Synthetic Seismograms

A large part of this course will describe techniques for computing *synthetic seismograms* for realistic Earth models. In general, we wish to know how to calculate what would be recorded by a seismograph at a specified receiver location, given an exact specification of the seismic source and the Earth model through which the seismic waves propagate. This is a well-defined forward problem which, in principle, can be solved exactly. However, in practice it is difficult to achieve a perfect fit to real data. Inaccuracies in the synthetic seismograms can be separated into two parts:

1. Inaccuracies arising from approximations in the theory used to compute the synthetic seismograms. Examples of this would include many applications of ray theory which do not properly account for head waves, diffracted waves, or the coupling between different waves types at long periods. Another computational error is the grid dispersion that occurs in most finite difference schemes.

2. Errors caused by using a simplified Earth or source model. In this case the synthetic seismogram may be exact for the simplified model, but the model is an inadequate representation of the real problem. These simplifications might be necessary in order to apply a particular numerical technique, or might result from ignorance of many of the details of the model. Examples would include the use of 1D models that do not fully account for 3D structure, the assumption of a point source rather than a finite rupture, and neglecting the effects of attenuation in the calculations.

The first category of errors may be addressed by applying a more exact algorithm; although in practice limits on available computer power may prevent achieving the desired accuracy in the case of complicated models. The second category is more serious because often one simply does not know the properties of the Earth well enough to be able to model every wiggle in the observed seismograms. This is particularly true at high frequencies (0.5 Hz and above). For teleseismic arrivals, long-period body waves (15–50 s period) and surface waves (40–300 s period) can usually be fit very well with 1-D Earth models, whereas the coda of high-frequency body wave arrivals can only be modeled statistically (fitting the envelope function but not the individual wiggles).

Because of the linearity of the problem and the superposition principle (in which distributed sources can be described as the sum of multiple point sources), there is no great difficulty in modeling even very complicated sources (inverting for these sources, is, of course, far more difficult, but here we are only concerned with the forward problem). If the source can be exactly specified, then computing synthetics for a distributed source is only slightly more complicated than for a simple point source. By far the most difficult part in computing synthetic seismograms is solving for the propagation effects through realistic velocity structures. Only for a few grossly simplified models (e.g., whole space or half-spaces) are analytical solutions possible.

The part of the solution that connects the force distribution at the source with the displacements at the receiver is termed the elastodynamic Green’s function (e.g., see p. 166 of *Introduction to Seismology*, henceforth abbreviated ITS). Computation of the Green’s function is the key part of the synthetic seismogram calculation because this part must
take into account all of the elastic properties of the material and the appropriate boundary conditions.

There are a large number of different methods for computing synthetic seismograms. Most of these fall into the following categories:

1. Finite difference and finite element methods that use brute-force computer power to solve the wave equation over a discrete set of grid points or model elements. These have the great advantage of being able to handle models of arbitrary complexity. Their computational cost grows with the number of required grid points; more points are required for 3-D models (vs. 2-D) and for higher frequencies.

2. Ray theoretical methods in which ray geometries are explicitly specified and ray paths are computed. These methods include simple (or geometrical) ray theory, WKBJ, and so-called ”generalized” ray theory. They are most useful at high frequencies for which the ray theoretical approximation is most accurate.

3. Homogeneous layer methods in which the model consists of a series of horizontal layers with constant properties within each layer. Matrix methods are then used to connect the solutions between layers. Examples of this approach include “reflectivity” and “wavenumber integration.” These methods yield an exact solution but can become computationally intensive at high frequencies because a large number of layers are required to accurately simulate continuous velocity gradients. Unlike finite difference and ray theoretical methods, homogeneous layer techniques are restricted to 1-D Earth models. However, spherically symmetric models can be computed using the Flat Earth Transformation.

4. Normal mode summation methods in which the standing waves (eigenvectors) of the spherical Earth are computed and then summed to generate synthetic seismograms. This is the most natural and complete way to compute synthetic seismograms for the spherical Earth, but is computationally intensive, particularly at high frequencies. Generalization to 3-D Earth models requires including coupling between modes; this is generally done using asymptotic approximations and greatly increases the complexity of the algorithm.

There is no single “best” way to compute synthetic seismograms as each method has its own advantages and disadvantages. The method of choice will depend upon the particular problem to be addressed and the available computer power; thus it is useful to be aware of the full repertoire of techniques. In this course (227C), we will focus on methods for computing body-wave synthetic seismograms at relatively high frequencies. We will not discuss surface waves or normal modes; these are covered in the low-frequency seismology class (227B) that is taught in alternate years with 227C.

**Finite Difference Methods**

Finite difference methods specify a model at a series of grid points and then solve the wave equation numerically. Advantages of finite differences are that an exact solution is provided and models can be of arbitrary complexity. Complicated analytical techniques are not required, although the speed of the algorithm depends upon the skill of the computer
programmer in developing efficient code. The method is currently practical on ordinary computers for 2-D models or for 3-D models at relatively low frequencies and short ranges.

Let us begin with some examples of how finite difference schemes work. Begin with the momentum equation:

\[
\rho \frac{\partial^2 \mathbf{u}}{\partial t^2} = \nabla \cdot \mathbf{\tau}
\]  
(1.1)

Now let \( \mathbf{u} = (u_x, u_y, u_z) = (u, v, w) \) and recall that \( (\nabla \cdot \mathbf{\tau})_i = \partial_j \tau_{ij} \). For the two-dimensional case of SH-waves propagating in the \( x-z \) plane, displacement only occurs in the \( y \)-direction (i.e. \( \mathbf{u} = (0, v, 0) \)) and we can write:

\[
\rho \frac{\partial^2 v}{\partial t^2} = \partial_j \tau_{yj} = \frac{\partial \tau_{yx}}{\partial x} + \frac{\partial \tau_{yz}}{\partial z}
\]  
(1.2)

Note that \( \frac{\partial}{\partial y} = 0 \) for the two-dimensional problem. Now recall (ITS 3.10) which relates stress to displacement for isotropic media:

\[
\tau_{ij} = \lambda \delta_{ij} \partial_k u_k + \mu (\partial_i u_j + \partial_j u_i)
\]  
(1.3)

Using this equation we can obtain expressions for \( \tau_{yx} \) and \( \tau_{yz} \):

\[
\tau_{yx} = \mu \frac{\partial v}{\partial x}
\]
\[
\tau_{yz} = \mu \frac{\partial v}{\partial z}
\]  
(1.4)

Substituting into (1.2), we obtain:

\[
\rho \frac{\partial^2 v}{\partial t^2} = \frac{\partial}{\partial x} \left[ \mu \frac{\partial v}{\partial x} \right] + \frac{\partial}{\partial z} \left[ \mu \frac{\partial v}{\partial z} \right]
\]  
(1.5)

Note that for one-dimensional wave propagation in the \( x \)-direction \( \frac{\partial}{\partial z} = 0 \) and the SH equation reduces to:

\[
\rho(x) \frac{\partial^2 v}{\partial t^2} = \frac{\partial}{\partial x} \left[ \mu(x) \frac{\partial v}{\partial x} \right]
\]  
(1.6)

This is equivalent to equation (13.129) in Aki and Richards. A similar equation exists for one-dimensional \( P \)-wave propagation if the \( \mu(x) \) is replaced with \( \lambda(x) + 2\mu(x) \) and the displacements in the \( y \)-direction \( (v) \) are replaced with displacements in the \( x \)-direction \( (u) \).

We can avoid the double time derivative and the space derivatives of \( \mu \) if we use the particle velocity \( \dot{v} \) and stress \( \tau = \mu \partial v / \partial x \) as variables. We then have the simultaneous equations:

\[
\frac{\partial \dot{v}}{\partial t} = \frac{1}{\rho(x)} \frac{\partial \tau}{\partial x}
\]
\[
\frac{\partial \tau}{\partial t} = \mu(x) \frac{\partial \dot{v}}{\partial x}
\]  
(1.7)
A solution to these equations can be obtained directly using finite-difference approximations for the derivatives. In order to design a stable finite-difference algorithm, it is important to use centered finite difference operators. To see this, consider the Taylor series expansion of a function $\phi(x)$

$$\phi(x + \Delta x) = \phi(x) + \frac{\partial \phi}{\partial x} \Delta x + \frac{1}{2} \frac{\partial^2 \phi}{\partial x^2} (\Delta x)^2 + \frac{1}{6} \frac{\partial^3 \phi}{\partial x^3} (\Delta x)^3 + \text{higher order terms} \quad (1.8)$$

If we solve this equation for $\frac{\partial \phi}{\partial x}$, we obtain

$$\frac{\partial \phi}{\partial x} = \frac{1}{\Delta x} \left[ \phi(x + \Delta x) - \phi(x) \right] - \frac{1}{2} \frac{\partial^2 \phi}{\partial x^2} \Delta x - \frac{1}{6} \frac{\partial^3 \phi}{\partial x^3} (\Delta x)^2 - \ldots \quad (1.9)$$

and we see that the simple approximation

$$\frac{\partial \phi}{\partial x} = \frac{1}{\Delta x} \left[ \phi(x + \Delta x) - \phi(x) \right] \quad (1.10)$$

will have a leading truncation error proportional to $\Delta x$. To obtain a better approximation, consider the expansion for $\phi(x - \Delta x)$

$$\phi(x - \Delta x) = \phi(x) - \frac{\partial \phi}{\partial x} \Delta x + \frac{1}{2} \frac{\partial^2 \phi}{\partial x^2} (\Delta x)^2 - \frac{1}{6} \frac{\partial^3 \phi}{\partial x^3} (\Delta x)^3 + \text{higher order terms} \quad (1.11)$$

Solving for $\frac{\partial \phi}{\partial x}$, we obtain

$$\frac{\partial \phi}{\partial x} = \frac{1}{\Delta x} \left[ \phi(x) - \phi(x - \Delta x) \right] + \frac{1}{2} \frac{\partial^2 \phi}{\partial x^2} \Delta x - \frac{1}{6} \frac{\partial^3 \phi}{\partial x^3} (\Delta x)^2 - \ldots \quad (1.12)$$

Averaging (1.9) and (1.12), we obtain

$$\frac{\partial \phi}{\partial x} = \frac{1}{2\Delta x} \left[ \phi(x + \Delta x) - \phi(x - \Delta x) \right] - \frac{1}{3} \frac{\partial^3 \phi}{\partial x^3} (\Delta x)^2 - \ldots \quad (1.13)$$

and we see that the central difference formula

$$\frac{\partial \phi}{\partial x} = \frac{1}{2\Delta x} \left[ \phi(x + \Delta x) - \phi(x - \Delta x) \right] \quad (1.14)$$

has an error of order $(\Delta x)^2$. For small values of $\Delta x$, these errors will be much smaller than those obtained using (1.10).
To show how a centered finite difference approach can be used to solve (1.7), consider the \(xt\) plane sampled at points \((i\Delta t, j\Delta x)\), where \(i\) and \(j\) are integers.

We can then write

\[
\frac{\dot{v}_{j+1}^i - \dot{v}_{j}^{i-1}}{2\Delta t} = \frac{1}{\rho_j} \frac{\tau_{j+1}^i - \tau_{j-1}^i}{2\Delta x}
\]

\[
\frac{\tau_{j+1}^i - \tau_{j}^{i-1}}{2\Delta t} = \frac{\mu_j}{\rho_j} \frac{v_{j+1}^i - v_{j}^{i-1}}{2\Delta x}
\]

This approach will be stable provided the time-mesh interval \(\Delta t\) is smaller than or equal to \(\Delta x/c_j\), where \(c_j = \sqrt{\mu_j/\rho_j}\) is the local wave velocity.

An even better algorithm uses a *staggered grid* approach in which the velocities and stresses are computed at different grid points, offset by half a grid length in both \(x\) and \(t\).
\[ \frac{\dot{v}_j^{i+1/2} - \dot{v}_j^{i-1/2}}{\Delta t} = \frac{1}{\rho_j} \frac{\tau_{j+1/2}^i - \tau_{j-1/2}^i}{\Delta x}, \]

As discussed in Aki and Richards (p. 777), the error in this approximation is four times smaller than in (1.15) because the sampling interval has been halved.

Now let us consider the two-dimensional P-SV system. In this case \( \mathbf{u} = (u, 0, w) \) and we can write:

\[
\begin{align*}
\rho \frac{\partial^2 u}{\partial t^2} &= \partial_j \tau_{xj} = \frac{\partial \tau_{xx}}{\partial x} + \frac{\partial \tau_{xz}}{\partial z} \\
\rho \frac{\partial^2 w}{\partial t^2} &= \partial_j \tau_{zj} = \frac{\partial \tau_{zx}}{\partial x} + \frac{\partial \tau_{zz}}{\partial z}
\end{align*}
\]

Using (1.3) we can obtain expressions for \( \tau_{xx}, \tau_{xz}, \) and \( \tau_{zz} \):

\[
\begin{align*}
\tau_{xx} &= \lambda \left[ \frac{\partial u}{\partial x} + \frac{\partial w}{\partial z} \right] + \mu \left[ 2 \frac{\partial u}{\partial x} \right] \\
&= (\lambda + 2\mu) \frac{\partial u}{\partial x} + \lambda \frac{\partial w}{\partial z} \\
\tau_{xz} &= \mu \left[ \frac{\partial u}{\partial z} + \frac{\partial w}{\partial x} \right] \\
\tau_{zz} &= \lambda \left[ \frac{\partial u}{\partial x} + \frac{\partial w}{\partial z} \right] + \mu \left[ 2 \frac{\partial w}{\partial z} \right] \\
&= (\lambda + 2\mu) \frac{\partial w}{\partial z} + \lambda \frac{\partial u}{\partial x}
\end{align*}
\]

Equations (1.2) and (1.4) are a coupled system of equations for two-dimensional SH-wave propagation, while (1.17) and (1.18) are the equations for P-SV wave propagation. As in the one-dimensional case, it is often convenient to take time derivatives of the equations for the stress (1.4) and (1.18), so that we can express everything in terms of \( (\dot{u}, \dot{v}, \dot{w}) = \frac{\partial \mathbf{u}}{\partial t} \). In this case the SH equations become:

\[
\begin{align*}
\frac{\partial \dot{v}}{\partial t} &= \frac{\partial \tau_{yx}}{\partial x} + \frac{\partial \tau_{yz}}{\partial z} \\
\frac{\partial \tau_{yx}}{\partial t} &= \mu \frac{\partial \dot{v}}{\partial x} \\
\frac{\partial \tau_{yz}}{\partial t} &= \mu \frac{\partial \dot{v}}{\partial z}
\end{align*}
\]

(1.19)
and the P-SV equations become:

\[
\begin{align*}
\rho \frac{\partial \dot{u}}{\partial t} &= \frac{\partial \tau_{xx}}{\partial x} + \frac{\partial \tau_{xz}}{\partial z} \\
\rho \frac{\partial \dot{w}}{\partial t} &= \frac{\partial \tau_{xx}}{\partial x} + \frac{\partial \tau_{zz}}{\partial z} \\
\frac{\partial \tau_{xx}}{\partial t} &= (\lambda + 2\mu) \frac{\partial \dot{u}}{\partial x} + \lambda \frac{\partial \dot{w}}{\partial z} \\
\frac{\partial \tau_{xz}}{\partial t} &= \mu \left[ \frac{\partial \dot{u}}{\partial z} + \frac{\partial \dot{w}}{\partial x} \right] \\
\frac{\partial \tau_{zz}}{\partial t} &= (\lambda + 2\mu) \frac{\partial \dot{w}}{\partial z} + \lambda \frac{\partial \dot{u}}{\partial x}
\end{align*}
\] (1.20)

These are first-order systems of equations in velocity and stress which can be solved numerically. In this case, the elastic properties, \( \rho, \lambda, \) and \( \mu \) are specified at a series of model grid points. With suitable starting conditions, the velocities and stresses are also defined at grid points. The program then calculates the required spatial derivatives of the stresses in order to compute the velocities at time \( t + \Delta t \). The spatial derivatives of these velocities than allow the computation of new values for the stresses. This cycle is then repeated.

Finite difference methods vary depending upon how the temporal and spatial derivatives in these equations are calculated. Simple first-order differencing schemes for the spatial derivatives are fast and easy to program, but require more grid points per wavelength to achieve accuracy comparable to higher-order differencing schemes. The best finite difference programs use the staggered grid approach in which the velocities and stresses are computed at different grid points.

A few points to keep in mind:

1. Finite difference programs run much faster if the required arrays fit into memory and don’t need to be swapped to and from disk. Thus machines with large memories are desirable. Higher-order finite difference schemes generally also have an advantage because fewer grid points per wavelength are required for accurate results.

2. Simple first-order differencing schemes typically require about 20 grid points per wavelength. Higher-order schemes require less grid points per wavelength. So called pseudo-spectral methods are equivalent to very high order differencing methods and require only 2 grid points per wavelength. However, models with sharp velocity discontinuities often require more than 2 points per wavelength, so some of the advantage of the higher order methods is lost in this case.

3. Finite difference methods are most useful at relatively low frequencies and short ranges. Models typically are no larger than a few tens of wavelengths.

4. An important aspect of finite difference methods is devising absorbing boundary conditions to prevent annoying reflections from the edges of the model. This is a nontrivial problem and many papers have been written discussing various techniques. Most of these
methods work o.k. for waves hitting the boundaries at near normal incidence, but have problems for grazing incidence angles.

(5) As computer speed and memory size increases, finite difference methods will become practical even on desktop machines. Eventually they will probably largely replace many of the alternative methods.

(6) Finite-element programs have advantages over finite differences in applying boundary conditions. Currently the most developed finite element program in seismology is the implementation of the spectral-element method by Komatitsch, Tromp and coworkers (e.g., Komatitsch et al., 2002). This program is designed to run in parallel on large computing clusters, such as the Beowulf PC cluster at Caltech or the "Earth Simulator" in Japan.
Additional reading

Aki and Richards, section 13.6.