simpegseis Documentation

Release 0.0.1

Seogi Kang

May 04, 2017

Contents

SimPEG (Simulation and Parameter Estimation in Geophysics) is a python package for simulation and gradient based parameter estimation in the context of geoscience applications. simpegSeis uses SimPEG as the framework for the forward modeling and inversion of seismic geophysical problem.

We consider two fundamental governing equations for seismic wave propagation: acoustic and elastic wave equations in both time and frequency domains. We initially discretize those problems to compute forward responses. Obvious next step is to solve inverse problems

We welcome any people who want to contribute this project developing forward modeling and inversion package of seimic data, simpegSeis.

CHAPTER 1

Acoustic wave equation

Backgrounds

Time domain acoustic wave equation in 1st order form:

$$\mu^{-1}\frac{\partial \vec{u}}{\partial t} = \nabla\phi$$

$$\rho\frac{\partial \phi}{\partial t} = \nabla\cdot\vec{u} + \frac{\partial s}{\partial t}\delta(\vec{r}-\vec{r}_s)$$

where \(\rho\) is the density, \(\mu\) is the adiabatic compression modulus, \(\delta(\vec{r}-\vec{r}_s)\) represents point source, and \(\phi\) is the pressure.

By combining two equations we have 2nd order form:

$$\rho\frac{\partial^2 \phi}{\partial t^2} - \div\mu\phi = \frac{\partial^2 s}{\partial t^2}\delta(\vec{r}-\vec{r}_s)$$

By applying Fourier transform to frequency domain we have:

$$-\rho\omega^2\phi - \div\mu\phi = -\omega^2 s\delta(\vec{r}-\vec{r}_s)$$

where \(\omega = 2\pi f\) is angular frequency. Assuming contant \(\mu\) and \(\rho\) we have Helmholtz equation:

$$({}^2 + k^2)\phi = \mu^{-1}\omega^2 s\delta(\vec{r}-\vec{r}_s)$$

where \(k = \omega\sqrt{\frac{\rho}{\mu}}=\frac{\omega}{v} \) is the wave propagation constant.

Discretization of problem

To compute the solution of time or frequency dependent partial differential equation (PDE) as discussed above, we need to discretize those equations in both time and space. Our domain in real situation (i.e., earth) can be considered as infinite in space. However, in discrete space where we compute the solution of those PDE should be finite. Therefore, we need to implement artificial boundary conditions to get rid of out going wave, which propagates beyond our domain of interest. We first implement sponge boundary condition then perfectly matched layer (PML) boundary condition.

Sponge boundary

A fundamental idea of sponge boundary condition is adding damping term to acoustic wave equation:

$$\rho\frac{\partial^2\phi}{\partial t^2} + c\frac{\partial\phi}{\partial t} - \div\mu\phi = \frac{\partial^2 s}{\partial t^2}\delta(\vec{r}-\vec{r}_s)$$

Similarly in frequency domain we have:

$$-\rho\omega^2\phi + \imath\omega c\phi - \div\mu\phi = -\omega^2 s\delta(\vec{r}-\vec{r}_s)$$

The 1st order form of time domain equation can be written as:

$$\begin{aligned}\mu^{-1}\frac{\partial\vec{u}}{\partial t} &= \nabla\phi\\ \rho\frac{\partial\phi}{\partial t} + c\phi - \nabla\cdot\vec{u} &= +\frac{\partial s}{\partial t}\delta(\vec{r}-\vec{r}_s)\end{aligned}$$

We discretize 2nd order acoustic wave equation in time (CITE) with central difference:

$$\rho\triangle t^{-2}(\phi^{n+1} - 2\phi^n - \phi^{n-1}) + \frac{c}{2}(\phi^{n+1} - \phi^{n-1}) - \nabla\cdot\phi^n = q$$

where

$$q = \triangle t^{-2}(s^{n+1} - 2s^n - s^{n-1})\delta(\vec{r}-\vec{r}_s)$$

By rearranging them, we have:

$$\phi^{n+1} = (1+\frac{c\rho}{2}\triangle t)^{-1}[2\phi^n - \phi^{n-1} + \frac{c\rho}{2}\triangle t\phi^{n-1} + \triangle t^2\rho\div\mu\phi^n + \triangle t^2\rho q]$$

By letting \(f = (1+\frac{c \rho}{2}\triangle t)^{-1}\), we can substitute \(c = \frac{1-f}{f}\triangle t^{-1}2 \rho\). Range of \(f \) for time and frequency domain are 0.98-1 and 0.9-1, respectively. And 35 cells are used for the sponge boundary.

Using staggered grid, we discretize acoustic wave equation in time (CITE):

$$\mu^{-1}\triangle t^{-1}(\rho\triangle t^{-1}(\phi^{n+1} - \phi^n) + \rho\sigma\phi^n - \nabla\cdot\vec{u}^{n+1/2} = +\triangle t^{-1}(s^{n+} - s^n)$$

where \(\sigma = 2 \frac{1-f}{f} \triangle t^{-1}\).

Using mimetic finite volume approach, we spatially discretize above equations:

$$\begin{aligned}\mathbf{diag}(^T\mu^{-1})\frac{\mathbf{u}^{n+1/2} - \mathbf{u}^{n-1/2}}{\triangle t} + \mathbf{diag}(^T\mu^{-1}\sigma)\mathbf{u}^{n+1/2} &= \mathbf{Grad}\phi^n\\ \mathbf{diag}(\rho)\frac{\phi^{n+1} - \phi^n}{\triangle t} + \mathbf{diag}(\rho\sigma)\phi^n - \mathbf{Div}\mathbf{u}^{n+1/2} &= +\mathbf{diag}(\mathbf{vol})\frac{\mathbf{s}^{n+1} - \mathbf{s}^n}{\triangle t}\end{aligned}$$

where \(\mathbf{Div}\) and \(\mathbf{Grad}\) and discrete differential operators and \(\Acf\) is averaging operator from cell face to center. \(\mu\), \(\rho\) and \(\sigma\) are defined on the cell center, and \(\mathbf{vol} \) is volume of each cell. When we compute these, we first compute:

$$\mathbf{u}^{n+1/2} = \mathbf{u}^{n-1/2} - \triangle t\mathbf{diag}(\sigma)\mathbf{u}^{n-1/2} + \triangle t\mathbf{diag}(^T\mu^{-1})^{-1}\mathbf{Grad}\phi^n$$

Then we compute:

$$\phi^{n+1} = \phi^n - \triangle t\mathbf{diag}(\sigma)\phi^n + \mathbf{diag}(\rho)^{-1}\triangle t[\mathbf{Div}\mathbf{u}^{n+1/2} + \mathbf{diag}(\mathbf{vol})^{-1}\frac{\mathbf{s}^{n+1} - \mathbf{s}^n}{\triangle t}]$$

Note: Choice of \(\sigma \) in sponge boundary condition case can be expanded to PML case.

$$\sigma = 2\frac{1-f}{f}\triangle t^{-1}$$

And range of \(f \) is 0.98-1, which can be useful reference property.

PML boundary

PML has two fundamental factors: a. matching the impedance and b. damping. These purposes can be realized by considering solution of Helmholtz equation on complex plane:

$$\tilde{x} = x + \frac{1}{\imath\omega}\int_0^x \sigma^x(\xi)d\xi$$
$$\partial x = \frac{\imath\omega}{\sigma^x + \imath\omega}\partial\tilde{x}$$

By parameterizing the physical coordinate as

$$\tilde{x} = \tilde{x}(x)$$
$$x < 0 \ : \ \text{real axis}$$
$$x > 0 \ : \ \Im[\tilde{x}] < 0 \ \text{(decaying term)}$$

Another core treatment of PML is decomposing \(\phi \) as:

$$\phi_d = \begin{bmatrix}\phi^x \\ \phi^y \\ \phi^z\end{bmatrix}$$
$$\phi = [1,1,1]\phi_d = \phi^x + \phi^y + \phi^z$$

Substituting those yields:

$$\imath\omega\mu^{-1}\Phi \begin{bmatrix}\frac{1}{\sigma^x+\imath\omega}\frac{\partial\phi}{\partial x} \\ \frac{1}{\sigma^y+\imath\omega}\frac{\partial\phi}{\partial y} \\ \frac{1}{\sigma^z+\imath\omega}\frac{\partial\phi}{\partial z}\end{bmatrix}$$
$$\imath\omega\rho\begin{bmatrix}\phi^x \\ \phi^y \\ \phi^z\end{bmatrix} - \rho\imath\omega\begin{bmatrix}\frac{1}{\sigma^x+\imath\omega}\frac{\partial u^x}{\partial x} \\ \frac{1}{\sigma^y+\imath\omega}\frac{\partial u^y}{\partial y} \\ \frac{1}{\sigma^z+\imath\omega}\frac{\partial u^z}{\partial z}\end{bmatrix} = \frac{1}{3}\imath\omega\delta(\vec{r} - \vec{r}_s)\begin{bmatrix}s \\ s \\ s\end{bmatrix}$$

With some linear algebra:

$$\mu^{-1}(\imath\omega\Phi\Sigma\Phi) = \phi$$
$$\imath\omega\rho\begin{bmatrix}\phi^x \\ \phi^y \\ \phi^z\end{bmatrix} - \rho\begin{bmatrix}\sigma^x\phi^x \\ \sigma^y\phi^y \\ \sigma^z\phi^z\end{bmatrix} + \begin{bmatrix}u^x \\ u^y \\ u^z\end{bmatrix} = \frac{1}{3}\imath\omega\delta(\vec{r} - \vec{r}_s)\begin{bmatrix}s \\ s \\ s\end{bmatrix}$$

where

$$\Sigma = \begin{bmatrix} \sigma^x & 0 & 0 \\ 0 & \sigma^y & 0 \\ 0 & 0 & \sigma^z \end{bmatrix}$$

In time domain we have:

$$\mu^{-1}(\frac{\partial \vec{u}}{\partial t} + \Sigma \Phi) = \nabla \phi$$

$$\rho \frac{\partial \phi_d}{\partial t} + \rho \Sigma \phi_d - \begin{bmatrix} \frac{\partial u^x}{\partial x} \\ \frac{\partial u^y}{\partial y} \\ \frac{\partial u^z}{\partial z} \end{bmatrix} = \frac{1}{3} \imath \omega \delta(\vec{r} - \vec{r}_s) \begin{bmatrix} s \\ s \\ s \end{bmatrix}$$

We discretize above equations in both space and time:

$$\triangle t^{-1}(\mathbf{u}^{n+1/2} - \mathbf{u}^{n-1/2}) + \mathbf{\Sigma}^f \mathbf{u}^{n-1/2} - \mathbf{Grad} \mathbf{I}_d \phi_d^n = 0$$

$$\mathbf{\Omega}^{cc} \triangle t^{-1}(\phi_d^{n+1} - \phi_d^n) + \mathbf{\Omega}^{cc} \mathbf{\Sigma}^{cc} \phi_d - \mathbf{Div}_{vec} \mathbf{u} = \triangle t^{-1} \mathbf{diag}(\mathbf{vol})^{-1} (\mathbf{s}_d^{n+1} - \mathbf{s}_d^n)$$

where

$$= \mathbf{diag}(^T \mu^{-1}), \ \mathbf{\Sigma}^f = \mathbf{diag}(^T_{vec} \begin{bmatrix} \sigma^x \\ \sigma^y \\ \sigma^z \end{bmatrix})$$

$$\mathbf{I}_d = [\mathbf{I}^{cc}, \mathbf{I}^{cc}, \mathbf{I}^{cc}], \ \mathbf{\Omega}^{cc} = \begin{bmatrix} \rho & 0 & 0 \\ 0 & \rho & 0 \\ 0 & 0 & \rho \end{bmatrix},$$

$$\mathbf{\Sigma}^{cc} = \begin{bmatrix} \sigma^x & 0 & 0 \\ 0 & \sigma^y & 0 \\ 0 & 0 & \sigma^z \end{bmatrix}$$

$$\mathbf{s}_d = \frac{1}{3} \begin{bmatrix} \mathbf{s} \\ \mathbf{s} \\ \mathbf{s} \end{bmatrix}$$

Similarly we first compute:

$$\mathbf{u}^{n+1/2} = \mathbf{u}^{n-1/2} - \triangle t \mathbf{\Sigma}^f \mathbf{u}^{n-1/2} + \triangle t^{-1} \mathbf{Grad} \mathbf{I}_d \phi_d^n$$

Then we compute:

$$\phi_d^{n+1} = \phi_d^n - \triangle t \mathbf{\Sigma}^{cc} \phi_d + \mathbf{\Omega}^{cc\ -1} \triangle t \mathbf{Div}_{vec} \mathbf{u} + \mathbf{\Omega}^{cc\ -1} \mathbf{diag}(\mathbf{vol})^{-1} (\mathbf{s}_d^{n+1} - \mathbf{s}_d^n)$$

Stability conditions

Stability of forward modeling have two fundamental factors: cell size and time step size (\(\triangle t\)). First, we determine cell size based on the number of cell per wavelength (\(G\)):

$$G = \frac{\lambda}{\triangle x} \approx 16$$

where

$$\lambda = \frac{v_{min}}{f_{main}}$$

Second, we determine \(\triangle t\)

$$\triangle t = \frac{\triangle x}{v_{max}} c$$

where \(c \) is a proper constant.

Notebooks

1. **How to run acoustic wave modeling in simpegSeis** Examples shown are generated through this ipython notebook.

CHAPTER 2

Elastic wave

CHAPTER 3

License

License

CHAPTER 4

Project Index & Search

- genindex
- modindex
- search