SW

Seismic Wave Propagation Modeling and Inversion

author: Phil Bording (bordingcsep1.phy.ornl.gov)

1 Introduction to Wave Propagation

The propagation of energy via waves is a familiar phenomenon in our everyday life. The particular waves to be studied here are seismic waves which are intentionally created to image the interior of the earth [1], [8] and [21]. Our three dimensional earth consists of more than the geological structures we are accustomed to thinking about, much of the earth is fluid or fluid-like. Here the principal fluids of interest are hydrocarbons. The other essential fluid to consider is water. The fluid-like materials include the many gases trapped in earth, gases like carbon dioxide, helium, and natural gas. Actually, we may not be aware of the less visible subsurface structure, but all of the surface geology you can observe, and more, exists in some form under the surface.

To find accumulations of petroleum requires an intimate knowledge of the subsurface geology, the history of the material source and the structure of the subsurface. A reservoir requires porosity, a sealing mechanism, and a hydrocarbon source. The storage capacity is dependent on the porosity, the seal prevents leakage of the hydrocarbons, and the source generates the hydrocarbons. Note, the source rocks are not always the same as the reservoir rocks.

To produce hydrocarbons the reservoir must be found and be capable of producing fluids. The interconnections of the porous spaces, the permeability, permits flow of the gases and liquids. Tightly connected porous spaces are difficult producers, but well connected spaces have good permeability and are productive. The oil exploration process finds possible drilling locations, and the actual drilling of a well is used to test the geological hypothesis of hydrocarbon existence.

We shall study the seismic method for determining the subsurface structure. The geophysical technique is to generate artificial seismic waves and record their reflections from impedance differences within the earth. Echoes come from the reflections created by relatively hard surfaces where the impedance changes.

Mathematically, the simplest hyperbolic partial differential equation is the constant density acoustic wave equation. The basis for using this particular wave equation [2] will be developed further. Because the computational effort of solving three dimensional problems ([16] Chapter 10 and [15] Chapter 1) exceeds most computing environments, this study will primarily focus on one and two dimensional problems. All the techniques and algorithms presented here can be directly extended to three dimensions.

The constant density assumption simplifies model representation: only a sound speed is required. The earth density variation is important for modeling and imaging. However, neglecting density variation will still provide a useful wave equation. One additional comment about earth parameters: surface measurements using physical methods use potential fields, gravity for example. These physical measurements are of a different scale compared to reflection seismology. The seismic data wavelength has sufficient resolution for structural imaging. Resolution of the seismic experiment is a direct function of the wavelength; the shorter the wave length the higher the resolution.

The acoustic assumption is also in contrast to the elastic assumption. A fluid medium supports the propagation of a pressure wave. An elastic medium supports both shear and pressure waves. Marine seismic data are collected using water borne receivers or receivers which rest on the ocean bottom. Propagation of shear energy is restricted to solid media, and this makes land seismic measurement the principal generator of elastic seismic data. However, it is possible to find mode converted elastic information within an acoustic marine seismic dataset. In an elastic medium when waves impinge on a reflector, both shear and pressure waves are created, i.e. there is a mode conversion.

Seismic pressure waves are recorded by using a *geophone*, a microphone on a spike. The geophone has a small weight which is spring mounted with a *magnet and a coil of wire.* The pressure wave from the vibrating earth generates a vertical displacement moving the coil while the weight tries to resist this motion. The magnetic field generates a voltage proportional to the earth acceleration. Elastic waves have three components of displacement. *Three geophones* are aligned in orthogonal directions to measure the pressure wave and the two shear waves.

The seismic wave is man-made and requires a significant amount of energy to propagate any distance in the earth. Dynamite charges are used to generate the primary pressure wave (P-wave) and some shear wave energy (S-wave).

2 The Modeling Domain and Wave Equations

The acoustic wave equation in two dimensions relates the spatial derivatives to the time derivative. Using the acoustic approximation the wave equation is derived in Section 4. The dimensionality of the equation requires a media velocity $C(x, z)$, the sound speed. In the two dimensional model of Figure 1 the surface coordinate is x and the depth coordinate is

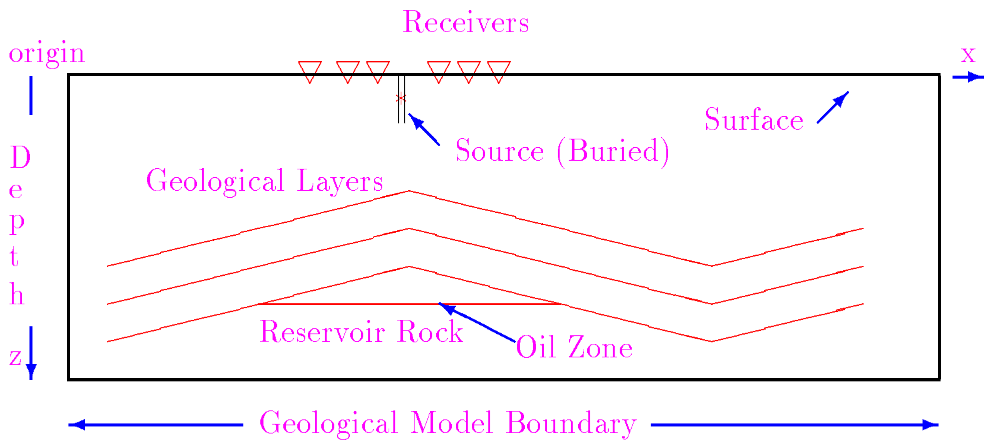

Figure 1: Geological Model

z with the positive axis down, a tradition in the oil industry. This half space is defined for $z_{max} \geq z \geq 0$ and for a bounded $x_{min} \leq x \leq x_{max}$.

The pressure wave field is Ψ and the seismic source is $src(t)$. The inhomogeneous constant density two dimensional wave equation is

$$\frac{1}{C(x,z)^2}\frac{\partial^2 \Psi}{\partial t^2} = \left[\frac{\partial^2 \Psi}{\partial x^2} + \frac{\partial^2 \Psi}{\partial z^2}\right] + src(t). \tag{1}$$

This equation is hyperbolic and the inhomogeneity is due to the variable model velocities. The seismic source $src(t)$ is applied at or near the surface. Sources are sometimes buried to improve the signal strength and to reduce noise generated by the near surface. Receivers are placed in a line parallel to the expected principal dip of the subsurface. This is done to reduce the energy scattering by out of plane reflectors.

The rest of this chapter is organized in the following manner:

- The first order wave equation is presented.
- The wave equation is developed from first principles.
- Model results for a one dimensional problem are used to illustrate the reflection coefficient.
- A matrix formulation for computing difference weights is reviewed.
- The seismic source is presented as a numerical method to produce realistic seismic waves with a finite duration in time and limited in bandwidth.
- A geological model is defined and two methods are presented which reduce unwanted reflections from the model boundaries.
- The exploding reflector model is introduced as a method to generate a synthetic seismogram.
- The "reverse" time imaging concept is presented to process the synthetic seismic data.

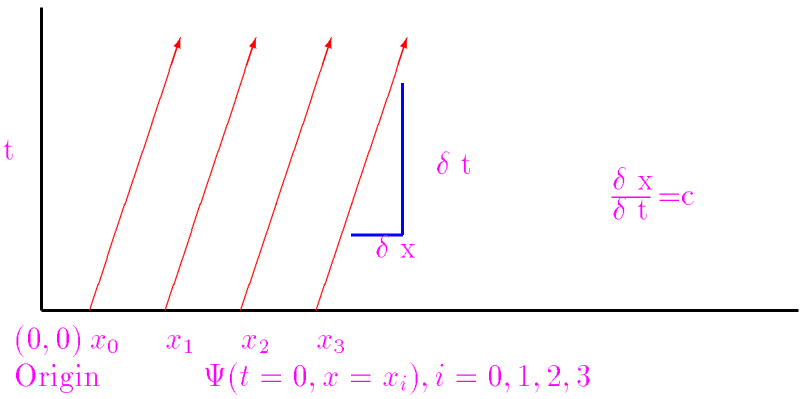

Figure 2: First Order Wave Equation

- The seismic modeling and imaging codes are introduced with operating instructions and several example models.

The seismic data are too complex to visualize as numbers so visualization tools are provided to help in understanding the modeling and imaging process.

3 First Order Wave Equation

Conceptually, a first order hyperbolic wave equation is the simplest wave propagation model. The characteristic solution [20] in the time–space domain for the homogeneous, constant velocity model is illustrated in Figure 2. The first order hyperbolic system, Equation 2, has an analytic solution, $\Psi(x-ct)$. The initial condition is $\Psi(t=0, x=x_i)$ and the slope of the characteristic solution is the media velocity c.

$$\frac{\partial \Psi}{\partial x} + \alpha \frac{1}{c} \frac{\partial \Psi}{\partial t} = 0 \tag{2}$$

The sign of α, $\alpha = \pm 1$, of Equation 2 determines the propagation direction of the wave. If α is positive, the wave propagates to the right. Bidirectional waves are a property of the second order hyperbolic wave equation. To develop a non-reflecting boundary condition, Reynolds [18] used first order systems similar to Equation 2. This method of Reynolds is presented in Section 15.

4 Wave Equation

To develop the wave equation, Equation 1, from first principles we will consider the disturbance of a fluid-like medium. The conservation of mass and momentum provide the basis for development of the acoustic wave equation ([14], [5], [6], and [3]). The mass density is ρ, the particle velocity is Ψ, and the fluid pressure is P. The three spatial coordinates are x_i $(i = 1,2,3)$ for the domain Ω. Particle velocities Ψ_i are for each direction x_i.

The conservation of momentum is

$$\frac{\partial \rho \Psi_i}{\partial x_j}\Psi_j + \frac{\partial \rho \Psi_i}{\partial t} - \frac{\partial \sigma_{ij}}{\partial x_j} = 0. \tag{3}$$

The conservation of mass, the continuity equation, is

$$\frac{\partial \rho \Psi_i}{\partial x_i} + \frac{\partial \rho}{\partial t} = 0. \tag{4}$$

The incompressible fluid flow equation can be derived from the Navier–Stokes equations. The form used here is Euler's equation when the viscosity is zero and uses D as the *Substantial Derivative*

$$\rho \frac{D\Psi}{Dt} = f_b - \nabla P. \tag{5}$$

The body forces are negligible, $f_b = 0$. Notice that the repeated index used in the equations indicate a tensor summation.

The force $S(t)$ in Equation 6 acts as a source function upon the domain Ω. Any force acting within the domain causes pressure and density changes, and the fluid nature of the medium will create an equilibrium restoring force. These changes in density and pressure are small.

$$\frac{\partial \rho \Psi_i}{\partial x_j}\Psi_j + \frac{\partial \rho \Psi_i}{\partial t} + \frac{\partial P}{\partial x_i} = S(t). \tag{6}$$

We consider small perturbations Δ in the density ρ, the particle velocity Ψ, and the pressure P from the initial rest conditions which are labeled with subscript 0. The perturbation Equations 7, 8, and 9 are for Ψ_t, ρ_t, and P_t, the particle velocity, density, and pressure respectively.

$$\Psi_t = \Psi_0 + \Delta\Psi \tag{7}$$

$$\rho_t = \rho_0 + \Delta\rho \tag{8}$$

$$P_t = P_0 + \Delta P \tag{9}$$

The initial particle velocity is $\Psi_0 = 0$ because the domain Ω is at rest.

The density perturbation is known as the *acoustic approximation* [11]. A fluid has a pressure which is a function of density, temperature, and gravity forces. We shall assume that the gravity forces are relatively constant over the domain Ω and do not exert any differential force on the fluid. The earth temperature field does vary as a function of position and in time, but the temperature changes are very small compared to the time it takes to make seismic measurements. Hence, we will assume that only the density is important and that the stress within the fluid is related to the strain as a function of density.

First, δ_{ij} is defined as the Kronecker delta function,

$$\delta_{ij} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{10}$$

The stress matrix is σ_{ij}

$$\sigma_{ij} = -P(\rho)\delta_{ij} \tag{11}$$

Now using Euler's Equation (Eq. 5) with the neglected body forces

$$\rho \frac{D\Psi}{Dt} = -\nabla P_t. \tag{12}$$

and realizing that P_0 is constant, we have

$$\rho \frac{D\Psi}{Dt} = -\nabla \Delta P. \tag{13}$$

Now the initial medium is at rest and has no convective acceleration, which permits changing the form of the derivatives in Equation 13

$$\rho \frac{\partial \Psi}{\partial t} = -\nabla \Delta P. \tag{14}$$

Next Ψ is the gradient of Φ, and we can also appreciate that the product of $\Delta\rho$ and the gradient of Φ will be small

$$\nabla \Phi = \Psi \tag{15}$$

and

$$\rho_0 \frac{\partial \nabla \Phi}{\partial t} = -\nabla \Delta P. \tag{16}$$

Next we assume the derivatives of time and space can be exchanged

$$\nabla \rho_0 \frac{\partial \Phi}{\partial t} = -\nabla \Delta P. \tag{17}$$

Removing the operator Δ gives

$$\rho_0 \frac{\partial \Phi}{\partial t} = -\Delta P. \tag{18}$$

The compressibility C and the bulk modulus of elasticity K are defined in terms of a unit volume V and ΔV.

$$C = \frac{\Delta\rho}{\rho} = -\frac{\Delta V}{V} \tag{19}$$

$$\Delta P = -K \frac{\Delta V}{V} \tag{20}$$

$$\Delta\Delta P = K \frac{\Delta\Delta\rho}{\rho} \tag{21}$$

Now computing the derivative with respect to time in Equation 21, the change in pressure is related to the change in density

$$\frac{\partial \Delta P}{\partial t} = \frac{K}{\rho_0} \frac{\partial \Delta\rho}{\partial t} \tag{22}$$

$$\nabla^2\Phi = -\frac{1}{K}\frac{\partial \Delta P}{\partial t}. \tag{23}$$

Using Equation 4 to conserve mass and Equations 17 and 23,

$$\frac{\partial \rho \Psi_i}{\partial x_i} = -\frac{\partial \Delta\rho}{\partial t} \approx \rho_0 \nabla^2\Phi. \tag{24}$$

Taking the time derivative of Equation 18,

$$\frac{\partial}{\partial t}\rho_0\frac{\partial \Phi}{\partial t} = -\frac{\partial \Delta P}{\partial t}. \tag{25}$$

Finally, the acoustic wave equation is

$$\nabla^2\Phi = \frac{1}{v^2}\frac{\partial^2\Phi}{\partial t^2} \tag{26}$$

where $v = \sqrt{\frac{K}{\rho_0}}$ is the sound speed of the medium.

5 One Dimensional Problem

The one dimensional wave equation with a homogeneous velocity function has an analytic solution [22]. The wave propagates in two directions; solutions are of the form $\Psi_0(x-ct)+\Psi_1(x+ct)$. The method of characteristics can be used to formulate this analytic solution. The numerical solution [10] has the advantage of solving the inhomogeneous problem, which is awkward analytically but feasible in one dimension. After mastering the requirements of one dimensional modeling, the extensions required for two and three dimensional models are not difficult. Inhomogeneous modeling in two and three dimensions requires a numerical method; analytic methods are not capable of modeling most complex media.

$$\frac{1}{C(x)^2}\frac{\partial^2\Psi}{\partial t^2} = \left[\frac{\partial^2\Psi}{\partial x^2}\right] + src(t). \tag{27}$$

The model used to demonstrate the propagation of waves in one dimension is shown in Figure 3. The source is placed at the center and the signal is propagated in both directions from the source. This model is symmetric and is homogeneous. No reflected waves will be generated and this wave is plotted at time equals 2.0 seconds as displayed in Figure 4.

If the model is modified, assume that the velocity is increased by 50 percent to the right 2000 meters from the source. Then when the initial source wave impinges on the impedance a reflected wave will be generated. This inhomogeneous model is plotted in Figure 5 and the seismic waves are plotted in Figure 6. The strength of the reflected wave is determined by the reflection coefficient [19]. Given the velocities and densities the reflection coefficient R is

$$R = \frac{\rho_1 v_1 - \rho_0 v_0}{\rho_1 v_1 + \rho_0 v_0}. \tag{28}$$

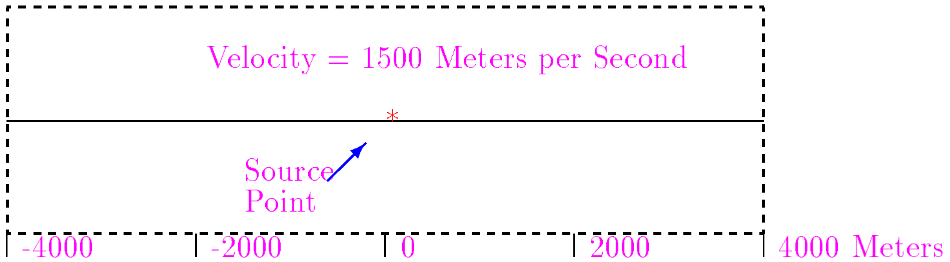

Figure 3: Homogeneous One Dimensional Model

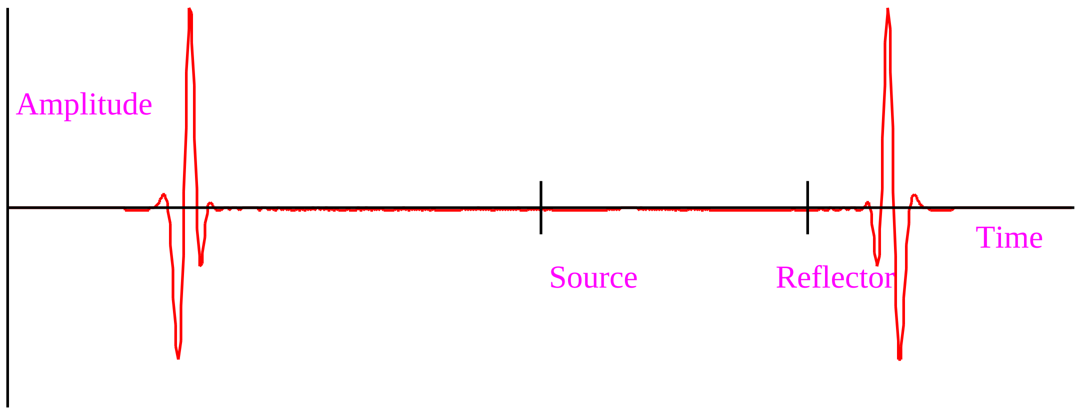

Figure 4: Homogeneous Waves

The subscript 0 is for the incident medium and the subscript 1 is for the transmitting medium. When the density is a constant this simplifies to

$$R = \frac{v_1 - v_0}{v_1 + v_0}. \tag{29}$$

The transmitted energy is T, $T = 1 - R$. If the velocity function increases with depth, the reflection coefficient is positive; if it decreases, the reflection coefficient becomes negative. The sign change is physically realized as a polarity change in the reflected signal which is observed in Figure 7.

The inhomogeneous model is modified with a velocity reduction of 50 percent to 750 m/s at the same 2000 meters from the right. This result is shown in Figure 7.

The reflection coefficients for the velocity changes are

$$R_+ = \frac{2250 - 1500}{2250 + 1500} = 0.200 \tag{30}$$

$$R_- = \frac{750 - 1500}{750 + 1500} = -0.333 \tag{31}$$

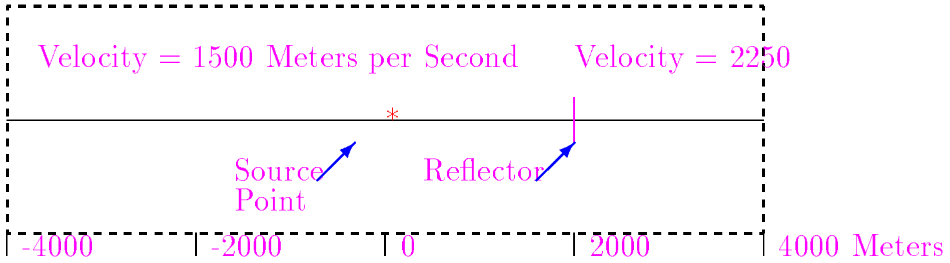

Figure 5: Inhomogeneous One Dimensional Model

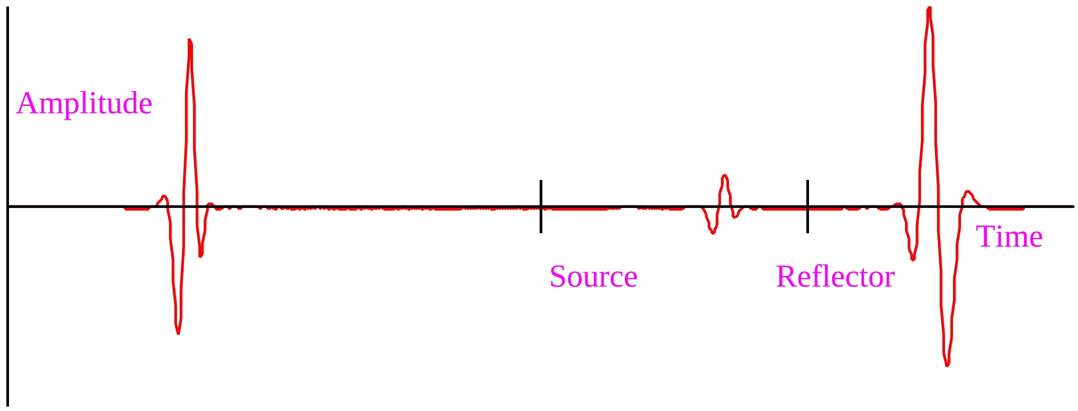

Figure 6: Reflected Waves, Velocity Increased 50%

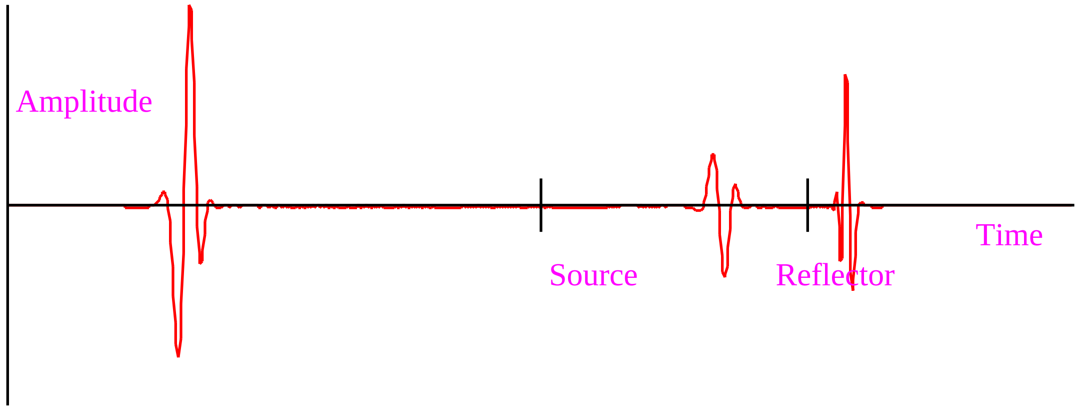

Figure 7: Reflected Waves, Velocity Decreased, 50%

The corresponding transmission coefficients for the velocity changes are

$$T_+ = 1 - R_+ = 0.800 \tag{32}$$

$$T_- = 1 - R_- = 1.333 \tag{33}$$

If we examine Figures 6 and 7, the reflections show both the amplitude differences and the sign differences of Equations 30 and 31. Likewise, the transmission coefficients of Equations 32 and 33 compare well with the figures.

6 Derivative Coefficients

The time and space derivatives must be discretized from a continuous function to a discrete function. Methods used to compute derivatives for seismic modeling include Taylor Series, Chebechev, Fourier Transforms, and Padé. The Taylor Series (TS) methods will be developed here; they are reasonably good approximations but not optimal. The theories for optimal operators are beyond the scope of this effort. The TS method assumes that a function known at point a can be extended to point b if sufficient number of derivatives exist and are known at point a. The truncation error for the series expansion has a maximum within the approximation interval.

The formulation preferred is a matrix representation for computing the coefficients of the Taylor Series. Given

$$h = b - a \tag{34}$$

then

$$f(b) = f(a) + f'(a)h + \frac{f''(a)h^2}{2!} + \frac{f'''(a)h^3}{3!} \cdots \tag{35}$$

This computation of $f(b)$ does not explain how to compute a derivative approximation. But if the point b is extended to a series of uniformly spaced points each a multiple of h from a and we write the TS expansion as an expression, we get

$$\begin{array}{rcllll} f(h) & = & f(0) & +f'(0)h & +\dfrac{f''(0)h^2}{2} & \ldots \\ f(0) & = & f(0) & & & \\ f(-h) & = & f(0) & -f'(0)h & +\dfrac{f''(0)h^2}{2} & \ldots \end{array} \tag{36}$$

Assume the difference operator is centered at the origin. Now formulating this as a matrix equation using terms up to the second derivative, we have

$$\begin{pmatrix} f(h) \\ f(0) \\ f(-h) \end{pmatrix} = \begin{pmatrix} 1 & h & \frac{h^2}{2} \\ 1 & 0 & 0 \\ 1 & -h & \frac{h^2}{2} \end{pmatrix} \begin{pmatrix} f(0) \\ f'(0) \\ f''(0) \end{pmatrix} \tag{37}$$

Thus an invertible matrix equation is generated. Care must be taken to use an appropriate value for h; the simplest method is to factor the h terms out with the functions. Otherwise, for longer operators the matrix can be numerically unstable and difficult to invert. The h terms contribute to a poor condition number matrix. The result is general and will generate difference operators for any length.

Solving Equation 37 for the $f(0)$ and derivative terms gives

$$\begin{pmatrix} f(0) \\ f'(0) \\ f''(0) \end{pmatrix} = \frac{1}{h^2} \begin{pmatrix} 0 & h^2 & 0 \\ \frac{h}{2} & 0 & -\frac{h}{2} \\ 1 & -2 & 1 \end{pmatrix} \begin{pmatrix} f(h) \\ f(0) \\ f(-h) \end{pmatrix} \tag{38}$$

If we modify the matrix terms by factoring out the terms in h in Equation 37, we get a simpler matrix which will have a better condition number:

$$\begin{pmatrix} f(h) \\ f(0) \\ f(-h) \end{pmatrix} = \begin{pmatrix} 1 & 1 & \frac{1}{2} \\ 1 & 0 & 0 \\ 1 & -1 & \frac{1}{2} \end{pmatrix} \begin{pmatrix} f(0)h^0 \\ f'(0)h^1 \\ f''(0)h^2 \end{pmatrix} \tag{39}$$

This can be solved like Equation 38, and we see the familiar weights for the first and second derivatives:

$$\begin{pmatrix} f(0)h^0 \\ f'(0)h^1 \\ f''(0)h^2 \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 \\ \frac{1}{2} & 0 & -\frac{1}{2} \\ 1 & -2 & 1 \end{pmatrix} \begin{pmatrix} f(h) \\ f(0) \\ f(-h) \end{pmatrix} \tag{40}$$

The second derivative equation is

$$f''(0) = \frac{1}{h^2}\left[f(h) - 2f(0) + f(-h)\right]. \tag{41}$$

The coefficients for the derivative approximations with different grid spacings, or with more grid points, can be computed with this method.

7 Seismic Source Function

To generate seismic waves a source function is required. This section develops the concept of a frequency band limited source function. Real seismic sources use an energy impulse or a vibration source to generate waves in the earth. Impulse sources include dynamite, a drop weight, a sledge hammer, a shot gun, or a rifle. The actual source used is dependent on the desired signal to noise ratio, the human environment, the target depth desired and the geological environment. The vast majority of land seismic data are generated by two methods, dynamite placed in shot holes drilled into the earth or by a truck mounted vibrating mass which shakes the earth in a vertical or horizontal direction. The method we will use in our simulation of seismic exploration is the impulsive dynamite source.

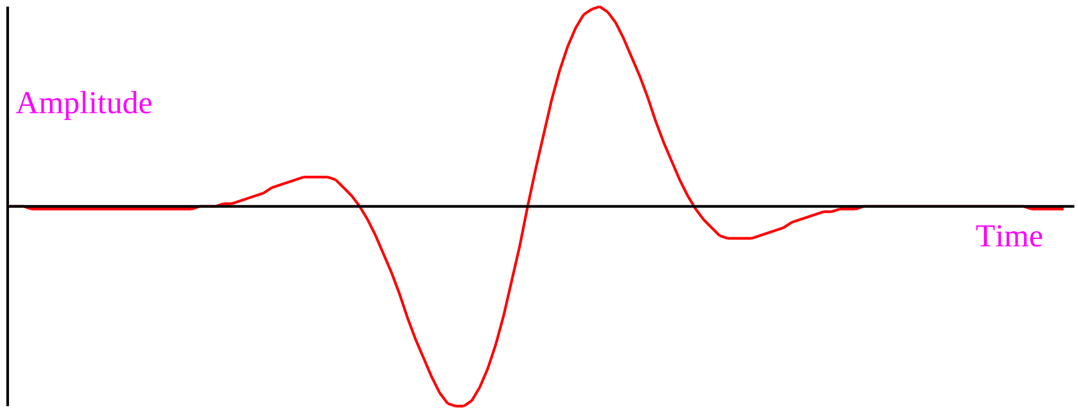

Figure 8: Source Function

The source is finite in duration. This requires a time varying function. The following exponential equation is suitable, where α and β determine the maximum value and the length of time, t_{max}. The required frequency content for the model is f_{max}; a typical value is 30 Hertz.

$$d(t) = \frac{\exp(-\alpha t^2)}{\beta}. \tag{42}$$

Now the source needs an oscillatory function and the symmetric Sine will do nicely,

$$s(t) = \mathrm{Sine}\,(2\pi f_{max} t). \tag{43}$$

Our simple seismic source is the product of $d(t)$ and $s(t)$

$$\alpha = 2\pi f_{max} \tag{44}$$

$$\beta = 9 \tag{45}$$

$$src(t) = \mathrm{Sine}\,(2\pi f_{max} t)\,\frac{\exp(-\alpha t^2)}{\beta}. \tag{46}$$

The plot in Figure 8 is for the source function, $src(t)$.

8 Wave Propagation Example

With a source function and derivative approximations it is possible to model waves, and Figure 9 is an example of a propagating wave. The model is a rectangular box with the source placed at the center. The model velocity is uniform and no edge boundary condition is applied. Boundary conditions are presented later.

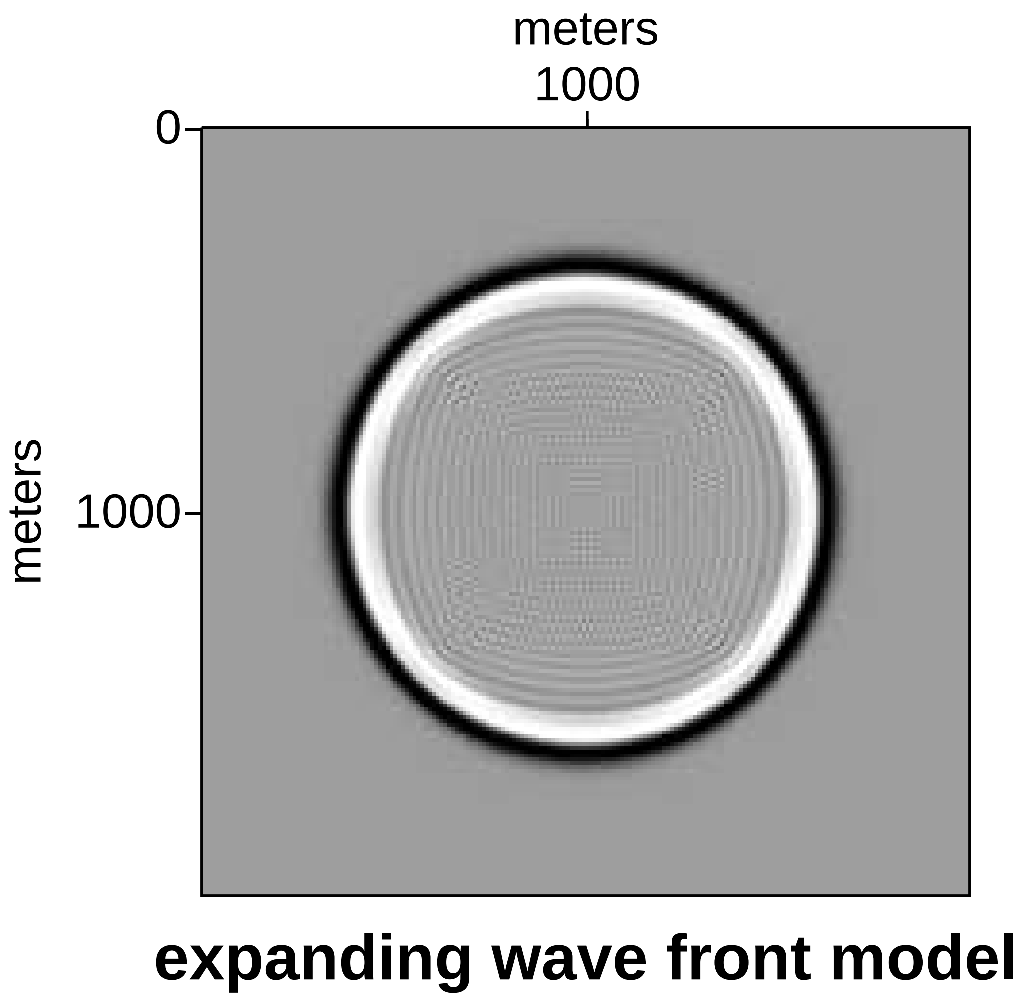

Figure 9: Two Dimensional Model Data, an Expanding Wave Front

8.1 The Finite Difference Approximation

The finite difference equation, based on second order in time and second order in space approximations, is

$$P_{i,j}^{n+1} - 2P_{i,j}^{n} + P_{i,j}^{n-1} = \alpha^2 \left[P_{i+1,j}^{n} + P_{i-1,j}^{n} + P_{i,j+1}^{n} + P_{i,j-1}^{n} - 4P_{i,j}^{n} \right] + src^n. \tag{47}$$

Equation 47 is the discrete approximation to the wave equation shown in Equation 1. The choice of the right hand side (RHS) time value as n describes an explicit time marching scheme where the $n+1$ is a function of n and $n-1$:

$$P_{i,j}^{n+1} = 2P_{i,j}^{n} - P_{i,j}^{n-1} + \alpha^2 \left[P_{i+1,j}^{n} + P_{i-1,j}^{n} + P_{i,j+1}^{n} + P_{i,j-1}^{n} - 4P_{i,j}^{n} \right] + src^n. \tag{48}$$

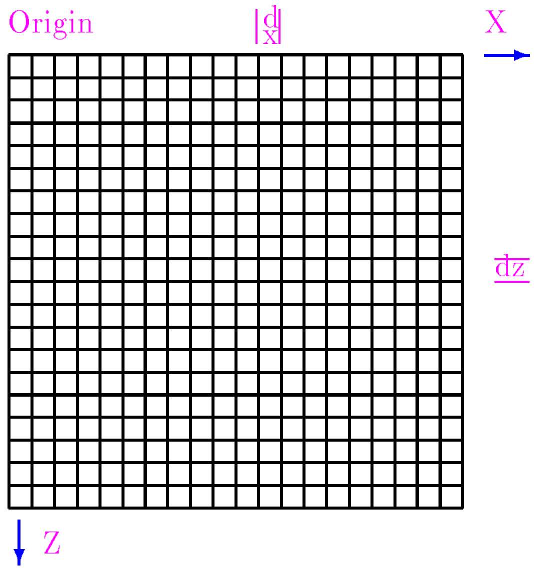

Figure 10: Two Dimensional Mesh

If the RHS was chosen to have an $n + 1$ time value, the resulting difference equation would have been implicit. The implicit form requires solving a system of equations at each time step and has a significant increase in computational complexity. The explicit form Equation 48 is used here with good results, and in a later section the stability of the explicit method is considered.

8.2 The Meshed Grid

In one dimension, the concept of a mesh is a bit difficult, however in two and three dimensions it is vital to finite difference methods. The one dimensional method used a uniform step dx, and in the two dimensional method we will again use a uniform step $h = dx = dz$ in both directions as in Figure 10.

The continuous velocity function $C(x, z)$ is discretized into an average value to each square of the mesh and is assumed to be an appropriate approximation. This assumption is valid if h is small compared to the wavelengths of propagation.

9 Stability Condition

The stability of the finite difference method is essential. The scheme in Equation 48 is explicit in time; for each new step the wave values are determined from the previous values. The mesh spacing is h. The size of the time step dt is limited; information cannot be propagated across the mesh faster than the mesh velocity. The mesh velocity is h/dt. Hence, the time step δt must be bounded, and for the difference equation this limit is

$$\frac{c\delta t}{h} \leq \frac{1}{\sqrt{n_d}} \tag{49}$$

where n_d is the number of spatial dimensions.

In two dimensions the stability condition is

$$\frac{c\delta t}{h} \leq \frac{1}{\sqrt{2}}. \tag{50}$$

The maximum time step size is bounded by

$$\delta t \leq \frac{h}{c\sqrt{2}}. \tag{51}$$

10 Seismic Modeling in Two Dimensions

10.1 Shot Record Modeling

Assuming a model velocity structure $C(x, z)$ is known and the source location $S_{shot}(x, z)$ is specified, it is possible to simulate a sequence of shot records. These individual shot records are recorded as data $D_{shot}(x, t)$ and processed like field data.

The geometry of the source location, the receiver separation, and the number of receivers used depends on a number of factors. Close sample spacing in the seismic experiment provides better data quality and improves the signal to noise ratio. The subsurface structure scatters energy in different directions. If the structure has significant dip then long receiver *spreads* are used. The cost of shots and drill holes also influences the geometry of seismic data collection.

These geometry considerations are not as critical in seismic modeling because the physical field constraints do not apply. The amount of data generated by a seismic model can be a problem, and it is not always wise to record every grid point along the receiver line for every time step. Sufficient data must be recorded to prevent aliasing the data [19]; the Nyquist limit is two grid points per wave length. Care must be taken to collect data which is not aliased either in space or time. While two grid points per wavelength is the theoretical limit, careful experimenters use at least three.

The routine seismic processing sequence will attempt to flatten the time recordings and sum into a stack section. This stacking process reduces the amount of data by a significant amount. The individual shot records are corrected for offset, the distance from the source location, and then summed. This process is termed a moveout correction. The equivalent seismic process would be to use only one receiver for every shot point location and record this data. This coincident source and receiver position is known as zero-offset data and would not need moveout correction.

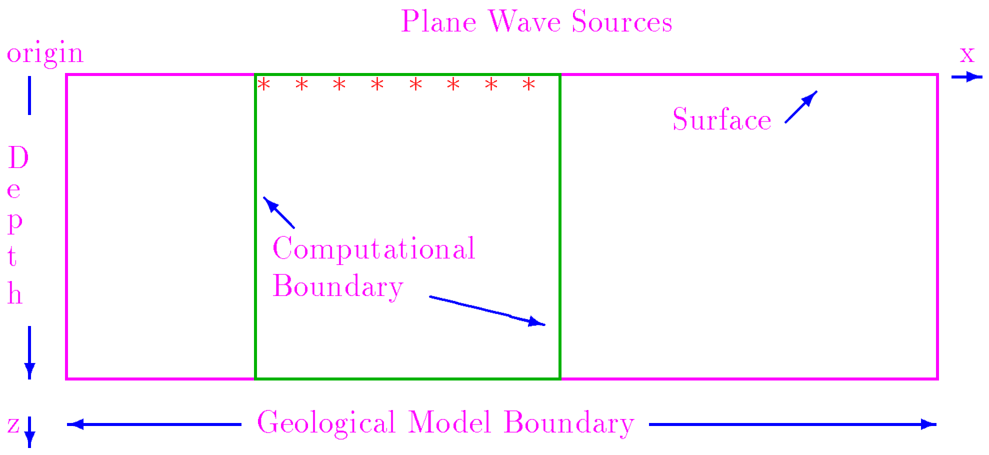

Figure 11: Plane Wave at the Surface

10.2 Model Complexity

The grid spacing is n_x and n_z in the x and z direction, respectively. The number of time steps are n_t and the computational effort is w_s, or floating point operations per step. The number of individual shot records is n_s. The n terms are about the same magnitude and w_s is considerably less in value, typically around 25; w_s is the operations count of the finite difference operator. The complexity of shot record modeling is $O(n_x n_z n_t w_s n_s)$ and the complexity of zero-offset modeling is $O(n_x n_z n_t w_s)$ which is $O(n_s)$ less.

10.3 Zero-Offset and Plane Waves

The equivalent of the zero-offset data is to excite all the surface sources at the same time. In an actual field experiment this is difficult, because then all sources must be set off at exactly the same time. The recording is for a single experiment, and the signal enhancement benefit of multiple shots is lost.

The simulation of a normal plane wave, where all surface sources are excited at the same time and send energy in the form of a wave into the earth model, is illustrated in Figure 11. This plane wave moves down until a reflector is struck, and then the reflected wave travels the same path back to the receiver and the transmitted wave continues on. The wave front is distorted by the velocity field and bends and twists according to Snell's law, Equation 52,

$$\frac{\sin\theta_i}{v(x,z)_i} = \frac{\sin\theta_{i+1}}{v(x,z)_{i+1}} \tag{52}$$

where the θ angles are between the reflector normal and the wavefront normal. Above the reflector is the i^{th} velocity and below is the $i+1^{th}$ velocity.

The exploding reflector concept is the realization that if the downgoing wave travels the same path as the reflected upgoing wave, then only one wave is needed. The upgoing wave is recorded and hence is the important one. By making each reflector position a source point

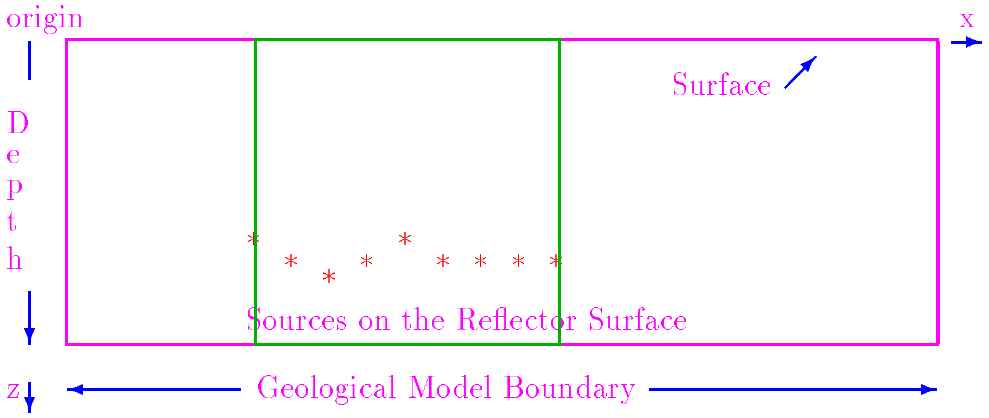

Figure 12: Exploding Reflector at Depth

as in Figure 12 it is possible to generate zero-offset data. The two way travel times are generated by halving the interval velocity of the model. Thus it is possible to generate a stacked seismic section model using a finite difference method and the exploding reflector method. The magnitude of the reflector source must be adjusted to correct for the energy losses as the wave travels in the model. The benefit of less complexity is less computing for the exploding reflector method.

The plane wave does not generate all the waves observed in seismic field data. For example, a diffraction is a reflected wave from a point impedance and in a homogeneous medium they appear as circular wave fronts.

10.4 Example Models

Three models are used to demonstrate the usefulness of this method. They are a dipping layer model, a reef model, and a salt model. Each model illustrates some difficulty associated with the seismic data processing method.

- Dipping layers move the expected midpoint; the exploding reflector source point is not at the midpoint between source and receiver. The dipping layer model is shown in Figure 20.
- The reef model has a seal with a high porosity zone associated with a natural gas or oil accumulation. This is indicated by a higher than expected reflection coefficient and a phase reversal of the signal. The reef model is shown in Figure 23.
- The salt structure has the feature of a slow moving slug of salt rising up through layered sediments and trapping around the flanks pockets of hydrocarbon. The signals from such an intrusion are reflected far from the expected vertical position, which makes it very difficult to correctly position the exploration well site. Salt has a very high sound velocity, which further complicates accurate processing of the data. The salt model is shown in Figure 26.

After studying the seismic modeling process we shall find a way to process this zero-offset data and image it. This imaging process we shall call migration, the effect of moving data mispositioned in time to a place in depth. The exploding reflector modeling data are plotted in Figures 21, 24, and 27 for the dipping layer, reef, and salt models, respectively.

10.5 Model Building

The models are simple line segments with a z-intercept and a slope and overlapped circles placed in sequence. Each interface line segment is used as a boundary and all velocity grid values below the line segment are changed to the new value. The circles are used to modify these grid values, with all points within the circle of specified radius and origin acquiring a new velocity value. The sequence of lines and circles can vary to create interesting models.

11 Imaging, Two Dimensions

The partial differential equation for modeling, the wave equation, assumes a quiet background as the initial condition. What would happen if the recorded seismic data were introduced into the wave equation as a boundary condition. The data would propagate into the earth from the receiver positions very much like the exploding reflector model process if the reflector was at the surface. If the wave equation is formulated as a backward in time equation, the boundary condition data can be moved down in depth until time reaches zero. This zero time is called the imaging condition, $t = 0$.

We shall see that this method works and propagates the surface recorded data back into depth. What is missing is the energy which left the sides and bottom of the image field. It is possible to record along the sides of an image field, but this is not done very often. The bottom data are quite difficult and expensive to obtain. Essentially, the only data routinely available are the surface seismic shot record data, which is stacked to improve the signal to noise ratio. This stacked data can then be migrated. Sounds so simple, but what is the earth velocity model to be used? If we examine Equation 53, we see that the velocity term is essential. We shall assume the velocities are known.

Equations 53 and 54 exchange the P^{n+1} and P^{n-1} terms. The source term src^n is removed, and the boundary condition bc^n is applied every time step. This backward in time migration of seismic data is called *Reverse Time Migration.* The modeling finite difference equation is

$$P_{i,j}^{n+1} = 2P_{i,j}^{n} - P_{i,j}^{n-1} + \alpha^2 \left[P_{i+1,j}^{n} + P_{i-1,j}^{n} + P_{i,j+1}^{n} + P_{i,j-1}^{n} - 4P_{i,j}^{n}\right] + src^n. \tag{53}$$

The imaging finite difference equation is

$$P_{i,j}^{n-1} = 2P_{i,j}^{n} - P_{i,j}^{n+1} + \alpha^2 \left[P_{i+1,j}^{n} + P_{i-1,j}^{n} + P_{i,j+1}^{n} + P_{i,j-1}^{n} - 4P_{i,j}^{n}\right] + bc^n. \tag{54}$$

where

$$\alpha^2 = \frac{c(x,z)^2 \delta t^2}{h^2} \tag{55}$$

The three seismic models which were generated using the exploding reflector model are migrated using Equation 55. The *dipping layer*, *reef*, and *salt models* have all been computed using the initial known velocity model and are shown in Figures 22, 25, and 28, respectively. Study the moving wave fields and see if the recorded reflections move into the earth in an understandable way. As you might want to compare the migration process with the modeling process, notice that waves exiting the model sides do not appear in the migration. What does this do to the image reconstruction? Would any other imaging method be more successful in reconstruction of the missing data?

Obtaining velocity data for the model is the next step in the process. Let's begin with a simple marine data case, data which was collected over water. The velocity of water is approximately 1500 meters per second. Only the depth of the water is unknown and can be computed from the seismic data.

However, on land we are in for a terrible surprise. The weathered earth is just that, and the low velocity layer has truly variable properties. Just look across your favorite park and you will see stream beds with sand and gravel, hills with bedrock, fertile fields with soils, all places where the earth is really different. All these differences are seen in the near surface seismic data. Here both the velocity and depth are unknown and will be determined from the seismic data.

After the first layer either on land or at sea we still have mystery to unravel: what is the velocity of the rest of the subsurface? This we shall leave for another day. I will say that using the unstacked data and more powerful mathematics it is possible to do a very credible job of velocity analysis.

12 Seismic Data Display

An X-windows tool Suxwigb is available from the Center for Wave Phenomena at Colorado School of Mines. Seismic Unix, or SU, is a public domain seismic trace processing package of which Suxwigb is a part.

This seismic trace display has the ability to display wiggle traces and in/out zoom. This is a public domain software product and is included with the codes of this chapter. The input file required by Suxwigb is generated by ftn2su and is an unformatted Fortran file. The Ftn2su file is a formatted Fortran file organized trace by trace. These input files are quite large and SU permits Unix piping of process output.

The reader is encouraged to *test seismic display* before proceeding.

13 Introduction to Boundary Conditions

The model configuration in Figure 13 describes a finite bounded area in two dimensions. When a propagated wave arrives in the neighborhood of the boundary, we must consider the effects of the reflected wave. Are reflected waves wanted? Is a reflected wave a physical phenomenon, or is it an artifact? What are the consequences of the finite nature of the model?

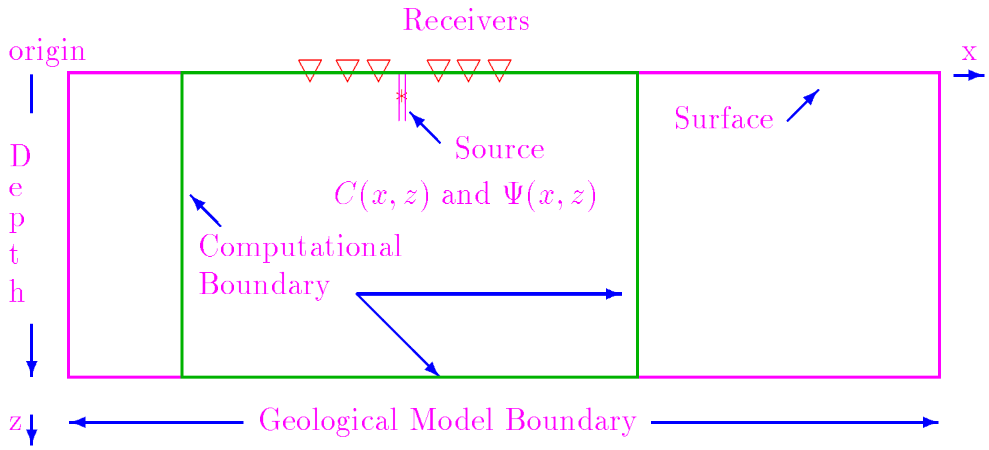

Figure 13: Geological Model

What are the directions of the arriving waves, and is the numerical method sufficiently robust to eliminate the reflected waves? We will assume the model is sufficient in size to include all significant reflection events within the time recording window. The actual geological structure which is being modeled might be quite large compared with the area of the computational model of Figure 13.

The earth surface has a density contrast; the density of air is quite different from that of rock, water, and soil. This upper surface has a high reflectivity coefficient and is correctly modeled as a hard surface boundary. The sides and bottom of a model usually are treated as transmissive boundaries. The terminology used for these boundaries is absorbing, infinite, or transmissive, which all mean essentially the same thing, a non-reflecting boundary.

We shall consider two useful types of boundary condition methods for non-reflecting, and one method for reflecting, boundaries. The simplest of these methods is for the reflecting boundary.

14 Reflecting Boundaries

Assume a flat surface and an incident wave which is propagated by the acoustic wave equation. The finite difference equation, Equation 56, uses a stencil which places a fictitious node or nodes outside the model. These nodes are initially zero and are kept zero for all model time; the superscript n is the time step. This is illustrated in Figure 14 for a flat surface boundary and the second order difference stencil.

$$\Psi^{n+1}(z < 0) = \Psi^{n}(z < 0) \equiv 0 \tag{56}$$

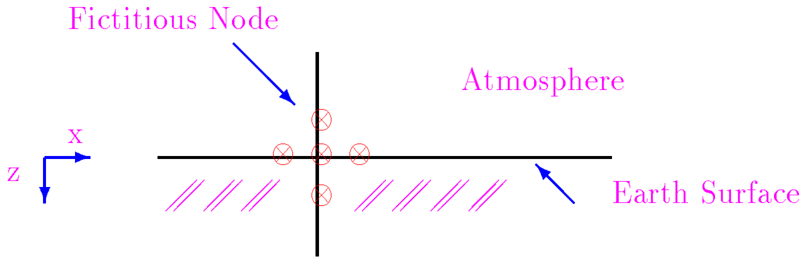

Figure 14: Boundary Node Stencil

15 Non-Reflecting Boundaries

The non-reflecting boundary is much more difficult, and two methods are presented here. The first is based on the one way wave equation as derived by factoring the one dimensional wave equation [18] into two one way wave equations. This assumes the primary wave direction is normal to the boundary. The partial differential operators are factored, and these wave equations propagate in the direction of the sign of Equation 59.

$$\frac{\partial^2\Psi}{\partial x^2} = \frac{1}{C(x,z)^2}\left(\frac{\partial^2\Psi}{\partial t^2}\right). \tag{57}$$

$$\left(\frac{\partial^2}{\partial x^2} - \frac{1}{C(x,z)^2}\frac{\partial^2}{\partial t^2}\right)\Psi = 0. \tag{58}$$

Factoring the operator of the previous equation gives

$$\left(\frac{\partial}{\partial x} + \frac{1}{C(x,z)}\frac{\partial}{\partial t}\right)\left(\frac{\partial}{\partial x} - \frac{1}{C(x,z)}\frac{\partial}{\partial t}\right)\Psi = 0. \tag{59}$$

Now if the product of two terms in Equation 59 is zero, then either term could be zero, and for the left side of the model we have Equation 60:

$$\left(\frac{\partial}{\partial x} - \frac{1}{C(x,z)}\frac{\partial}{\partial t}\right)\Psi = 0. \tag{60}$$

These first order hyperbolic systems are the same as those in Section 3. If the one way wave equation is applied each time step, then any error E_r from the previous time step can be computed using Equation 61. This error can be corrected in the current application of the boundary condition. This is Equation 62, which is for the left side of the model as shown in Figure 15.

$$\left(\frac{\partial}{\partial x} - \frac{1}{C(x,z)}\frac{\partial}{\partial t}\right)\Psi^n_{x=\delta_x} = E_r. \tag{61}$$

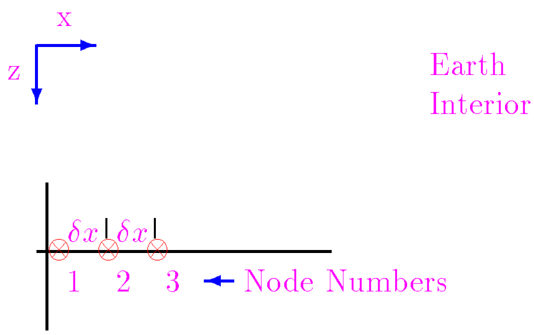

Figure 15: Model Boundary Node Stencil, Left Side

The approximation is improved by incorporation of an error term from the previous time step; the error is computed in the current time step for a node displaced by one grid space inside the model as in Equation 62. The resulting boundary condition is shown in Equations 63, 64, 65, and 66 for all four sides of a rectangular model, left, right, top and bottom respectively.

$$\left(\frac{\partial}{\partial x} - \frac{1}{C(x,z)}\frac{\partial}{\partial t}\right)\Psi^{n+1}_{x=0} - E_r = 0. \tag{62}$$

The left side boundary equation is

$$\left(\frac{\partial}{\partial x} - \frac{1}{C(x,z)}\frac{\partial}{\partial t}\right)\Psi^{n+1}_{x=0} - \left(\frac{\partial}{\partial x} - \frac{1}{C(x,z)}\frac{\partial}{\partial t}\right)\Psi^{n}_{x=\delta x} = 0. \tag{63}$$

The right side boundary equation is

$$\left(\frac{\partial}{\partial x} + \frac{1}{C(x,z)}\frac{\partial}{\partial t}\right)\Psi^{n+1}_{x=xmax} - \left(\frac{\partial}{\partial x} + \frac{1}{C(x,z)}\frac{\partial}{\partial t}\right)\Psi^{n}_{x=xmax-\delta x} = 0. \tag{64}$$

The top side boundary equation is

$$\left(\frac{\partial}{\partial z} - \frac{1}{C(x,z)}\frac{\partial}{\partial t}\right)\Psi^{n+1}_{z=0} - \left(\frac{\partial}{\partial z} - \frac{1}{C(x,z)}\frac{\partial}{\partial t}\right)\Psi^{n}_{z=\delta z} = 0. \tag{65}$$

The bottom side boundary equation is

$$\left(\frac{\partial}{\partial z} + \frac{1}{C(x,z)}\frac{\partial}{\partial t}\right)\Psi^{n+1}_{z=zmax} - \left(\frac{\partial}{\partial z} + \frac{1}{C(x,z)}\frac{\partial}{\partial z}\right)\Psi^{n}_{z=zmax-\delta z} = 0. \tag{66}$$

The finite difference equations shown as Fortran code are Code Fragments I, II, III, and IV, respectively. These difference equations are all second order in space and second order in time. The difference approximations for the boundary conditions must match or nearly match those of the wave equation [17]. If the degree of approximation for the wave equation is k, the boundary condition approximation must be k or $k-1$.

The left side boundary condition loop is

```
        do 140 j=2,nzd-1
        u3(1,j) = u2(1,j)+u2(2,j)-u1(2,j)+
    x  cb(1,j)*(u2(2,j)-u2(1,j)-u1(3,j)+u1(2,j))
 140    continue
```

Code Fragment I

The $u3, u2, u1$ wave field arrays are for the $n+1$, n, $n-1$ time steps respectively. The cb array contains the spatial coefficient, $c(x,z)\delta t/h$.

The right side boundary condition loop is

```
        do 160 j=2,nzd-1
        u3(nxd,j) = u2(nxd,j)+u2(nxd-1,j)-u1(nxd-1,j)-
    x   cb(nxd,j)*(u2(nxd,j)-u2(nxd-1,j)-
    x   u1(nxd-1,j)+u1(nxd-2,j))
 160    continue
```

Code Fragment II

The top side boundary condition loop is

```
        do 150 i=2,nxd-1
        u3(i,nzd) = u2(i,nzd)+u2(i,nzd-1)-u1(i,nzd-1)-
    x  cb(i,nzd)*(u2(i,nzd)-u2(i,nzd-1)-
    x     u1(i,nzd-1)+u1(i,nzd-2))
150     continue
```

Code Fragment III

The bottom side boundary condition loop is

```
        do 130 i=2,nxd-1
        u3(i,nzd) = u2(i,nzd)+u2(i,nzd-1)-u1(i,nzd-1)-
    x  cb(i,nzd)*(u2(i,nzd)-u2(i,nzd-1)-
    x     u1(i,nzd-1)+u1(i,nzd-2))
130     continue
```

Code Fragment IV

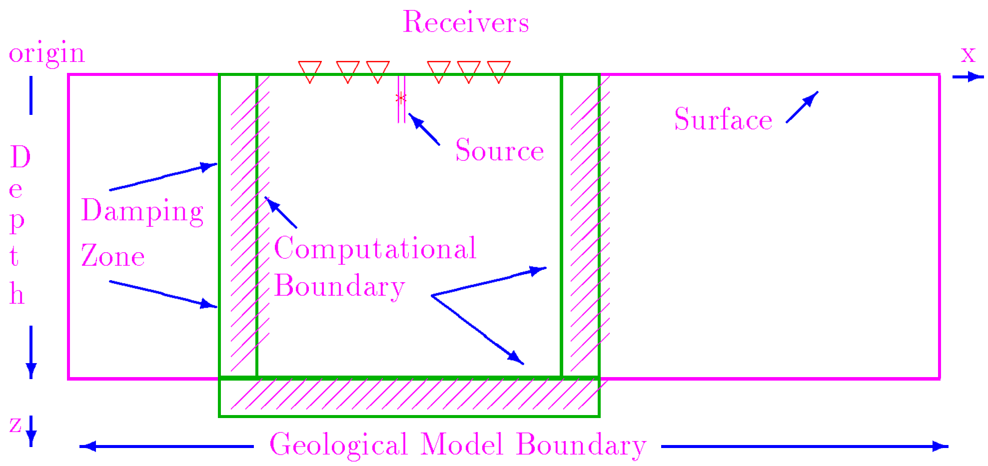

Figure 16: Computational Model Damping Zone

The difficulty the one way wave equation has for waves which arrive at angles other than normal could be fixed by making an angle dependent method. The wave field is used to determine an incoming wave direction, and the wave field is decomposed into two parts, normal and tangential. If more than one wave is active in the boundary region, it makes the direction and decomposition quite complex. A considerable amount of work has been done in developing boundary conditions and angle dependent boundary conditions; students are not encouraged to plow this old ground. The damping method, presented next, is suitable for most modeling methods and overcomes the angle dependent difficulty in an elegant way; "it does not care where the wave comes from". Further, it is independent of the degree of approximation for the wave equation.

16 Damping Zone Method

The alternative to wave equation methods is a numerical damping zone, as in Figure 16, which reduces the wave strength over a grid region near the boundary. The idea is to slowly and smoothly apply a weight which will be effective in the aggregate. The product $W = \Pi w(x_w)$, $W \leq \epsilon$, is computed to a small factor ϵ with the property of certain smoothness at both edges. The smoothness is required to reduce the unwanted reflections.

The weight for the left side of the computational model, $0 \geq x_w \geq x_b$, is computed where x_b is extended into the model for a set of grid points. The number of grid points is typically 20, and the values of Table 1 are used for the weight function where $W = 0.04$.

The weights presented by Cerjan [4] are $w(x_w) = \exp^{-[0.015(20-i)]^2}$, where i is the node number for the left side and corresponds to the x_w. The grid index i could be different for the other sides. The weights are applied to all the nodes in the region bounded by x_b to the current time step data, and to the previous time step data after the interior finite differences

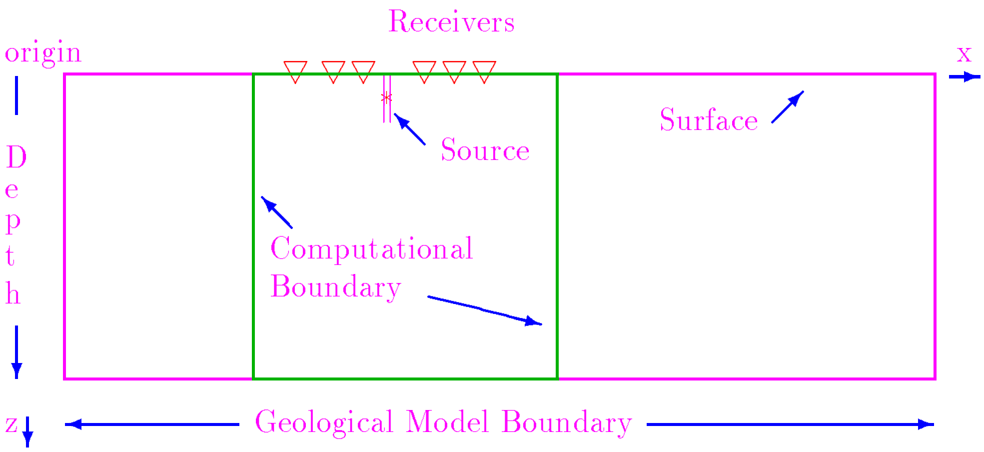

Figure 17: Geological Model

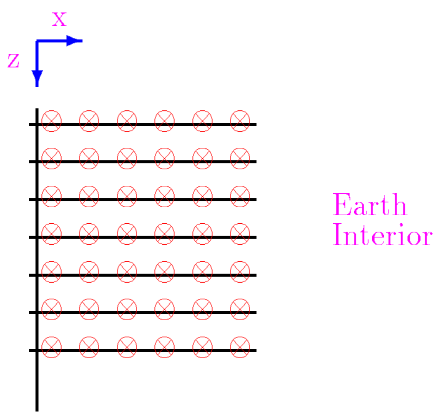

Figure 18: Model Boundary Damping Nodes, Left Side

Table 1: Boundary Weights

Grid Point	Weight
1	0.00445
2	0.00775
3	0.01310
4	0.02149
5	0.03422
6	0.05287
7	0.07926
8	0.11533
9	0.16284
10	0.22313
11	0.29671
12	0.38289
13	0.47951
14	0.58275
15	0.68729
16	0.78663
17	0.87372
18	0.94176
19	0.98511
20	1.00000

are computed.

$$\hat{\Psi}^{n+1}(x_w) = \Psi^{n+1}(x_w)w(x_w) \tag{67}$$

$$\hat{\Psi}^{n}(x_w) = \Psi^{n}(x_w)w(x_w) \tag{68}$$

17 Seismic Model Codes and Results

17.1 VELMOD Program

The *velmod* program generates a velocity data file for the *seismic* and *revtime* programs. A sinusoidal structure is placed over a flat layer. The primary feature of this artificial model is the varying dip structure as shown in Figure 19. The velocities of this model have maximum

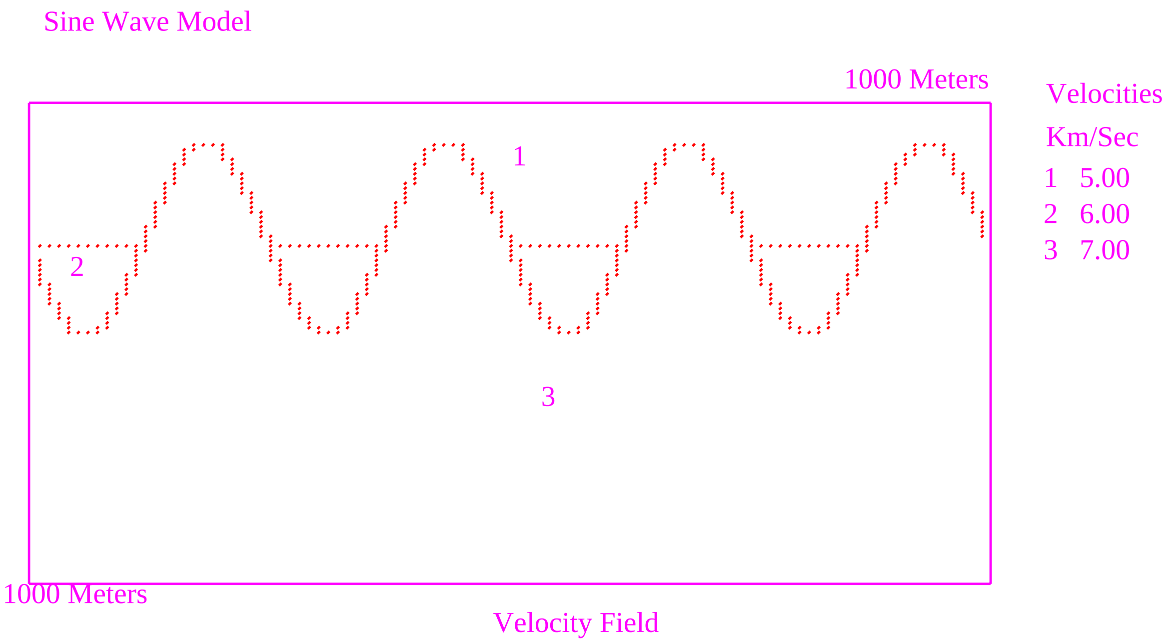

Figure 19: Sine Wave Model

to minimum ratio of 3. A portion of the *velmod.f* program listing is shown in PROGRAM CODE I. The number of cycles in the model is set by the variable Cycle.

```
c
c              r p bording
c
c          parameter (nxd=500,nzd=300)
           parameter (nxd=101,nzd=101)
c          parameter (nxd=1024,nzd=1024)
c          parameter (nxd=50,nzd=50)
c
           dimension  src(nxd,nzd)
           dimension  vel(nxd,nzd)
c
c
           do 10 i=1,nxd
            do 20 j=1,nzd
            vel(i,j) = 1500.0
            src(i,j) = 0.0
 20        continue
c
c       add sine wave
c
```

```
          xx = i
          xt = nxd
          xr = xx/xt
          n90 =  70
          jzz = (nzd-n90)+20*sin(4.*6.28*xr)
          do 15 j=jzz,nzd
          vel(i,j) = 3000.0
 15       continue
          do 16 j=jzz+40,nzd
          vel(i,j) = 1500.0
 16       continue
 10       continue
          do 18 i=nxd/4,nxd-nxd/4
c
c       add sine wave
c
          xx = i
          xx = xx * 1.0
          xt = nxd
          xr = xx/xt
c
          jzz = (nzd-20)-10*cos(6.28*xr-3.14*0.5)
 18       continue
c

          vmin = vel(1,1)
          vmax = vel(1,1)
           do 41 i=1,nxd
           do 40 j=1,nzd
           if( j .ne. nzd ) then
               if(abs( vel(i,j) - vel(i,j+1)) .gt. 5.0 ) then
               src(i,j) = 1.0
               endif
           endif
           if( vel(i,j) .gt. vmax ) vmax = vel(i,j)
           if( vel(i,j) .lt. vmin ) vmin = vel(i,j)
 40        continue
 41        continue
c
          open(12,file='mvel.dat',form='unformatted')
          open(10,file='velocity.dat',form='unformatted')
          do  30 i=1,nzd
```

```
           write(12) (vel(j,i),j=1,nxd)
           write(10) (vel(j,i),j=1,nxd)
 30       continue
          close(10)
          write(6,210) vmin,vmax
 210      format( ' min, max velocity ',2f12.4)
          close(12)

          open(10,file='reflector.dat',form='unformatted')
          do  60 i=1,nzd
           write(10) (src(j,i),j=1,nxd)
 60       continue
          close(10)
          open(10,file='reflector.su',form='formatted')
          la = 1
          write(10,200) nzd,nxd,la
 200      format(3i4)
          do i=1,nxd
           do j=1,nzd
            write(10,201) src(i,j)
 201        format(e13.6)
           enddo
          enddo
          close(10)
          end
```

PROGRAM CODE I, Sine Wave Model

17.2 Dipping Model

17.2.1 DIP Program

The *dip* program generates a velocity data file for the *seismic* and *revtime* programs. This is the simplest model provided and illustrates the reverse time method. The layers are nearly flat, but some slight dip is included. This geology is similar to regions of West Texas. The velocities of this model have maximum to minimum ratio of 2. A portion of the *dip.f* program listing is shown in PROGRAM CODE II.

```
c
c        dipping layer model building program
c
c
          parameter(nx=800,nz=800)
```

```
c
          dimension vv(nx,nz)
          dimension vz(20)
c
          dimension  am(20),bm(20)
c
c         the velocity at depth is determined by line segments
c
          nline = 8
c
          bm(1) = 500.0
          bm(2) = 2000.0
          bm(3) = 3000.0
          bm(4) = 6000.0
          bm(5) = 8000.0
          bm(6) = 11000.0
          bm(7) = 14000.0
          bm(8) = 16000.0
c
c
          am(1) = -0.05
          am(2) = -0.07
          am(3) = -0.05
          am(4) = -0.05
          am(5) = +0.02
          am(6) = -0.03
          am(7) = -0.06
          am(8) = -0.00
c
          vz(1) = 2000.0
          vz(2) = 2400.0
          vz(3) = 3200.0
          vz(4) = 3000.0
          vz(5) = 3500.0
          vz(6) = 3200.0
          vz(7) = 3600.0
          vz(8) = 3800.0
c
          dx  = 25.0
          dz  = 25.0
c
c         create initial water layer
```

```
c
           do i=1,nx
            do j=1,nz
            vv(i,j) = 1500.0
            enddo
           enddo
c
c          now layer by layer make velocities under each
c             line
c
c          each new layer over writes that below
c
           do k=1,nline
            do i=1,nx
             do j=1,nz
             x = (i-1)*dx
             z = (j-1)*dz
             zl = am(k)*x + bm(k)
c
             if( z .gt. zl ) then
               vv(i,j) = vz(k)
             endif
            enddo
           enddo
          enddo
c
c           write model to file
c
             open(16,file='dip.dat',form='unformatted')
             do j=1,nz
             write(16) (vv(i,j),i=1,nx)
             enddo
             close(16)
             end
```

PROGRAM CODE II, The Dipping Model

17.2.2 Dipping Layer Model Results

The dipping layer model velocity field and exploding reflector source locations are shown in Figures 20 and 21, respectively. The computed model results are shown in Figures 22 and 23.

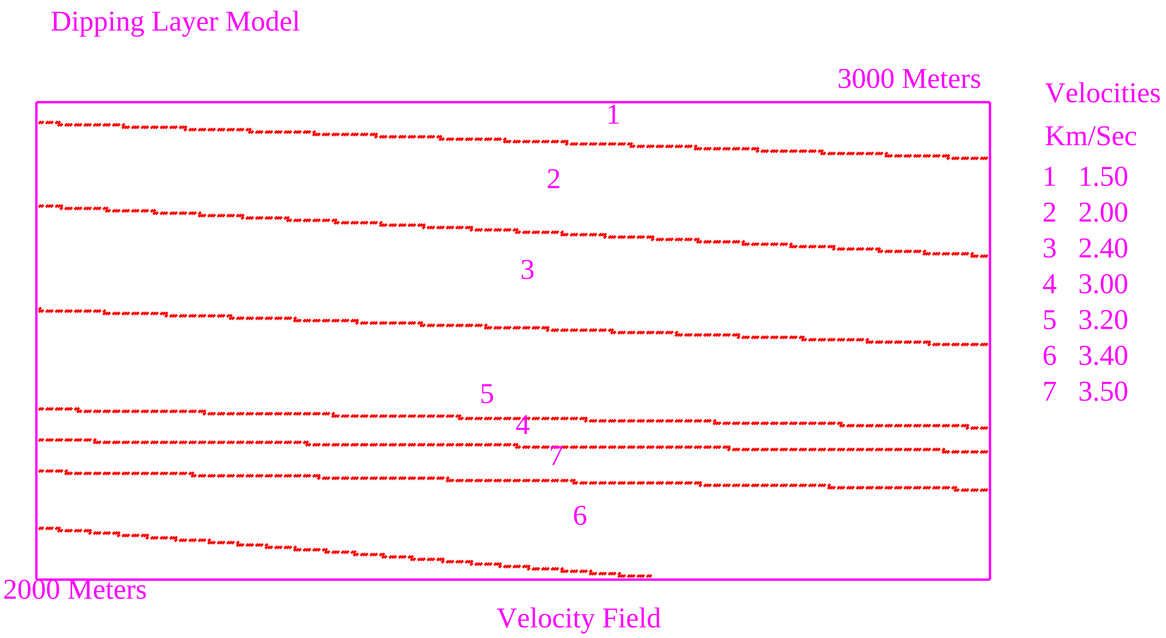

Figure 20: Dipping Layer Model, Velocity Field. See Section 17.3.3 for details of how the figure was created.

The process by which the model was generated and the figures created is described in Section 17.3.3.

17.3 Reef Model

17.3.1 REEF Program

The *reef* program generates a velocity data file for the *seismic* and *revtime* programs. Reef structures are characterized by reasonably flat layers which change abruptly laterally. The reef section is usually narrow in one dimension and long in the other dimension. Reefs like this model can be found in the Alberta province of Canada. The reflections change across the reef in amplitude, and reflections from layers below show time push down or pull ups depending on the lateral velocity differences. If the velocity in the reef is higher than in adjacent sections, then the times from the reflections from the bottom of this layer will pull up over the reef section. The velocities of this model have maximum to minimum ratio of 3. A portion of the *reef.f* program listing is shown in PROGRAM CODE III.

```
c
c reef model building program
c
............. code deleted here .....................
c          the velocity at depth is determined by line segments
c
```

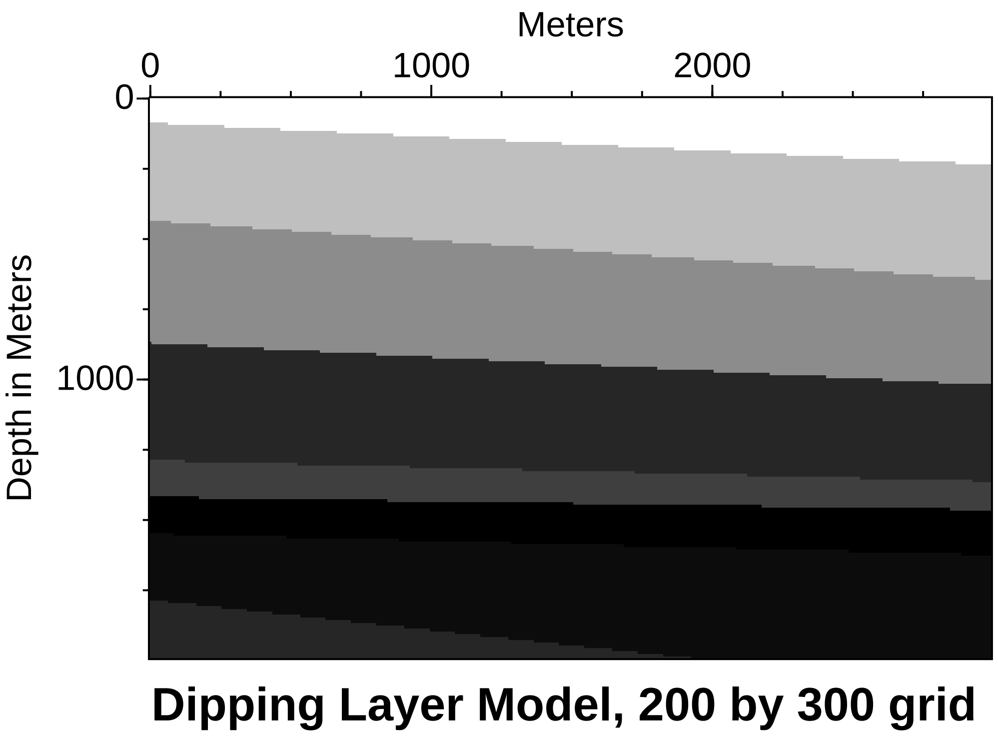

Figure 21: Dipping Layer Model, Exploding Reflector Source Locations. See Section 17.3.3 for details of how the figure was created.

```
          nline = 6
c
          bm(1) =  150.0
          bm(2) =  400.0
          bm(3) =  700.0
          bm(4) =  900.0
          bm(5) =  1300.0
          bm(6) =  1600.0
c
          am(1) = +0.02
          am(2) = -0.03
          am(3) = -0.02
          am(4) = -0.02
          am(5) = -0.03
          am(6) = -0.00
c
```

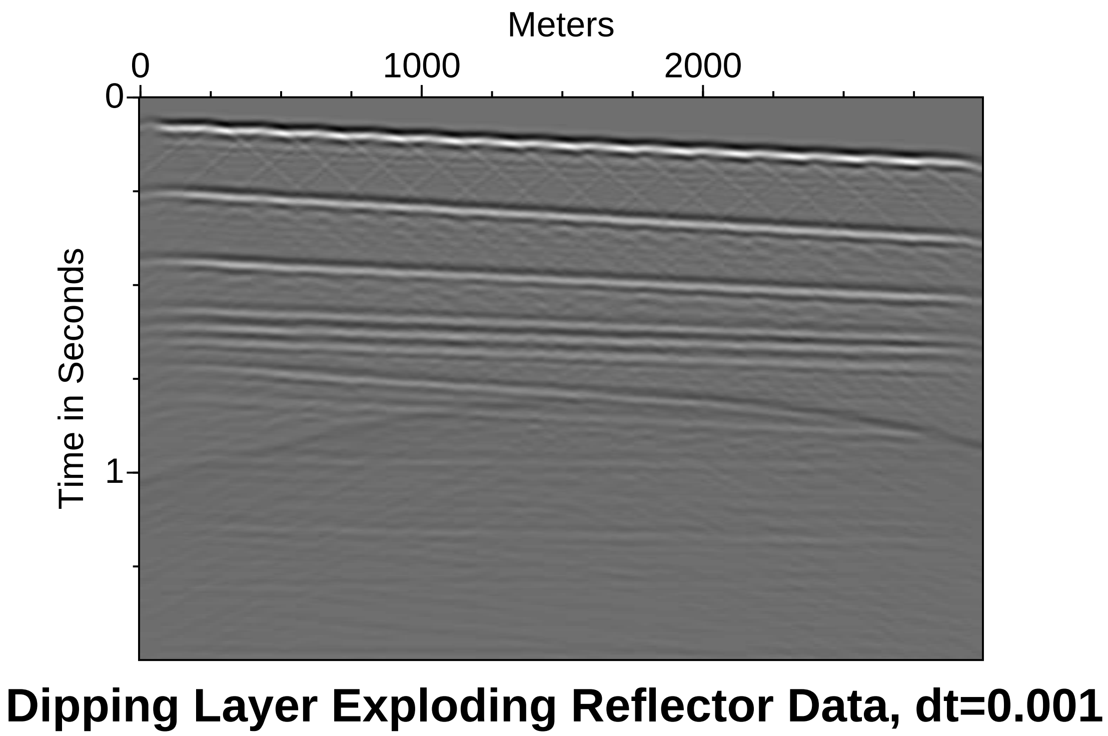

Figure 22: Forward Modeling of Dipping Layers. See Section 17.3.3 for details of how the figure was created.

```
          vz(1) = 2000.0
          vz(2) = 2600.0
          vz(3) = 3200.0
          vz(4) = 2000.0
          vz(5) = 3600.0
          vz(6) = 3900.0
c
          dx  = 5.0
          dz  = 5.0
c
c         reef circle
c
          l_reef = 3
          xc = (nx-1)*dx*0.5
          zc = 800.0
          rc = 320.0
```

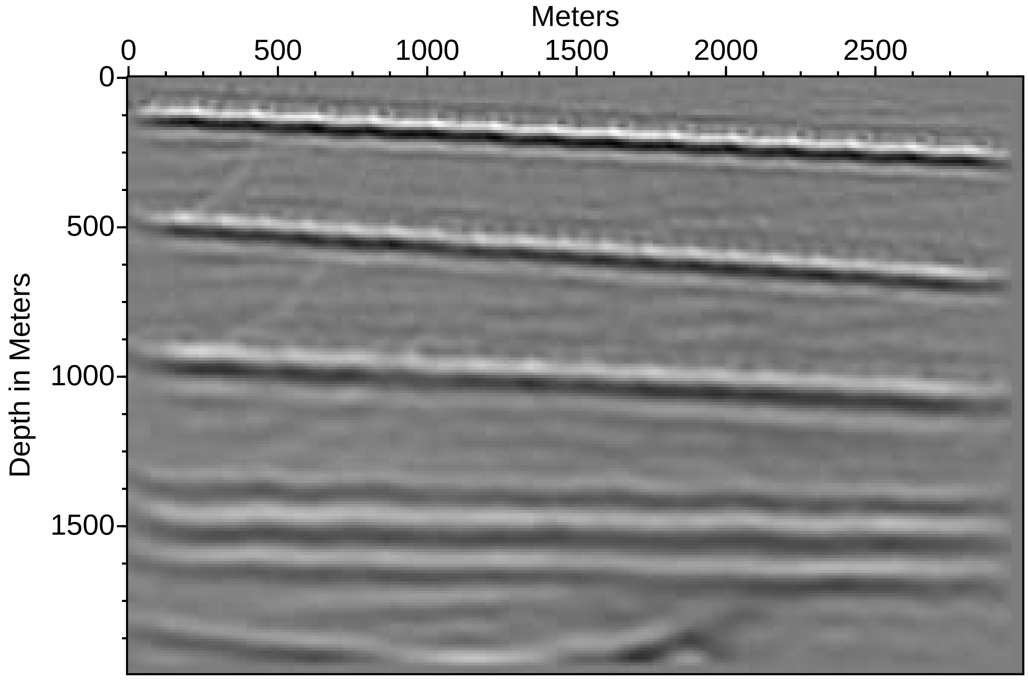

Figure 23: Reverse Time Imaging of Dipping Layers. See Section 17.3.3 for details of how the figure was created.

```
           salt = 4500.0
c
c          create initial water layer
c
           do i=1,nx
            do j=1,nz
            src(i,j) = 0.0
            vv(i,j) = 1500.0
            enddo
           enddo
c
c         now place a circle for the reef
c
           ilmax = nx
           irmax = 1
c
           do i=1,nx
```

```
          do j=1,nz
          x = (i-1)*dx
          z = (j-1)*dz
          r = sqrt((x-xc)**2+(z-zc)**2)
c
c         if point is inside circle it's the reef
c
         if( r .lt.  rc ) then
             if( i .lt. ilmax ) ilmax = i
             if( i .gt. irmax ) irmax = i
         endif
         enddo
        enddo
c
c        now layer by layer make velocities under each
c           line
c
c        each new layer over writes that below
c
         do k=1,nline
          do i=1,nx
           do j=1,nz
           x = (i-1)*dx
           z = (j-1)*dz
           zl = am(k)*x + bm(k)
c
           if( z .gt. zl ) then
            vmult = 1.0
            if(l_reef .eq. k) then
             if( i .gt. ilmax .and. i .lt. irmax ) vmult = 1.4
             if( i .gt. irmax ) vmult = 1.2
            endif
             vv(i,j) = vz(k)*vmult
           endif
          enddo
         enddo
        enddo
c
.............. code deleted here ..................
```

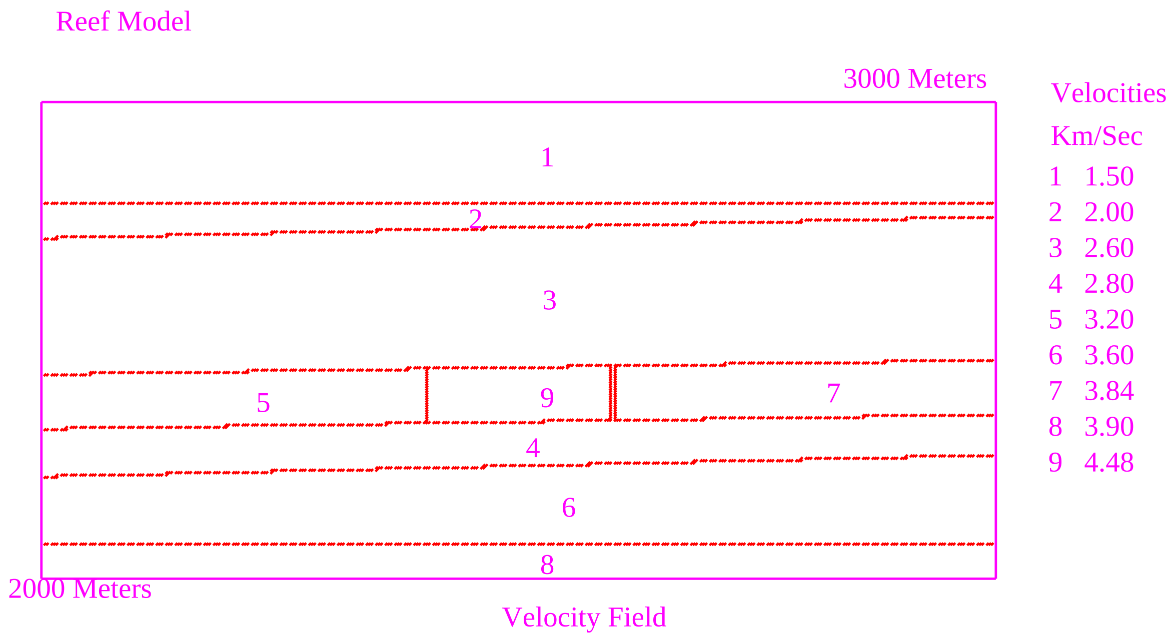

Figure 24: Reef Model, Velocity Field. See Section 17.3.3 for details of how the figure was created.

17.3.2 Reef Model Results

The reef model velocity field and exploding reflector source locations are shown in Figures 24 and 25, respectively. The computed model results are shown in Figures 26 and 27.

The process by which the model was generated and the figures created is described in the next section.

17.3.3 Model and Figure Generation

The Reef velocity model can be generated using the *reef.f* program and the *model.dat* file. The user must specify the grid size, grid spacing, the length of time for the model run and the time step. A model name is also required.

The output can be viewed with a postscript viewer such as *ghostview*. A Unix shell program *reef_reef.sh* is provided, and it generates a *reef_reef.ps* file. The first process of the shell is to convert the unformatted output file from *reef.f* into an SU format file using *suaddhead*. The shell uses SU program *supsimage* to convert the SU file to postscript.

The seismic image file with the time data is generated using the *seismic.f* program. The output file is processed by *reef_seismic.sh*, which is a shell program that is similar to the *reef_reef.sh* shell program. It converts the program output into an SU file and then into a .ps file.

The reverse time imaging which converts the time data back into depth data is generated using the *revtime.f* program. The output file is processed by the *reef_revtime.sh* shell similarly to *reef_reef.sh*. The depth image data file is converted from a Fortran unformatted

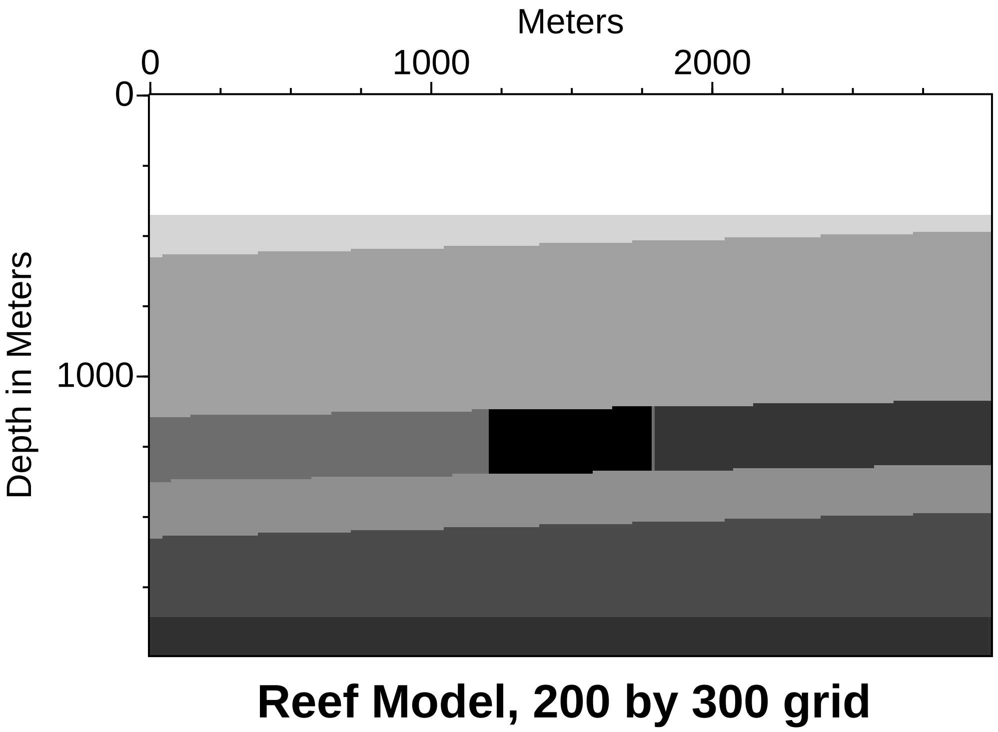

Figure 25: Reef Model, Exploding Reflector Source Locations. See Section 17.3.3 for details of how the figure was created.

file to an SU file and then to a .ps file.

The Salt Intrusion model and the Dipping layer models have Unix scripts and programs identical to those for the Reef model. The names of the files used for all three models are shown in Table 2.

A comment about the shell programs described above: they all require parameters to create output data images. Any changes to *model.dat* will probably require changes in the shell programs.

17.4 Salt Model

17.4.1 SALT Program

The *salt* program generates a velocity data file for the *seismic* and *revtime* programs. Some of the truly interesting geological structures are those made by salt and are typically found in the Gulf of Mexico. *salt* computes a salt body model; the principal feature is the high velocity contrast between the salt and the surrounding geological media. This particular salt model is an intrusion, a slowly moving feature pushing up and aside the inplace media; these

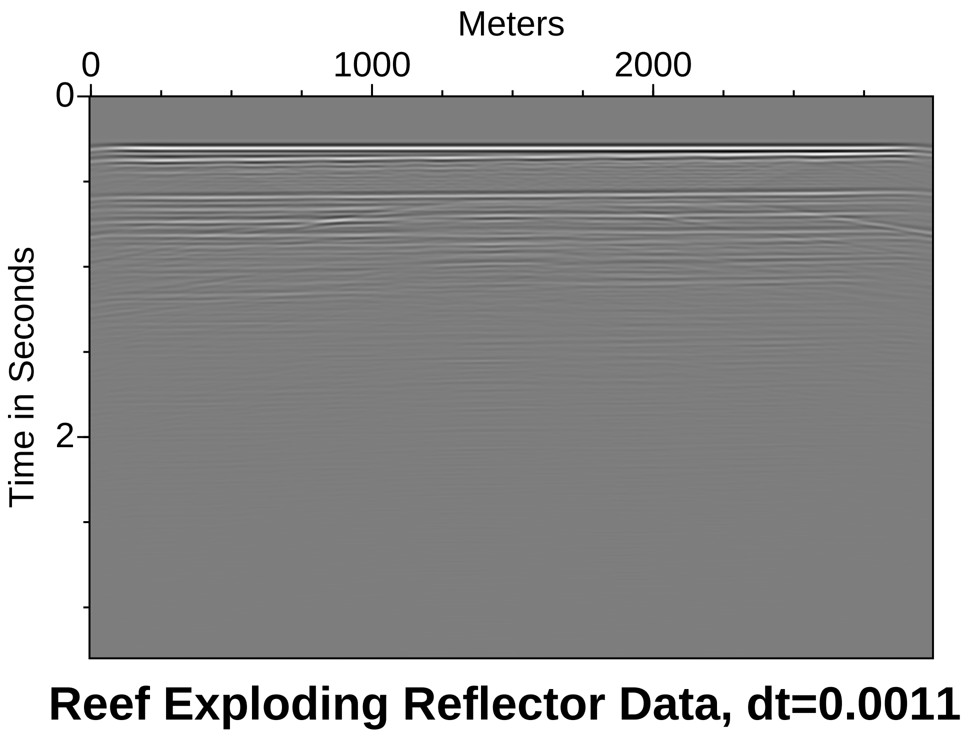

Figure 26: Forward Modeling of Reef. See Section 17.3.3 for details of how the figure was created.

Table 2: Fortran Codes and Shell Scripts Used in Seismic Modeling and Imaging of Data

Program Organization			
	Model Building	Seismic Modeling	Reverse Time Imaging
Reef	reef.f	seismic.f	revtime.f
Salt	saltv.f	seismic.f	revtime.f
Dip	dip.f	seismic.f	revtime.f
Reef	reef_reef.sh	reef_seismic.sh	reef_revtime.sh
Salt	salt_salt.sh	salt_seismic.sh	salt_revtime.sh
Dip	dip_dip.sh	dip_seismic.sh	dip_revtime.sh

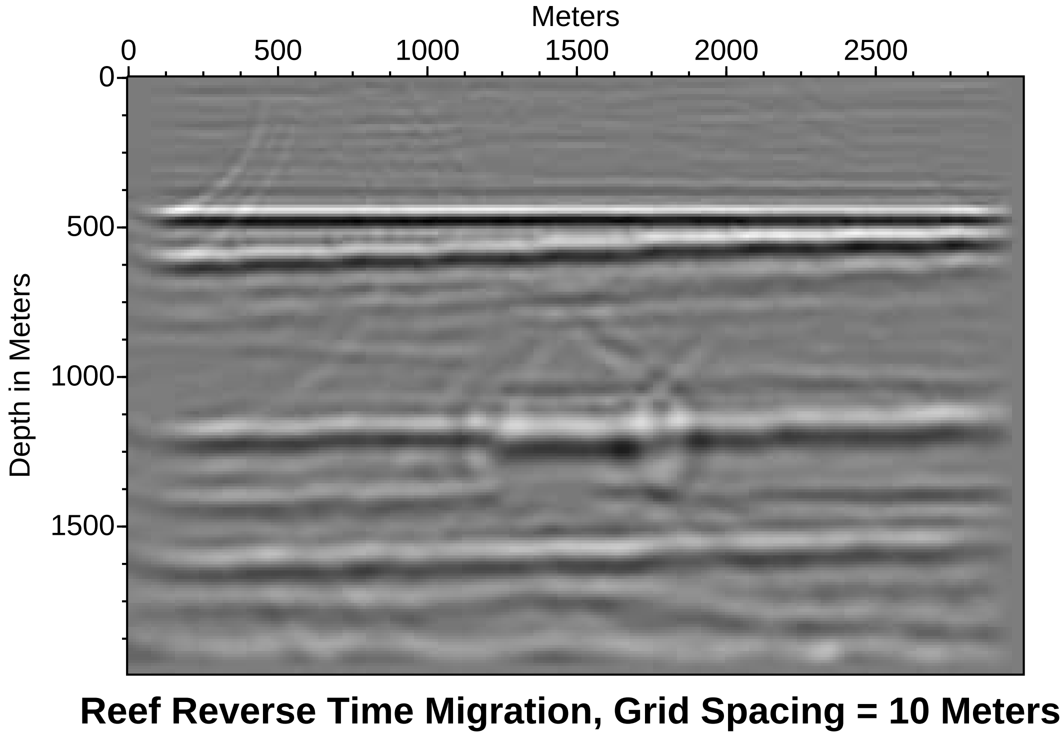

Figure 27: Reverse Time Imaging of Reef. See Section 17.3.3 for details of how the figure was created.

geological changes take millions of years to form. The regions near the salt which are lifted or tilted make excellent reservoirs. The velocities of this model have maximum to minimum ratio of 4. A portion of the *saltv.f* program listing is shown in PROGRAM CODE IV.

Subsalt exploration is receiving considerable attention, and the imaging of subsalt geological structures is becoming a more routine process. Two difficulties of subsalt imaging are the roughness of the top of the salt layer and the small reflection coefficients. The roughness scatters the incoming wave field, and the small reflection coefficient keeps energy from reflecting. Both of these mechanisms reduce the amount of available energy for reflections.

```
c
c        salt model building program
c
....................... code deleted here .......................
c
c          the velocity at depth is determined by line segments
c
           nline = 5
```

```
c
          bm(1) = 500.0
          bm(2) = 700.0
          bm(3) = 9000.0
          bm(4) = 1000.0
          bm(5) = 1600.0
c
          am(1) = 0.0
          am(2) = -0.17
          am(3) = -0.15
          am(4) = -0.26
          am(5) = -0.05
c
          vz(1) = 2000.0
          vz(2) = 3000.0
          vz(3) = 3200.0
          vz(4) = 3500.0
          vz(5) = 3700.0
c
          dx = 20.0
          dz = 20.0
c
c         salt circle
c
          xc(1) = (nx-1)*dx*0.5
          zc(1) = 1000.0
          rc(1) = 300.0
c
          xc(2) = (nx-1)*dx*0.5-12*dx
          zc(2) = 1100.0
          rc(2) = 300.0
c
          xc(3) = (nx-1)*dx*0.5-12*dx
          zc(3) = 1200.0
          rc(3) = 400.0
c
          xc(4) = (nx-1)*dx*0.5-12*dx
          zc(4) = 1300.0
          rc(4) = 400.0
c
          xc(5) = (nx-1)*dx*0.5-12*dx
          zc(5) = 1400.0
```

```
          rc(5) = 300.0
c
          xc(6) = (nx-1)*dx*0.5-12*dx
          zc(6) = 1500.0
          rc(6) = 200.0
c
          xc(7) = (nx-1)*dx*0.5-12*dx
          zc(7) = 1550.0
          rc(7) = 200.0
c
          xc(8) = (nx-1)*dx*0.5-12*dx
          zc(8) = 1600.0
          rc(8) = 200.0
c
          xc(9) = (nx-1)*dx*0.5-25*dx
          zc(9) = 1750.0
          rc(9) = 200.0
c
          salt = 4500.0
c
c         create initial water layer
c
          do i=1,nx
           do j=1,nz
           vv(i,j) = 1500.0
           src(i,j) = 0.0
           enddo
          enddo
c
c         now layer by layer make velocities under each
c            line
c
c         each new layer over writes that below
c
          do k=1,nline
           sign = 1.0
           do i=1,nx
            if(i .gt. nx/2 ) sign = -1.0
            do j=1,nz
            x = (i-1)*dx
            z = (j-1)*dz
            zl = sign*am(k)*x + bm(k)
```

```
c
            if( z .gt. zl ) then
              vv(i,j) = vz(k)
            endif
           enddo
          enddo
         enddo
c
c        now place a circle for the top of salt
c

          ilmax = nx
          irmax = 1
c
         do ic=1,9
          do i=1,nx
           do j=1,nz
           x = (i-1)*dx
           z = (j-1)*dz
           r = sqrt((x-xc(ic))**2+(z-zc(ic))**2)
c
c          if point is inside circle it's salt
c
          if( r .lt.  rc(ic) ) then
              vv(i,j) = salt
              if( i .lt. ilmax ) ilmax = i
              if( i .gt. irmax ) irmax = i
          endif
          enddo
         enddo
        enddo

c
c        now drop a line from the circle center
c          to the bottom of the model
c
c          and create the flanks of the salt
c
            ixc = xc(1)/dx
            izc = zc(1)/dz
            izc2 = 0.7*(nz-izc)+izc
c
```

```
            ileft = ilmax+60
            iright= irmax-65
            iil = +1
            iir = -1
            ilmin = ixc - float(ixc - ileft)*2.6
            irmin = ixc + float(iright - ixc)*2.4

            write(6,*) ixc,izc,izc2
            write(6,*) ileft,iright,iil,iir,ilmin,irmin

            mml = 2
            do j=izc,nz-40
             write(6,*) ileft,iright
             ilf = min(ileft,iright)
             irt = max(ileft,iright)
             do i=ilf,irt
c             vv(i,j) = salt
              enddo
              ileft = ileft + iil*mml
              iright= iright+ iir*mml
              if( ileft .gt. ilmin) ileft = ilmin
              if( iright .lt. irmin) iright = irmin
              if( j .gt. izc2 ) then
                mml = 5
              endif
            enddo
................... code deleted here ....................
```

PROGRAM CODE IV, The Salt Model

17.4.2 Salt Model Results

The salt model velocity field and exploding reflector source locations are shown in Figures 28 and 29, respectively. The computed model results are shown in Figures 30 and 31.

The process by which the model was generated and the figures created is described in Section 17.3.3.

18 Exercises

Exercise 1 *Fourier Transform of the source function.*

The source is a function of time. Compute the Fourier Transform of the source function and plot the frequency content.

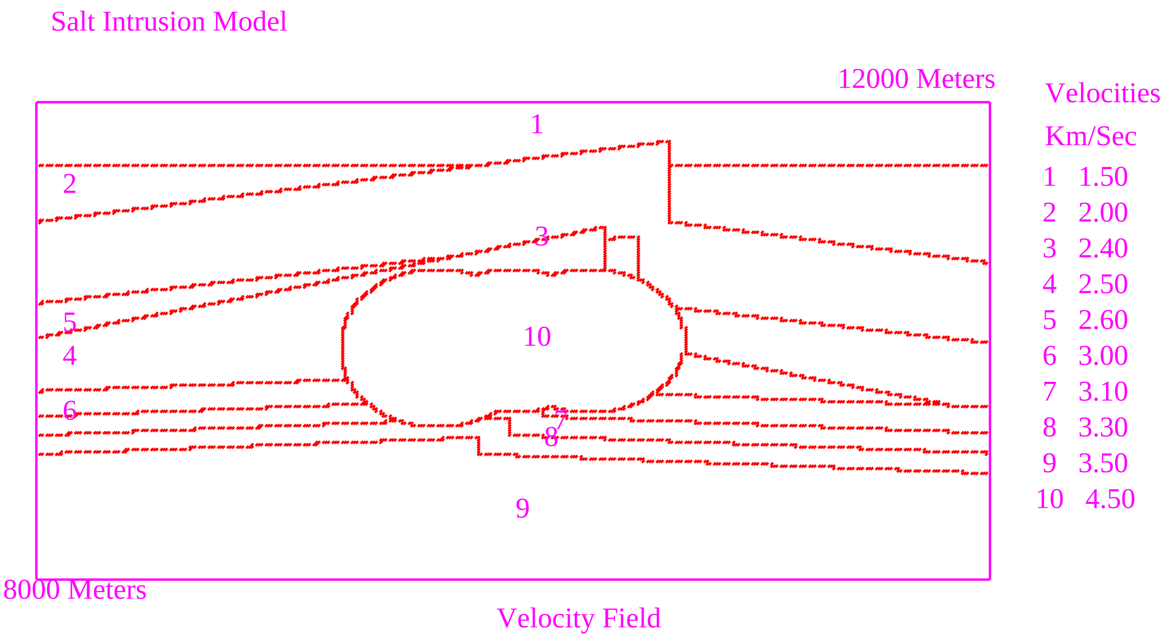

Figure 28: Salt Model, Velocity Field. See Section 17.3.3 for details of how the figure was created.

Exercise 2 *Modification of codes to use damping weights and sponges.*

The damping zone and weights are presented in the text. Convert the SEISMIC.f and REVTIME.f programs to use damping weights. Compare the plotted results of the two methods. Compute the additional computational effort for using sponges. Can anything be done to reduce this work?

Exercise 3 *Use of the DIP model with a generated velocity field.*

Modify the DIP.f program to generate a $C(z)$ model and use this velocity field to image a DIP model. How does the $C(z)$ result compare to the image generated using the exact model velocity field. $C(z) = V_0 + V_{gradient} * Z$.

Exercise 4 *Taylor Series weights for a fourth order finite difference operator.*

Using the matrix method, compute the Taylor Series weights for a fourth order finite difference operator. If possible, use a symbolic method like MATLAB.

Exercise 5 *Stability condition for the second order finite difference method in two and three dimensions.*

Mathematically derive the stability condition for the second order finite difference method in two and three dimensions. Reproduce the analysis for a fourth order method from Problem 4.

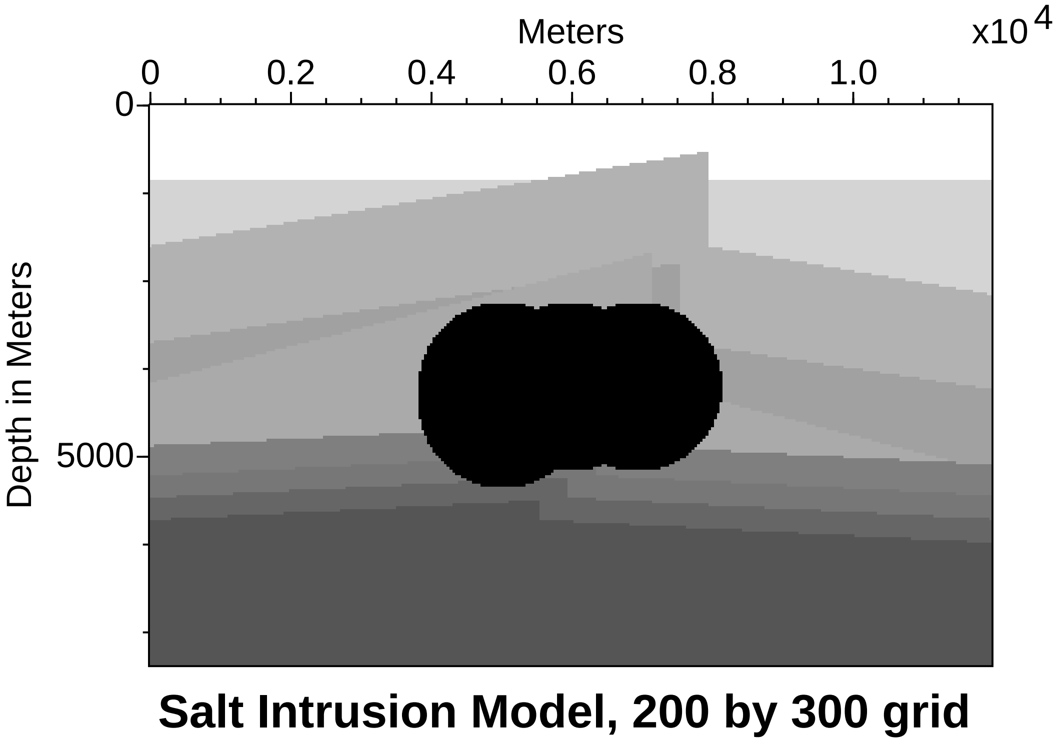

Figure 29: Salt Model, Exploding Reflector Source Locations. See Section 17.3.3 for details of how the figure was created.

Exercise 6 *REVTIME.f with higher order finite difference operator.*

Using the Sponge version of the program, add the higher order finite difference operator to REVTIME.f. Consider changing the boundary condition data so the grid spacing can be made larger. What happens to the complexity if the grid spacing is larger and the operator is longer?

Exercise 7 *Creating a new model.*

Modify one of the model building programs to produce a new model. Create two modeling teams; each is to make a model and generate seismic data. Exchanging data, each team should attempt to image the other's data without knowledge of the exact model.

Exercise 8 *Violation of the the stability condition.*

Attempt to run SEISMIC.f and violate the stability condition. What happens?

Exercise 9 *Reflection coefficients for a model.*

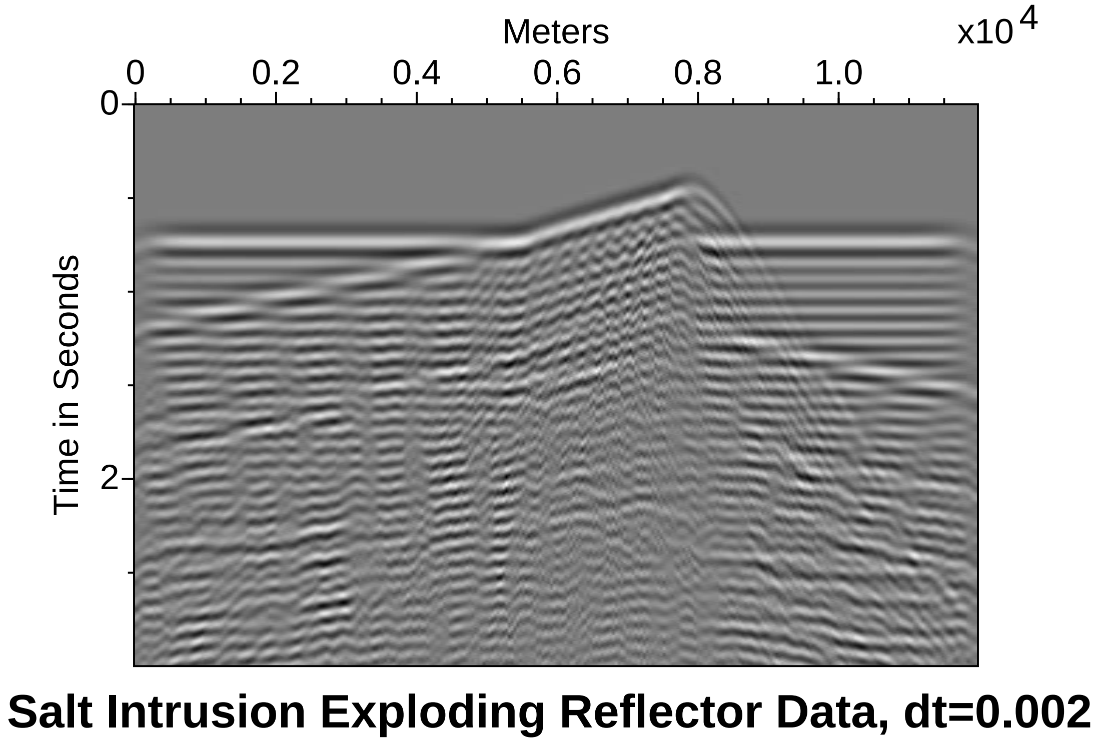

Figure 30: Forward Modeling of Salt. See Section 17.3.3 for details of how the figure was created.

Compute the reflection coefficients for a model, and adjust the REVTIME.f source weight function to create sources which match the reflection coefficients. Why would this give a better image?

Exercise 10 *Construction of ray paths.*

Geometrically construct ray paths from two sources to two receivers. Case A) use two flat layers; Case B) use two dipping layers. Use Snell's law to bend the rays. The P-Wave velocity of the two layers must be different.

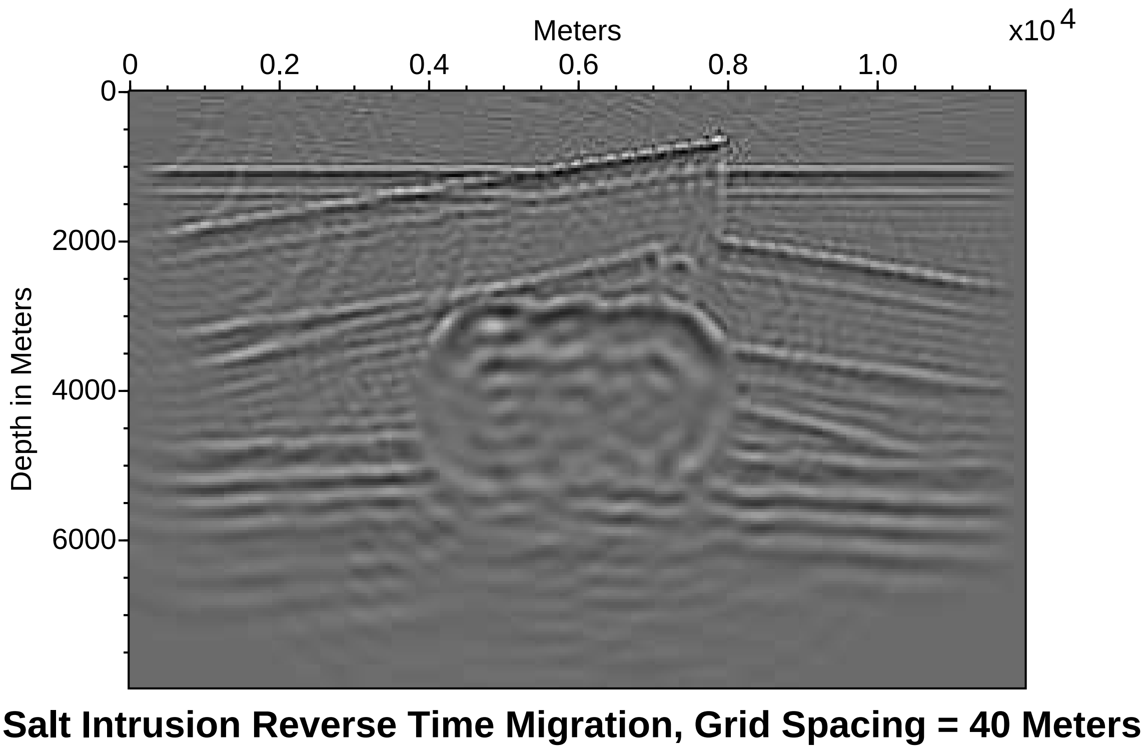

Figure 31: Reverse Time Imaging of Salt. See Section 17.3.3 for details of how the figure was created.

References

[1] Aki, K., and Richards, P. G., 1980, *Quantitative Seismology*, Volume I and II, W. H. Freeman and Co., New York.

[2] Alford, R. M., Kelly, K. R., and Boore, D. M., 1974, *Accuracy of Finite-difference Modeling of the Acoustic Wave Equation*, Geophysics, **39**, 834–842.

[3] Artley, J., 1965, *Fields and Configurations*, Holt, Rinehart, and Winston, Inc., New York.

[4] Cerjan, C., Kosloff, D., Kosloff, R., and Reshef, M., 1985, *A Nonreflecting Boundary Condition for Discrete Acoustic and Elastic Wave Equations*, Geophysics, **50**, 705–708.

[5] Chew, W. C., 1990, *Waves and Fields in Inhomogeneous Media*, Van Nostrand Reinhold, New York.

[6] Chung, T. J., 1978, *Finite Element Analysis in Fluid Dynamics*, McGraw–Hill, New York.

[7] Claerbout, J. F., 1976, *Fundamentals of Geophysical Data Processing*, McGraw–Hill, New York.

[8] Claerbout, J. F., 1985, *Imaging the Earth's Interior*, Blackwell Scientific Publications, Oxford, England.

[9] Clayton, R. and Enquist, B., 1977, *Absorbing Boundary Conditions for Acoustic and Elastic Wave Equations*, Bull. Seis. Soc. Am., **67:6**, 1529–1540.

[10] Dablain, M. A., 1986, *The Application of High-Order Differencing to the Scalar Wave Equation*, Geophysics, **51**, 54–66.

[11] Hunter, S. C., 1976, *Mechanics of Continuous Media*, Ellis Horwood, Sussex, England.

[12] Kelly, K. R., Ward, R. W., Treitel Sven, and Alford, R. M., 1976, *Synthetic Seismograms: A Finite Difference Approach*, Geophysics, **41**, 2–27.

[13] Mufti, I. R., 1985, *Seismic Modeling in the Implicit Mode*, Geophysical Prospecting, **33**, 619–656.

[14] Morse, P. M. and Feshbach, H., 1953, *Methods of Theoretical Physics*, McGraw–Hill Book Company, Inc., New York.

[15] Baker, L. J., 1989, *Supercomputers in Seismic Exploration*, Pergamon Press, Oxford, 1–10.

[16] Mufti, I. R., 1989, *Supercomputers in Seismic Exploration*, Pergamon Press, Oxford, 229–251.

[17] Oliger, J., 1976, *Hybrid Difference Methods for the Initial Boundary-Value Problem for Hyperbolic Equations*, Mathematics of Computation, Vol. 30, 136, 724–738.

[18] Reynolds, A. C., 1978, *Boundary Conditions for the Numerical Solution of Wave Propagation Problems*, Geophysics, **43**, 1099–1110.

[19] Robinson, E. A. and Treitel, Sven, 1980, *Geophysical Signal Analysis*, Prentice Hall, Englewood Cliffs, New Jersey.

[20] Strikwerda, J. C., 1989, *Finite Difference Schemes and Partial Differential Equations*, Wadsworth & Brooks/Cole, Belmont, Calif.

[21] Telford, W. M., Geldart, L. P., Sheriff, R. E., and Keys, D. A., 1976, *Applied Geophysics*, Cambridge University Press, Cambridge.

[22] Weinberger, H. F., *A First Course in Partial Differential Equations*, John Wiley and Sons, New York.