Optimally Accurate Nearly Analytic Discrete Scheme for Wave-Field Simulation in 3D Anisotropic Media

by Dinghui Yang, Guojie Song, and Ming Lu

Abstract We present a new numerical method for elastic wave modeling in 3D isotropic and anisotropic media, which is called the 3D optimal nearly analytic discrete method (ONADM) in this article. This work is an extension of the 2D ONADM (Yang et al., 2004) that models acoustic and elastic waves propagating in 2D media. The formulation is derived by using a multivariable truncated Taylor series expansion and high-order interpolation approximations. Our 3DONADM enables wave propagation to be simulated in three dimensions through isotropic and anisotropic models. Promising numerical results show that the error of the ONADM for the 3D case is less than those of the conventional finite-difference (FD) method and the so-called Lax–Wendroff correction (LWC) schemes, measured quantitatively by the root-mean-square deviation from analytical solution. The seismic wave fields in the 3D isotropic and anisotropic media are simulated and compared with those obtained by using the fourth-order LWC method and exact solutions for the acoustic wave case. We show that, compared with the conventional FD method and the LWC schemes, the ONADM for the 3D case can reduce significantly the storage space and computational costs. Numerical experiments illustrate that the ONADM provides a useful tool for the 3D large-scale isotropic and anisotropic problems and it can suppress effectively numerical dispersions caused by discretizing the 3D wave equations when too-coarse grids are used, which is the same as the 2D ONADM. Numerical modeling also implies that simultaneously using both the wave displacement and its gradients to approximate the high-order derivatives is important for both decreasing the numerical dispersion caused by the discretization of wave equations and compensating the important wave-field information included in wave-displacement gradients.

Introduction

Analyses of seismic data to determine earth structure require accurate and efficient numerical techniques for modeling seismic wave fields. Finite-difference methods for solving wave equations have provided useful tools in exploration seismology. As a deterministic processing tool or as an interpretive aid, the finite-difference technique has met with resistance, however, because of its intensive use of central processing unit (CPU) time and its need for large amounts of direct-access memory or seriously numerical dispersion when too few samples per wavelength are used or when the models have a large velocity contrast. Although the numerical dispersion caused by the discretization of the wave equation can be eliminated by using finer spatial grids and small-time sampling, it will result in large computational costs and large amounts of storage. It is therefore desirable to develop accurate and efficient computational methods, especially for the large-scale computing and 3D anisotropic problems.

Many works have attempted to derive more accurate methods by redefining the operators for spatial and temporal differentiation to minimize numerical dispersion or to make some special treatments to eliminate numerical dispersion (Kosloff and Baysal, 1982; Dablain, 1986; Fei and Larner, 1995; Zhang et al., 1999; Mizutani et al., 2000; Yang et al., 2002b; Zheng and Zhang, 2005; and others). However, the developed methods for eliminating the numerical dispersion have some disadvantages. For example, although the numerical dispersions can be suppressed using the so-called flux-corrected transport (FCT) technique (Fei and Larner, 1995; Zhang et al., 1999; Yang et al., 2002a; Zheng et al., 2006), the FCT method is typically unable to fully recover the resolution lost by the numerical dispersion when the spatial sampling becomes too coarse (Yang et al., 2002a). The pseudospectral method (PSM) (Kosloff and Baysal, 1982; Fornberg, 1987) is attractive because the space operators are exact up to the Nyquist frequency, but it requires the Fast Fourier transform (FFT) of wave field to be made, which is computationally expensive for 3D anisotropic models and
has the difficulties of handling nonperiodic boundary conditions and the nonlocality on memory access of the FFT, For which the PSM is difficult to implement on massively parallel computers (Mizutani et al. , 2000). The high-order finite-difference (FD) methods such as high-order FD scheme (e.g., Fornberg, 1990; Igel et al., 1995), high-order compact FD scheme (Lele, 1992), or the so-called Lax–Wendroff correction (LWC) scheme (Lax and Wendroff, 1964; Dablain, 1986; Robertsson et al., 1994; Blanch and Robertsson, 1997) with local operators compared with the PSM are another choice of suppressing the numerical dispersion. But the numerical dispersion or undesirable ripple also affects the performance of the so-called high-order FD methods. For example, the tenth-order compact FD schemes (e.g., Wang et al., 2002), which usually use more grids than low-order schemes, also suffer from numerical dispersions. The demand of more grids in high-order FD methods also prevents the algorithms from efficient parallel implementation and artificial boundary treatment like the PSM. A remedy is to use local operators with high accuracy that are suitable for fast parallel implementation when we develop numerical methods with less numerical dispersion to generate synthetic seismograms. The staggered-grid FD methods with local operators (Virieux, 1986; Fornberg, 1990; Igel et al., 1995) can further reduce the numerical dispersion, but they still suffer from the numerical dispersion when too few samples per wavelength are used (Sei and Symes, 1994; Yang et al., 2002c). Actually, Virieux (1986) had concluded that “the main limitations of our stress-velocity finite-difference method come from the numerical dispersion and the finite numerical size of the grid.” For the anisotropic case the staggered-grid FD method may result in the numerical anisotropy and induce additional error of the wave properties (Igel et al., 1995); because of some of the elements of the stress and strain, tensors must be interpolated to calculate the Hook sum for the strain-stress, staggered-grid FD.

Recently, a so-called optimal nearly analytic discrete method (ONADM) with local operators attacking the numerical dispersion for 1D and 2D acoustic and elastic equations was proposed (Yang et al., 2004), which was an improved version of the “nearly analytic discrete method” (NADM) (Yang et al., 2003) initially suggested by Kondoh (1991) and applied to solve 1D and 2D parabolic and hyperbolic equations (Kondoh et al., 1994). Furthermore, its theoretical problems such as the numerical dispersion, stability conditions, and errors are studied for 1D and 2D acoustic equations in more recent work (Yang et al., 2006). The ONADM (or NADM) uses a truncated Taylor expansion with respect to time to analytically approximate the wave displacement and its gradients at grid points. Meanwhile, it uses simultaneously both the wave displacement and its gradients to determine the high-order space derivatives involved in these truncated Taylor formulae. This structure method is different from other FD methods that use only the wave displacements to approximate high-order derivatives for discretizing the original wave equation. The difference and the relation between the LWC or high-order compact difference schemes were discussed in detail in our earlier work (Yang et al., 2004).

As shown by the numerical results in our recent work (Yang et al., 2006), the ONADM is very efficient in large-scale seismogram synthetics because it can effectively suppress the numerical dispersion caused by discretizing the wave equation using the local interpolation compensation for the truncated Taylor series, whereas the conventional FD schemes with second- and fourth-order accuracies and LWC methods suffer from numerical dispersion near large velocity contrast or when too few samples per wavelength are used (Fei and Larner, 1995; Wang et al., 2002; Yang et al., 2002a, 2006). The efficient synthetics together with algorithms for waveform inversion can help us better understand the earth structure.

In general, our previous works have evaluated proposed computational schemes by presenting theoretical derivations or conducting numerical tests for the 1D and 2D cases (Yang et al., 2003, 2004, 2006, 2007). However, seismologists use computational methods to study wave propagation in 3D models that approximate the actual 3D Earth, which is anisotropic. It is therefore essential to derive the 3D ONADM for investigating the actual Earth structure.

The main purpose of this article is to present the 3D ONADM for modeling the elastic wave propagating in 3D anisotropic media, which is actually an extension of the 2D ONADM proposed by Yang et al. (2004) for simulating acoustic and elastic waves propagating in 2D media. Although the extension is not a difficult task, it is still far from straightforward because of the complexity of the formulation of 3D anisotropic problem. In contrast to previous studies, the derivations in this article are also based on the perturbation method and interpolating approximation of higher-degree multivariable polynomial. To derive the 3D ONADM, we first use the fourth-order partial differential equations with respect to time $t$ and convert the high-order time derivative into the high-order spatial derivative, which is similar to the LWC methods (Lax and Wendroff, 1964; Dablain, 1986; and others). Then, we design the ONADM for the 3D anisotropic case based on the truncated Taylor expansion and multivariable interpolation relations. To verify the validity of the 3D ONADM, we compare numerically the error of the 3D ONADM with those of the conventional second-order FD scheme and the LWC method with fourth-order accuracy for the initially 3D value problem of elastic wave propagation. Meanwhile, multicomponent seismic wave fields in isotropic and transversely isotropic media are simulated to test the ONADM and to compare with the LWC method.

**Formulation of 3D ONADM**

In this section, following the 2D NADM and the 2D ONADM (Yang et al., 2003, 2004) we design the 3D
ONADM for the anisotropic case by using the Taylor expansion and multivariable high-degree interpolation method.

In 3D anisotropic media, the wave equation, describing the elastic wave propagation, is written as

\[
\frac{\partial^2 u_i}{\partial t^2} + f_i = \rho \frac{\partial^4 u_i}{\partial x^4}, \quad (1)
\]

\[
\sigma_{ij} = \frac{1}{2} c_{ijkl}(x, y, z) \left( \frac{\partial u_k}{\partial x_j} + \frac{\partial u_j}{\partial x_k} \right), \quad (2)
\]

where \( i, j, k, l = 1, 2, 3 \), \( \rho = \rho(x, y, z) \) is the density, \( u_i \) and \( f_i \) denote the displacement component, and the force-source component in the \( i \)th direction. The fourth-order stiffness tensor \( c_{ijkl}(x, y, z) \) may up to 21 independent elastic constants for the anisotropic 3D case. In particular, for the isotropic and transversely isotropic cases the 21 independent elastic constants are reduced to two Lamé constants (\( \lambda \) and \( \mu \)) and five constants \( c_{11}, c_{13}, c_{14}, c_{44}, \) and \( c_{66} \), respectively.

For simplicity, we use the notations similar to those in the original NADM (Yang et al., 2003), that is,

\[
p_i = \frac{1}{\rho} \left( \frac{\partial \sigma_{ij}}{\partial x_j} + f_i \right) = \frac{\partial^2 u_i}{\partial t^2}, \quad i = 1, 2, \ldots, 3,
\]

\[
U = (u_1, u_2, u_3)^T, \quad P = (p_1, p_2, p_3)^T,
\]

\[
\bar{U} = \left[ \frac{\partial U}{\partial x}, \frac{\partial U}{\partial y}, \frac{\partial U}{\partial z} \right]^T, \quad \bar{P} = \left[ \frac{\partial P}{\partial x}, \frac{\partial P}{\partial y}, \frac{\partial P}{\partial z} \right]^T.
\]

Following the 2D ONADM (Yang et al., 2004), for the 3D case we similarly have the following formula:

\[
\bar{U}_{i,j,k}^{n+1} = 2\bar{U}_{i,j,k}^n - \bar{U}_{i,j,k}^{n-1} + (\Delta t)^2 \bar{P}_{i,j,k}^n + \frac{(\Delta t)^4}{12} \left( \frac{\partial^2 P}{\partial t^2} \right)_{i,j,k}^n, \quad (4)
\]

where \( n \) is a nonnegative integer, \((i, j, k)\) denotes a spatial grid point, \( \Delta t \) denotes the time increment. To avoid the implicit schemes and cost of storage derived from direct central-differencing of the high-order time derivatives of \( \bar{U} \) that included implicitly in \( \left( \frac{\partial^2 P}{\partial t^2} \right) \) (equation 4), we convert the high-order time derivative to the spatial derivatives \( \left( \frac{\partial^{l+m+s} U}{\partial x^l \partial y^m \partial z^s} \right)_{i,j,k} \) \((2 \leq l + m + s \leq 5)\) using equation (3). The transform used here is similar to the high-order FD methods or the so-called Lax–Wendroff correction (LWC) or optimal FD methods, where the original wave equation is used to convert the high-order error terms in Taylor expansions to spatial derivatives that can be handled explicitly, thereby increasing the accuracy of the method significantly (Dablain, 1986; Blanch and Robertsson, 1997; Takeuchi and Geller, 2000). However, the way that the ONADMs approximate the high-order spatial derivatives is completely different from the later methods. It is also different from the optimal FD scheme based on a predictor-corrector method (Geller and Takeuchi, 1998; Takeuchi and Geller, 2000). The high-order FD methods (Dablain, 1986) (or the so-called LWC) use only the wave displacement to determine the high-order spatial derivatives and, thus, it is hard to capture the wave-field information characterized by the gradient of wave displacement, whereas the ONADMs uses both the wave displacement and its gradients to determine the high-order spatial derivatives. This allows the algorithm to capture more wave-field information included in both the wave displacement and its gradients (Yang et al., 2006). To be more specific, following the 2D ONADM (Yang et al., 2004) and the “analysis thought” for solving the 2D hyperbolic and parabolic equations (Komodoh et al., 1994), we introduce the following interpolation function on variables \( X, Y, \) and \( Z \):

\[
G(X, Y, Z) = \sum_{r=0}^{M} \frac{1}{r!} \left( X \frac{\partial}{\partial x} + Y \frac{\partial}{\partial y} + Z \frac{\partial}{\partial z} \right)^r U, \quad (5a)
\]

and its partial derivatives with respect to \( X, Y, \) and \( Z \) are

\[
G_x(X, Y, Z) = \sum_{r=0}^{M-1} \frac{1}{r!} \left( X \frac{\partial}{\partial x} + Y \frac{\partial}{\partial y} + Z \frac{\partial}{\partial z} \right)^r \frac{\partial}{\partial x} U, \quad (5b)
\]

\[
G_y(X, Y, Z) = \sum_{r=0}^{M-1} \frac{1}{r!} \left( X \frac{\partial}{\partial x} + Y \frac{\partial}{\partial y} + Z \frac{\partial}{\partial z} \right)^r \frac{\partial}{\partial y} U, \quad (5c)
\]

\[
G_z(X, Y, Z) = \sum_{r=0}^{M-1} \frac{1}{r!} \left( X \frac{\partial}{\partial x} + Y \frac{\partial}{\partial y} + Z \frac{\partial}{\partial z} \right)^r \frac{\partial}{\partial z} U, \quad (5d)
\]

where \( M \) is an integer and can be determined in accordance with the designed method. In our study, we choose \( M = 5 \) because the highest partial derivatives of the displacement \( U \) with respect to \( x, y, \) and \( z \) in the computational formula (4) are fifth order.

To determine the values of the higher-order partial derivatives \( \left( \frac{\partial^{l+m+s} U}{\partial x^l \partial y^m \partial z^s} \right)_{i,j,k} \) \((2 \leq l + m + s \leq 5)\) included in equation (4) by the wave displacement \( U_{i+p,j+q,k+r} \) and its first-order partial derivatives such as \( \left( \frac{\partial U}{\partial x} \right)_{i+p,j+q,k+r} \), \( \left( \frac{\partial U}{\partial y} \right)_{i+p,j+q,k+r} \), \( \left( \frac{\partial U}{\partial z} \right)_{i+p,j+q,k+r} \) \((p, q, r = -1, 0, 1)\). Let the function \( G(X, Y, Z) \) and its partial derivatives \( G_x(X, Y, Z), G_y(X, Y, Z), \) and \( G_z(X, Y, Z) \) satisfy the following interpolation conditions between the grid point \((i, j, k)\) and its neighboring nodes \((i+p, j+q, k+r)\) \((p, q, r = -1, 0, 1)\), as shown in Figure 1.

At the grid point \((i-1, j, k)\):

\[
[G(-\Delta x, 0, 0)]_{i-1,j,k}^n = U_{i-1,j,k}^n, \quad (6a)
\]

\[
[G_x(-\Delta x, 0, 0)]_{i-1,j,k}^n = \left( \frac{\partial}{\partial x} \right)_{i-1,j,k}^n U^u, \quad (6b)
\]
Uously, we can obtain the high-order derivatives of $u$ at rest 24 nodes.

Similarly, we can easily write the rest connection relations of the fourth-order LWC and the second-order FD methods.

At the point $(i-1, j+1, k)$:

$$[G(-\Delta x, 0, 0)]_{i,j,k}^n = \left(\frac{\partial}{\partial y} U\right)_{i-1,j+1,k}^n, \quad \text{and} \quad (6c)$$

$$[G(-\Delta x, 0, 0)]_{i,j,k}^n = \left(\frac{\partial}{\partial z} U\right)_{i-1,j+1,k}^n. \quad (6d)$$

At the point $(i-1, j+1, k)$:

$$[G(-\Delta x, 0, 0)]_{i,j,k}^n = \left(\frac{\partial}{\partial y} U\right)_{i-1,j+1,k}^n. \quad (6e)$$

$$[G(-\Delta x, 0, 0)]_{i,j,k}^n = \left(\frac{\partial}{\partial z} U\right)_{i-1,j+1,k}^n. \quad (6f)$$

$$[G(-\Delta x, 0, 0)]_{i,j,k}^n = \left(\frac{\partial}{\partial y} U\right)_{i-1,j+1,k}^n, \quad \text{and} \quad (6g)$$

$$[G(-\Delta x, 0, 0)]_{i,j,k}^n = \left(\frac{\partial}{\partial z} U\right)_{i-1,j+1,k}^n. \quad (6h)$$

Similarly, we can easily write the rest connection relations at rest 24 nodes. $\Delta x$, $\Delta y$, and $\Delta z$ are the spatial increments in the $x$, $y$, and $z$ directions, respectively.

According to the interpolation relations given previously, we can obtain the high-order derivatives of $U$ with respect to $x$, $y$, and $z$. Evaluations of the high-order partial derivatives of the displacement $U$ included in equation (4) at the grid point $(i, j, k)$ are listed in the Appendix.

Error Analysis and Stability

To investigate the accuracy of the 3D ONADM, we estimate the theoretical error and numerically compare against the fourth-order LWC and the second-order FD methods.

Theoretical Error

Using the Taylor series expansion, we find that the errors of $\partial^{l+m+s}U/\partial x^l\partial y^m\partial z^s$ on $2 \leq l + m + s \leq 3$ and $4 \leq l + m + s \leq 5$ are, respectively, $O(h_1^l + h_2^m + h_3^s)$ and $O(h_1^l + h_2^m + h_3^s)$ (where $h_1 = \Delta x$, $h_2 = \Delta y$, $h_3 = \Delta z$) caused by the interpolation approximation. Owing to using formula (4), which is similar to the LWC method (Dablain, 1986), the temporal error, caused by the discretization of the temporal partial derivative, is $O(\Delta t^2)$. Therefore, we find that the error introduced by the ONADM is in the order of $O(\Delta t^2 + h_1^l + h_2^m + h_3^s)$ due to the use of equation (4) and the interpolation approximation, that is, the ONADM is accurate fourth-order in space and time. To further illustrate the accuracy of the ONADM, in the following subsection we compare the numerical results of the ONADM and different methods such as the conventional FD and fourth-order LWC methods with the exact solution of the 3D acoustic wave equation for the homogeneous media.

Numerical Error

Consider the following 3D initial problem:

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2} = \frac{1}{c^2} \frac{\partial^2 u}{\partial t^2}, \quad (7a)$$

$$u(0, x, y, z) = \cos\left(-\frac{2\pi f_0}{c}(l_0 \cdot x + m_0 \cdot y + n_0 \cdot z)\right), \quad \text{and} \quad (7b)$$

$$\frac{\partial u(0, x, y, z)}{\partial t} = -2\pi f_0 \sin\left(-\frac{2\pi f_0}{c}(l_0 \cdot x + m_0 \cdot y + n_0 \cdot z)\right), \quad (7c)$$

where $f_0$ denotes the frequency, $c$ is the wave velocity, and the vector $(l_0, m_0, n_0)$ is the direction of the incident wave at time $t = 0$, which are chosen by $c = 2.5$ km/sec and $(l_0, m_0, n_0) = (1/\sqrt{3}, 1/\sqrt{3}, 1/\sqrt{3})$ in our present experiment.

Obviously, the analytical solution for the initial problem (7) is

$$u(t, x, y, z) = \cos\left[2\pi f_0 \left(t - \frac{x}{c}l_0 - \frac{y}{c}m_0 - \frac{z}{c}n_0\right)\right].$$

For comparison, we use the second-order FD method and the so-called LWC (fourth-order compact scheme [Dablain, 1986]) for equation (7).

In our calculation, the parameters are chosen in the following way. The grid number $N_x = N_y = N_z = 101$ in the $x$, $y$, and $z$ directions, the frequency $f_0 = 15$ Hz. The relative error ($E_r$), which is the ratio of the rms of the residual ($u_{r,ik}^n - u(t_n, x_i, y_j, z_k)$) and the rms of the exact solution $u(t_n, x_i, y_j, z_k)$, is defined by
Figures 2, 3, and 4 are the curves of the relative error $E_r$ versus time corresponding to different spatial and time-step sizes, where three lines of $E_r$ for the second-order center scheme (line, ---), the fourth-order LWC (line, —), and the ONADM (line – – •) are shown in a semilog scale. For these figures, the maximum relative errors for different cases are listed in Table 1 ($\Delta x = \Delta y = \Delta z = h$). From these error curves and Table 1, we find that $E_r$ increases corresponding to the increase in the time and/or spatial increments for all the three methods. Figures 2, 3, and 4 show that the ONADM has the highest accuracy among them. From the maximum relative errors listed in Table 1 and the error curves shown in Figures 2, 3, and 4, computed by the LWC and the ONADM, we can find that the errors of the ONADM are the same degree of magnitude as those of the fourth-order LWC scheme. This illustrates that the ONADM has same accuracy as the fourth-order LWC method. Our numerical results are consistent with what we derived in our theoretical analyses.

Stability of ONADM

It is well known that the time increment $\Delta t$ must be less than or equal to the Courant limit to keep numerical calculation stable. In our recent work (Yang et al., 2006), we have derived the following stability conditions:

For the 1D acoustic case, $\Delta t \leq \frac{\alpha_{\text{max}}}{c} \frac{h}{c} = 0.754 \frac{h}{c}$, and

For the 2D acoustic case, $\Delta t \leq \frac{\alpha_{\text{max}}}{c} \frac{h}{c} = 0.527 \frac{h}{c}$.
where \( \alpha_{\text{max}} \) denotes the maximum value of the Courant number defined by \( \alpha = c \Delta t / h \) (Sei and Symes, 1994; Dablain, 1986) in which \( h = \Delta x \) for the 1D case and \( h = \Delta x = \Delta z \) for the 2D case is the spatial step size, which keeps the numerical calculation stable. For the 3D homogeneous case, through the Fourier analyses (Richtmyer and Morton, 1967), as for the 2D case presented in Yang et al. (2006), we can similarly obtain the stability condition for solving the acoustic equation (7a) by using the ONADM. The algebraic details for the stability of the 3D ONADM are tedious and here we only give the stability condition under the case of our consideration \( \Delta x = \Delta y = \Delta z = h \) as follows:

\[
\Delta t \leq \alpha_{\text{max}} \frac{h}{c} \approx 0.46 \frac{h}{c}, \tag{9}
\]

When the 3D ONADM is applied to solve the 3D elastic wave equation, we roughly estimate that the temporal increment should satisfy the following stability condition:

\[
\Delta t \leq \Delta t_{\text{max}} \approx 0.46 \frac{h}{c_{\text{max}}}, \tag{10}
\]

where \( \Delta t_{\text{max}} \) denotes the maximum temporal increment and \( c_{\text{max}} \) is the maximum \( P \)-wave velocity.

### Efficiency and Wave Fields Modeling

In this section, we investigate the efficiency of the ONADM for the 3D isotropic and transversely isotropic cases and compare it with the so-called LWC method (Dablain, 1986) by wave-field modeling. The different spatial sampling rates are chosen so that we test the effects of sampling rate based on the Nyquist frequency, which is defined (Dablain, 1986) by

\[
\Delta x = \frac{v_{\text{min}}}{f_{\text{Nyquist}} \cdot G}, \tag{11}
\]

where \( v_{\text{min}} \) denotes the minimum \( S \)-wave (or quasi-\( S \)-wave) velocity, \( f_{\text{Nyquist}} \) is the Nyquist frequency, and \( G \) denotes the number of grid points per minimum \( S \) (or quasi-\( S \)) wavelength at the upper half-power frequency of the source or the number of grid points needed to cover the Nyquist frequency for nondispersive propagation (Dablain, 1986).

### Computational Efficiency

To examine the efficiency of the ONADM, we solve numerically the following 3D acoustic wave equation and compare numerical results computed using the ONADM and the so-called LWC method:

\[
\frac{\partial^2 u}{\partial t^2} = c^2 \left( \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2} \right) + f, \tag{12}
\]

where \( f \) is the force source that has the following form (Zahra-dnik et al., 1993):

\[
f(t) = \sin(2\pi f_0 t) \exp(-4\pi^2 f_0^2 t^2/16), \tag{13}
\]

with a peak frequency of \( f_0 = 15 \) Hz, and an explosive source is located at the center of the computational domain. In this subsection, the computation of generating wave-field snapshots is performed on PC clusters with 1 GB memory.

In the numerical experiment, we choose the time increment \( \Delta t = 0.005 \) sec, the acoustic velocity \( c = 4 \) km/sec, and the spatial increments \( \Delta x = \Delta y = \Delta z = 50 \) m resulting in the number of grid points per wavelength \( G \approx 2.7 \). The computational domain is \( 0 \leq x \leq 3 \) km, \( 0 \leq y \leq 3 \) km, and \( 0 \leq z \leq 3 \) km, and the number of grid points is \( 61 \times 61 \times 61 \). The snapshots of seismic wave fields at \( t = 0.3 \) sec in the \( xy \) plane, modeled by the ONADM and the fourth-order LWC, are presented in Figure 5. We can see that the wavefronts of seismic waves simulated by two kinds of methods at the same time are basically identical (see Fig. 5a and b), and it took the ONADM and the LWC about 7 and 6 sec to generate Figure 5a and b, respectively. However, the snapshot in Figure 5b simulated by the fourth-order LWC presents strong numerical dispersion, and the corresponding result in Figure 5a computed by the ONADM shows that the ONADM has almost no numerical dispersion even if the space increment chosen is 50 m without any additional treatments. It indicates that the ONADM enables wave propagation to be simulated in 3D models through using the coarse computation grids.

To further test the efficiency of the ONADM, the snapshot, computed by the fourth-order LWC on the fine-grid step (\( \Delta x = \Delta y = \Delta z = 10 \) m and \( \Delta t = 0.001 \) sec) so that the grid dispersion caused by the LWC is eliminated, is shown in Figure 6, whereas Figure 5a is generated on the coarse-

### Table 1

<table>
<thead>
<tr>
<th>Case</th>
<th>( h )</th>
<th>( \Delta t )</th>
<th>Second-Order FD</th>
<th>Fourth-Order LWC</th>
<th>ONADM</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10 m</td>
<td>( 10^{-4} ) sec</td>
<td>12.5322</td>
<td>0.0706</td>
<td>0.0528</td>
</tr>
<tr>
<td>2</td>
<td>20 m</td>
<td>( 3 \times 10^{-4} ) sec</td>
<td>65.8149</td>
<td>1.5501</td>
<td>0.8599</td>
</tr>
<tr>
<td>3</td>
<td>30 m</td>
<td>( 5 \times 10^{-4} ) sec</td>
<td>130.9485</td>
<td>7.4675</td>
<td>3.7395</td>
</tr>
</tbody>
</table>

This table compares numerical results computed using the ONADM and the LWC method for different cases and different methods.
grid step ($\Delta x = \Delta y = \Delta z = 50$ m and $\Delta t = 0.005$ sec) resulting in the same Courant number. Figures 5a and 6 show, in general, that the ONADM on a coarse grid can provide an identical result to the LWC on a finer grid. In other words, the ONADM on a coarse grid can obtain, in general, the identical solution because the high-order LWC was verified to be convergent (Dablain, 1986), but it needs less memory and lower computational costs. Actually, it took the ONADM about 7 sec CPU time to generate Figure 5a, whereas generating Figure 6 under the same computer environment costs the LWC roughly 3910 sec. In other words, the computational speed of the ONADM for generating Figure 5a is about 559 times the fourth-order LWC on a fine grid to achieve the same accuracy as that of the ONADM.

Note that the storage space required for computation in the ONADM is also different from that of the LWC method. The ONADM needs 12 arrays to store wave displacement $u_{i,j,k}^{n+1}$, $u_{i,j,k}^{n}$, and its gradients at each spatial grid point, and the number of grid points is $61 \times 61 \times 61$ on a coarse grid for generating Figure 5a. The fourth-order LWC needs only three arrays to store the wave displacement at each grid point, but the number of grid points on a fine grid for generating Figure 6 is $301 \times 301 \times 301$ for the fourth-order LWC. It indicates that the ONADM requires only about 3.3% of the storage space used by the fourth-order LWC. When computing a value at a grid point, the ONADM involves only 3 grid points in a direction whereas the fourth-order LWC needs 5 grid points (Dablain, 1986). The demand of more grids in high-order LWC methods prevents the algorithms from efficient parallel implementation and artificial boundary treatment.

Wave Fields Modeling

Isotropic Model. To investigate the validity of the ONADM for the 3D elastic isotropic case, the elastic constants are chosen by $\lambda = 4.75$ GPa, $\mu = 3.75$ GPa, and the density $\rho = 2.1$ g/cm$^3$. The spatial and time increments are $\Delta x = \Delta y = \Delta z = 30$ m and $\Delta t = 0.003$ sec, respectively, and the number of grid points is $101 \times 101 \times 101$. The rest parameters are the same as those used in the acoustic experiment previously. For this case the number of grid points per $\lambda$ wavelength is about 2.7.

Figure 7 shows the snapshots of the $u_1$ component in the $x$ direction at $t = 0.45$ sec on a coarse grid ($\Delta x = \Delta y = \Delta z = 30$ m), generated by the ONADM, where Figure 7a, b, and c present the wave-field snapshots of the $u_1$ component in $xy$, $xz$, and $yz$ planes, respectively. In Figure 7 the snapshots in $xy$ and $xz$ planes show very clear wavefronts of the $P$ and $S$ waves and the snapshot (Fig. 7c) in the vertical plane ($yz$ plane) for the $x$-direction component of the displacement shows very weak $P$ wave and strong $S$ wave. The same waves can be found from the wave-field snapshots of $u_2$ and $u_3$ components, which are omitted here. The computation of generating the results in Figure 7 were performed on a Pentium 4 with 512 MB memory, and it took about 285 min.
VTI Model. In this example, we choose the transversely isotropic model with a vertical symmetry axis (VTI) and medium parameters listed in Table 2. The spatial and time increments are $\Delta x = \Delta y = \Delta z = 20$ m and $\Delta t = 0.002$ sec, respectively. The computational domain is $0 \leq x \leq 2$ km, $0 \leq y \leq 2$ km, and $0 \leq z \leq 2$ km, and the rest computational parameters and computer environment are the same as those in isotropic model.

The wave-field snapshots for three components $u_1$, $u_2$, and $u_3$ at time 0.3 sec are given in Figures 8, 9, and 10: Figure 8 presents the snapshots of the $u_1$ component in $xy$, $xz$, and $yz$ planes, whereas Figures 9 and 10 show the snapshots of displacement components $u_2$ and $u_3$ in three planes. The wave-field snapshots in the $xy$ plane (transverse plane) for the displacement three components, shown in Figures 8a, 9a, and 10a, show that the wavefronts of $P$ and $S$ waves are a circle in the VTI medium, whereas other snapshots in Figures 8, 9, and 10 show that the wavefronts of $P$- and $S$-waves are an ellipse and the $qP$ and $qSV$ waves show the directional dependence on propagation velocity. The $qSV$ wavefronts can have cusps and triplications depending on the value of $c_{13}$ (Faria and Stoffa, 1994). Triplications can be observed in the horizontal component $qSV$ wavefronts (shown in Figs. 8b and 9c) in the $xz$ plane for the $u_1$ component, in the $yz$ plane for the $u_2$ component, and in the vertical component $qSV$ wavefronts presented in Figure 10b and c. Moreover, we can also observe that the arrival times of $qSH$ and $qSV$ waves and shear-wave splitting in the VTI medium are different through comparing Figures 8c and 9b with Figures 8b, 9c, 10b, and 10c. On a Pentium 4 with 512 MB memory, it took the ONADM about 305 min to generate Figures 8, 9, and 10.

Two-Layered Model. To further investigate the efficiency of the 3D ONADM, we choose a two-layer medium model with acoustic wave velocities 2 km/sec and 4 km/sec in the upper and lower layers, respectively. The number of grid points is $151 \times 151 \times 151$, the model size is 4.5 km $\times$ 4.5 km $\times$ 4.5 km, and the source is located at $(x_s, y_s, z_s) = (2.25$ km, $2.25$ km, $1.95$ km). The time increment is $\Delta t = 0.003$ sec, and the spatial increments are $\Delta x = \Delta y = \Delta z = 30$ m, resulting in the number of grid points per minimum wavelength of about 2.2.

Figure 11 is the wave-field snapshot at $t = 0.6$ sec in the $xz$ plane, generated by the 3D ONADM. From the wave-field snapshot, we can clearly observe the reflection of the acoustic wave from the inner interface. The wave-field snapshot also shows that the ONADM has almost no numerical dispersions without any additional treatments, even if the number of grid points per minimum wavelength is about 2.2 and the model velocity contrasts between adjacent layers are two times. The computation of generating the result in Figure 11 was performed on PC clusters with 1 GB memory, and it took the ONADM about 9 min to finish the job.

Discussion and Conclusions

The ONADM for 3D elastic media (including isotropic and anisotropic cases), which is an extension of the 2D ONADM for solving acoustic and elastic wave equations (Yang et al., 2004), is developed via the Taylor series expansion and the high-degree multivariable interpolation approximation, that is, the time derivatives are approximated analytically by a truncated Taylor series and the high-order space derivatives are calculated using the multivariable interpolation approximation. On the basis of such a structure,
Figure 8. Snapshots of seismic wave fields at time 0.3 sec for the $x$ direction displacement component ($u_1$) in the VTI medium, generated by the ONADM, for $xy$ plane (a), $xz$ plane (b), and $yz$ plane (c).

Figure 9. Snapshots of seismic wave fields at time 0.3 sec for the $y$-direction displacement component ($u_2$) in the VTI medium, generated by the ONADM, for $xy$ plane (a), $xz$ plane (b), and $yz$ plane (c).

Figure 10. Snapshots of seismic wave fields at time 0.3 sec for the $z$-direction displacement component ($u_3$) in the VTI medium, generated by the ONADM, for $xy$ plane (a), $xz$ plane (b), and $yz$ plane (c).
we have to first convert these high-order time derivatives to the spatial derivatives, which is similar to the high-order FD or so-called LWC methods (Lax and Wendroff, 1964; Dablain, 1986). However, the ONADM in approximating the high-order spatial derivatives is different from these high-order FD, LWC, and compact FD methods stated previously that only use the wave displacement at some grid points to approximate the high-order spatial derivatives or directly discretizing the original wave equation. This ONADM uses simultaneously both the wave displacement and its gradients to approximate the high-order derivatives (see formulae [A1] to [A14]). In other words, when computing \( U_{i,j,k}^{n+1} \), the ONADM uses not only the values of the displacement \( U \) at the mesh point \((i, j, k)\) and its neighboring grid points, but also the values of the displacement gradients. As a result, the ONADM retains more wave-field information in both the function \( U \) and its gradient \( \nabla U \). Therefore, the ONADM effectively suppress the loss of wave-field information included in the higher-order terms of the Taylor expansion, further resulting in fewer numerical dispersions (Yang et al., 2004) and great numerical accuracy because of the small coefficient \((1/360)\) presented in the theoretical error term (see Appendix D in Yang et al., 2006). In fact, the ONADM has the same theoretical accuracy as the fourth-order LWC scheme; the numerical error of the ONADM is less than those of the conventional second-order FDM and fourth-order LWC for the three models that we presented in Figures 2, 3, and 4. Wave-field modeling illustrates that the ONADM can effectively suppress numerical dispersions even for the fewer space-sampling points (e.g., 2.2 points in our previous experiments) per minimum wavelength and larger velocity contrasts (two times in previous two-layered model) between adjacent layers. These numerical results also imply that simultaneously using both the wave displacement and its gradients to approximate the high-order derivatives is important for decreasing the numerical dispersion caused by the discretization of wave equations because wave displacement gradients include important wave-field information. On the other hand, using these connection relations such as equations (6b)–(6d), (6f)–(6h), and those omitted in our present paper between the grid point \((i, j, k)\) and its neighboring nodes \((i+p, j+q, k+r)\) \((p, q, r = -1, 0, 1)\) keeps the continuity of gradients. The continuity and high accuracy (fourth-order accuracy in space) of gradients improve automatically the continuity of the stresses that are the linear combinations of gradients or the Hook sum, further resulting in the ONADM having less numerical dispersion. It suggests that we should consider the wave-gradient field and the use of connection relations such as equations (6b)–(6d), (6f)–(6h), and so on, as we design a new numerical method to solve the 3D acoustic and elastic wave equations.

It appears that the CPU time of the ONADM is more than that of the high-order LWC, but in fact, because the ONADM yields less numerical dispersion than the LWC with fourth-order accuracy, we can afford to increase the time increment through adopting coarser spatial increments to achieve the same accuracy as that of the LWC on a finer grid with smaller timesteps. Hence, the total CPU time of the ONADM does not exceed that of the LWC. Our numerical experiments illustrate that the computational speed of the ONADM is about 559 times of the fourth-order LWC on a finer grid to achieve the same accuracy as that of the ONADM. In addition, the ONADM involves the added gradient fields, resulting in added storage space, but the required memory of the ONADM is actually about 3.3% of the storage space of the fourth-order LWC on a finer grid to obtain the same accuracy as that of the ONADM on a coarser grid.

Provided a suitable time increment, not only stable calculations can be kept, but computational speed can be improved while the spatial increment is a constant. We have obtained the stability criterion (9) based on the Fourier analysis method (Richtmyer and Morton, 1967) when the ONADM is applied to numerically solve the 3D acoustic and elastic equations. The stability condition for a heterogeneous medium cannot be directly determined by (9) and (10), but could be approximated by using a local homogeneous method. Our conjecture is that criterion (10) is approximately correct for a heterogeneous medium if the maximal value \( c_{\text{max}} \) of the wave velocity is used.

Also, consider local difference operators for parallel computing because the nearest-neighbor communication is extremely fast, and 3D large anisotropic models are feasible because of the intrinsic parallelism of the conservation equations. As we have seen, the ONADM is very fast and requires a small amount of storage space as only three nearest grid points are involved in a direction while the LWC and high-order compact methods need more grid points in a direction. The property of the ONADM requiring fewer grid points is very useful to both implementing the parallel calculations and treating with the artificial boundary. In fact, we easily find from the structure of the ONADM that the computational cost of the ONADM depends mainly on calculating these...
values of both \( \frac{d^{l+m+s}}{dx^{m}y^{n}z^{o}} U_{i,j,k} \) (\( 2 \leq l + m + s \leq 5 \)), and the computations of partial derivatives for different displacement components are decoupled and can be simultaneously computed. Therefore, further increases in efficiency are feasible through parallel computer program so that the computations can be performed on multiple workstations or on a parallel computer. These problems and other problems such as 3D absorbing boundary conditions and 3D heterogeneous ONADM will be in our forthcoming study.

Acknowledgments

We thank Fred F. Hollitz (associate editor) and two anonymous reviewers for their constructive comments and helpful suggestions that greatly contributed to improving the manuscript. This work was supported by the National Natural Sciences Foundation of China (Grant 40574014) and the Foundation of Tsinghua University (Grant 49705057).

References


Dablain, M. A. (1986). The application of high-order differentiating to the scalar wave equation, Geophysics 51, 54-66.


Appendix

Evaluation of High-Order Derivatives

For simplicity, we define the unchanged operator I, second central difference operator \( \delta^2 \), and the displacement operator \( \mathcal{E} \) as follows:

\[
\begin{align*}
U_{i,j,k}^m &= U_{i,j,k}^m, \\
\delta^2 U_{i,j,k}^m &= U_{i+1,j+1,k+1}^m - 2U_{i,j,k}^m + U_{i-1,j-1,k-1}^m, \\
\mathcal{E}_x U_{i,j,k}^m &= U_{i+1,j,k}^m - U_{i-1,j,k}^m, \\
\mathcal{E}_y U_{i,j,k}^m &= U_{i,j+1,k}^m - U_{i,j-1,k}^m, \\
\mathcal{E}_z U_{i,j,k}^m &= U_{i,j,k+1}^m - U_{i,j,k-1}^m.
\end{align*}
\]

Using the interpolation function (5) and the interpolation
relations from equations (6a) to (6h) and those omitted equations, we can determine the high-order spatial derivatives expressed by the displacement and its gradient as follows:

\[
\begin{aligned}
\left( \frac{\partial^2 U}{\partial g^2} \right)_{i,j,k}^n &= \frac{2}{\Delta g^2} \delta^2_{i,j,k} U_{i,j,k}^m - \frac{1}{2\Delta g} \left( E_g^1 - E_g^{-1} \right) \left( \frac{\partial U}{\partial g} \right)_{i,j,k}^n, \quad g = x, y, z, \text{ and (A1)}
\end{aligned}
\]

\[
\begin{aligned}
\left( \frac{\partial^2 U}{\partial \varnothing^2} \right)_{i,j,k}^n &= \frac{1}{\Delta \varnothing^2} \left( E_{\varnothing}^1 - E_{\varnothing}^{-1} \right) \left( \frac{\partial U}{\partial \varnothing} \right)_{i,j,k}^n \\
&+ \frac{1}{2\varnothing} \left( E_{\varnothing}^1 - E_{\varnothing}^{-1} \right) \left( \frac{\partial U}{\partial \varnothing} \right)_{i,j,k}^n - \frac{1}{4\Delta \varnothing \Delta \varnothing} \left( E_{\varnothing}^1 \right) \left( \frac{\partial U}{\partial \varnothing} \right)_{i,j,k}^n \\
&+ E_g^{-1} E_g^{-1} - E_{\varnothing}^1 E_{\varnothing}^{-1} - E_{\varnothing}^1 E_{\varnothing}^{-1} U_{i,j,k}^m, \quad \text{corresponding to three cases: } g = x \text{ and } e = y, g = y \text{ and } e = z, \text{ and } g = z \text{ and } e = x, \text{ and defining } \left( \frac{\partial^3 U}{\partial g^2 \partial \varnothing} \right)_{i,j,k}^n = \left( \frac{\partial^3 U}{\partial \varnothing^2 \partial \varnothing} \right)_{i,j,k}^n. 
\end{aligned}
\]

\[
\begin{aligned}
\left( \frac{\partial^3 U}{\partial g^2 \partial \varnothing} \right)_{i,j,k}^n &= \frac{15}{2\Delta g} \left( E_{\varnothing}^1 - E_{\varnothing}^{-1} \right) \left( \frac{\partial U}{\partial g} \right)_{i,j,k}^n - \frac{3}{2\Delta g^2} \left( E_{\varnothing}^1 \right) \left( \frac{\partial U}{\partial g} \right)_{i,j,k}^n + 8i + E_g^{-1} \left( \frac{\partial U}{\partial g} \right)_{i,j,k}^n, \quad g = x, y, z, \text{ and (A3)}
\end{aligned}
\]

\[
\begin{aligned}
\left( \frac{\partial^3 U}{\partial g^2 \partial \varnothing^2} \right)_{i,j,k}^n &= \frac{1}{2\Delta g \Delta \varnothing} \left( - E_{\varnothing}^1 E_{\varnothing}^{-1} - E_g^{-1} E_{\varnothing}^{-1} \right) \\
&+ \frac{1}{\Delta \varnothing} \left( E_{\varnothing}^1 + E_{\varnothing}^{-1} \right) \left( \frac{\partial U}{\partial g} \right)_{i,j,k}^n + \frac{1}{\Delta g} \left( \frac{\partial U}{\partial \varnothing} \right)_{i,j,k}^n \\
&+ \frac{1}{4\Delta g \Delta \varnothing} \left( 5E_g^1 \right) \left( 5E_{\varnothing}^1 \right) - 5E_g E_{\varnothing}^{-1} \left( E_{\varnothing}^1 E_{\varnothing}^{-1} - E_g^{-1} E_{\varnothing}^{-1} \right) \\
&+ E_g^1 E_{\varnothing}^{-1} - E_g^{-1} E_{\varnothing}^1 - 6E_g^1 \\
&+ 6E_g^{-1} - 4E_g^1 + 4E_g^{-1} U_{i,j,k}^m, \quad \text{corresponding to three cases: } g = x, e = y, \text{ and } d = z; g = y, e = z, \text{ and } d = x; \text{ and } g = z, e = x, \text{ and } d = y. \text{ And defining}
\end{aligned}
\]

\[
\begin{aligned}
\left( \frac{\partial^3 U}{\partial \varnothing \partial \varnothing^2} \right)_{i,j,k}^n &= \frac{15}{2\Delta g} \left( E_g^1 \right) \left( \frac{\partial U}{\partial \varnothing} \right)_{i,j,k}^n + \frac{1}{4\Delta \varnothing \Delta \varnothing} \left( E_g^1 \right) \left( \frac{\partial U}{\partial \varnothing} \right)_{i,j,k}^n - \frac{3}{2\Delta g^2} \left( E_g^1 \right) \left( \frac{\partial U}{\partial \varnothing} \right)_{i,j,k}^n + 8i + E_g^{-1} \left( \frac{\partial U}{\partial \varnothing} \right)_{i,j,k}^n, \quad g = x, y, z, \text{ and (A3)}
\end{aligned}
\]

\[
\begin{aligned}
\left( \frac{\partial^3 U}{\partial g \partial \varnothing^2} \right)_{i,j,k}^n &= \frac{1}{2\Delta g \Delta \varnothing} \left( - E_{\varnothing}^1 E_{\varnothing}^{-1} - E_g^{-1} E_{\varnothing}^{-1} \right) \\
&+ \frac{1}{\Delta \varnothing} \left( E_{\varnothing}^1 + E_{\varnothing}^{-1} \right) \left( \frac{\partial U}{\partial g} \right)_{i,j,k}^n + \frac{1}{\Delta g} \left( \frac{\partial U}{\partial \varnothing} \right)_{i,j,k}^n \\
&+ \frac{1}{4\Delta g \Delta \varnothing} \left( 5E_g^1 \right) \left( 5E_{\varnothing}^1 \right) - 5E_g E_{\varnothing}^{-1} \left( E_{\varnothing}^1 E_{\varnothing}^{-1} - E_g^{-1} E_{\varnothing}^{-1} \right) \\
&+ E_g^1 E_{\varnothing}^{-1} - E_g^{-1} E_{\varnothing}^1 - 6E_g^1 \\
&+ 6E_g^{-1} - 4E_g^1 + 4E_g^{-1} U_{i,j,k}^m, \quad \text{corresponding to three cases: } g = x, e = y, \text{ and } d = z; g = y, e = z, \text{ and } d = x; \text{ and } g = z, e = x, \text{ and } d = y. \text{ And defining}
\end{aligned}
\]
\[
\left( \frac{\partial^2 U}{\partial g^2 \partial e} \right)_{i,j,k}^n = \frac{6}{(\Delta g)^3(\Delta e)} \left( E_g^3 E_e^3 + E_e^{-1} E_g^{-1} E_d^{-1} \right) \\
+ 2\delta_e^2 - E_g^2 - E_g^{-1}(\frac{\partial U}{\partial g})_{i,j,k}^n \\
- \frac{3}{(\Delta g)^4} (5E_g^3 E_e^3 - 5E_e^{-1} E_g^{-1} E_d^{-1}) \\
+ E_g^4 E_e^{-1} - E_g^{-1} E_e^{-1} E_d^{-1} \\
+ 6E_e^{-1} - 4E_e^{-1} + 4E_e^{-1} U_{i,j,k}^n \quad \text{and} \\
\left( \frac{\partial^2 U}{\partial g^2 \partial e^2} \right)_{i,j,k}^n = \frac{3}{(\Delta g)^3(\Delta e)} \left( E_g^3 - E_g^{-1} \right) \delta_e^2 U_{i,j,k}^n \\
- \frac{6}{(\Delta g)^3(\Delta e)^2} \delta_e^2 \left( \frac{\partial U}{\partial g} \right)_{i,j,k}^n, \quad (A12)
\]

where \( g, e = x, y, \text{ and } z, \) \( g \neq e, \) and defining

\[
\left( \frac{\partial^2 U}{\partial g^2 \partial e^2} \right)_{i,j,k}^n = \left( \frac{\partial^2 U}{\partial g^2 \partial e^3} \right)_{i,j,k}^n, \quad \left( \frac{\partial^2 U}{\partial g^2 \partial e^2} \right)_{i,j,k}^n = \left( \frac{\partial^2 U}{\partial e^2 \partial e} \right)_{i,j,k}^n.
\]

\[
\left( \frac{\partial^2 U}{\partial g^2 \partial e} \right)_{i,j,k}^n = \frac{3}{4(\Delta g)^3(\Delta e)(\Delta d)} \left( E_g E_e E_d^3 - E_e^{-1} E_g^{-1} E_d^{-1} \right) \\
+ E_g^{-1} E_e^{-1} E_d^{-1} - E_g^2 E_e E_d^{-1} \\
+ E_g^{-1} E_e E_d^{-1} - E_e^2 E_g E_d^{-1} \\
+ E_g E_e^{-1} E_d^{-1} - E_e^2 E_g^2 E_d U_{i,j,k}^n \\
- \frac{3}{2(\Delta g)^3(\Delta e)(\Delta d)} \left( E_g E_d + E_e E_d^{-1} \right) \\
- E_g E_d^{-1} - E_e E_d^{-1} \left( \frac{\partial U}{\partial g} \right)_{i,j,k}^n \quad \text{and} \\
\left( \frac{\partial^2 U}{\partial g^2 \partial e} \right)_{i,j,k}^n = \frac{1}{(\Delta g)(\Delta e)^2} \delta_e^2 \left( \frac{\partial U}{\partial d} \right)_{i,j,k}^n, \quad (A14)
\]

\[
(E_g^3 + E_g^{-1} - 2I) \delta_e^2 \left( \frac{\partial U}{\partial d} \right)_{i,j,k}^n,
\]

corresponding to three cases: \( g = x, e = y, \) and \( d = z; \) \( g = y, e = x, \) and \( d = z; \) and \( g = z, e = x, \) and \( d = y. \)

And defining \( \left( \frac{\partial^2 U}{\partial \alpha \partial \beta \partial \gamma} \right)_{i,j,k}^n = \left( \frac{\partial^2 U}{\partial \alpha \partial \beta \partial \gamma} \right)_{i,j,k}^n, \)
\( \left( \frac{\partial^2 U}{\partial \alpha \partial \beta \partial \gamma} \right)_{i,j,k}^n = \left( \frac{\partial^2 U}{\partial \alpha \partial \beta \partial \gamma} \right)_{i,j,k}^n. \)