

Traveltime modeling in 2D isotropic media using the fast marching method

Chaiwoot Boonyasiriwat

ABSTRACT

In this work, I implement the fast marching method (FMM) for first-arrival traveltime modeling in 2D isotropic media. Both MATLAB and C codes have been developed and will later be used in first-arrival traveltime inversion, and Kirchhoff modeling and migration. In the future, I will extend this implementation to anisotropic case.

INTRODUCTION

Traveltime modeling is an important process which has many applications including traveltime inversion and Kirchhoff migration. The efficiency of traveltime modeling is very crucial to that of traveltime inversion and Kirchhoff migration, especially in 3D applications, since traveltime modeling must be carried out several times.

In this work, the computationally efficient fast marching method proposed by Sethian and Popovici (1999) for solving the eikonal equation was implemented in both MATLAB and C. The MATLAB is not computationally efficient compared to the C code which uses the heap data structure to store the traveltimes of the constructed wave front.

FAST MARCHING METHOD

In this section, I briefly describe the fast marching method. The full details of the method is described in Sethian and Popovici (1999).

The eikonal equation governs the traveltime of seismic wave propagating from one point to any point in the medium. In this work, we only consider the isotropic eikonal equation which can be directly derived from the acoustic wave equation by using the high frequency approximation. The reader is referred to Shearer (1999) for the detailed derivation.

The eikonal equation can be written as

$$|\nabla T(\mathbf{r})|^2 = \frac{1}{v^2(\mathbf{r})} = s^2(\mathbf{r}), \quad (1)$$

where $T(\mathbf{r})$ is traveltime at position \mathbf{r} , $v(\mathbf{r})$ is the wave velocity, and $s(\mathbf{r})$ is the slowness which is the reciprocal of velocity.

Let's consider the eikonal equation in one dimension, given by

$$\left(\frac{\partial T(x)}{\partial x}\right)^2 = \frac{1}{v^2(x)} = s^2(x). \quad (2)$$

Sethian and Popovici (1999) proposed to approximate the spatial derivative using the upwind finite difference

$$\left(\frac{\partial T(x)}{\partial x}\right)^2 \approx (\max(D_i^{+x}T, 0)^2 + \min(D_i^{-x}T, 0)^2), \quad (3)$$

where D_i^{+x} and D_i^{-x} are the standard finite-difference notions defined by

$$D_i^{+x} = \frac{T_{i+1} - T_i}{\Delta x}, \quad (4)$$

and

$$D_i^{-x} = \frac{T_i - T_{i-1}}{\Delta x}. \quad (5)$$

Here, T_i is the value of T on a grid at point $i\Delta x$ with a uniform grid spacing Δx . The unwind scheme given in equation 3 automatically chooses grid points in the approximation from the flow direction of information (Sethian and Popovici, 1999). This upwind scheme is very important since conventional scheme will lead to unstable results.

In two dimensions, the eikonal equation can be written as

$$\left(\frac{\partial T(x, z)}{\partial x}\right)^2 + \left(\frac{\partial T(x, z)}{\partial z}\right)^2 = \frac{1}{v^2(x, z)} = s^2(x, z), \quad (6)$$

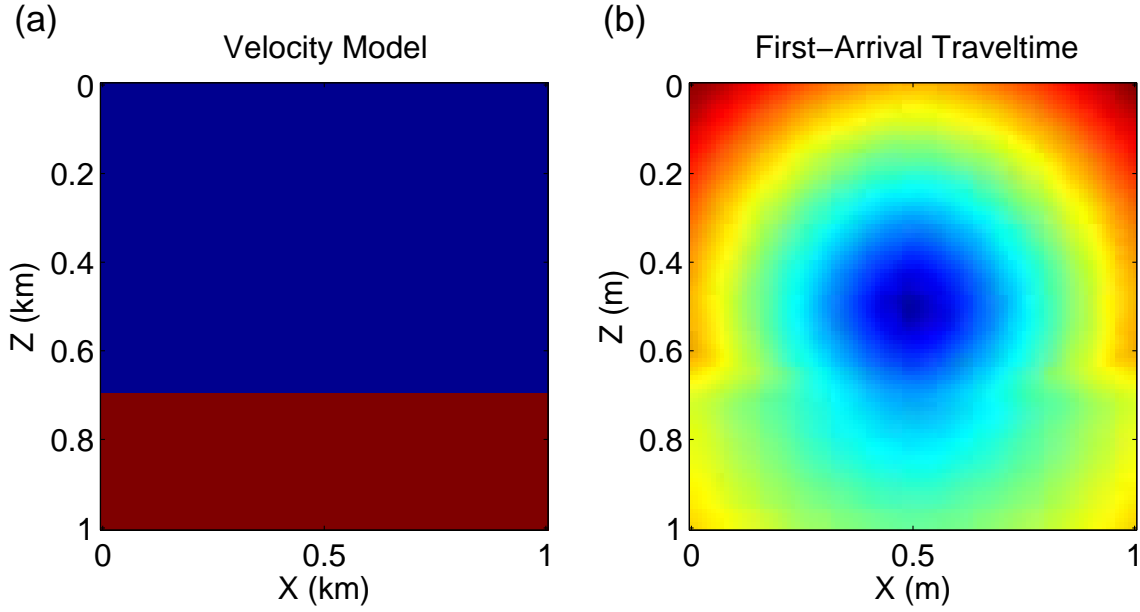


Figure 1: (a) Two-layer velocity model. (b) First-arrival traveltime obtained using the fast marching method.

and the left hand side can be approximated as

$$\begin{aligned} & \left(\frac{\partial T(x, z)}{\partial x} \right)^2 + \left(\frac{\partial T(x, z)}{\partial z} \right)^2 \\ & \approx (\max(D_{ij}^{+x} T, 0)^2 + \min(D_{ij}^{-x} T, 0)^2) + \\ & \quad (\max(D_{ij}^{+z} T, 0)^2 + \min(D_{ij}^{-z} T, 0)^2), \end{aligned} \quad (7)$$

where j is the index of the grid point in the z -direction.

Now I present the algorithm for implementing the fast marching method. First, the traveltimes of grid points at and surrounding the source location are computed explicitly. Then, grid point at or nearest to the source location is set as the **accepted value**, whose traveltime value can no longer change, while the grid points surrounding the source point are set as the **narrow band**, whose traveltime value can still be changed. The traveltime values of grid points in the narrow band are then sorted and store in a heap data structure. The binary heap is usually used in this case. The top of the heap is the smallest traveltime value which corresponds to a grid point on a wave front we are constructing. This grid point is then set as **accepted**. Then, the traveltimes of the neighboring grid points of the newly accepted point are computed for the grid points originally outside of the narrow band, or recomputed for the grid points already in the narrow band. Then all these neighboring points are (re)added to the narrow band. The sorting process is then carried on to find the new grid point with the smallest traveltime value. The wavefront is then expanded iteratively until all grid points are accepted.

SYNTHETIC DATA EXAMPLE

The fast marching is applied to compute first-arrival traveltimes from a point source located in a 2D isotropic, two-layer model (Figure 1a). The computed traveltimes are shown in Figure 1b.

SUMMARY

The fast marching method has been implemented in both MATLAB and C and applied to a synthetic model. In the future, I will extend it to an anisotropic case.

ACKNOWLEDGMENTS

I am grateful for the support from PTTEP.

REFERENCES

- Sethian, J. A. and A. M. Popovici, 1999, 3-D traveltime computation using the fast marching method: *Geophysics*, **64**, 516–523.
- Shearer, P., 1999, *Introduction to seismology*: Cambridge University Press.