Finite Difference Method for Modeling Seismic Wave Propagation

Sarah Verros

April 28, 2015

Abstract

The purpose of this project is to model the propagation of seismic waves in a homogeneous, elastic medium. The method that will be used to accomplish this will be a second order finite difference method that has been derived to handle mixed spatial derivatives. The problem will be in a 3D domain and will simulate waves propagating in materials such as sandstone and peridotite to determine how the material properties affect wave propagation.

1 Introduction

Seismic events may cause strong motion shaking in the form of a propagating wave in a wide radius around the source. Capturing the motion of these waves on the earth can provide valuable information on the material properties of the earth or even give information for how the earth will shake in future earthquake scenarios. Numerical methods provide a tool to approximate the motion of the earth during an earthquake and can be useful in studying the structure of the earth and various seismic phenomena. One such numerical method is the finite difference method (FDM) which approximates the solutions to differential equations and may give close solutions to a wide variety of partial differential equations. The focus of this project will be to capture the displacement of the earth during a seismic event using a finite-difference time-domain method to approximate the solution to the equation of motion along with a constitutive relation.

2 Background

Seismic events, such as earthquakes, can be powerful and destructive sources of energy. Although we now have tools to measure the ground motion intensity during seismic events, there is still no method of earthquake prediction and in most cases, little can be done to prevent loss near the source once the event has occurred. Understanding how seismic waves move through different materials can prevent loss of life and property by providing information on how and where to construct buildings, homes, and other structures. It is now known that the areas with the highest population density tend to be the areas impacted the most when a seismic event occurs. This is because large cities are most often built on soft soil, which amplifies the ground motion shaking. Thus, understanding how the seismic waves travel through the earth can determine what safety precautions should be taken for specific regions.

The modeling of seismic waves on the earth was first approximated using FDMs in the 1960's. Since then, several methods have been developed that accommodate for different sources, material properties, attenuation, and boundary conditions, each capable of giving valuable information about the earth and seismic phenomena. The developed finite difference methods provide approximations to the equation of motion given by

$$\rho \ddot{u}_i - \sigma_{ij,j} - f_i = 0$$

with boundary condition

$$p_i = \sigma_{ij} n_j$$

where ρ is the material density, $\ddot{u}_i$ represents the particle acceleration in the ith component, $\sigma_{ij,j}$ represents partial of the stress tensor with respect to the jth component, p_i is the traction vector, and n_j is a normal vector. Applying the equation of motion requires that we have information on how the stress and strain of the medium are related. These relations are called constitutive laws. For this problem we will use Cauchy's generalization of Hooke's law given by

$$\sigma_{ij} = \kappa\epsilon_{kk}\delta_{ij} + 2\mu(\epsilon_{ij} - \frac{1}{3}\epsilon_{kk}\delta_{ij})$$

where κ, μ are the bulk and shear moduli, ϵ is the strain, and δ_{ij} is the Kronecker delta function. Since λ, the elastic modulus, is equivalent to $\lambda = \kappa - \frac{2}{3}\mu$, this expression may also be written in the form

$$\sigma_{ij} = \lambda\epsilon_{kk}\delta_{ij} + 2\mu\epsilon_{ij}$$

Combining the equation of motion and the constitutive law, we may solve the equation of motion for the parameter of interest, in this case the displacement, to get that

$$\rho\ddot{u}_i = [(\kappa - \frac{2}{3}\mu)u_{k,k}]_{,i} + (\mu u_{i,j})_{,j} + (\mu u_{j,i})_{,j} + f_i$$

where for component i, $j, k \in i, j, k$. Using this equation we seek to model the behavior of the displacement in a homogeneous material using finite differences as described in the next section.

3 Finite Difference Methods

In order to approximate solutions to the equation for displacement derived from the equation of motion, we will use a second order finite-difference time-domain method on a standard grid. The functions we are looking to approximate are the components of the displacement vector (u, v, w) which we will denote U, V, and W for the x, y,and z components respectively. Then we will approximate each of these functions at the spatial coordinates (x, y, z) and at each time t. To decimate these values, for $x \in [\alpha_x, \beta_x], y \in [\alpha_y, \beta_y], z \in [\alpha_z, \beta_z]$, let

$$\alpha_x = x_1 < x_2 < \cdots < x_M = \beta_x$$

$$\alpha_y = y_1 < y_2 < \cdots < y_N = \beta_y$$

$$\alpha_z = z_1 < z_2 < \cdots < z_K = \beta_z$$

where M, N, K are the number of grid points in each direction and the spatial step is h for all coordinates. Additionally, for $t \in [0, T]$ let

$$0 = t_1 < t_2 < \cdots < t_L = T$$

where L is the number of time steps and Δt is the temporal step.

We will define the following notation:

$$\phi(x_i, y_j, z_k, t_m) = \phi^m_{i,j,k}$$

where $\phi(x_i, y_j, z_k, t_m)$ represents the value of function $\phi \in U, V, W$ at spatial coordinates $x = x_i, y = y_j, z = z_k$ and at time $t = t_m$.

The FDM approximations that we will use are the central difference approximation to first derivatives and the second order approximation to the second derivative. These are given by

$$\frac{d\phi}{dx}(x) = \frac{\phi(x+h) - \phi(x-h)}{2h}$$

and

$$\frac{d^2\phi}{dx^2}(x) = \frac{\phi(x-h) - 2\phi(x) + \phi(x+h)}{h^2}$$

where $\phi \in \{U, V, W\}$ and h is the spatial step. Using these approximations we will define the FDM in the next section.

4 Derivation of the Method

We are considering deriving a finite difference method for the following equation:

$$\rho\ddot{u}_i = [(\kappa - \frac{2}{3}\mu)u_{k,k}]_{,i} + (\mu u_{i,j})_{,j} + (\mu u_{j,i})_{,j} + f_i$$

where u is the displacement, ρ is the density, κ and μ are Lame parameters, and f is the forcing function. The notation, $_{,i}$ denotes the partial derivative with respect to the i^{th} component, and $\ddot{u}$ represents the second temporal derivative of the function u. We will use the second order approximation for the time derivative described by equation CITE. In order to approximate the spatial derivatives, we must find a scheme that will account for the mixed spatial derivatives present in the equation.

We begin by defining an auxiliary function,

$$\phi = \mu u_{,x}$$

and seek to define $\phi_{,z}$ the mixed derivative and $\phi_{,x}$ the non-mixed derivative. Using the central difference formula, we have a second order approximation for $\phi_{,x}$ as

$$\phi_{,x}|_{i,j,k} = \frac{\phi_{i+1/2,j,k} + \phi_{i-1/2,j,k}}{h}$$

using half spatial steps, h. To find the appropriate approximations for $\phi_{i+1/2,j,k}$ and $\phi_{i-1/2,j,k}$, we will take the following steps:

$$\phi = \mu u_{,x} \Leftrightarrow \frac{\phi}{\mu} = u_{,x}$$

Integrating with respect to x gives

$$\int_{x_{i,j,k}}^{x_{i+1,j,k}} \frac{\phi}{\mu} dx = \int_{x_{i,j,k}}^{x_{i+1,j,k}} u_{,x} dx$$

Using the mean value theorem, we will approximate $\int_{x_{i,j,k}}^{x_{i+1,j,k}} \phi dx$ with $\phi_{i+1/2,j,k}$ and thus the integral becomes

$$\phi_{i+1/2,j,k} \int_{x_{i,j,k}}^{x_{i+1,j,k}} \frac{1}{\mu} dx = U_{i+1,j,k} - U_{i,j,k}.$$

We will define

$$\mu^x_{i+1/2,j,k} = [\frac{1}{h} \int_{x_{i,j,k}}^{x_{i+1,j,k}} \frac{1}{\mu} dx]^{-1}.$$

Then by dividing we get

$$\phi_{i+1/2,j,k} = \frac{1}{h} \mu^x_{i+1/2,j,k} (U_{i+1,j,k} - U_{i,j,k}).$$

Similarly we have that

$$\phi_{i-1/2,j,k} = \frac{1}{h} \mu^x_{i-1/2,j,k} (U_{i,j,k} - U_{i-1,j,k}).$$

Thus we have derived a form for the non-mixed spatial derivative, $\phi_{,x}$. To obtain an approximation for the mixed spatial derivative, $\phi_{,z}$ we begin with the second order, central difference formula

$$\phi_{,z}|_{i,j,k} = \frac{\phi_{i,j,k+1/2} - \phi_{i,j,k-1/2}}{h}$$

where $\phi = \mu u_{,x}$. Now we seek to find approximations to $\phi_{i,j,k+1/2}$ and $\phi_{i,j,k-1/2}$. Then

$$\int_{z_{i,j,k-1/2}}^{z_{i,j,k+1/2}} \frac{\phi}{\mu} dz = \int_{z_{i,j,k-1/2}}^{z_{i,j,k+1/2}} u_{,x} dz.$$

We will define

$$\mu^z_{i,j,k+1/2} = [\frac{1}{h} \int_{z_{i,j,k}}^{z_{i,j,k+1}} \frac{1}{\mu} dz]^{-1}.$$

Then using the mean value theorem and approximating the right hand side, we have

$$\frac{h}{\mu^z_{i,j,k+1/2}}\phi_{i,j,k+1/2} = hu_{,x}|_{i,j,k+1/2}.$$

We will approximate the derivative $u_{,x}|_{i,j,k+1/2}$ using

$$u_{,x}|_{i,j,k+1/2} = \frac{1}{4h}[U_{i+1,j,k+1} - U_{i-1,j,k+1} + U_{i+1,j,k-1} - U_{i-1,j,k-1}]$$

Finally, combining these we have that

$$\phi_{i,j,k+1/2} = \frac{1}{4h}\mu_{i,j,k+1/2}(U_{i+1,j,k+1} - U_{i-1,j,k+1} + U_{i+1,j,k-1} - U_{i-1,j,k-1}).$$

Now we will use these to find the approximation for

$$\rho\ddot{u}_i = [(\kappa - \frac{2}{3}\mu)u_{k,k}]_{,i} + (\mu u_{i,j})_{,j} + (\mu u_{j,i})_{,j} + f_i.$$

Then employing the approximation for the temporal derivative,

$$\rho\ddot{u}_i = \rho\frac{U^{m+1}_{i,j,k} - 2U^m_{i,j,k} + U^{m-1}_{i,j,k}}{dt^2}$$

and $[(\kappa - \frac{2}{3}\mu)u_{k,k}]_{,i} + (\mu u_{i,j})_{,j} + (\mu u_{j,i})_{,j} + f_i$ is approximated by

$$\begin{aligned}
&[(\kappa - \frac{2}{3}\mu)u_{k,k}]_{,i} + (\mu u_{i,j})_{,j} + (\mu u_{j,i})_{,j} + f_i = \\
&\qquad \frac{1}{h^2}[\lambda^x_{i+1,j,k}(U^m_{i+1,j,k} - U^m_{i,j,k}) - \lambda^x_{i-1,j,k}(U^m_{i,j,k} - U^m_{i-1,j,k}) \\
&\qquad + 2(\mu^x_{i+1,j,k}(U^m_{i+1,j,k} - U^m_{i,j,k}) - \mu^x_{i-1,j,k}(U^m_{i,j,k} - U^m_{i-1,j,k})) \\
&\qquad + \mu^y_{i,j+1,k}(U^m_{i,j+1,k} - U^m_{i,j,k}) - \mu^y_{i,j-1,k}(U^m_{i,j,k} - U^m_{i,j-1,k}) \\
&\qquad + \mu^z_{i,j,k+1}(U^m_{i,j,k+1} - U^m_{i,j,k}) - \mu^z_{i,j,k-1}(U^m_{i,j,k} - U^m_{i,j,k-1}) \\
&\qquad + \frac{1}{4}[\lambda^y_{i,j+1,k}(V^m_{i,j+1,k} + V^m_{i+1,j+1,k} - V^m_{i,j-1,k} - V^m_{i+1,j-1,k}) \\
&\qquad - \lambda^y_{i,j-1,k}(V^m_{i-1,j+1,k} + V^m_{i,j+1,k} - V^m_{i-1,j-1,k} - V^m_{i,j-1,k}) \\
&\qquad + \lambda^z_{i,j,k+1}(W^m_{i,j,k+1} + W^m_{i+1,j,k+1} - W^m_{i,j,k-1} - W^m_{i+1,j,k-1}) \\
&\qquad - \lambda^z_{i,j,k-1}(W^m_{i-1,j,k+1} + W^m_{i,j,k+1} - W^m_{i-1,j,k-1} - W^m_{i,j,k-1}) \\
&\qquad + \mu^y_{i+1,j,k}(V^m_{i+1,j,k} + V^m_{i+1,j+1,k} - V^m_{i-1,j,k} - V^m_{i-1,j-1,k}) \\
&\qquad - \mu^y_{i-1,j,k}(V^m_{i+1,j-1,k} + V^m_{i+1,j,k} - V^m_{i+1,j-1,k} - V^m_{i-1,j,k}) \\
&\qquad + \mu^z_{i+1,j,k}(W^m_{i+1,j,k} + W^m_{i+1,j,k+1} - W^m_{i-1,j,k} - W^m_{i-1,j,k-1}) \\
&\qquad - \mu^z_{i-1,j,k}(W^m_{i-1,j,k+1} + W^m_{i,j,k+1} - W^m_{i-1,j,k-1} - W^m_{i,j,k-1})]] \\
&\qquad + F^{x,m}_{i,j,k}
\end{aligned}$$

where $\kappa =$, We will take advantage of the following notation:

$$L_{\gamma,\gamma}(a,\phi) = \frac{1}{h^2}[a^{\gamma+}(\phi^m_+ - \phi^m) - a^{\gamma-}(\phi^m - \phi^m_-)]$$

where $\gamma \in \{x,y,z\}$, $a \in \{\mu,\lambda\}$ $\phi \in \{U,V,W\}$, and the notation $\gamma\pm$ and $\phi_\pm$ refers to plus or minus one in the γ component. Additionally,

$$a^{\gamma+} = \frac{1}{h}[\int_{\gamma_n}^{\gamma_{n+1}} \frac{1}{a}d\gamma]^{-1}.$$

We will also use the notation

$$L_{\gamma,\eta}(a,\phi) = \frac{1}{4h^2}[a^{\eta+}(\phi^m_{2+} + \phi^m_{3+} - \phi^m_{2-} - \phi^m_{3-}) - a^{\eta-}(\phi^m_{1+} + \phi^m_{2+} - \phi^m_{1-} - \phi^m_{2-})]$$

for the mixed derivatives with $\pm1, \pm2, \pm3$ standing for

	xz	xy	yz	zx	yx	zy
$\pm$ 1	$i \pm 1$, j, k-1	$i \pm 1$, j-1, k	i, $j \pm 1$, k-1	i-1, j, $k \pm 1$	i-1, $j \pm 1$, k	i, j-1, $k \pm 1$
$\pm$ 2	$i \pm 1$, j, k	$i \pm 1$, j, k	i, $j \pm 1$, k	i, j, $k \pm 1$	i, $j \pm 1$, k	i, j, $k \pm 1$
$\pm$ 3	$i \pm 1$, j, k+1	$i \pm 1$, j+1, k	i, $j \pm 1$, k+1	i+1, j, $k \pm 1$	i+1, $j \pm 1$, k	i, j+1, $k \pm 1$

Using this notation, we may write the method in the concise form:

$$\begin{aligned}
U_{i,j,k}^{m+1} &= 2U_{i,j,k}^{m} - U_{i,j,k}^{m-1} + \frac{\Delta t^2}{\rho_{i,j,k}}[\\
&L_{xx}(\lambda, U) + 2L_{xx}(\mu, U) + L_{yy}(\mu, U) + L_{zz}(\mu, U) \\
&L_{yx}(\lambda, V) + L_{zx}(\lambda, W) + L_{xy}(\mu, V) + L_{xz}(\mu, W)] + F_{i,j,k}^{x,m} \\
V_{i,j,k}^{m+1} &= 2V_{i,j,k}^{m} - V_{i,j,k}^{m-1} + \frac{\Delta t^2}{\rho_{i,j,k}}[\\
&L_{xx}(\mu, V) + L_{yy}(\lambda, V) + 2L_{yy}(\mu, V) + L_{zz}(\mu, V) \\
&L_{yx}(\mu, U) + L_{xy}(\lambda, U) + L_{zy}(\mu, W) + L_{yz}(\mu, W)] + F_{i,j,k}^{y,m} \\
U_{i,j,k}^{m+1} &= 2U_{i,j,k}^{m} - U_{i,j,k}^{m-1} + \frac{\Delta t^2}{\rho_{i,j,k}}[\\
&L_{xx}(\mu, W) + L_{yy}(\mu, W) + L_{zz}(\mu, W) + 2L_{zz}(\mu, W) \\
&L_{zx}(\lambda, U) + L_{zy}(\lambda, V) + L_{xz}(\mu, U) + L_{yz}(\lambda, V)] + F_{i,j,k}^{z,m}
\end{aligned}$$

5 Stability Analysis

We will derive the stability condition for a second order central difference formula on the general equation of motion in an isotropic, homogeneous medium.

$$\rho \ddot{u}_i = \sigma_{ij,j} + f_i$$

Let $\alpha = \sqrt{\frac{\lambda+2\mu}{\rho}}$ and $\beta = \sqrt{\frac{\mu}{\rho}}$. Then the equation of motion becomes the set of equations:

$$\begin{aligned}
u_{tti} &= \alpha^2 u_{xxi} + \beta^2 u_{yyi} + \beta^2 u_{zzi} + (\alpha^2 - \beta^2) u_{xyi} + (\alpha^2 - \beta^2) u_{xzi} + f_i \\
u_{ttj} &= \beta^2 u_{xxj} + \alpha^2 u_{yyj} + \beta^2 u_{zzj} + (\alpha^2 - \beta^2) u_{xyj} + (\alpha^2 - \beta^2) u_{yzj} + f_j \\
u_{ttk} &= \beta^2 u_{xxk} + \beta^2 u_{yyi} + \alpha^2 u_{zzk} + (\alpha^2 - \beta^2) u_{xzk} + (\alpha^2 - \beta^2) u_{yzk} + f_k
\end{aligned}$$

Then we may rewrite the equation of motion in the following form:

$$\mathbf{u}_{tt} = A\mathbf{u}_{xx} + B\mathbf{u}_{yy} + C\mathbf{u}_{zz} + D\mathbf{u}_{xy} + E\mathbf{u}_{xz} + H\mathbf{u}_{yz} + \mathbf{F}$$

where

$$A = \begin{bmatrix} \alpha^2 & 0 & 0 \\ 0 & \beta^2 & 0 \\ 0 & 0 & \beta^2 \end{bmatrix} \quad B = \begin{bmatrix} \beta^2 & 0 & 0 \\ 0 & \alpha^2 & 0 \\ 0 & 0 & \beta^2 \end{bmatrix} \quad C = \begin{bmatrix} \beta^2 & 0 & 0 \\ 0 & \beta^2 & 0 \\ 0 & 0 & \alpha^2 \end{bmatrix}$$

$$D = \begin{bmatrix} 0 & \alpha^2 - \beta^2 & 0 \\ \alpha^2 - \beta^2 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad E = \begin{bmatrix} 0 & 0 & \alpha^2 - \beta^2 \\ 0 & 0 & 0 \\ \alpha^2 - \beta^2 & 0 & 0 \end{bmatrix} \quad H = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & \alpha^2 - \beta^2 \\ 0 & \alpha^2 - \beta^2 & 0 \end{bmatrix}$$

where $\mathbf{u}$ is the displacement vector, α is the compressional wave velocity, and β is the shear wave velocity. Using the central difference formula

$$\begin{aligned}\frac{u^{t+\Delta t}_{x,y,z} - 2u^t_{x,y,z} + u^{t-\Delta t}_{x,y,z}}{\Delta t} &= \frac{A}{\Delta x^2}(u^t_{x+\Delta x,y,z} - 2u^t_{x,y,z} + u^t_{x-\Delta x,y,z}) \\ &+ \frac{B}{\Delta y^2}(u^t_{x,y+\Delta y,z} - 2u^t_{x,y,z} + u^t_{x,y-\Delta y,z}) \\ &+ \frac{C}{\Delta z^2}(u^t_{x,y,z+\Delta z} - 2u^t_{x,y,z} + u^t_{x,y,z-\Delta z}) \\ &+ \frac{D}{4\Delta x\Delta y}[(u^t_{x+\Delta x,y+\Delta y,z} - u^t_{x+\Delta x,y-\Delta y,z} - u^t_{x-\Delta x,y+\Delta y,z} + u^t_{x-\Delta x,y-\Delta y,z}) \\ &+ \frac{E}{4\Delta x\Delta z}(u^t_{i+\Delta x,j,k+\Delta z} - u^t_{x+\Delta x,y,z-\Delta z} - u^t_{x-\Delta x,y,z+\Delta z} + u^t_{x-\Delta x,y,z-\Delta z}) \\ &+ \frac{H}{4\Delta y\Delta z}(u^t_{i,j+\Delta y,k+\Delta z} - u^t_{x,j+\Delta y,z-\Delta z} - u^t_{x,y-\Delta y,z+\Delta z} + u^t_{x,y-\Delta y,z-\Delta z})\end{aligned}$$

Let $\epsilon_1 = \frac{\Delta t}{\Delta x}$, $\epsilon_2 = \frac{\Delta t}{\Delta y}$, $\epsilon_3 = \frac{\Delta t}{\Delta z}$. Then this may be rewritten as

$$\begin{aligned}u^{t+\Delta t}_{x.y.z} &= 2(1 - A\epsilon_2^2 - B\epsilon_2^2 - C\epsilon_3^2)u^t_{x,y,z} - u^{t-\Delta t}_{x,y,z} \\ &+ A\epsilon_1^2(u^t_{x+\Delta x,y,z} + u^t_{x-\Delta x,y,z}) + B\epsilon_2^2(u^t_{x,y+\Delta y,z} + u^t_{x,y-\Delta y,z}) + C\epsilon_3^2(u^t_{x,y,z+\Delta z} + u^t_{x,y,z-\Delta z}) \\ &+ \frac{D}{4}\epsilon_1\epsilon_2(u^t_{x+\Delta x,y+\Delta y,z} - u^t_{x+\Delta x,y-\Delta y,z} - u^t_{x-\Delta x,y+\Delta y,z} + u^t_{x-\Delta x,y-\Delta y,z}) \\ &+ \frac{E}{4}\epsilon_1\epsilon_3(u^t_{x+\Delta x,y,z+\Delta z} - u^t_{x+\Delta x,y,z-\Delta z} - u^t_{x-\Delta x,y,z+\Delta z} + u^t_{x-\Delta x,y,z-\Delta z}) \\ &+ \frac{H}{4}\epsilon_2\epsilon_2(u^t_{x,y+\Delta y,z+\Delta x} - u^t_{x,y-\Delta y,z+\Delta x} - u^t_{x,y+\Delta y,z-\Delta x} + u^t_{x,y-\Delta y,z-\Delta x})\end{aligned}$$

We will use Von Neumann analysis to investigate the stability of this scheme. Let $u^t_{x,y,z} = \alpha e^{i(\gamma j\Delta x + \delta k\Delta y + \epsilon m\Delta z)}\chi^n$, where $j = \frac{x}{\Delta x}, k = \frac{y}{\Delta y}, m = \frac{z}{\Delta z}, n = \frac{t}{\Delta t}$. Then substituting this into the equation we have

$$\begin{aligned}\chi(\alpha e^{i(\gamma j\Delta x+\delta k\Delta y+\epsilon m\Delta z)}\chi^n) &= 2(1 - A\epsilon_1^2 - B\epsilon_2^2 - C\epsilon_3^2)\alpha e^{i(\gamma j\Delta x+\delta k\Delta y+\epsilon m\Delta z)}\chi^n \\ &- \chi^{-1}(\alpha e^{i(\gamma j\Delta x+\delta k\Delta y+\epsilon m\Delta z)}\chi^n) \\ &+ A\epsilon_1^2(e^{i\gamma\Delta x} + e^{-i\gamma\Delta x})\alpha e^{i(\gamma j\Delta x+\delta k\Delta y+\epsilon m\Delta z)}\chi^n \\ &+ B\epsilon_1^2(e^{i\delta\Delta y} + e^{-i\delta\Delta y})\alpha e^{i(\gamma j\Delta x+\delta k\Delta y+\epsilon m\Delta z)}\chi^n \\ &+ C\epsilon_1^2(e^{i\epsilon\Delta z} + e^{-i\epsilon\Delta z})\alpha e^{i(\gamma j\Delta x+\delta k\Delta y+\epsilon m\Delta z)}\chi^n \\ &+ \frac{D}{4}\epsilon_1\epsilon_2(e^{i\gamma\Delta x+i\delta\Delta y} - +e^{-i\gamma\Delta x+i\delta\Delta y} - e^{i\gamma\Delta x-i\delta\Delta y} \\ &+ e^{-i\gamma\Delta x-i\delta\Delta y})\alpha e^{i(\gamma j\Delta x+\delta k\Delta y+\epsilon m\Delta z)}\chi^n \\ &+ \frac{E}{4}\epsilon_1\epsilon_3(e^{i\gamma\Delta x+i\epsilon\Delta z} - e^{-i\gamma\Delta x+i\epsilon\Delta z} - e^{i\gamma\Delta x-i\epsilon\Delta z} \\ &+ e^{-i\gamma\Delta x-i\epsilon\Delta z})\alpha e^{i(\gamma j\Delta x+\delta k\Delta y+\epsilon m\Delta z)}\chi^n \\ &+ \frac{H}{4}\epsilon_2\epsilon_3(e^{i\delta\Delta y+i\epsilon\Delta z} - e^{-i\delta\Delta y+i\epsilon\Delta z} - e^{i\delta\Delta y-i\epsilon\Delta z} \\ &+ e^{-i\delta\Delta y-i\epsilon\Delta z})\alpha e^{i(\gamma j\Delta x+\delta k\Delta y+\epsilon m\Delta z)}\chi^n\end{aligned}$$

Dividing by $e^{i(\gamma j\Delta x+\delta k\Delta y+\epsilon m\Delta z)}\chi^n$ gives

$$\begin{aligned}\alpha\chi = \alpha(2(1-A\epsilon_1^2-B\epsilon_2^2-C\epsilon_3^2)-\chi^{-1}\\
+A\epsilon_1^2(e^{i\gamma\Delta x}+e^{-i\gamma\Delta x})+B\epsilon_1^2(e^{i\delta\Delta y}+e^{-i\delta\Delta y})+C\epsilon_1^2(e^{i\epsilon\Delta z}+e^{-i\epsilon\Delta z})\\
+\frac{D}{4}\epsilon_1\epsilon_2(e^{i\gamma\Delta x+i\delta\Delta y}-e^{-i\gamma\Delta x+i\delta\Delta y}-e^{i\gamma\Delta x-i\delta\Delta y}+e^{-i\gamma\Delta x-i\delta\Delta y})\\
+\frac{E}{4}\epsilon_1\epsilon_3(e^{i\gamma\Delta x+i\epsilon\Delta z}-e^{-i\gamma\Delta x+i\epsilon\Delta z}-e^{i\gamma\Delta x-i\epsilon\Delta z}+e^{-i\gamma\Delta x-i\epsilon\Delta z})\\
+\frac{H}{4}\epsilon_2\epsilon_3(e^{i\delta\Delta y+i\epsilon\Delta z}-e^{-i\delta\Delta y+i\epsilon\Delta z}-e^{i\delta\Delta y-i\epsilon\Delta z}+e^{-i\delta\Delta y-i\epsilon\Delta z}))\end{aligned}$$

Let $\rho=\frac{\gamma\Delta x}{2}, q=\frac{\delta\Delta y}{2}, s=\frac{\epsilon\Delta z}{2}$ Then this equation is simplified to

$$[(-\chi+2-\chi^{-1})I-4(\epsilon_1^2A\sin^2(\rho)+\epsilon_2^2B\sin^2(q)+\epsilon_3^2C\sin^2(s)+$$

$$\epsilon_1\epsilon_2D\sin(\rho)\sin(q)\cos(\rho)\cos(q)+\epsilon_1\epsilon_3E\sin(\rho)\sin(q)\cos(\rho)\cos(q)+\epsilon_2\epsilon_3H\sin(\rho)\sin(q)\cos(\rho)\cos(q))]\alpha=0$$

where I is the identity matrix. We may rewrite the equation in matrix form:

$$\kappa_1=\begin{bmatrix}-\chi+2-\chi^{-1}-4(\alpha^2\epsilon_1^2\sin^2(\rho)+\beta^2\epsilon_2^2\sin^2(q)+\beta^2\epsilon_2^2\sin^2(s))\\ 4(\alpha^2-\beta^2)\epsilon_1\epsilon_2\sin(\rho)\sin(q)\cos(\rho)\cos(q)\\ 4(\alpha^2-\beta^2)\epsilon_1\epsilon_3\sin(\rho)\sin(s)\cos(\rho)\cos(s)\end{bmatrix}$$

$$\kappa_2=\begin{bmatrix}4(\alpha^2-\beta^2)\epsilon_1\epsilon_2\sin(\rho)\sin(q)\cos(\rho)\cos(q)\\ -\chi+2-\chi^{-1}-4(\beta^2\epsilon_1^2\sin^2(\rho)+\alpha^2\epsilon_2^2\sin^2(q)+\beta^2\epsilon_3^2\sin^2(s))\\ 4(\alpha^2-\beta^2)\epsilon_2\epsilon_3\sin(q)\sin(s)\cos(q)\cos(s)\end{bmatrix}$$

$$\kappa_2=\begin{bmatrix}4(\alpha^2-\beta^2)\epsilon_1\epsilon_3\sin(\rho)\sin(s)\cos(\rho)\cos(s)\\ 4(\alpha^2-\beta^2)\epsilon_2\epsilon_3\sin(q)\sin(s)\cos(q)\cos(s)\\ -\chi+2-\chi^{-1}-4(\beta^2\epsilon_1^2\sin^2(\rho)+\beta^2\epsilon_2^2\sin^2(q)+\alpha^2\epsilon_3^2\sin^2(s))\end{bmatrix}$$

where the matrix of coefficients $\kappa=[\kappa_1|\kappa_2|\kappa_3]$. Then we will have that the equation above will be satisfied if the determinant of the matrix of coefficients is zero, more specifically we will look at the magnitude of the coefficients. Let

$$\begin{aligned}C_1&=1-2(\epsilon_1^2\alpha^2\sin^2(\rho)+\epsilon_2^2\beta^2\sin^2(q)+\epsilon_3^2\beta^2\sin^2(s))\\
C_2&=1-2(\epsilon_1^2\beta^2\sin^2(\rho)+\epsilon_2^2\alpha^2\sin^2(q)+\epsilon_3^2\beta^2\sin^2(s))\\
C_3&=1-2(\epsilon_1^2\beta^2\sin^2(\rho)+\epsilon_2^2\beta^2\sin^2(q)+\epsilon_3^2\alpha^2\sin^2(s))\\
C_4&=(4\epsilon_1\epsilon_2(\alpha^2-\beta^2)\sin(\rho)\sin(q)\cos(\rho)\cos(q))^2\\
C_5&=(4\epsilon_1\epsilon_3(\alpha^2-\beta^2)\sin(\rho)\sin(s)\cos(\rho)\cos(s))^2\\
C_6&=(4\epsilon_2\epsilon_3(\alpha^2-\beta^2)\sin(q)\sin(s)\cos(q)\cos(s))^2\end{aligned}$$

Then this matrix may be written as

$$\begin{bmatrix}\chi^2-2C_1\chi+1 & \chi C_4^{1/2} & \chi C_5^{1/2}\\ \chi C_4^{1/2} & \chi^2-2C_2\chi+1 & \chi C_6^{1/2}\\ \chi C_5^{1/2} & \chi C_6^{1/2} & \chi^2-2C_3\chi+1\end{bmatrix}$$

and the determinant is found by

$$(\chi^2-2C_1\chi+1)[(\chi^2-2C_2\chi+1)(\chi^2-2C_3\chi+1)-(\chi C_6^{1/2})(\chi C_6^{1/2})]$$

$$-\chi C_4^{1/2}[(\chi C_4^{1/2})(\chi^2-2C_3\chi+1)-(\chi C_5^{1/2})(\chi C_6^{1/2})]$$

$$+\chi C_5^{1/2}[(\chi C_4^{1/2})(\chi C_6^{1/2})-(\chi^2-2C_2\chi+1)(\chi C_5^{1/2})]$$

Then for this equation to be zero, we may rewrite this equation as

$$(\chi^2 - 2E_1\chi + 1)(\chi^2 - 2E_2\chi + 1)(\chi^2 - 2E_3\chi + 1) = 0$$

where

$$E_1 + E_2 + E_3 = C_1 + C_2 + C_3 = k_1$$
$$E_1E_2 + E_1E_3 + E_2E_3 = C_1C_2 + C_1C_3 + C_2C_3 - (C_4 + C_5 + C_6)/4 = k_2$$
$$E_1E_2E_3 = C_1C_2C_3 - \frac{1}{4}(C_1C_6 + C_2C_5 + C_3C_4 + \sqrt{C_4C_5C_6}) = k_3$$

Then in order for χ to be bounded by unity, $|E_i| \leq 1$ for $i = 1, 2, 3$. Additionally, for E_i, we see that

$$E_i^3 - k_1E_i^2 + k_2E_i - k_3 = E_i^3 - E_i^2E_1 - E_i^2E_2 - E_i^2E_3 + E_iE_1E_2 + E_iE_1E_3 + E_iE_2E_3 - E_1E_2E_3 = 0$$

for $i = 1, 2, 3$. Equivalently, each E_i is a solution to the equation

$$x^2 - k_1x^2 + k_2x - k_3 = 0$$

Using the known solution to the cubic equation we have that $E_i - k_1/3 = y$ where $y = 2(\frac{-(k_2-k_1^3)^3}{27})^{1/6}\cos(\frac{\theta+2n\pi}{3})$ and $\theta = \cos^{-1}((2k_1^3/27 - k_1k_2/3 + k_3)/(2(a_1^2/3 - a2)^{3/2}/\sqrt{27})$. Then $|E_i| \leq 1 \Rightarrow |y + k_1/3| \leq 1$ which is satisfied if $2(\frac{-(k_2-k_1^2/3)^3}{27})^{1/6} + k_1/3 \leq 1$ and $k_1 \geq 0$. Since $k_1 \geq 0$, $2(\frac{-(k_2-k_1^2/3)^3}{27})^{1/6} \leq -k_1/3$ and since $-k_1/3 \geq 0$, the stability condition becomes:

$$2(\frac{-(k_2 - k_1^2/3)^3}{27})^{1/6} \leq 0 \Leftrightarrow -(k_2 - k_1^2/3)^3 \leq 0 \Leftrightarrow (k_2 - k_1^2/3)^3 \geq 0 \Leftrightarrow k_2 - k_1^2/3 \geq 0$$

This condition may be rewritten as $C_1C_2 + C_2C_3 + C_1C_3 - (C_4 + C_5 + C_6)/4 \geq (1/3) * (C_1 + C_2 + C_3)^2$. Solving this yields

$$\alpha^2(\epsilon_1^2 + \epsilon_2^2 + \epsilon_3^2) \leq \frac{3}{2(1 + 2\beta^2/\alpha^2)}$$

where $\beta/\alpha \geq 1/2$. Under the assumption that $\Delta x = \Delta y = \Delta z = h$, this condition is equivalent to

$$\Delta t^2 \leq \frac{h^2}{2\alpha^2(1 + 2\beta^2/\alpha^2)}$$

where $\alpha = \sqrt{\frac{\lambda+2\mu}{\rho}}$ is the P-wave velocity and $\beta = \sqrt{\frac{\mu}{\rho}}$ is the S-wave velocity

6 Problem Specifics

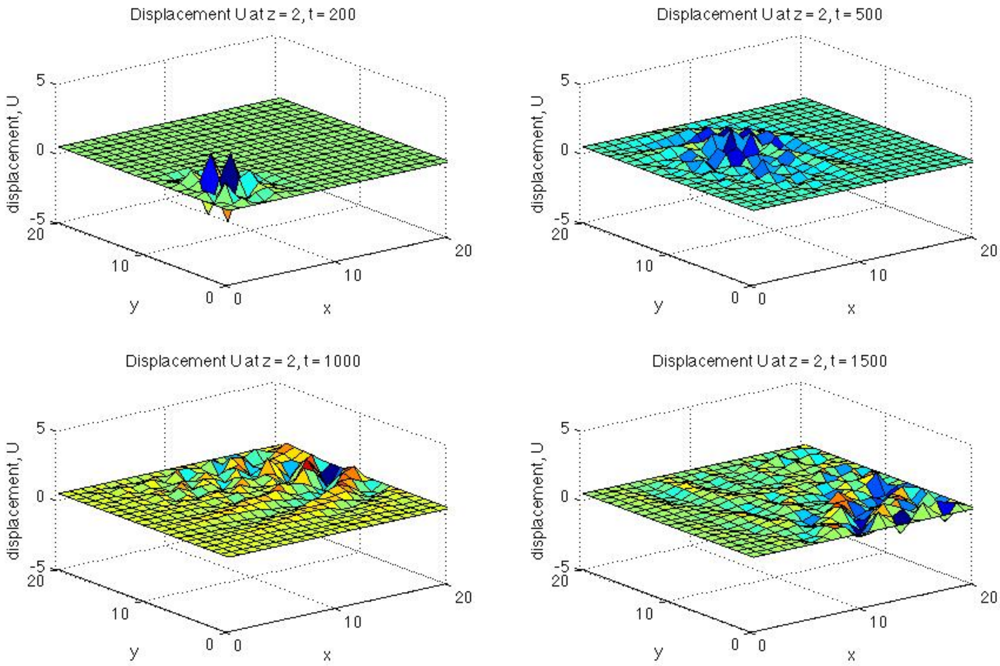
The method will be coded using the FDM outlined in section 4. The parameter values chosen for the simulations are given below.

parameter	value	interpretation
h	0.5 m	spatial discretization parameter
Δt	0.05 sec	temporal discretization parameter
α_x, β_x	0,10 m	min and max x values
α_y, β_y	0,10 m	min and max y values
α_z, β_z	0,1 m	min and max z values
T	50 sec	max time value
ρ	2500 kg/m^3	material density
λ	12.66 10^9 Pa	elastic modulus
μ	17 10^9 Pa	shear modulus
F^x	$\begin{cases} 0 & x_n, y_m, z_k \neq 2h \\ \sin(\pi t_m), m = 1, \cdots, 4 & \text{otherwise} \end{cases}$	force in the x component
F^y	0	force in the y component
F^z	0	force in the z component

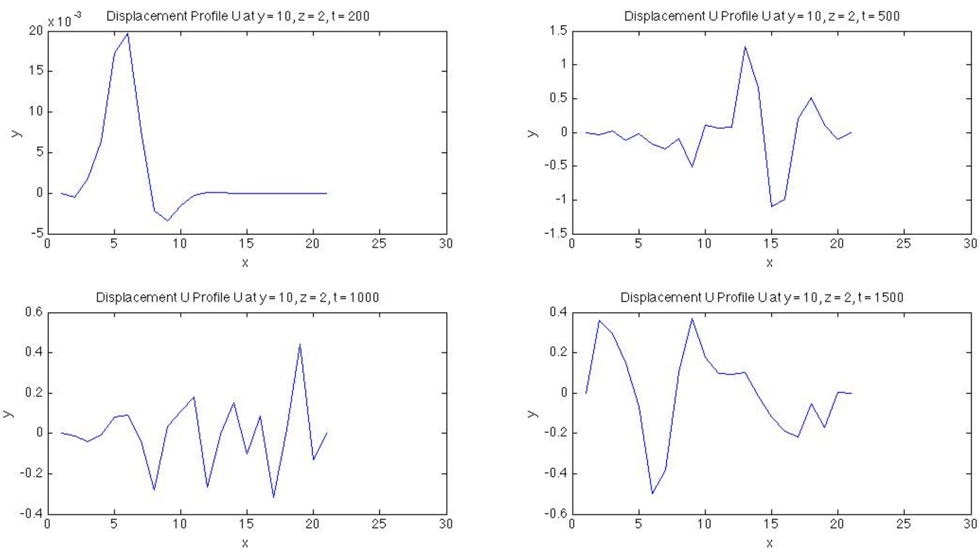
The parameter values given are consistent with the material sandstone. In order to investigate how the motion of the waves is dependent on the material, the code will also be run using the material properties of peridotite, a much harder rock. The parameters for peridotite are: $\rho = 3300$ kg/$m^3, \lambda = 86 \times 10^9Pa, \mu = 63 \times 10^9$ Pa

7 Computational Results

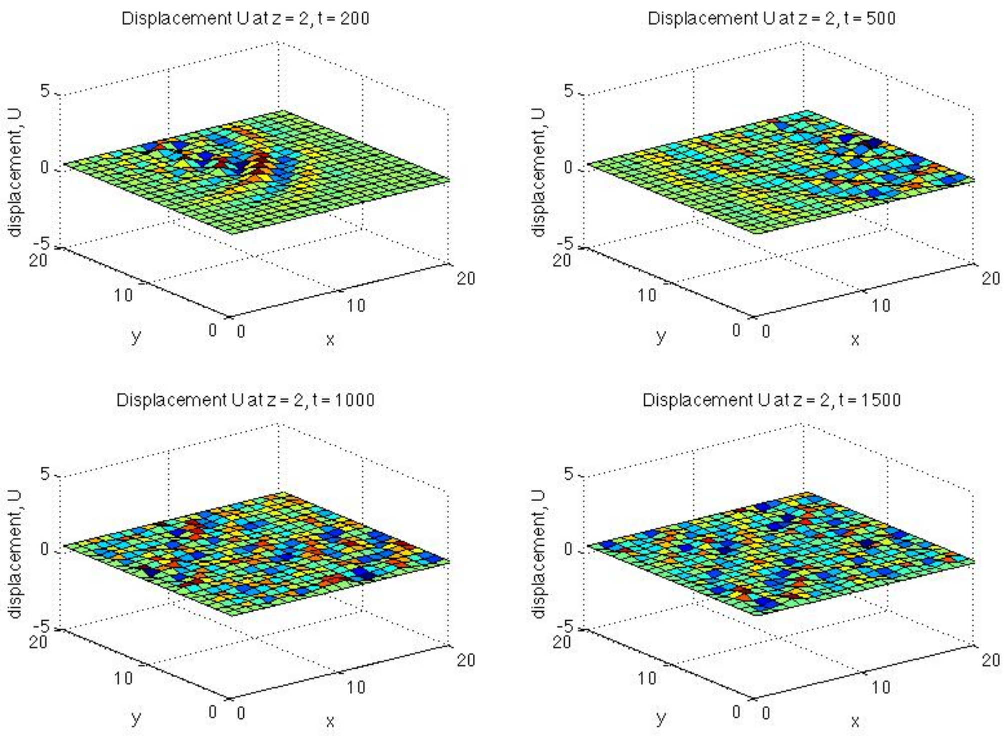
In order to see how the waves propagate in the sandstone material, the following graphs depict the displacement in the x direction for $z_n = 2h$ at different times. We can see the that force in the x direction creates an initial displacement which then propagates through the domain. Since the boundaries are non-absorbent, the waves reflect off the boundaries and interact moving throughout the domain. The magnitude of the wave heights for this material is approximately 2.5 m.



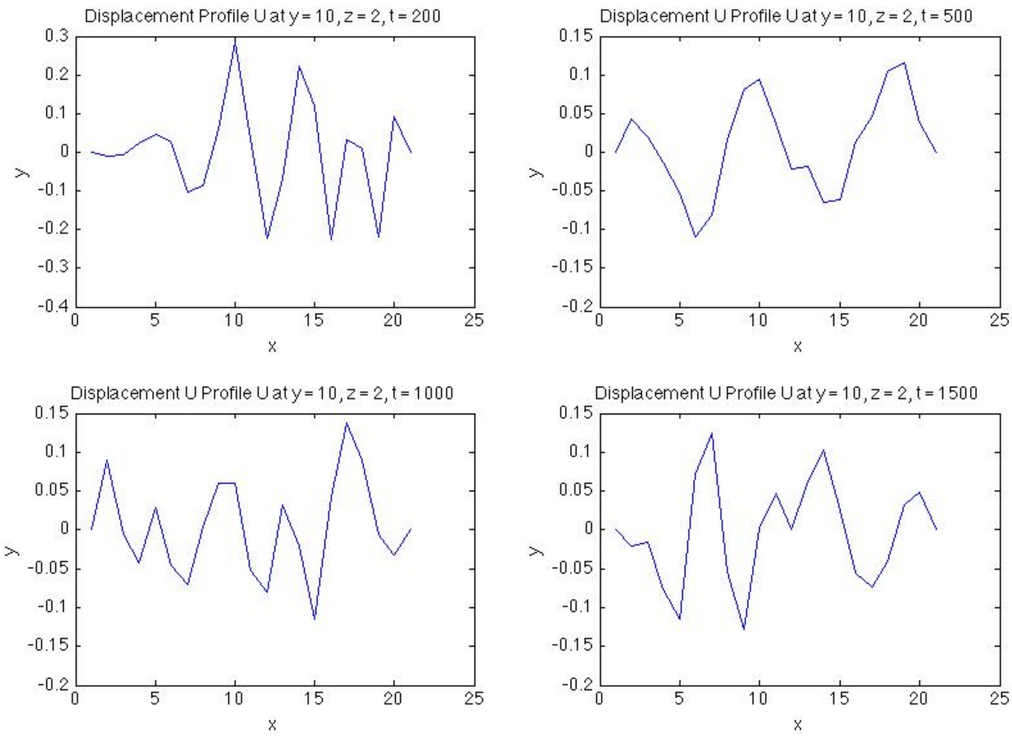
The next graph depicts the profile of the displacement at various times along the line $y = 10h, z = 2h$. The edges in this graph are sharp due to the large step size. We can see the initial wave propagating through the material at time $t = 200\Delta t$, and the interaction of the waves thereafter. On these graphs we can see the the magnitude begins at approximately 10^{-2} and grows to about 0.4.



The following graphs were generated using the material properties of peridotite. The first depicts the displacement in the x component for $z = 2h$ for various times. These graphs, displaying the same spatial and temporal data as the sandstone graphs, show that the waves do not propagate well in the material. In fact, comparing the two sets of graphs it appears that the waves propagate faster in the peridotite but the amplitude of the waves decreases over time.



The second figure which displays a profile of the displacement at various times along the line $y = 10h, z = 2h$ confirms this. We can see that the initial displacement has a magnitude of about 0.3 m; however, unlike in the sandstone model, the magnitude decreases at further time steps.



These data confirm what has been suggested about the manner with which waves propagate through softer and hard rock: waves propagate and are amplified in soft rock (i.e. sandstone) while they are dampened by hard rock (i.e. peridotite). Even though the waves are dampened, we can still see movement in the graph. This is because the PDE is hyperbolic and the boundary conditions are non-absorbing, and so the waves continue to propagate for a long period of time.

8 Conclusion

Numerical methods provide a valuable tool to approximate the motion of the earth and may be useful in studying various seismic phenomena. The focus of this project was to approximate the solution to the equation of motion along with a constitutive relation in order to model the displacement of the earth. Using finite difference methods to approximate the motion of seismic waves can be a valuable resource for many reasons. While much is known about the behavior of seismic waves, there is still much to be gained from using numerical methods to approximate the motion of waves. The work done here served to simply confirm what has been previously proven, that waves are amplified in softer rock and dampened in harder rock; however this method could be extended to model various types of scenarios, source and object elements. For example, adding in components of fault structure, dispersion, or inhomogeneous material properties would allow this work to apply to other problems.

References

[1] Engdahl B. Kind R. Bormann, P. *Seismic Wave Propagation and Earth Models. In: Bormann, P. (Ed.) New Manual of Seismological Observatory Practice (NMSOP).* Postdam: Deutsches Geo-ForschungsZentrum GFZ, 2009.

[2] Kristek J. Galis M. Pazak P. Balazovjech M. Moczo, P. The finite difference and finite element modeling of seismic wave propagation and earthquake motion. *Acta Physica Slovaca*, 57(2):177–406, April 2007.

Code

```
clear; clc;

h = 0.5;

stability = h/(sqrt(2*(  ((0.21 + 2*17)/2500) + (17/2500) )));

dt = 0.05;
x = 0:h:10;
y = x;
z = 0:h:1;
t = 0:dt:75;

M = length(x);
N = length(y);
R = length(z);
T = length(t);

U = zeros(M,N,R,T);
V = zeros(M,N,R,T);
W = zeros(M,N,R,T);

rho = 2500*ones(M,N,R);
lambda = 12.66*ones(M,N,R);
mu = 17*ones(M,N,R);

% rho = 3300*ones(M,N,R);
% lambda = 86*ones(M,N,R);
% mu = 63*ones(M,N,R);
Fx = zeros(M,N,R,T);
Fy = Fx;
Fz = Fx;

for m = 1:4
    Fx(2,2,2,m) = sin(pi*t(m));
end



for m = 2:T
    for i = 2:M-1
        for j = 2:N-1
            for k = 2:R-1
                U(i,j,k,m+1) = 2*U(i,j,k,m) - U(i,j,k,m-1)+...
                    (dt^2/rho(i,j,k))*...
                    (L_op1(lambda, U(:,:,:,m), 1, h, i,j,k)+...
                    L_op1(mu, U(:,:,:,m), 1, h, i,j,k)+...
                    L_op1(mu, U(:,:,:,m), 2, h, i,j,k)+...
                    L_op1(mu, U(:,:,:,m), 3, h, i,j,k)+...
                    L_op2(mu, V(:,:,:,m), 2, 1, h, i,j,k)+...
                    L_op2(mu, W(:,:,:,m), 3, 1, h, i,j,k)+...
                    L_op2(lambda, V(:,:,:,m), 1, 2, h, i,j,k)+...
                    L_op2(lambda, W(:,:,:,m), 1, 3, h, i,j,k))+...
                    Fx(i,j,k,m);
                V(i,j,k,m+1) = 2*V(i,j,k,m) - V(i,j,k,m-1)+...
                    (dt^2/rho(i,j,k))*...
                    (L_op1(lambda, V(:,:,:,m), 1, h, i,j,k)+...
                    L_op1(mu, V(:,:,:,m), 1, h, i,j,k)+...
                    L_op1(mu, V(:,:,:,m), 2, h, i,j,k)+...
                    L_op1(mu, V(:,:,:,m), 3, h, i,j,k)+...
                    L_op2(mu, U(:,:,:,m), 2, 1, h, i,j,k)+...
                    L_op2(mu, U(:,:,:,m), 3, 1, h, i,j,k)+...
                    L_op2(lambda, W(:,:,:,m), 3, 2, h, i,j,k)+...
                    L_op2(lambda, W(:,:,:,m), 2, 3, h, i,j,k))+...
                    Fy(i,j,k,m);
                W(i,j,k,m+1) = 2*W(i,j,k,m) - W(i,j,k,m-1)+...
                    (dt^2/rho(i,j,k))*...
                    (L_op1(lambda, W(:,:,:,m), 1, h, i,j,k)+...
                    L_op1(mu, W(:,:,:,m), 1, h, i,j,k)+...
                    L_op1(mu, W(:,:,:,m), 2, h, i,j,k)+...
                    L_op1(mu, W(:,:,:,m), 3, h, i,j,k)+...
                    L_op2(mu, U(:,:,:,m), 3, 1, h, i,j,k)+...
                    L_op2(mu, V(:,:,:,m), 3, 2, h, i,j,k)+...
                    L_op2(lambda, U(:,:,:,m), 1, 3, h, i,j,k)+...
                    L_op2(lambda, V(:,:,:,m), 2, 3, h, i,j,k))+...
                    Fz(i,j,k,m);

            end
        end
    end
end
%%
loops = 4001;
F(loops) = struct('cdata',[],'colormap',[]);

for j = 1:loops
    surf(U(:,:,2,j))
    zlim([-10 10])
    drawnow
    F(j) = getframe;

end

movie(F,2)
%%
% subplot(2,2,1), surf(U(:,:,2,200))
% title('Displacement U at z = 2, t = 200'), xlabel('x'), ylabel('y'), zlabel('displacement, U')
% xlim([0 20])
% ylim([0 20])
% zlim([-5 5])
% subplot(2,2,2), surf(U(:,:,2,500))
% title('Displacement U at z = 2, t = 500'), xlabel('x'), ylabel('y'), zlabel('displacement, U')
% xlim([0 20])
% ylim([0 20])
% zlim([-5 5])
% subplot(2,2,3), surf(U(:,:,2,1000))
% title('Displacement U at z = 2, t = 1000'), xlabel('x'), ylabel('y'), zlabel('displacement, U')
```

```
% xlim([0 20])
% ylim([0 20])
% zlim([-5 5])
% subplot(2,2,4), surf(U(:,:,2,1500))
% title('Displacement U at z = 2, t = 1500'), xlabel('x'), ylabel('y'), zlabel('displacement, U')
% xlim([0 20])
% ylim([0 20])
% zlim([-5 5])

ind = 1:21;

subplot(2,2,1), plot(ind, U(:,10,2,200))
title('Displacement Profile U at y = 10, z = 2, t = 200'), xlabel('x'), ylabel('y'), zlabel('displacement, U')
subplot(2,2,2), plot(ind, U(:,10,2,500))
title('Displacement U Profile U at y = 10, z = 2, t = 500'), xlabel('x'), ylabel('y'), zlabel('displacement, U')
subplot(2,2,3), plot(ind, U(:,10,2,1000))
title('Displacement U Profile U at y = 10, z = 2, t = 1000'), xlabel('x'), ylabel('y'), zlabel('displacement, U')
subplot(2,2,4), plot(ind, U(:,10,2,1500))
title('Displacement U Profile U at y = 10, z = 2, t = 1500'), xlabel('x'), ylabel('y'), zlabel('displacement, U')


function out = a_gamma(a, h, sign, n, i, j, k)

if n == 1
    int = (1/h)*(1/a(i,j,k))*(1/a(i+sign, j, k))*h/2;
elseif n == 2
    int = (1/h)*(1/a(i,j,k))*(1/a(i, j+sign, k))*h/2;
elseif n == 3
    int = (1/h)*(1/a(i,j,k))*(1/a(i, j, k+sign))*h/2;
end
 out = 1/int;


function out = L_op1(arg1, arg2, gamma, h, i,j,k)

a1 = a_gamma(arg1, h, 1, gamma, i, j, k);
a2 = a_gamma(arg1, h, -1, gamma, i, j, k);

if gamma == 1
    phi1 = arg2(i+1, j, k);
    phi3 = arg2(i-1, j, k);
elseif gamma == 2
    phi1 = arg2(i, j+1, k);
    phi3 = arg2(i, j-1, k);
elseif gamma == 3
    phi1 = arg2(i, j, k+1);
    phi3 = arg2(i, j, k-1);
end

phi2 = arg2(i,j,k);

out = (1/h^2)*(a1*(phi1-phi2)-a2*(phi2-phi3));


function out = L_op2(arg1, arg2, gamma, ada, h, i,j,k)

a1 = a_gamma(arg1, h, 1, ada, i, j, k);
a2 = a_gamma(arg1, h, -1, ada, i, j, k);

if gamma == 1
    if ada == 2
        phi1 = arg2(i+1, j, k);
        phi2 = arg2(i+1, j+1, k);
        phi3 = arg2(i-1, j, k);
        phi4 = arg2(i-1, j+1, k);
        phi5 = arg2(i+1, j-1, k);
        phi7 = arg2(i-1, j-1, k);
    elseif ada == 3
        phi1 = arg2(i+1, j, k);
        phi2 = arg2(i+1, j, k+1);
        phi3 = arg2(i-1, j, k);
        phi4 = arg2(i-1, j, k+1);
        phi5 = arg2(i+1, j, k-1);
        phi7 = arg2(i-1, j, k-1);
    end
elseif gamma == 2
    if ada == 1
        phi1 = arg2(i, j+1, k);
        phi2 = arg2(i+1, j+1, k);
        phi3 = arg2(i, j-1, k);
        phi4 = arg2(i+1, j-1, k);
        phi5 = arg2(i-1, j+1, k);
        phi7 = arg2(i-1, j-1, k);
    elseif ada == 3
        phi1 = arg2(i, j+1, k);
        phi2 = arg2(i, j+1, k+1);
        phi3 = arg2(i, j-1, k);
        phi4 = arg2(i, j-1, k+1);
        phi5 = arg2(i, j+1, k-1);
        phi7 = arg2(i, j-1, k-1);
    end
elseif gamma == 3
    if ada == 1
        phi1 = arg2(i, j, k+1);
        phi2 = arg2(i+1, j, k+1);
        phi3 = arg2(i, j, k-1);
        phi4 = arg2(i+1, j, k-1);
        phi5 = arg2(i-1, j, k+1);
        phi7 = arg2(i-1, j, k-1);
    elseif ada == 2
        phi1 = arg2(i, j, k+1);
        phi2 = arg2(i, j+1, k+1);
        phi3 = arg2(i, j, k-1);
        phi4 = arg2(i, j+1, k-1);
        phi5 = arg2(i, j-1, k+1);
        phi7 = arg2(i, j-1, k-1);
    end
end


out = (1/4*h^2)*(a1*(phi1+phi2-phi3-phi4)-a2*(phi5+phi1-phi7-phi3));
```

```
clear; clc;


for k = 0:2
h = 0.5/2^k;

stability = h/(sqrt(2*(  ((0.21 + 2*17)/2500) + (17/2500) )));

dt = h/10;%0.05;
x = 0:h:10;
y = x;
z = 0:h:1;
t = 0:dt:10;

M = length(x);
N = length(y);
R = length(z);
T = length(t);

U = zeros(M,N,R,T);
V = zeros(M,N,R,T);
W = zeros(M,N,R,T);

rho = 3300*ones(M,N,R);
lambda = 86*ones(M,N,R);
mu = 63*ones(M,N,R);
Fx = zeros(M,N,R,T);
Fy = Fx;
Fz = Fx;

for m = 1:4
    Fx(2,2,2,m) = sin(pi*t(m));
end



for m = 2:T
    for i = 2:M-1
        for j = 2:N-1
            for k = 2:R-1
                U(i,j,k,m+1) = 2*U(i,j,k,m) - U(i,j,k,m-1)+...
                    (dt^2/rho(i,j,k))*...
                    (L_op1(lambda, U(:,:,:,m), 1, h, i,j,k)+...
                    L_op1(mu, U(:,:,:,m), 1, h, i,j,k)+...
                    L_op1(mu, U(:,:,:,m), 2, h, i,j,k)+...
                    L_op1(mu, U(:,:,:,m), 3, h, i,j,k)+...
                    L_op2(mu, V(:,:,:,m), 2, 1, h, i,j,k)+...
                    L_op2(mu, W(:,:,:,m), 3, 1, h, i,j,k)+...
                    L_op2(lambda, V(:,:,:,m), 1, 2, h, i,j,k)+...
                    L_op2(lambda, W(:,:,:,m), 1, 3, h, i,j,k))+...
                    Fx(i,j,k,m);
                V(i,j,k,m+1) = 2*V(i,j,k,m) - V(i,j,k,m-1)+...
                    (dt^2/rho(i,j,k))*...
                    (L_op1(lambda, V(:,:,:,m), 1, h, i,j,k)+...
                    L_op1(mu, V(:,:,:,m), 1, h, i,j,k)+...
                    L_op1(mu, V(:,:,:,m), 2, h, i,j,k)+...
                    L_op1(mu, V(:,:,:,m), 3, h, i,j,k)+...
                    L_op2(mu, U(:,:,:,m), 2, 1, h, i,j,k)+...
                    L_op2(mu, U(:,:,:,m), 3, 1, h, i,j,k)+...
                    L_op2(lambda, W(:,:,:,m), 3, 2, h, i,j,k)+...
                    L_op2(lambda, W(:,:,:,m), 2, 3, h, i,j,k))+...
                    Fy(i,j,k,m);
                W(i,j,k,m+1) = 2*W(i,j,k,m) - W(i,j,k,m-1)+...
                    (dt^2/rho(i,j,k))*...
                    (L_op1(lambda, W(:,:,:,m), 1, h, i,j,k)+...
                    L_op1(mu, W(:,:,:,m), 1, h, i,j,k)+...
                    L_op1(mu, W(:,:,:,m), 2, h, i,j,k)+...
                    L_op1(mu, W(:,:,:,m), 3, h, i,j,k)+...
                    L_op2(mu, U(:,:,:,m), 3, 1, h, i,j,k)+...
                    L_op2(mu, V(:,:,:,m), 3, 2, h, i,j,k)+...
                    L_op2(lambda, U(:,:,:,m), 1, 3, h, i,j,k)+...
                    L_op2(lambda, V(:,:,:,m), 2, 3, h, i,j,k))+...
                    Fz(i,j,k,m);

            end
        end
    end
end

values(k) = U(2,2,2,end);
fprintf('Finishing %d', k)
end
%%
```