*Stanford Exploration Project, Report SERGEY, November 9, 2000, pages 477–**??***

Stanford Exploration Project, Report SERGEY, November 9, 2000, pages 477–??

Implementing the fast marching eikonal solver: Spherical versus Cartesian coordinates

Tariq Alkhalifah and Sergey Fomel[1]

ABSTRACT

Spherical coordinates are a natural orthogonal system to describe wavefronts emanating from a point source. While a regular grid distribution in the Cartesian coordinate system tends to undersample the wavefront description near the source (the highest wavefront curvature) and oversample it away from the source, spherical coordinates, in general, provide a more balanced grid distribution to characterize such wavefronts. Our numerical implementation confirms that the recently introduced fast marching algorithm is both a highly efficient and an unconditionally stable eikonal solver. However, its first-order approximation of traveltime derivatives can induce relatively large traveltime errors for waves propagating in a diagonal direction with respect to the coordinate system. Examples, including the infamous Marmousi model, show that a spherical coordinate implementation of the method results in far fewer errors in traveltime calculation than the conventional Cartesian coordinate implementation, and with practically no loss in computational advantages.

INTRODUCTION

A recently introduced method for solving the eikonal equation, named the *fast marching* method (Sethian, 1996; Sethian and Popovici, 1997), has two very intriguing features: it is unconditionally stable and, at the same time, highly efficient. Essentially, the method is based on solving the eikonal equation on a Cartesian grid along the wavefront, starting with those points with minimum traveltime—an idea similar to—the method of *expanding wavefronts*(Qin et al., 1992). Additionally, a minimum traveltime tree is constructed using fast algorithms (heap sorting) and can be maintained at a computational cost proportional to $\log N$, where N is the number of grid points in the computational domain. As a result, the cost of the eikonal solver is roughly proportional to $N \log N$. The minimum-time requirement ensures the method's stability no matter how complicated the velocity model is.

However, this new solver has a problem. It is based on a first-order approximation of the traveltime derivatives with respect to position. This low-order approximation can result in relatively large traveltime errors, particularly in two cases:

[1]**email:** tariq@sep.stanford.edu, sergey@sep.stanford.edu

- When there is large wavefront curvature, such as near the source
- When the wavefront propagation is diagonal to the grid orientation

Higher-order approximations of the traveltime derivatives could reduce these errors, but at a considerably larger cost.

In this paper, we investigate a possibility of reducing the computational errors by implementing the fast marching solver in spherical coordinates. The ability to handle turning waves is usually considered the main advantage of solving the eikonal equation in polar coordinates (van Trier and Symes, 1991; Popovici, 1991; Schneider, 1993; Fowler, 1994). Turning waves, however, do not present a problem with this new scheme; the method can, by design, handle wavefronts at any angle, provided that they correspond to the first arrival. Nevertheless, the importance of the polar coordinates in the fast marching approach should not be underestimated. Compared with the Cartesian-coordinates implementation of the fast marching algorithm, in simple and complex velocity models, we show that the polar coordinate implementation can considerably improve the accuracy of the solution at practically no loss in the computational efficiency.

FINITE-DIFFERENCE SOLUTION OF THE EIKONAL EQUATION

Using the high-frequency approximation, wavefronts in a 3-D model can be described by the eikonal equation,

$$\left(\frac{\partial t}{\partial x}\right)^2 + \left(\frac{\partial t}{\partial x}\right)^2 + \left(\frac{\partial t}{\partial x}\right)^2 = s^2(x,y,z), \tag{1}$$

where t is the traveltime, and s is the wave slowness in a 3-D model. For arbitrary slowness models the eikonal equation is solved numerically using finite-difference schemes introduced by Vidale (1990).

Numerically solving the eikonal equation is probably the most efficient method of obtaining wavefront traveltimes in arbitrary velocity models. One reason for the efficiency is that we can conveniently solve the eikonal equation over a regular grid, which eliminates the need for the interpolation commonly used with other methods, such as ray tracing.

A major drawback of using conventional methods to solve the eikonal equation numerically, is that we only evaluate the fastest arrival solution, not necessarily the most energetic (Vidale, 1990). This results in less than acceptable traveltime computation for imaging in complex media (Geoltrain and Brac, 1993). Another drawback of the eikonal solvers is that the conventional solvers are generally unstable in complex media (Popovici, 1991). Stability usually came at higher price (a finer grid) to the implementation. In addition, conventional eikonal solvers cannot treat turning waves. Using polar coordinates, though it does not eliminate the turning wave problem, allows for wavefronts overturning at a considerable angle, on the condition that they do not overturn on the radial axis. The eikonal equation can also be

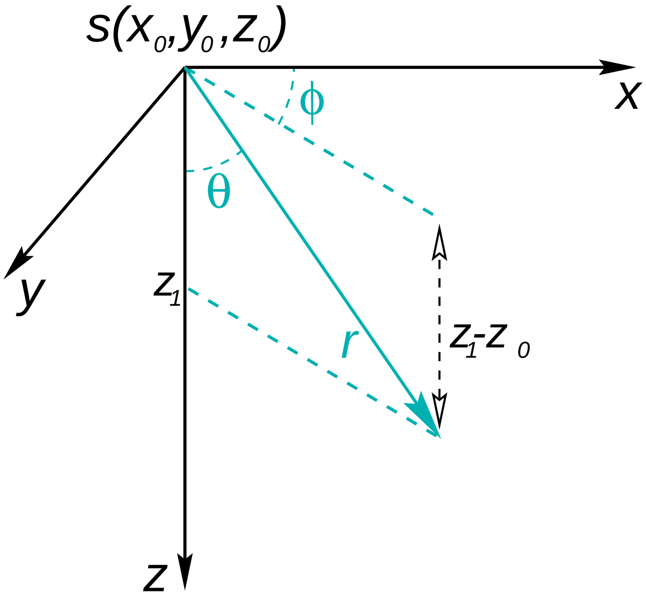

Figure 1: A spherical coordinate system given by r, θ, and ϕ. The source, $s(x_0, y_0, z_0)$, is at the origin of the spherical coordinates where $r = 0$. The parameter r ($=\sqrt{(x-x_0)^2+(y-y_0)^2+(z-z_0)^2}$) is the distance from the source to the point of interest along the wavefront, which is also the magnitude of the vector. The parameter ϕ is the angle between the x-axis and the projection of the vector onto the $x-y$ plane. The parameter θ is the angle between the z-axis and the vector along a vertical plane. fmpolar-polar-dif [NR]

described in spherical coordinates (r, θ, and ϕ; see Figure 1), as follows:

$$\left(\frac{\partial t}{\partial r}\right)^2 + \frac{1}{r^2}\left(\frac{\partial t}{\partial \theta}\right)^2 + \frac{1}{r^2\sin^2\theta}\left(\frac{\partial t}{\partial \phi}\right)^2 = s^2(r,\theta,\phi). \tag{2}$$

Note the $\sin\theta$ equals zero for vertical wave propagation, which pauses a problem for this eikonal equation. However, the change of t with respect to ϕ is, in this case, meaningless. Therefore, a flag is necessary to remove the ϕ term when waves are propagating vertically or nearly vertically. A small constant parameter ($\approx \epsilon$) added to $\sin\theta$ in the denominator of the ϕ term will ensure the eikonal equation has numerical stability. Other stability solutions are suggested by Schneider (1993) and Fowler (1994). Also, the variation of the grid size in polar coordinates poses a problem for the stability of conventional solvers, especially when encountering small scale inhomogeneities away from the source (Popovici, 1991; Fowler, 1994). These stability problems are eliminated by using the fast marching method, which is unconditionally stable. The spherical coordinate system, as we will see later, also helps improve the accuracy of the fast marching method.

In polar coordinates, we eliminate the ϕ terms so that the eikonal equation, described by r and θ, is given by

$$\left(\frac{\partial t}{\partial r}\right)^2 + \frac{1}{r^2}\left(\frac{\partial t}{\partial \theta}\right)^2 = s^2(r,\theta). \tag{3}$$

The only singularity in the polar coordinate form occurs at $r = 0$ which is easily avoidable by placing the source at r=0, and therefore, setting the time, t, to zero at r=0.

THE FAST MARCHING ALGORITHM

The fast marching method for numerically solving the eikonal equation is described in detail in Cao and Greenhalgh (1994) and Sethian (1996). In this section, we give a summary, as well as point out some of the advantages and drawbacks, of the approach.

Starting from a point source (we can also start with a plane or any closed surface of sources), we calculate the traveltime at the surrounding grid points analytically. Obtaining initial analytical solutions is necessary to reduce some of the first-order numerical errors, discussed in detail below. The original point source is set; that is its traveltime cannot be updated. The new traveltimes of the surrounding grid points are put in an array that constitutes the wavefront and are sorted from minimum to maximum traveltime values. The minimum traveltime, which is in front of this array, is extracted first, its value is set (cannot be updated), and all neighboring unset grid points are updated using the following formula (Sethian, 1996):

$$\begin{aligned} max(D_{ijk}^{-x}t,0)^2 &+ min(D_{ijk}^{+x}t,0)^2 + \\ max(D_{ijk}^{-y}t,0)^2 &+ min(D_{ijk}^{+y}t,0)^2 + \\ max(D_{ijk}^{-z}t,0)^2 &+ min(D_{ijk}^{+z}t,0)^2 = s_{ijk}^2, \end{aligned} \tag{4}$$

where D_{ij}^{-x} is the derivative of traveltime with respect to x at grid point i, j, k given by

$$D_{ijk}^{-x}t = \frac{t_{i,j,k} - t_{i-1,j,k}}{\Delta x},$$

and

$$D_{ijk}^{+x}t = \frac{t_{i+1,j,k} - t_{i,j,k}}{\Delta x}.$$

The same holds for D_{ijk}^{-y}, D_{ijk}^{+y}, and D_{ijk}^{-z}, D_{ijk}^{+z} for y and z, respectively. The traveltime t_{ijk} and the slowness s_{ijk} correspond to the grid point that is being updated. Solving for t_{ijk}, the time at a new grid point, requires solving a quadratic equation that has two solutions. We choose the solution that reduces to $t_{ijk} = t_{i-1,j,k} + \frac{\Delta x}{v}$, or $t_{ijk} = t_{i+1,j,k} + \frac{\Delta x}{v}$, when the wavefront travels horizontally. We do the same for all other unset points surrounding the initial extracted point. Each newly computed grid point is added to the wavefront array, and by using a highly efficient *heap method* the sorting from minimum to maximum traveltime is done promptly. Then. we extract the minimum once again, update all neighboring live grid points, and so on. Figure 2 shows schematic plots of the progress of this method along a 2-D Cartesian grid. The first-order nature of the fast marching method results in large errors for conventional sparse grid-point configuration. The largest of these errors occur at the first step of computation. To demonstrate, let us consider a simple example where the wave slowness is 1 km/s, and the grid spacing is 1 km. From Figure 3, the grid point A is the location of the source and its traveltime is, as a result, set to zero. Grid points B and C are part of the front with traveltimes accurately calculated at B and C (for both $t = 1$). Grid point B is extracted from the wavefront array and used to calculate the traveltime at D based on equation (4). As a result, the traveltime at D equals $1 + \frac{1}{\sqrt{2}}$ instead of the true value of $\sqrt{2}$. This difference

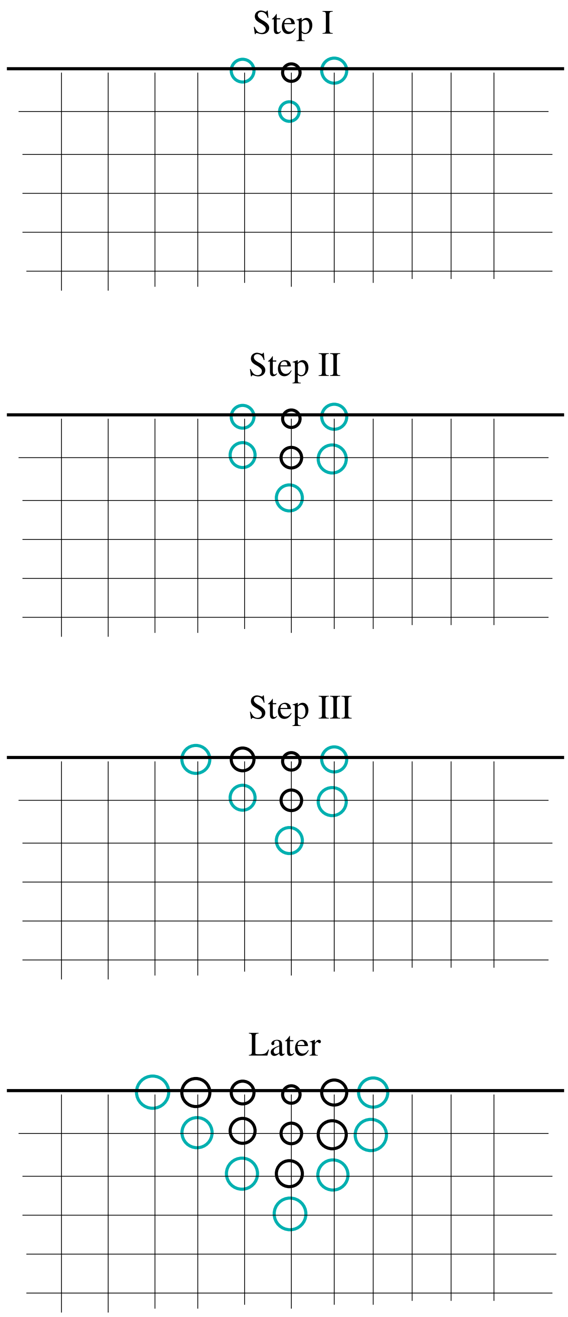

Figure 2: The steps taken to implement fast marching in Cartesian coordinates. Black circles imply computed traveltimes that are set, because of their minimum nature, at the time of extraction from the heap array. Gray circles constitute the front of the wave, and they are stored in the wavefront heap array and sorted from minimum to maximum traveltime value. The traveltime is schematically given by the size of the circle; the larger the radius, the greater the traveltime. The minimum is always extracted first from the heap array at each step, its traveltime is set (given a black circle), and surrounding grid points that are not set are computed and inserted into the wavefront heap array. We precede until all grid points are computed and set. fmpolar-march-cart [NR]

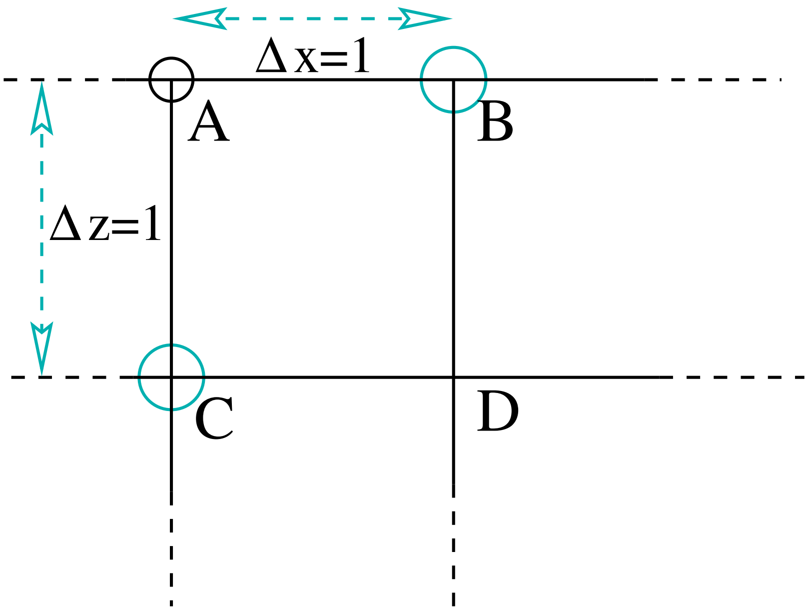

Figure 3: A simple Cartesian-coordinate grid configuration. In this model, the slowness equals 1. Grid A is set, and B is the minimum traveltime along the front that includes B and C. We then compute D as a neighbor of B. fmpolar-ABCD [NR]

corresponds to about 20 percent error in the traveltime calculation at D. It is also probably the maximum error allowed by this scheme, which occurs here because of the 45-degree wavefront propagation angle and the high wavefront curvature (near the source). This 20 percent error is always associated with the computation of traveltimes for the first layer of grid points around the source grid point. The next layer of grid points should have lower relative errors. If a plane wave (that is, with no curvature) passes through the same grid points in Figure 3 at a 45-degree wavefront propagation angle, the traveltime at D computed using equation (4) equals $\sqrt{2}$, which is accurate. Therefore, the amount of wavefront curvature with respect to the grid is another factor that affects the size of the error. As we propagate away from the source, the percentage of error reduces considerably. One way to avoid errors near the source is to compute the traveltime analytically at grid point D, and delay starting the fast marching algorithm until after computing the first group of grid points surrounding the source.

However, such first-order errors will always exist, as we will see in the numerical examples, regardless of how we compute the initial set of points. One way to reduce such errors, yet retain the main features of the method, is to using the spherical coordinate system, as we will see in the next section.

FAST MARCHING IN SPHERICAL COORDINATES

In spherical, or polar, coordinates, we are effectively propagating a plane wave on a regular grid. For homogeneous media, this plane wave will have a front that is always parallel to the θ-ϕ plane. As a result, traveltime calculation using equation (4) in spherical coordinates is always exact in homogeneous media. Figure 4 is similar to Figure 2, but using the polar coordinate system. The source is computed initially and set to zero for all angles θ. When stretched on a regular grid, all points at the surface $r = 0$ are set to zero. These points are

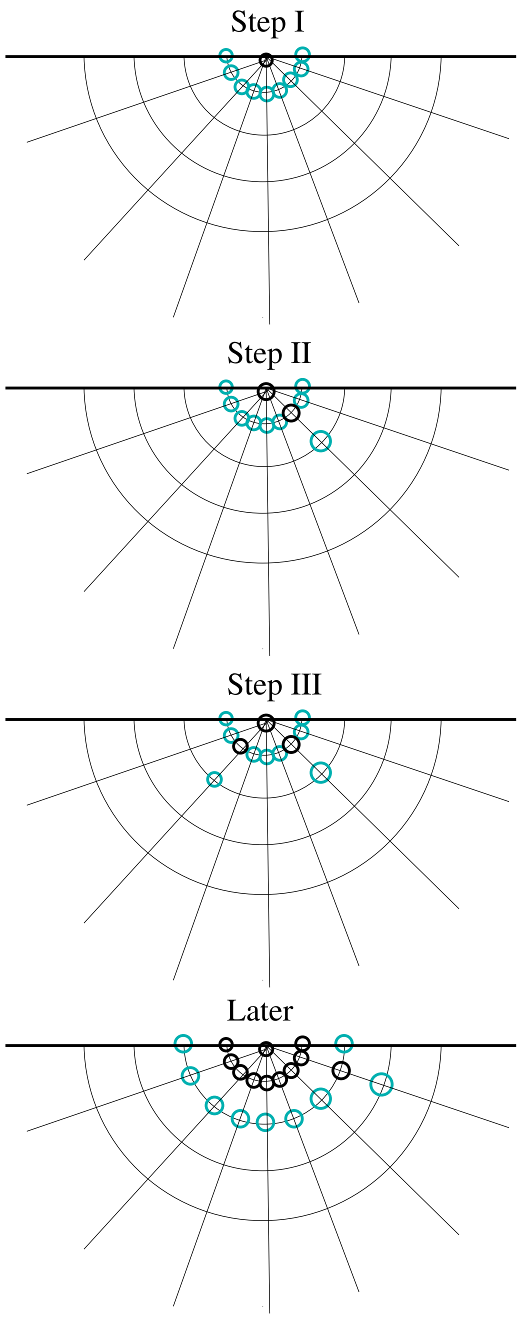

Figure 4: The steps taken to implement the fast marching in polar coordinates. This implementation resembles the Cartesian coordinate one shown in Figure 2, but with a different grid orientation. Black circles imply computed traveltimes that are set because of their minimum traveltime value along the front. Gray circles constitute the front, and their values are ordered in the heap array from minimum to maximum. The minimum is always extracted first from the heap array at each step, its traveltime is set (given a black circle), and all surrounding grid points that are not set are computed and put into the heap array. $\boxed{\text{fmpolar-march-pol}}$ [NR]

inserted in the wavefront array and sorted from minimum traveltime to maximum. In the case of the source grid point, the sorting step is unnecessary because all traveltimes are equal to zero. The minimum is then extracted, and the traveltimes for neighboring grid points are computed using the following relation:

$$\begin{aligned} max(D^{-r}_{ijk}\,t,0)^2 &+ min(D^{+r}_{ijk}\,t,0)^2 + \\ max(D^{-\theta}_{ijk}\,t,0)^2 &+ min(D^{+\theta}_{ijk}\,t,0)^2 + \\ max(D^{-\phi}_{ijk}\,t,0)^2 &+ min(D^{+\phi}_{ijk}\,t,0)^2 = s^2_{ijk}, \end{aligned} \tag{5}$$

where D^{-r}_{ijk} is the derivative of traveltime with respect to r at grid point i, j, k, given by

$$D^{-r}_{ijk}\,t = \frac{t_{i,j,k} - t_{i-1,j,k}}{\Delta r},$$

and

$$D^{+r}_{ijk}\,t = \frac{t_{i+1,j,k} - t_{i,j,k}}{\Delta r}.$$

The D^{θ}_{ijk} and D^{ϕ}_{ijk} derivatives are slightly different, given by

$$D^{-\theta}_{ijk}\,t = \frac{t_{i,j,k} - t_{i,j-1,k}}{r\Delta\theta},$$

$$D^{+\theta}_{ijk}\,t = \frac{t_{i,j+1,k} - t_{i,j,k}}{r\Delta\theta},$$

$$D^{-\phi}_{ijk}\,t = \frac{t_{i,j,k} - t_{i,j-1,k}}{r\sin\theta\,\Delta\phi},$$

and

$$D^{+\phi}_{ijk}\,t = \frac{t_{i,j+1,k} - t_{i,j,k}}{r\sin\theta\,\Delta\phi}.$$

Unlike the implementation in Cartesian coordinate, the heap array size in polar coordinates tends to be stable; as we extract a grid point, we usually insert another. In Cartesian coordinates, the heap array can become very large, especially in 3D media, as the wavefront expands. Figure 5 shows the size of the heap array for implementing the fast marching method on the model in Figure 7 using polar coordinates (solid curve) and Cartesian coordinates (dashed curve). Clearly, the heap array size in polar coordinate is practically the same through most of the computation, while the Cartesian version grows dramatically and then drops. The case becomes worse in 3-D media. Even with the polar coordinate system, the majority of the errors occur when the wavefront is at a 45 degree angle to the grid configuration. The time that the wavefront spends traveling diagonally along the polar coordinates is small, even in complex media, relative to that spent in the Cartesian coordinate system. Also, the wavefront curvature, another source of error in the scheme, though physically present because of a point source, does not computationally exist at the start of the wave along the spherical grid. As a result, the errors that accumulate at the start of wavefront propagation in the Cartesian coordinates are minimized in the spherical coordinate implementation.

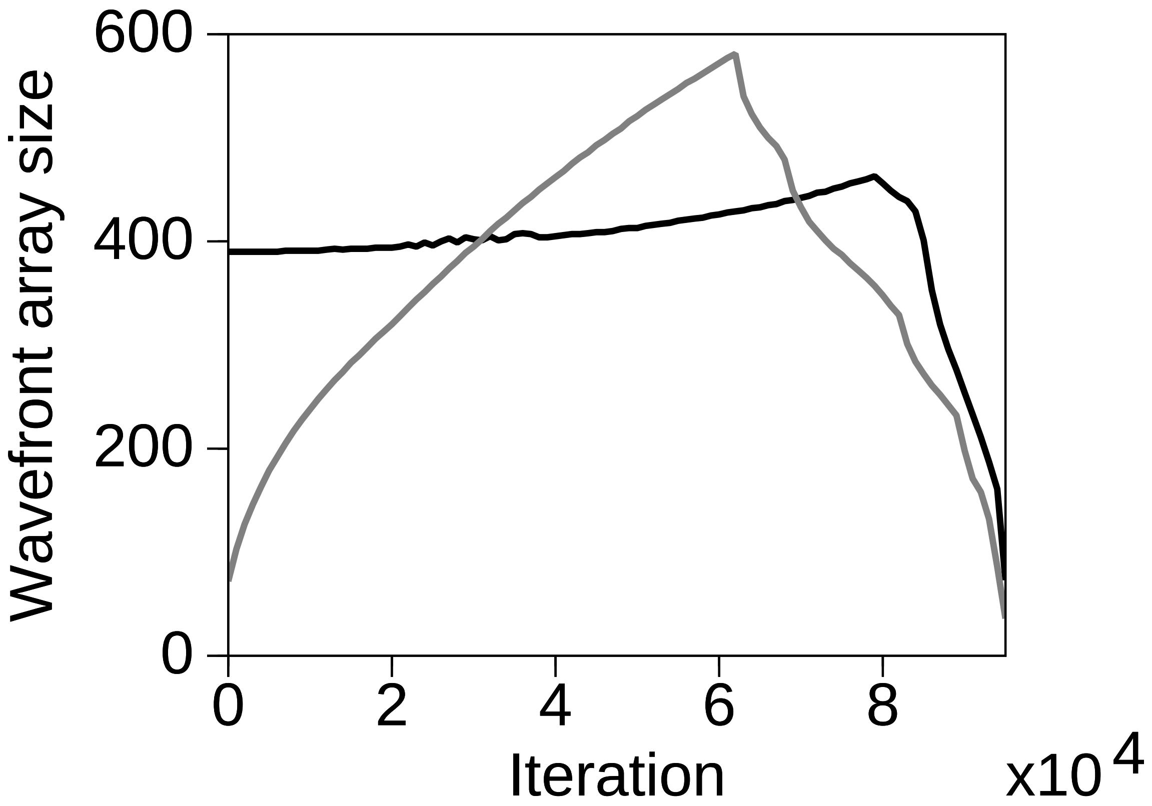

Figure 5: The wavefront heap array size as a function of the number of iterations for traveltimes calculated in the model shown in Figure 7. The black curve corresponds to the fast marching method in polar coordinates, while the gray curve corresponds to the method in Cartesian coordinates. fmpolar-count [NR]

NUMERICAL TESTS

Most of the numerical examples shown below are to demonstrate the advantages of using the spherical, or polar, coordinate system, over the Cartesian one, with this new efficient and unconditionally stable eikonal solver. The Cartesian coordinate implementation includes analytically solving for the first layer of grid points around the point source to reduce the wavefront curvature errors.

At the top of Figure 6, we show the traveltime in a homogeneous medium computed using a second-order in time, or first order in ray parameter, eikonal solver (van Trier and Symes, 1991), as well as using the grid adaptive scheme to achieve better stability (The code was built by Dave Hale, 1991). This eikonal solver, because of its higher-order accuracy, serves as the reference for testing the accuracy of the fast marching implementation in different coordinate systems. In addition, this particular second-order solver is exact in homogeneous media, because it is executed in polar coordinates. At the bottom of Figure 6, we show the traveltime difference, or error, between implementing the fast-marching method in Cartesian coordinates (left), polar coordinates (right), in contrast to the more accurate second-order scheme. As expected, the majority of the errors in the Cartesian coordinate implementation are concentrated around the 45-degree angle wave propagation. The errors also increase more rapidly near the source where the wavefront curvature is the largest. The polar coordinate implementation, on the other hand, is almost exact for homogeneous media. In this case, the waves propagate in a plane wave geometry with respect to the grid orientation.

Figure 7 shows traveltimes in a slightly more complicated velocity model. The traveltime contours computed using the various methods practically coincide. A closer look, in Figure 8, reveals, as in Figure 6, the details of the errors using the different coordinate schemes. The result of using the second-order eikonal solver is shown at the top, and the absolute traveltime errors from using the fast marching method in Cartesian coordinates (left), and polar coordinates (right) are shown at the bottom. The Cartesian and polar coordinate implementations have about the same computational cost; both are far faster than the more accurate second-order scheme. Clearly, the polar coordinate implementation has far fewer errors than the Cartesian coordinate one.

THE MARMOUSI MODEL

When the phrase *complex media* is used, the Marmousi model often comes to mind. The huge amount of folding and faulting induced in this model have created a rather interesting distribution of velocity anomalies and discontinuities. As a result, the Marmousi model has served as a performance measure used to evaluate various migration and traveltime calculation algorithms. In this section, we use the unsmoothed version of the Marmousi model to test the fast marching methods accuracy and stability. The lateral grid spacing was reduced to obtain a more efficient finite-difference solution of the acoustic wave equation, for comparison. Because the second-order scheme (van Trier and Symes, 1991), even with the adaptive measures, is not stable in such a complex model, we use a version of the fast marching eikonal

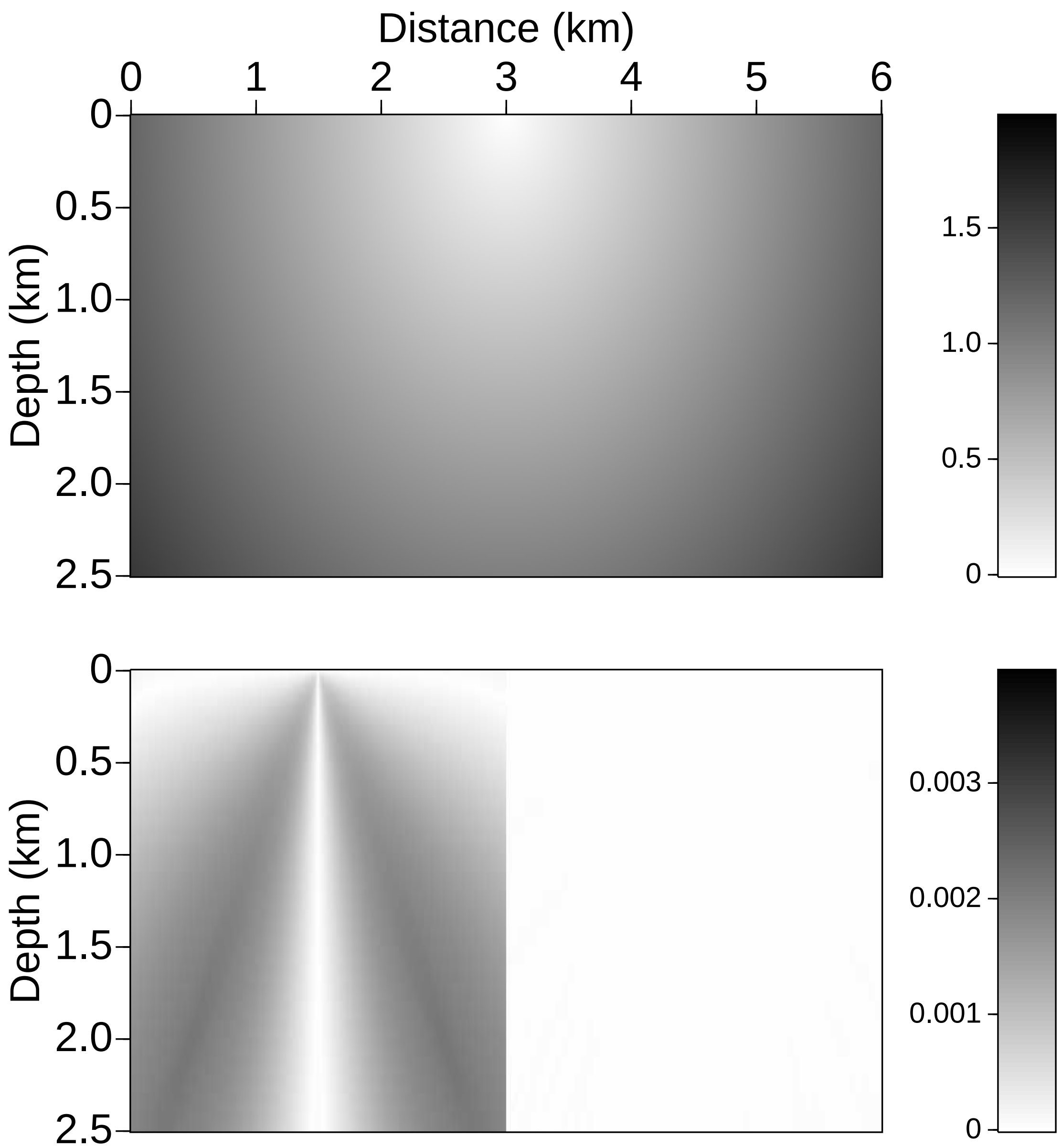

Figure 6: The top section shows traveltimes (in seconds) in a homogeneous medium with velocity equal to 2500 m/s, resulting from a source located at the surface at distance 3000 m. The bottom section shows the absolute difference, or errors, in seconds between traveltimes calculated by the fast marching method using the Cartesian coordinates (left) and the polar coordinates (right) in comparison to the more accurate second-order eikonal solver. In this homogeneous case, the polar coordinate fast marching method has practically no errors. fmpolar-wkbjvh [NR]

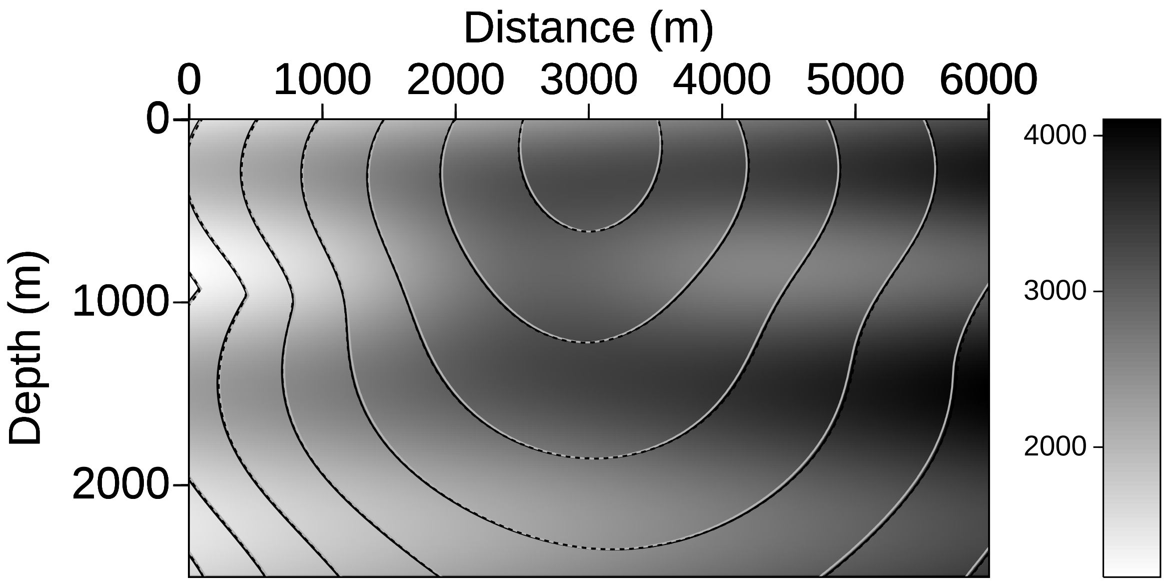

Figure 7: Contours of traveltimes resulting from a source placed on the surface at a distance of 3000 m. The traveltime contours are superimposed on the velocity model (in m/s). The solid black curves correspond to the solution of the more accurate second-order eikonal solver. The gray curves correspond to use of the fast marching eikonal solver in Cartesian coordinates; the dashed curves, in polar coordinates. The three curves are really close. fmpolar-vtcont-simpleg [NR]

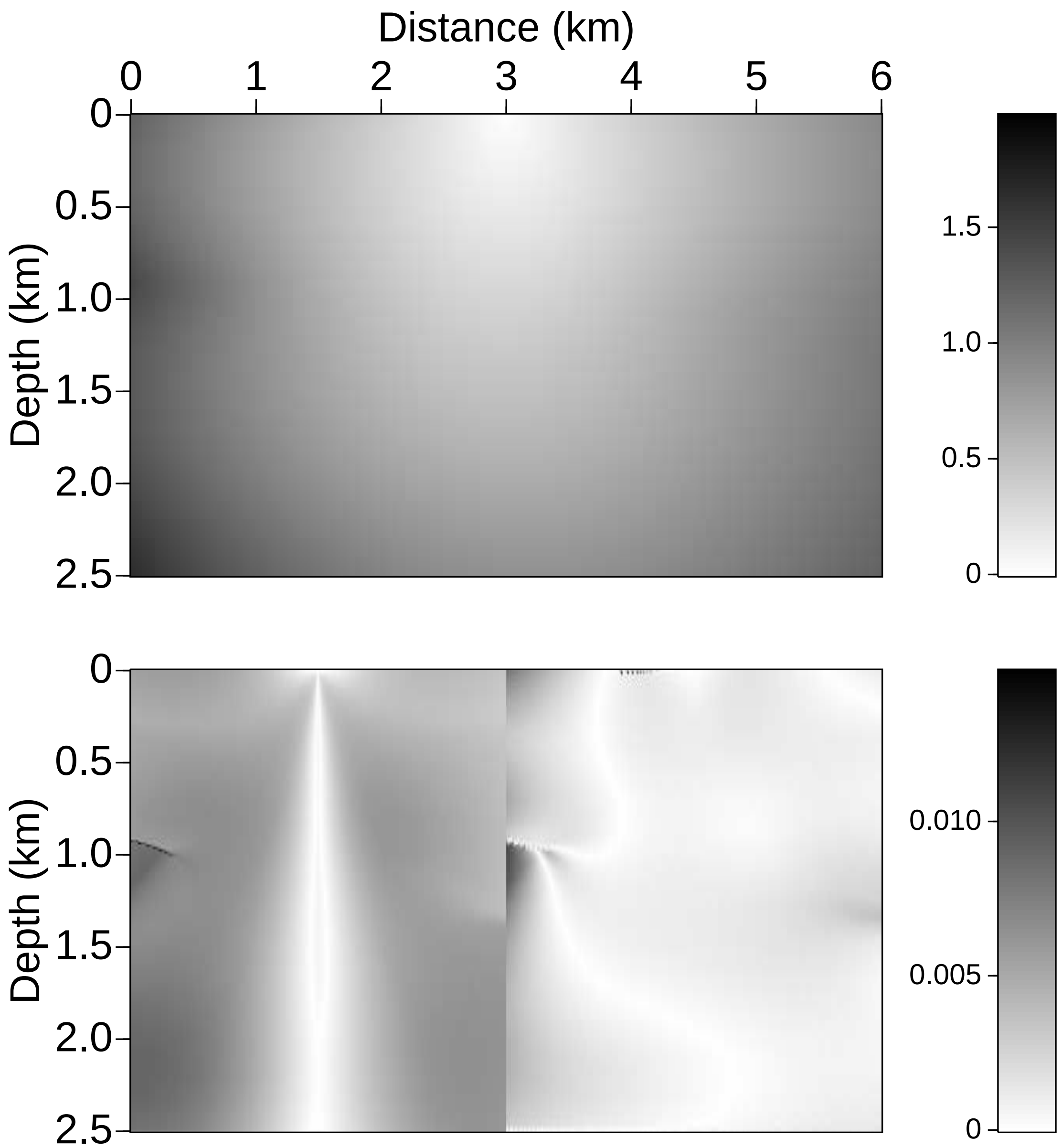

Figure 8: The top section shows the traveltimes, in seconds, for waves propagating in the model shown in Figure 7, resulting from a source located at the surface at a distance of 3000 m. The bottom sections show the absolute difference, or errors, in seconds between traveltime calculated by the fast marching method using the Cartesian coordinates (left) and the polar coordinates (right) in comparison to the more accurate second-order eikonal solver. fmpolar-wkbjvxz [NR]

solver, implemented on a very fine grid in polar coordinates to reduce the first-order traveltime derivate errors. The solution of this fine-grid implementation will serve as a reference used to compare the accuracies of Cartesian versus polar coordinate implementation of the method on a more practical grid spacing. Figure 9 shows contours of traveltime using the

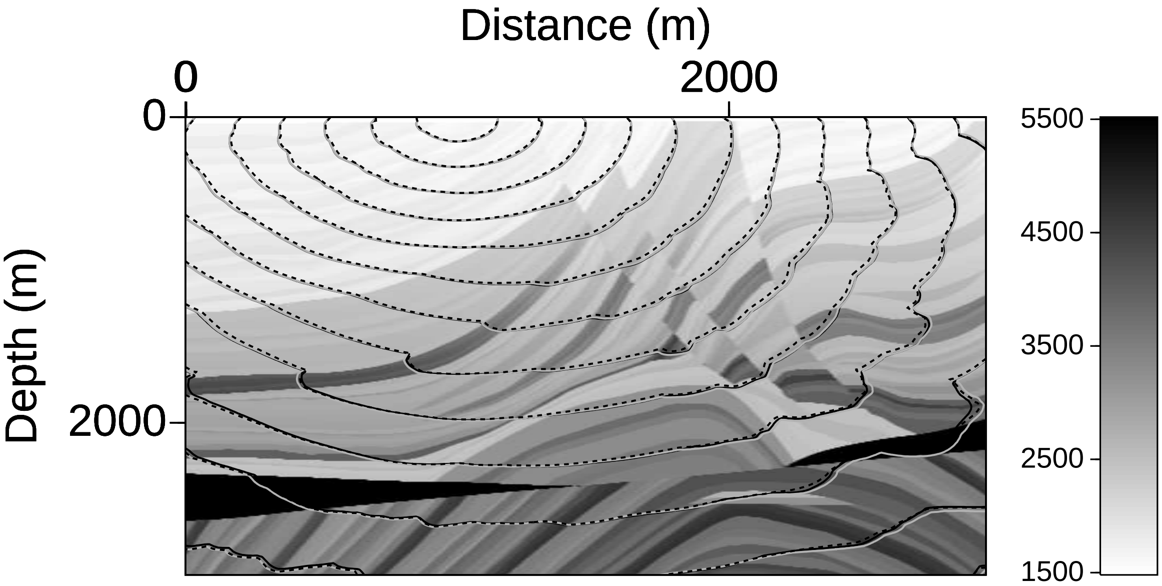

Figure 9: Contours of traveltimes in the Marmousi model caused by a source placed on the surface at a distance of 1000 m. The solid black curves correspond to the solution of the fast marching method implemented using a very fine grid (the reference solution). The dashed curves correspond to use of the fast marching solver in Cartesian coordinates; the gray curves, in polar coordinates. While the solid black and gray curves are close overall, the dashed curve corresponding to the Cartesian coordinate implementation has problems, especially near 45-degree wave propagation. In the background is the Marmousi velocity model described in m/s units. fmpolar-vtcontleg [NR]

fast marching method in Cartesian coordinates (dashed curves), and in polar coordinates (gray curves). The black curves show traveltimes for the more accurate reference solution based on a fine-grid implementation. For vertical and horizontal wave propagation, the various contours practically overlap. At a near 45-degree angular wave propagation, the dashed curves tend to predict faster times than the actual solution. In some regions, the polar coordinate implementation may give the worse results, for example, at distance 3000 m and depth 2000 m, but, overall, fast marching in polar coordinates gives better results than the Cartesian-coordinate implementation as Figure 10 shows. Both methods were executed on a 200MHz Pentuim processor and took less than two seconds of computer time. Figure 11 shows the one-second contour curves superimposed on a snapshot of the finite difference solution of the acoustic

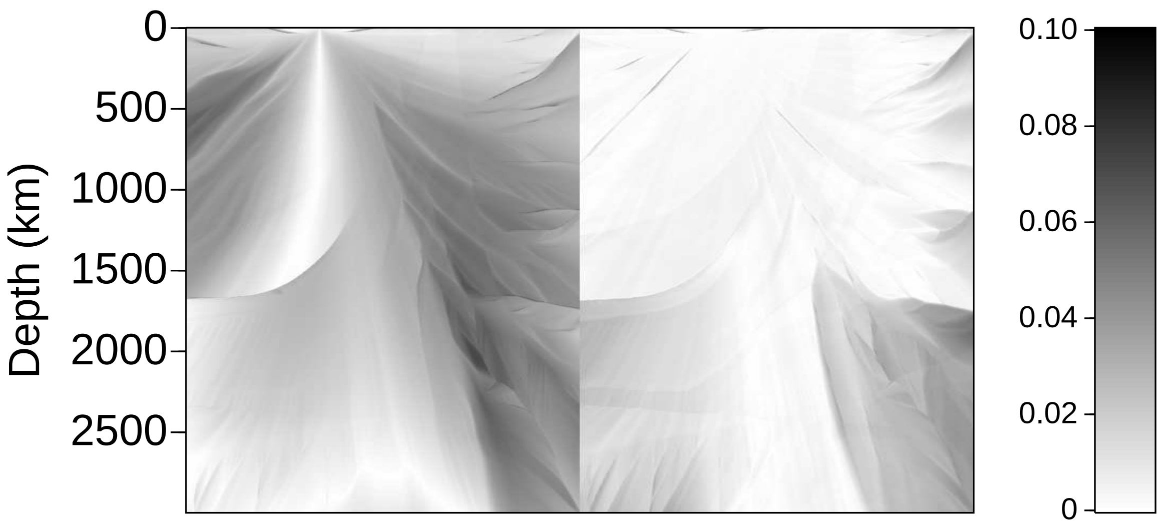

Figure 10: As in Figure 8 (bottom), but for the Marmousi model, the absolute difference, or errors, in seconds between traveltime calculated by the fast marching method using the Cartesian coordinates (left) in comparison to the polar coordinates (right) and the more accurate, finer grid implementation. fmpolar-diffvxzlmarmo1000 [NR]

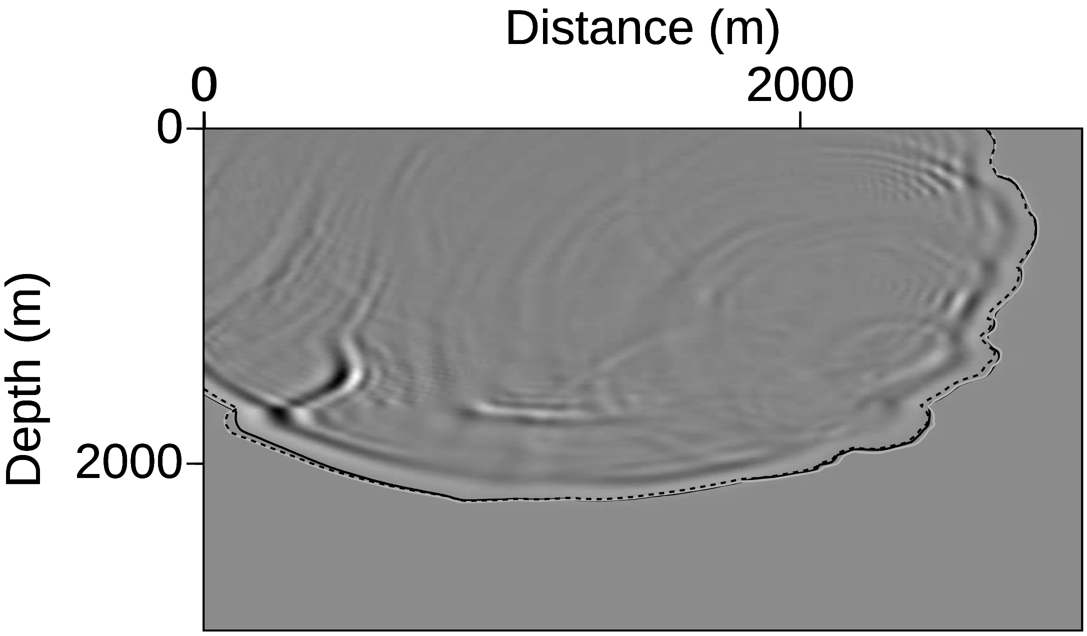

Figure 11: The one-second contour curves from Figure 9 laid over the finite-difference solution of the acoustic wave equation at time one second resulting from a source placed at the surface at 1000 m distance, and ignited as causal ricker wavelet at zero seconds. The contour curves nicely envelope the wave energy for almost all angles. fmpolar-tcont10 [NR]

wave equation at time one second caused by a source at time zero. The source is a ricker wavelet that starts at zero time with the peak value slightly delayed. As a result, the curves act as an envelope for the energy propagation, which in this case, matches most of the more energetic waves. (Later in this section, we show examples in which the most energetic waves are not necessarily the fastest predicted by the eikonal solver, and as a result the eikonal solution does not provide the desired solution.) Figure 12 shows contours of traveltime using

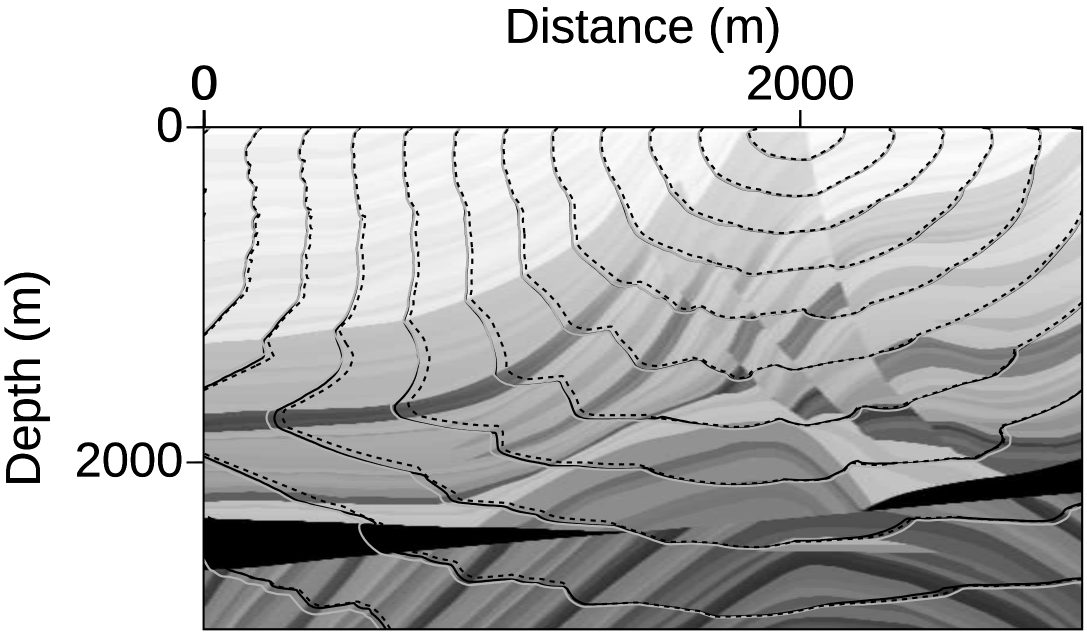

Figure 12: Contours of traveltimes in the Marmousi model resulting from a source placed on the surface above a more complicated region; at distance 2000 m. The solid black curves correspond to the solution of the fast marching method at a very fine grid—the reference solution. The dashed curves correspond to use of the fast marching eikonal solver in Cartesian coordinates; the gray curves, in polar coordinates. While the solid black and gray curves are close overall, the dashed curve corresponding to the Cartesian coordinate solution again has problems near 45-degree wave propagation. fmpolar-vtcont2000 [NR]

the fast marching method in Cartesian coordinates (dashed curves) and in polar coordinates (gray curves). The black curves show traveltimes for the more accurate reference solution. The source is placed above a complicated region of the Marmousi, which causes multi-arrival traveltimes. For vertical and horizontal wave propagation, the various contour curves practically coincide. At near 45-degree angular wave propagation, the dashed curves, which are the result of the Cartesian coordinate implementation, provide quite inaccurate traveltimes (see Figure 13). A closer look, given by Figure 14, reveals how much the Cartesian coordinate implementation hampered the results. The errors are as large as three percent in this area, which is clearly unacceptable. Finer grid coverage in the Cartesian-coordinate implementation will reduce such errors, but at a higher computational cost. At the same computational cost, the polar coordinate implementation of the fast marching method provides far better results even in complex models.

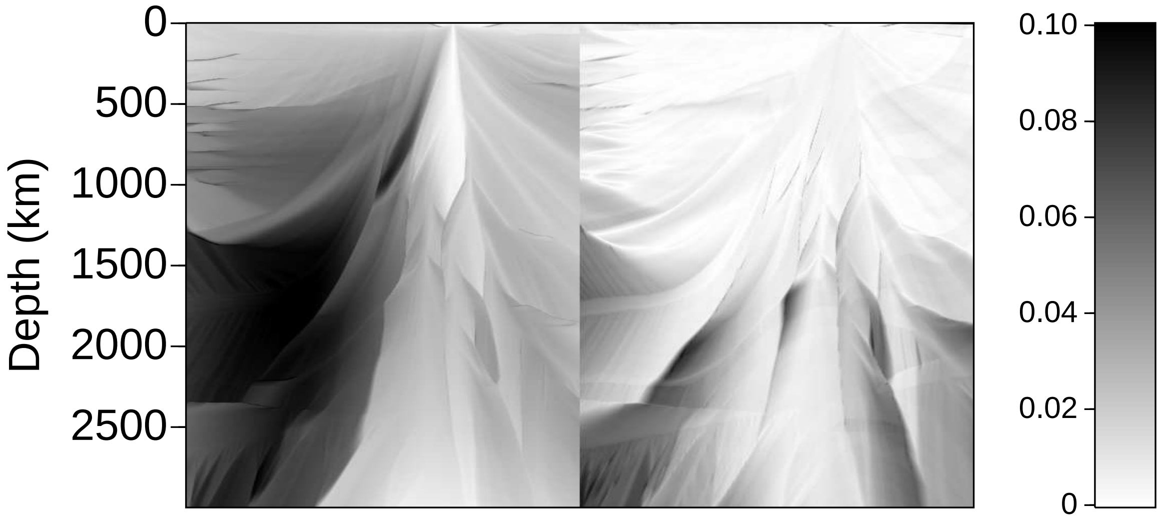

Figure 13: As in Figure 10, but with the source at 2000 m distance, the absolute difference, or errors, in seconds between traveltime calculated by the fast marching method using the Cartesian coordinates (left) and the polar coordinates (right) in comparison with the more accurate, finer grid implementation. fmpolar-diffvxzlmarmo [NR]

Figure 15 shows the 0.5 second contour curves superimposed on a snapshot of the finite difference solution of the acoustic wave equation at time 0.5 seconds caused by a source at time zero. The source is above a complicated area of the Marmousi model and some evidence of the departure of the eikonal solution from the most energetic solutions appears, especially for waves traveling vertically. A snapshot at later time, one second, (Figure 16) shows how much the eikonal solution departed from the most energetic solution. This departure results in less than desirable traveltimes when using the eikonal solution for a process like migration. However, this is the price we pay for such a highly efficient method of calculating traveltimes.

The above examples demonstrate that for an algorithm of the same cost, the accuracy of the polar (or spherical) implementation of the fast marching eikonal solver is far superior to the Cartesian version, even in complex models. Both methods are unconditionally stable with no limitations imposed on the direction of the wavefront propagation. Also, while the

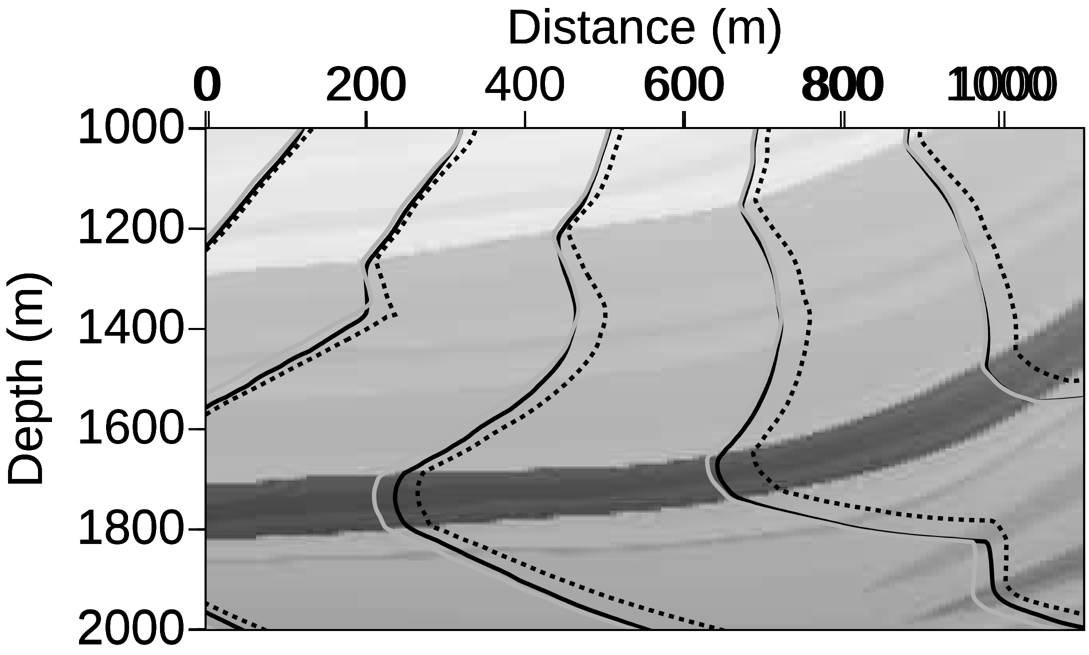

Figure 14: A detail of Figure 12 that demonstrates the accuracy of the polar coordinate solution given by the gray curves as opposed to the Cartesian one given by the dashed curves. The reference solution is indicated by the black curve. fmpolar-vtcont2000c [NR]

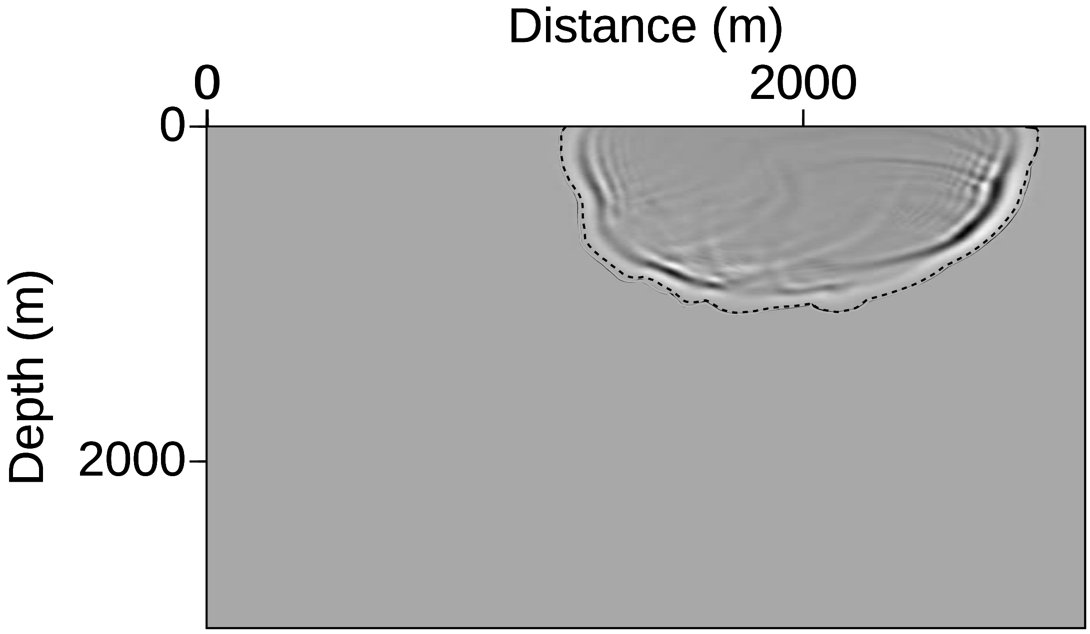

Figure 15: The 0.5 second contour curves from Figure 12 laid over the finite-difference solution of the acoustic wave equation at time 0.5 seconds, caused by a source placed at the surface at 2000 m distance, and ignited as a causal ricker wavelet at zero seconds. The contour curves nicely envelope the wave energy at this early time for almost all angles. fmpolar-tcont5-2000 [NR]

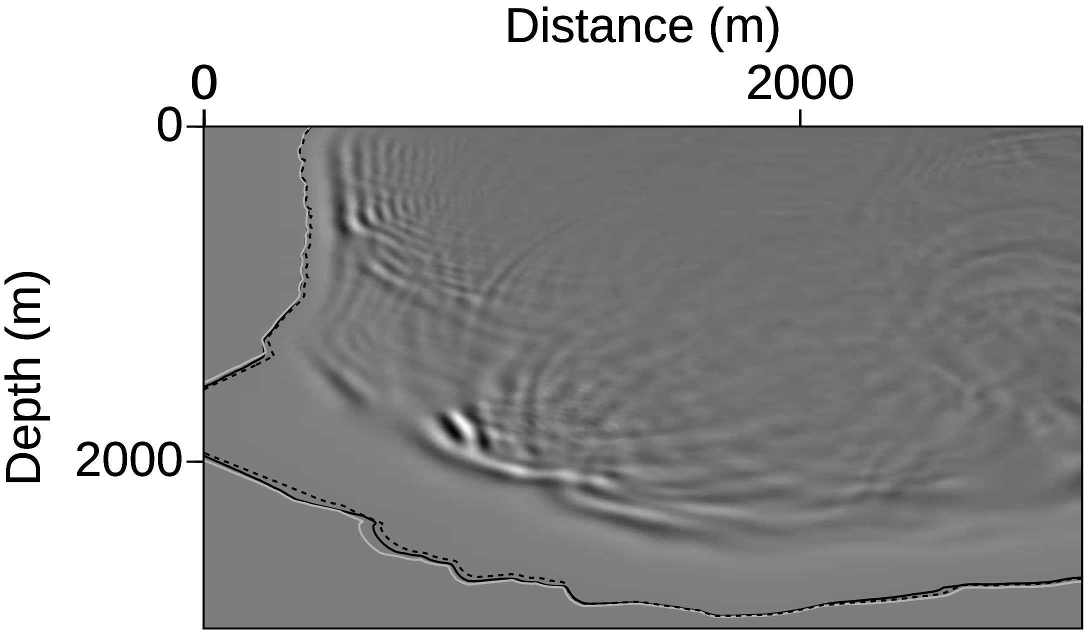

Figure 16: The one-second contour curves of Figure 12 laid over the finite-difference solution of the acoustic wave equation at time one second, caused by a source placed at the surface at 2000 m distance, and ignited as a causal ricker wavelet at zero seconds. In complex media, the fastest propagating energy is not necessarily the most energetic. fmpolar-tcont10-2000 [NR]

eikonal solver may provide reasonable solutions in some areas of the Marmousi models, the complexity of other areas results in less than optimal results.

CONCLUSIONS

Spherical coordinates are the most natural coordinate system in which to solve the eikonal equation in the case of a point source. The radiation of energy from a point source has an overall spherical shape, even in complex media. The fast marching method of solving the eikonal equation, though fast and unconditionally stable, is based on a first-order approximation of the traveltime derivatives. These approximations yield poor results at 45-degree angler wave propagation and at highly curved wavefronts. In Cartesian coordinates, such errors accumulate near the source where the curvature of the wavefront is at its highest. For a point source, both the curvature and the 45-degree angler propagation is reduced in polar coordinates. Even in complex media, wavefronts originating from a point source spend less time traveling diagonally with respect to the polar coordinate system than the Cartesian coordinate system. The Marmousi model is a prime example of the benefits of the polar coordinate system for solving the eikonal equation.

REFERENCES

Cao, S., and Greenhalgh, S. A., 1994, Finite-difference solution of the eikonal equation using an efficient, first-arrival wavefront tracking scheme: Geophysics, **59**, no. 4, 632–643.

Fowler, P. J., 1994, Finite-difference solutions of the 3-d eikonal equation in spherical coordinates: 64th Annual Internat. Mtg., Soc. Expl. Geophys., Expanded Abstracts, 1394–1397.

Geoltrain, S., and Brac, J., 1993, Can we image complex structures with first-arrival traveltime?: Geophysics, **58**, no. 4, 564–575.

Popovici, M., 1991, Finite difference travel time maps: SEP–**70**, 245–256.

Qin, F., Luo, Y., Olsen, K. B., Cai, W., and Schuster, G. T., 1992, Finite-difference solution of the eikonal equation along expanding wavefronts: Geophysics, **57**, no. 3, 478–487.

SchneiderW. A., J., 1993, Robust, efficient upwind finite-difference traveltime calculations in 3-d: 63rd Annual Internat. Mtg., Soc. Expl. Geophys., Expanded Abstracts, 1036–1039.

Sethian, J. A., and Popovici, A. M., 1997, Three-dimensional traveltime computation using the fast marching method: submitted to Geophysics.

Sethian, J. A., 1996, Level set methods: Cambridge University Press, New York.

van Trier, J., and Symes, W. W., 1991, Upwind finite-difference calculation of traveltimes: Geophysics, **56**, no. 6, 812–821.

Vidale, J. E., 1990, Finite-difference calculation of traveltimes in three dimensions: Geophysics, **55**, no. 5, 521–526.