Stanford Exploration Project, Report 72, December 18, 1997, pages 1–183

Stanford Exploration Project, Report 72, December 18, 1997, pages 1–183

Short Note

The local paraxial ray method: a proposal

Lin Zhang[1]

keywords: *ray tracing, amplitudes, modeling*

INTRODUCTION

The seismic traveltimes and amplitudes associated with the WKB Green's functions can be computed by using two different approaches. One approach uses the dynamic ray tracing method to find the traveltimes and amplitudes along rays (Červený, 1987), and then interpolates them onto a regular grid with the paraxial ray approximation (Beydoun and Keho, 1987). The other approach uses a finite difference method to extrapolate wavefronts so that the traveltimes and amplitudes are computed directly on a regular grid (Vidale, 1988; Vidale and Houston, 1990; Van Trier and Symes, 1991; Podvin and Lecomte, 1991; Zhang, 1991). The first approach is usually called the ray tracing method and the second the finite difference method.

The ray tracing method computes accurate traveltimes and amplitudes along rays. However, for complicated velocity media, the interpolation process may be theoretically invalid and computationally expensive. The finite difference method can efficiently and accurately compute the traveltimes in the presence of extremely severe, arbitrarily shaped velocity contrasts, but the amplitude calculation is limited to smoothly varying velocity structures. The finite difference method also requires that the grid sizes to be sufficiently small.

In this note, I propose a new method that combines the two approaches. Like the finite difference method, the new method does the local computation on a regular grid and the computation proceeds step by step. On the other hand, the local computation within a grid cell involves local dynamic ray tracing followed by local paraxial wavefront interpolation. Therefore, I call the new method the local paraxial ray method. This method turns into a ray tracing method if the whole velocity model is within one grid cell. And it becomes a finite difference method if the grid sizes are sufficiently so small that the assumptions of straight rays and plane wavefronts are locally valid.

This note describes the principles of the local paraxial ray method and discusses

[1]**email:** not available

several issues related to its implementation.

PARAXIAL RAY APPROXIMATION

Ray tracing is a technique for finding the coordinates of a ray. Suppose, by using this technique, that we find the coordinates of the ray. We can then define a new orthogonal coordinate system (s, n), named ray-centered coordinates. As Figure **??** shows, the coordinate s measures the arc-length along the ray, and the coordinate n is the normal distance from the ray at point s on the ray. Let us denote the relations between Cartesian coordinates and ray-centered coordinates as $x = \hat{x}(s, n)$ and $z = \hat{z}(s, n)$. Then, the coordinates of the ray are $(\hat{x}(s, 0), \hat{z}(s, 0))$. The traveltime on any point of the ray can be computed by integrating the slowness function along the ray, as follows:

$$\tau(s, 0) = \tau(s_0, 0) + \int_{s_0}^{s} m(\xi, 0) d\xi, \tag{1}$$

where $m(s, n)$ is the slowness function in ray-centered coordinates.

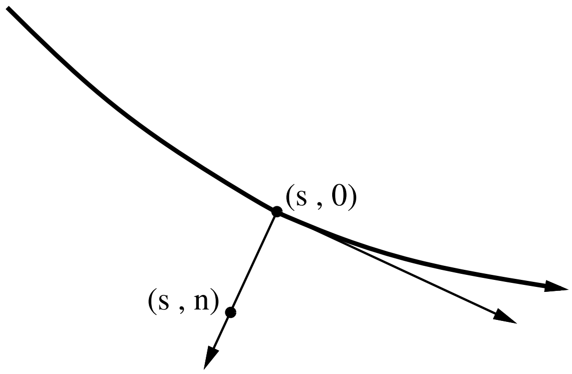

Figure 1: Ray-centered coordinates. lin3-rccoor [NR]

The amplitude calculation is more complicated than the traveltime calculation. Červený et al. (1977) showed that the amplitude function along the ray is related to a Jacobian function determined by

$$J(s, 0) = J(s_0, 0) \exp[\int_{s_0}^{s} \frac{M(\xi, 0)}{m(\xi, 0)} d\xi], \tag{2}$$

where $M(s, 0)$ is the second-order partial derivative of the traveltime with respect to n along the ray and can be found by solving the dynamic ray-tracing equation

$$\frac{dM}{ds} + \frac{1}{m} M^2 + m^2 \frac{\partial^2}{\partial n^2} \left(\frac{1}{m^2} \right) = 0 \tag{3}$$

along the ray.

To compute the traveltimes and amplitudes off but near the ray, we use the paraxial ray approximation. The resulted traveltime is

$$\tau(s,n) = \tau(s,0) + \frac{1}{2} M(s,0) n^2. \tag{4}$$

The direction of a ray follows the direction of the traveltime gradient that is determined by

$$\nabla \tau(s,n) = \left[m(s,0) + \frac{\partial m}{\partial n} n + \frac{1}{2} \frac{\partial M}{\partial s} n^2 \right] \vec{\mathbf{s}} + M(s,0) n \vec{\mathbf{n}}, \tag{5}$$

where $\vec{\mathbf{s}}$ and $\vec{\mathbf{n}}$ are unit vectors tangential and normal to the ray at $(s,0)$, respectively. Using the paraxial approximation also yields

$$M(s,n) = m(s,n) \frac{M(s,0)}{m(s,0)} \left[1 - \frac{1}{2} \frac{M^2(s,0)}{m^2(s,0)} n^2 \right] \tag{6}$$

and

$$J(s,n) = J(s,0) \left[1 + \frac{M^2(s,0)}{2m^2(s,0)} n^2 \right]. \tag{7}$$

The next section explains how these equations are used in the local paraxial ray method.

LOCAL WAVEFRONT EXTRAPOLATION

As in the case of the finite difference method, the description of the computational scheme of the local paraxial ray method consists of two parts. The first part addresses the local computation that determines how the traveltime and amplitude at a point are computed from the traveltimes and amplitudes at its neighbors. The second part explains the order in which the local computation proceeds. This note focuses on the description of the principles of the local computation. It should not be difficult to adapt these principles to the propagation schemes developed by Vidale and others.

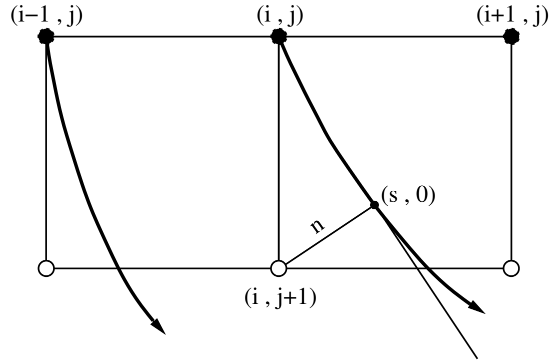

Figure 2: Local paraxial wavefront extrapolation. lin3-loray [NR]

The local computational scheme of the local paraxial ray method can be divided into two steps. As shown in Figure **??**, we assume that we know the traveltimes, the ray directions, the second-order partial derivatives of the traveltimes with respect to n, and the values of the Jacobian function at the grid points marked with dots, and we want to compute these quantities at the grid points marked with circles. The first step is to trace rays locally from every grid point marked with a dot and to compute traveltime $\tau(s, 0)$, function $M(s, 0)$, and Jacobian $J(s, 0)$ along each ray. To do so, we need to solve the ray equation and equation (3), and perform the integrations in equations (1) and (2). The second step interpolates these quantities computed along the rays onto the grid points marked with circles. Let us take grid point $(i, j+1)$ as an example. As indicated in Figure **??**, we can find ray-centered coordinates (s, n) of this point that are associated with the ray traced from grid point (i, j). If we assume that the grid sizes are small enough that the paraxial approximation is always valid, then we can use equation (4) to find the traveltime at grid point $(i, j+1)$ and use equation (7) to compute the value of the Jacobian function that is related to the amplitude at this point. Two other quantities are required in order to continue the local ray tracing from the grid point $(i, j+1)$. One is the ray direction at this point. The ray direction can be determined from the gradient of the traveltime that can be computed from equation (5). The other quantity is the value of function $M(s, n)$, which serves as the initial condition of the dynamic ray tracing equation at this point and can be computed from equation (6). If possible, we do the same computation for each ray passing by point $(i, j+1)$ and pick the result that is associated with the minimum traveltime. This completes the local computation.

DISCUSSION

In this section, I discuss three issues related to the implementation of the local paraxial ray method: the initial conditions, the representation of slowness model, and the choice of ray tracing algorithm.

Initial conditions

If we assume that the slowness model has a constant value m_0 in the region around source position (x_s, z_s), we can use the following initial conditions for the local paraxial ray method. The traveltimes are

$$\tau(x, z) = s_0 \sqrt{(x - x_s)^2 + (z - z_s)^2}.$$

The ray directions are specified by the angles of rays to the vertical, as follows:

$$\theta(x, z) = \arctan \frac{x - x_s}{z - z_s}.$$

We can also find that

$$M(x, z) = \frac{m_0}{\sqrt{(x - x_s)^2 + (z - z_s)^2}}$$

and

$$J(x,z) = \frac{1}{\sqrt{(x-x_s)^2 + (z-z_s)^2}}.$$

If the slowness varies in this region, the paraxial ray method should be used to find the initial conditions.

Representation of the slowness model

As with the ray tracing method, I assume that the slowness model is smoothly varying everywhere except along the specified interfaces where slowness discontinuities appear. To trace a ray within a grid cell in the smooth slowness region, we need to specify the slowness values in the cell on the basis of the given slowness samples at four corners of the cell. The constant slowness cell is the simplest one, but it causes problems in the amplitude calculation because of the slowness discontinuities formed around the boundaries of the cell. The constant gradient cell has the same potential problem as the constant slowness cell if the gradient of the slowness model is large. A simple continuous slowness model can be constructed by using the bilinear interpolation. The bilinear interpolation within a cell is defined as follows:

$$m(\alpha,\beta) = (1-\alpha)(1-\beta)m_{ij} + \alpha(1-\beta)m_{i+1j} + (1-\alpha)\beta m_{ij+1} + \alpha\beta m_{i+1j+1}, \quad (8)$$

where $0 \le \alpha \le 1$ and $0 \le \beta \le 1$. The components of the slowness gradient within the cell are

$$\left\{ \begin{array}{lcl} \dfrac{\partial m}{\partial x}(\alpha,\beta) & = & (1-\beta)\dfrac{m_{i+1j} - mij}{\Delta x} + \beta\dfrac{m_{i+1j+1} - m_{ij+1}}{\Delta x} \\ \dfrac{\partial m}{\partial z}(\alpha,\beta) & = & (1-\alpha)\dfrac{m_{ij+1} - mij}{\Delta z} + \alpha\dfrac{m_{i+1j+1} - m_{i+1j}}{\Delta z}, \end{array} \right. \quad (9)$$

which are not constant, and thus may complicate local ray tracing algorithm. Further numerical experiments are required to determine which type of cells should be used.

If a slowness cell contains a portion of an interface, we need to specify the dip of the interface as well as the slownesses at two sides of the interface.

Choice of the ray tracing algorithm

The local ray tracing algorithm should be as simple as possible because it has to be run many times. Ideally, we would like to have the analytical expressions for the coordinates of ray trajectories, which would allow us to find ray-centered coordinates efficiently. However, this would be difficult if we use bilinear interpolation within grid cells. Thus, we need to make approximations. The approximation with straight rays is not appropriate for slowness models interpolated bilinearly. I expect that approximating ray trajectories with quadratic functions should meet the needs.

CONCLUSIONS

I have described a new method, named local paraxial ray method, for calculating the traveltimes and amplitudes associated with the WKB Green's functions. The method combines the features of both conventional ray tracing method and the recently developed finite difference method; hence I expect that proper uses of the method should allow us to overcome some serious difficulties encountered by other methods.

REFERENCES

Beydoun, W.B., and Keho, T.H., 1987, The paraxial ray method: Geophysics, **52**, 1639–1653.

Červený, V., Molotkov, I.A., and Pšenčik, I., 1977, Ray methods in seismology: Univ. of Karlova Press.

Červený, V., 1987, Ray tracing algorithms in three-dimensional laterally varying layered structures: Seismic Tomography, edited by G. Nolet, Riedel Publishing Company, 99–134.

Podvin, P., and Lecomte, I., 1991, Finite difference computation of traveltimes in very contrasted velocity models: a massively parallel approach and its associated tools: Geophys. J. Int., **105**, 271–284.

Van Trier, J., and Symes, W.W., 1991, Upwind finite-difference calculation of traveltimes: Geophysics, **56**, 812–821.

Vidale, J.E., 1988, Finite-difference calculation of travel times: Bull. Seism. Soc. Am., **78**, no. 6, 2062–2076.

Vidale, J.E., and Houston, H., 1990, Rapid calculation of seismic amplitudes: Geophysics, **55**, 1504–1507.

Zhang, L., 1991, Finite difference calculation of Green's functions: SEP–**72**, 153–169.