared its simulation durations with those of an unoptimized SSG solver, which uses the 4th order elasticity tensor instead of the Voigt notation.

For the 2D model, our simulation covered a domain of 1000 meters by 1000 meters. This domain was meshed with a variable number of nodes in 2D, leading to distinct values of Δx_i. On the other hand, the 3D model was set within a fixed 1000-meter cube, characterized by $\Delta x_i = 10m$. Both models utilized an explosive 30Hz-central-frequency Ricker wavelet as the source and operated with a time interval of $\Delta t = 0.05ms$. We tested the solvers' performance, both in 2D and 3D, across various spatial orders.

From a computational perspective, the 2D tests were executed using 10 out of 64 available cores, while the 3D tests utilized 48 out of a total of 128 cores. All tests were conducted on a single computing node, powered by the AMD Epyc 7763 “Milan” CPUs operating at 2.2GHz. Given the inherent memory demands of 3D modeling, we also conducted a comparative analysis of memory usage for both solvers in the 3D scenarios. The results of these comparisons, including stencil setup times, computation durations, and memory consumption, are presented in the Figure 6.

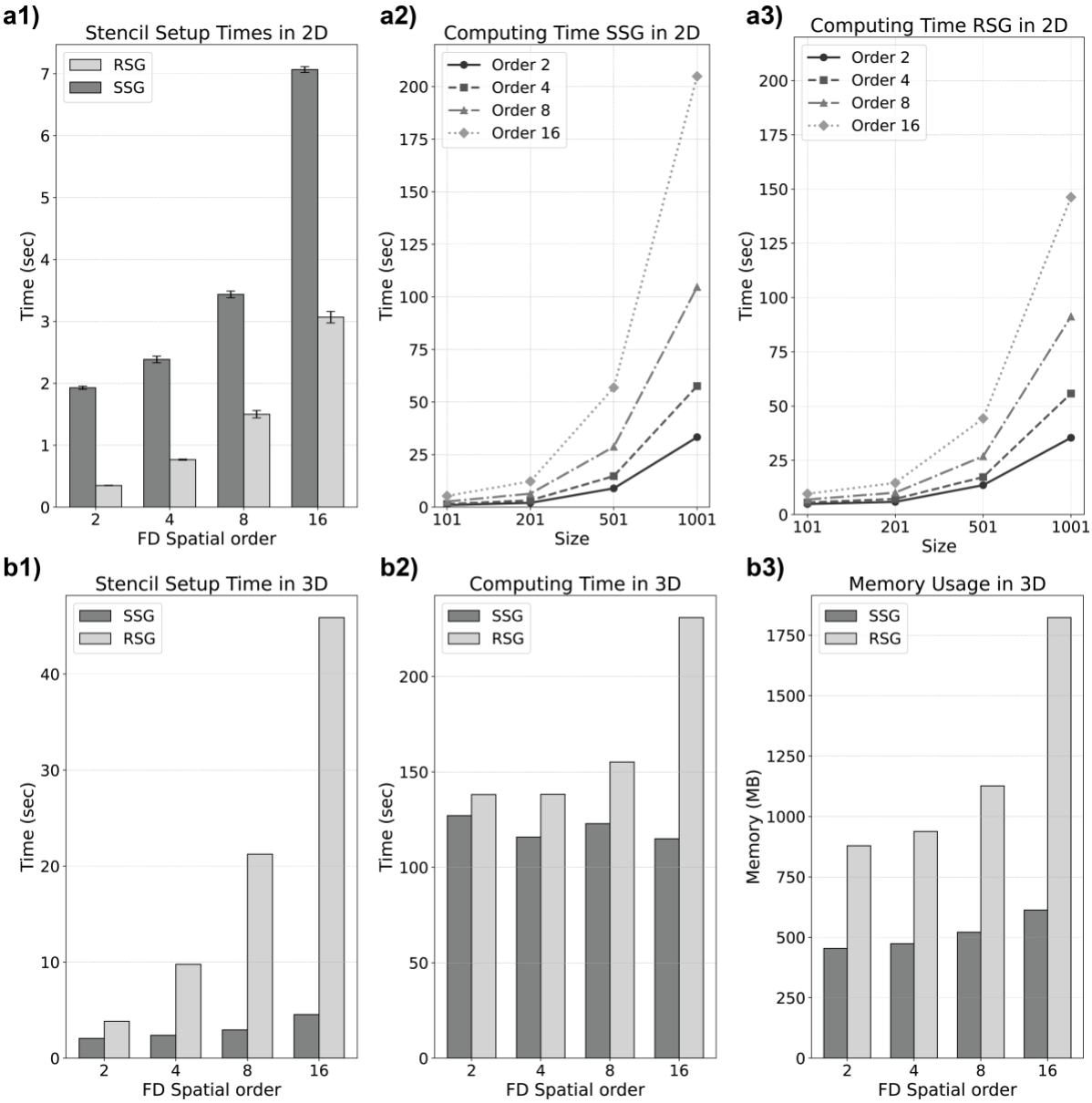

Figure 6. (a) 2D modeling with various model size and spatial order. Size denotes of points utilized in one direction for a square domain. (a1) Time spent on setup the 2D FD stencil as spatial order increases for SSG and RSG respectively. (a2) Time spent on computing SSG as size increases with varying spatial order. (a3) Time spent on computing RSG as size increases with varying spatial order. (b) 3D modeling with a fixed cube model in size (101×101×101) and varying spatial order. (b1) Time spent on setup the 3D FD stencil as spatial order increases for SSG and RSG respectively. (b2) Computing time comparison between SSG and RSG as spatial order increases. (b3) Memory usage comparison between SSG and RSG as spatial order increases.

5 Discussion

The initial stability tests (Figure 3) in isotropic and VTI media showed promising results for both RSG and SSG, primarily because the material's symmetry axis remains in alignment with the coordinate axes (Igel et al., 1995). However, the challenges observed in the TTI medium setup underscore the vulnerabilities of the SSG approach, especially when the symmetry axis is not aligned with the coordinate axes. The final test in the *a*-quartz environment further highlighted the robustness of RSG in handling complex anisotropic media, in stark contrast to the SSG's inability to manage the intricacies of such structures.

The rotated staggered grid (RSG) scheme benefits with the stably modeling complicated seismic anisotropy, on the other hand, suffers more from numerical dispersion, thus has less computational efficiency. While numerical dispersion is impossible entirely eliminate, it can certainly be minimized by (i) using finer grid spacings (nodes per wavelength), (ii) using higher order derivativse to increase accuracy, and (iii) reducing the central frequency of the source. (Saenger et al., 2000). However, lowering the frequency of the source means sacrifice both spatial and temporal resolution. RSG has more severe numerical dispersion than SSG, because its inherently number of grid points per wavelength is divided by a factor of square root of 2 for 2D, and square root of 3 for 3D (Saenger & Bohlen, 2004). To suppress the numerical dispersion, the first strategy (i), refining the spatial grid by decreasing the spacing is an effective in lessening numerical dispersion. However, in many practical scenarios, especially with 3D modeling, computational resources are constrained. It's crucial to recognize that even a modest reduction in grid size can escalate the computational cost exponentially. Since the spatial resolution increases, not only do the number of grid cells rise sharply in 3D, but the time step typically also needs to be made smaller which further intensifying the computational burden. The second (ii) of enhancing the spatial order emerges as an efficient strategy especially when computational power is limited. This method enhances the precision of derivative evaluations. It does more calculations, but it does not significantly increase memory consumption, offering a balance for scenarios with limited memory but ample processing power. For the approach (iii), lower the frequency content compromises resolution and interaction with small geological features, e.g., thin bed, small-scale

faults or fractures. With a major focus on approach (ii) and based on the dispersion observation on the waveform (Figures 4 and 5), we can infer:

1. RSG performs commendably in 2D. The reduction in nodes per wavelength divided by $\sqrt{2}$ may be countered by increasing the spatial order. This increment, as it was demonstrated in the performance test, does not substantially inflate computational costs.

2. 3D RSG presents more intricacies. While increasing the spatial order can mitigate the numerical dispersion to some extent, there seems to be a limit to its effectiveness. For accurate modeling of anisotropic media, it may be more prudent to rely on approach (i) and utilize a greater number of grid points.

Lastly, benchmarking the performance of a wave solver is inherently challenging, especially considering the potential variability when different platforms are utilized. However, the insights derived from the provided figure are instructive, given that both the RSG and SSG were executed under identical configurations on the same CPU/platform.

The stencil setup time is independent of the model size (Figure a1). It primarily involves preparing the stencil formula for updates and is thus solely dependent on the FD order. Interestingly, the optimized 2D RSG, which employs the Voigt notation and derivative operations reduction, results in a more concise stencil. Consequently, its stencil setup time is even shorter than that of the 2D SSG. Furthermore, the 2D RSG's computation time is exceptional. Across all four model sizes, the 2D RSG consistently outperforms the 2D SSG in computation time. This efficiency can be attributed to our implementation and optimization strategies, which effectively counterbalance the increased derivative operations inherent to the RSG.

In the 3D domain, the SSG's stencil setup time is considerably shorter and remains relatively consistent. Although RSG take longer time to setup the stencil when order increases, the RSG's performance aligns closely with the SSG for orders 2, 4, and 8 in terms of computing time, which indicates relative high order is suggested. The higher-order longer stencil is reusable and

can benefit the entire modeling with better accuracy, though it takes longer to setup. However, significant increasing computational times are observed for the order 16 RSG.

Memory usage is another critical aspect to consider. The SSG in this test operates with six parameters, including *vp*, *vs*, density ρ, and three Thomsen's parameters. In contrast, the RSG demands a total of 21 independent elastic constants. Given this disparity, it's unsurprising that the RSG consumes more memory. However, when evaluating the actual memory usage, the RSG's consumption is barely double that of the SSG, showcasing its efficiency.

In summarizing the performance, the 2D RSG stands out as an effective choice for simulating wave propagation in anisotropic media. Users can utilize it confidently in 2D, without major worries about computational costs. However, for those exploring the 3D RSG, it's important to recognize that its capability to model complex and general anisotropy comes with higher computational demands. Additionally, the 3D RSG's aggravated numerical dispersion may necessitate the use of finer grids, further elevating the computational requirements.

5 Conclusions

We have introduced an implementation of a rotated staggered grid (RSG) wave solver built upon a symbolic finite difference framework. This solver aims to simulate the elastic wave propagation in complex anisotropic media or structure and is theoretically applicable for any even-numbered spatial order. We address pertinent issues such as the source injection method to circumvent the checkerboard effect and provide computational optimizations. Our results highlight the RSG's superior stability, especially when dealing with lower anisotropic symmetries, outperforming the conventional standard staggered grid (SSG). Through our tests on numerical dispersion and waveform observations, we note that although 2D models are prone to numerical dispersion compared to SSG, its effects can potentially be mitigated using higher spatial orders. Based on the subsequent performance evaluation, the optimized 2D wave solver proves to be computationally efficient and could be applied to simulate any geologic problem with complicate anisotropic structures, without significant concerns about additional computational effort. In 3D, given the inherent fourfold increase in derivative operations relative to SSG, the RSG solver demonstrates improved performance, but remains computationally demanding. With the rising

significance of seismic anisotropy and the continuous advancement in computing capabilities, we anticipate that this challenge will become less prohibitive in the future.

Acknowledgments

O. Zhang was supported by Purdue University Teaching Assistantships and Brand Professor of Unconventional Resources Research Assistantships. Computational support was provided by Purdue Rosen Center for Advanced Computing (RCAC) with assistance of W-W. Tung greatly appreciated. The authors acknowledge the support form Devito developer and community with especially grateful to M. Louboutin and F. Luporini.

Open Research

The rotated staggered grid (RSG) wavesolver [software] and is available at Zhang & Schmitt (2023). It has the same formatted API or input arguments as other Devito seismic wavesolver class (e.g., acoustic, elastic, viscoacoustic and viscoelastic). A tutorial Jupyter Notebook is provided for a quick hands-on start. The tutorial performs a simple 2D modeling on a two-layered structure consists of an upper water layer and a lower tilted transverse isotropic (TTI) media. At the time of submission of the manuscript, the wavesolver is compatible with Devito version 4.6.2 to 4.8.3.

The Devito package [software] is available at Louboutin et al. (2019) and Luporini et al. (2020). The performance of 3D standard staggered grid is conducted with an open-source elastic solver [software] at Louboutin (2022) utilizing stiffness matrix. The data used for stability test, dispersion and performance in the study are also available at Purdue University Research Repository via doi:10.4231/R8N7-HX28 under GNU General Public License 3.0.

Other open-source software used in this study: Figures were made with Matplotlib [software] version 3.5.2 (Hunter, 2007) and wiggle plot tool [software] (Hall, 2020). Data processing with Numpy [software] version 1.21.5 (Harris et al., 2020).

Appendix A: Finite difference coefficient for arbitrary spatial order

The FD spatial order (i.e., how many adjacent points are used in evaluating the derivatives) is one critical factor controlling the modeling accuracy. Use of increasingly higher orders can suppress numerical dispersion (Marfurt, 1984) without dramatic increases in memory compared against using finer grids. Despite numerous efforts to identify optimal coefficients (Holberg, 1987; Lele, 1992; Liu and Sen, 2009), many approaches may require detailed knowledge about the model (Yang et al., 2015) or specific modifications to the FD stencil based on varying media (Wang et al., 2017). As a result, these methods might not be as universally applicable as the classical Taylor Expansion. With advancements in computing capabilities, including expanded memory and faster I/O, it's desirable for wave solvers to support increasingly high spatial orders. This allows users to select an order tailored to their available computational resources. In our implementation, we support arbitrary even-numbered spatial orders and determine the corresponding FD coefficients using Taylor Expansion.

For a 2D example, a series of adjacent points about x_0 are involved in derivative evaluations, whose location are distributed at $(x_0+ndl+0.5dl)$, where $n=-N, -N+1, \ldots, N-1, N$, in rotated direction, where $dl=\sqrt{dx^2+dz^2}$ (for 2D) and N is the even number spatial order. The finite-difference coefficients C can be determined by solving (Taylor, 2016):

$$\begin{bmatrix} C_1 \\ \vdots \\ C_N \end{bmatrix} = \frac{1}{dl} \begin{bmatrix} (-N-0.5)^0 & (-N+1-0.5)^0 & \cdots & (N-1+0.5)^0 & (N+0.5)^0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ (-N-0.5)^{n-1} & (-N+1-0.5)^{n-1} & \cdots & (N-1+0.5)^{n-1} & (N+0.5)^{n-1} \end{bmatrix} \begin{bmatrix} 1 \\ \vdots \\ 0 \end{bmatrix}$$

A more detail and general discussion to generate arbitrary spaced FD coefficient can refer to Fornberg (1988). Consequently, with this method, our implementation of the solver is capable to calculate the derivative of arbitrary even spatial order, while it is still limited to the computing power. The FD coefficients of order 2, 4, 6, 8, 16 and 24, which are used in this study, are summarized in Table A1.

Table A1. Finite-Difference coefficients for rotated staggered grid with spatial order of 2, 4, 8 ,16 and 24.

Spatial order	2	4	8	16	24
-12					1.74342842e-09
-11					-4.81004771e-08

-10					6.46359004e-07
-9					-5.65173186e-06
-8				8.52346420e-07	3.62967068e-05
-7				-1.70217111e-05	-1.83631414e-04
-6				1.66418878e-04	7.69431397e-04
-5				-1.07727117e-03	-2.79139604e-03
-4			6.97544643e-04	5.34238560e-03	9.22870290e-03
-3			-9.57031250e-03	-2.30363667e-02	-3.01471042e-02
-2		0.04166667	7.97526042e-02	1.06649846e-01	1.17238580e-01
-1	-1.0	-1.125	-1.19628906e+00	-1.23409107e+00	-1.24699536e+00
1	1.0	1.125	1.19628906e+00	1.23409107e+00	1.24699536e+00
2		-0.04166667	-7.97526042e-02	-1.06649846e-01	-1.17238917e-01
3			9.57031250e-03	2.30363667e-02	3.01471839e-02
4			-6.97544643e-04	-5.34238560e-03	-9.22874008e-03
5				1.07727117e-03	2.79141194e-03
6				-1.66418878e-04	-7.69437092e-04
7				1.70217111e-05	1.83633083e-04
8				-8.52346420e-07	-3.62970974e-05
9					5.65180201e-06
10					-6.46368086e-07
11					4.81012311e-08
12					-1.74345856e-09

References

Almqvist, B. S. G., & Mainprice, D. (2017). Seismic properties and anisotropy of the continental crust: Predictions based on mineral texture and rock microstructure: SEISMIC PROPERTIES OF THE CRUST. *Reviews of Geophysics*, *55*(2), 367–433. https://doi.org/10.1002/2016RG000552

Assad, J. M. (2005). The effect of orthorhombic anisotropy and its implication for oil recovery and reservoir exploitation. *Geophysical Prospecting*, *53*(1), 121–129.

Auld, B. A. (1973). *Acoustic fields and waves in solids*. Ripol Classic Publishing House.

600 Bansal, R., & Sen, M. K. (2008). Finite-difference modelling of S-wave splitting in anisotropic
601 media. *Geophysical Prospecting*, *56*(3), 293–312. https://doi.org/10.1111/j.1365-
602 2478.2007.00693.x
603 Bohlen, T., & Saenger, E. H. (2006). Accuracy of heterogeneous staggered-grid finite-difference
604 modeling of Rayleigh waves. *GEOPHYSICS*, *71*(4), T109–T115.
605 https://doi.org/10.1190/1.2213051
606 Chaillat, S., Bonnet, M., & Semblat, J. F. (2009). A new fast multi-domain BEM to model seismic
607 wave propagation and amplification in 3-D geological structures. *Geophysical Journal*
608 *International*, *177*(2), 509–531. https://doi.org/10.1111/j.1365-246X.2008.04041.x
609 Cheadle, S. P., Brown, R. J., & Lawton, D. C. (1991). Orthorhombic anisotropy: A physical seismic
610 modeling study. *Geophysics*, *56*(10), 1603–1613.
611 Chen, H., Wang, X., & Zhao, H. (2006). A rotated staggered grid finite-difference with the
612 absorbing boundary condition of a perfectly matched layer. *Chinese Science Bulletin*,
613 *51*(19), 2304–2314. https://doi.org/10.1007/s11434-006-2127-8
614 Crampin, S. (1984). An introduction to wave propagation in anisotropic media. *Geophysical*
615 *Journal International*, *76*(1), 17–28. https://doi.org/10.1111/j.1365-246X.1984.tb05018.x
616 Dormy, E., & Tarantola, A. (1995). Numerical simulation of elastic wave propagation using a finite
617 volume method. *Journal of Geophysical Research: Solid Earth*, *100*(B2), 2123–2133.
618 Fornberg, B. (1988). Generation of finite difference formulas on arbitrarily spaced grids.
619 *Mathematics of Computation*, *51*(184), 699–706.
620 Gao, K., & Huang, L. (2017). An improved rotated staggered-grid finite-difference method with
621 fourth-order temporal accuracy for elastic-wave modeling in anisotropic media. *Journal of*
622 *Computational Physics*, *350*, 361–386. https://doi.org/10.1016/j.jcp.2017.08.053
623 Graves, R. W. (1996). Simulating seismic wave propagation in 3D elastic media using staggered-
624 grid finite differences. *Bulletin of the Seismological Society of America*, *86*(4), 1091–1106.
625 Hall, M. (2020). Wiggle Plot Python Tool. In *GitHub repository*. GitHub.
626 https://gist.github.com/kwinkunks/f594b243e582666b5a808520e9add262
627 Harris, C. R., Millman, K. J., Walt, S. J. van der, Gommers, R., Virtanen, P., Cournapeau, D.,
628 Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., Kerkwijk, M.
629 H. van, Brett, M., Haldane, A., Río, J. F. del, Wiebe, M., Peterson, P., … Oliphant, T. E.

(2020). Array programming with NumPy. *Nature*, *585*(7825), 357–362. https://doi.org/10.1038/s41586-020-2649-2

Helbig, K., & Thomsen, L. (2005). 75-plus years of anisotropy in exploration and reservoir seismics: A historical review of concepts and methods. *GEOPHYSICS*, *70*(6), 9ND-23ND. https://doi.org/10.1190/1.2122407

Holberg, O. (1987). Computational aspects of the choice of operator and sampling interval for numerical differentiation in large-scale simulation of wave phenomena. Geophysical prospecting, 35(6), 629-655.

Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. *Computing in Science & Engineering*, *9*(3), 90–95. https://doi.org/10.1109/MCSE.2007.55

Hustedt, B., Operto, S., & Virieux, J. (2004). Mixed-grid and staggered-grid finite-difference methods for frequency-domain acoustic wave modelling. *Geophysical Journal International*, *157*(3), 1269–1296. https://doi.org/10.1111/j.1365-246X.2004.02289.x

Igel, H., Mora, P., & Riollet, B. (1995). Anisotropic wave propagation through finite-difference grids. *GEOPHYSICS*, *60*(4), 1203–1216. https://doi.org/10.1190/1.1443849

Iturrarán-Viveros, U., & Sánchez-Sesma, F. J. (2020). Seismic wave propagation in real media: Numerical modeling approaches. *Encyclopedia of Solid Earth Geophysics*, 1–13.

Komatitsch, D., & Tromp, J. (1999). Introduction to the spectral element method for three-dimensional seismic wave propagation. *Geophysical Journal International*, *139*(3), 806–822.

Kosloff, D. D., & Baysal, E. (1982). Forward modeling by a Fourier method. *Geophysics*, *47*(10), 1402–1412.

Krüger, O. S., Saenger, E. H., & Shapiro, S. A. (2005). Scattering and diffraction by a single crack: An accuracy analysis of the rotated staggered grid. *Geophysical Journal International*, *162*(1), 25–31. https://doi.org/10.1111/j.1365-246X.2005.02647.x

Kukreja, N., Louboutin, M., Vieira, F., Luporini, F., Lange, M., & Gorman, G. (2016). Devito: Automated fast finite difference computation. *arXiv:1608.08658 [Cs]*. http://arxiv.org/abs/1608.08658

Lele, S. K. (1992). Compact finite difference schemes with spectral-like resolution. *Journal of Computational Physics*, *103*(1), 16–42.

Levander, A. R. (1988). Fourth-order finite-difference P-SV seismograms. *Geophysics*, *53*(11), 1425–1436.

Lisitsa, V., & Vishnevskiy, D. (2010). Lebedev scheme for the numerical simulation of wave propagation in 3D anisotropic elasticity‡: Lebedev scheme for numerical simulation of wave propagation. *Geophysical Prospecting*, *58*(4), 619–635. https://doi.org/10.1111/j.1365-2478.2009.00862.x

Liu, Y., & Sen, M. K. (2009). An implicit staggered-grid finite-difference method for seismic modelling. *Geophysical Journal International*, *179*(1), 459–474. https://doi.org/10.1111/j.1365-246X.2009.04305.x

Louboutin, M. (2022). ElasticTTI. In *GitHub repository*. GitHub. https://github.com/slimgroup/ElasticTTI/tree/master

Louboutin, M., Lange, M., Luporini, F., Kukreja, N., Witte, P. A., Herrmann, F. J., Velesko, P., & Gorman, G. J. (2019). Devito (v3.1.0): An embedded domain-specific language for finite differences and geophysical exploration. *Geoscientific Model Development*, *12*(3), 1165–1187. https://doi.org/10.5194/gmd-12-1165-2019

Luporini, F., Louboutin, M., Lange, M., Kukreja, N., Witte, P., Hückelheim, J., Yount, C., Kelly, P. H. J., Herrmann, F. J., & Gorman, G. J. (2020). Architecture and Performance of Devito, a System for Automated Stencil Computation. *ACM Trans. Math. Softw.*, *46*(1). https://doi.org/10.1145/3374916

Malehmir, R., & Schmitt, D. R. (2017). Acoustic Reflectivity From Variously Oriented Orthorhombic Media: Analogies to Seismic Responses From a Fractured Anisotropic Crust. *Journal of Geophysical Research: Solid Earth*, *122*(12), 10,069-10,085. https://doi.org/10.1002/2017JB014160

Marfurt, K. J. (1984). Seismic modeling: A frequency-domain/finite-element approach. In *SEG Technical Program Expanded Abstracts 1984* (pp. 633–634). Society of Exploration Geophysicists.

Moczo, P., Robertsson, J. O., & Eisner, L. (2007). The finite-difference time-domain method for modeling of seismic wave propagation. *Advances in Geophysics*, *48*, 421–516.

O'Brien, G. S. (2010). 3D rotated and standard staggered finite-difference solutions to Biot's poroelastic wave equations: Stability condition and dispersion analysis. *GEOPHYSICS*, *75*(4), T111–T119. https://doi.org/10.1190/1.3432759

Ogi, H., Ohmori, T., Nakamura, N., & Hirao, M. (2006). Elastic, anelastic, and piezoelectric coefficients of α-quartz determined by resonance ultrasound spectroscopy. *Journal of Applied Physics*, *100*(5).

Operto, S., Virieux, J., Amestoy, P., L'Excellent, J.-Y., Giraud, L., & Ali, H. B. H. (2007). 3D finite-difference frequency-domain modeling of visco-acoustic wave propagation using a massively parallel direct solver: A feasibility study. *Geophysics*, *72*(5), SM195–SM211.

Saenger, E. H., Gold, N., & Shapiro, S. A. (2000). Modeling the propagation of elastic waves using a modified finite-difference grid. *Wave Motion*, *31*(1), 77–92. https://doi.org/10.1016/S0165-2125(99)00023-2

Schoenberg, M., & Sayers, C. M. (1995). Seismic anisotropy of fractured rock. *Geophysics*, *60*(1), 204–211.

Serón, F. J., Sanz, F. J., Kindelán, M., & Badal, J. I. (1990). Finite-element method for elastic wave propagation. *Communications in Applied Numerical Methods*, *6*(5), 359–368. https://doi.org/10.1002/cnm.1630060505

Sidler, R., Rubino, J. G., & Holliger, K. (2013). Quantitative comparison between simulations of seismic wave propagation in heterogeneous poro-elastic media and equivalent visco-elastic solids for marine-type environments. *Geophysical Journal International*, *193*(1), 463–474. https://doi.org/10.1093/gji/ggs125

Tape, C., Liu, Q., Maggi, A., & Tromp, J. (2009). Adjoint Tomography of the Southern California Crust. *Science*, *325*(5943), 988–992. https://doi.org/10.1126/science.1175298

Taylor, C. R. (2016). Finite Difference Coefficients Calculator. https://web.media.mit.edu/crtaylor/calculator.html

Thomsen, L. (1986). Weak elastic anisotropy. *GEOPHYSICS*, *51*(10), 1954–1966. https://doi.org/10.1190/1.1442051

Tromp, J., Tape, C., & Liu, Q. (2004). Seismic tomography, adjoint methods, time reversal and banana-doughnut kernels: Seismic tomography, adjoint methods, time reversal and banana-doughnut kernels. *Geophysical Journal International*, *160*(1), 195–216. https://doi.org/10.1111/j.1365-246X.2004.02453.x

721 Tsvankin, I. (1997). Anisotropic parameters and P-wave velocity for orthorhombic media.
722 *Geophysics*, *62*(4), 1292–1309.
723 Tsvankin, I., Gaiser, J., Grechka, V., Van Der Baan, M., & Thomsen, L. (2010). Seismic anisotropy
724 in exploration and reservoir characterization: An overview. *GEOPHYSICS*, *75*(5), 75A15-
725 75A29. https://doi.org/10.1190/1.3481775
726 Vavryčuk, V. (2015). Moment tensor decompositions revisited. *Journal of Seismology*, *19*(1), 231–
727 252. https://doi.org/10.1007/s10950-014-9463-y
728 Virieux, J. (1986). P-SV wave propagation in heterogeneous media: Velocity-stress finite-
729 difference method. *Geophysics*, *51*(4), 889–901.
730 Virieux, J., Etienne, V., Cruz-Atienza, V., Brossier, R., Chaljub, E., Coutant, O., Garambois, S.,
731 Mercerat, D., Prieux, V., Operto, S., Ribodetti, A., & Tago, J. (2012). Modelling Seismic
732 Wave Propagation for Geophysical Imaging. In M. Kanao (Ed.), *Seismic Waves—Research*
733 *and Analysis*. InTech. https://doi.org/10.5772/30219
734 Virieux, J., & Operto, S. (2009). An overview of full-waveform inversion in exploration
735 geophysics. *GEOPHYSICS*, *74*(6), WCC1–WCC26. https://doi.org/10.1190/1.3238367
736 Wang, K., Peng, S., Lu, Y., & Cui, X. (2020). The velocity-stress finite-difference method with a
737 rotated staggered grid applied to seismic wave propagation in a fractured medium.
738 *GEOPHYSICS*, *85*(2), T89–T100. https://doi.org/10.1190/geo2019-0186.1
739 Wang, W., Hu, T., Song, J., Li, Y., Li, J., & Zhang, Y. (2017). An optimized scheme of dispersion
740 suppression for elastic-wave variable-order rotated staggered-grid forward modeling.
741 *Journal of Geophysics and Engineering*, *14*(6), 1624–1638. https://doi.org/10.1088/1742-
742 2140/aa83a2
743 Yang, L., Yan, H., & Liu, H. (2015). Optimal rotated staggered-grid finite-difference schemes for
744 elastic wave modeling in TTI media. *Journal of Applied Geophysics*, *122*, 40–52.
745 https://doi.org/10.1016/j.jappgeo.2015.08.007
746 Zhang, O., & Schmitt, D. R. (2023). *A Rotated Staggered Grid (RSG) finite-difference elastic wave*
747 *solver based on Devito*. https://doi.org/doi:/10.4231/R8N7-HX28
748 Zheng, J., Meng, L., Sun, Y., & Peng, S. (2023). GPU-acceleration 3D rotated-staggered-grid
749 solutions to microseismic anisotropic wave equation with moment tensor implementation.
750 *International Journal of Mining Science and Technology*, *33*(4), 403–410.
751 https://doi.org/10.1016/j.ijmst.2022.09.017