Stanford Exploration Project, Report 70, December 18, 1997, pages 1–261

Finite-difference traveltime maps

Alexander M. Popovici [1]

keywords: *traveltime, velocity, algorithm, three-dimensional*

ABSTRACT

The finite-difference traveltime algorithm presented by Van Trier and Symes (1989; 1990) uses a first-order upwind finite-difference scheme described by Engquist and Osher (1980). The Engquist-Osher scheme has an underlying physical minimization condition important for maintaining the stability of the algorithm; it also introduces approximations in the estimation of the traveltime field in order to obtain greater computational speed at the expense of accuracy. The error can be reduced in a slower algorithm. I extend the algorithm in 3-D and show results for a constant velocity medium and slowly varying velocity medium. I find there are greater stability problems in 3-D than in 2-D for rapidly varying velocity models.

INTRODUCTION

Given the eikonal equation, there is an abundance of raytracing methods in literature. Finite-difference travel time schemes are one class of methods which calculate the first arrival of the wave-front from the eikonal equation. The finite-difference scheme introduced by Vidale (1988) has an explicit minimization condition (i.e., the time along a new computation front is calculated using the minimum time along the previous computation front), important for keeping the scheme stable. The Van Trier and Symes (1989; 1990) algorithm has a similar minimization condition which is applied locally to three points of the four point finite-difference stencil. I analyze their first-order upwind finite-difference scheme and show the implicit minimization condition and the errors introduced in favor of greater computational speed. I extend the algorithm to a 3-D medium and show results of traveltime fields for constant and slowly varying velocity media. The 3-D algorithm is much more susceptible to numerical instability when rapid variations in velocity are present.

[1]**email:** not available

THE BASIC FINITE-DIFFERENCE SCHEME

The Engquist-Osher scheme used by Van Trier and Symes is an interesting balance between computational speed and precision. Although the scheme introduces approximations in the calculation of the travel time field that can be adverse for accurate calculations of differential operators (which involve the local derivative of the travel-time field), the scheme is more stable than a straightforward finite-difference scheme. In spite of all the instability problems one may encounter using this algorithm, all of the simpler alternatives I have tried are stable only for much smaller grid sizes.

The fundamental algorithm described by Van Trier and Symes (1989) is based on the eikonal equation

$$u^2 + v^2 = s^2 \tag{1}$$

where

$$\begin{cases} u = \frac{\partial t(x,z)}{\partial x} \\ v = \frac{\partial t(x,z)}{\partial z} \end{cases}$$

and where $s(x, z)$ is the 2-dimensional slowness model and $t(x, z)$ is the travel time field. The second equation used is the equality of the partial derivatives of the fields $u(x, z)$ and $v(x, z)$,

$$\frac{\partial u}{\partial z} = \frac{\partial^2 t}{\partial x \partial z} = \frac{\partial v}{\partial x}. \tag{2}$$

In cylindrical coordinates the eikonal equation becomes

$$\frac{u^2}{r^2} + v^2 = s^2 \tag{3}$$

where

$$\begin{cases} u = \frac{\partial t(r,\theta)}{\partial \theta} \\ v = \frac{\partial t(r,\theta)}{\partial r} \end{cases}$$

and the mixed partial derivatives are

$$\frac{\partial u}{\partial r} = \frac{\partial^2 t}{\partial \theta \partial r} = \frac{\partial v}{\partial \theta}. \tag{4}$$

The finite difference implementation of equations (1), (2) and in cylindrical coordinates (3), (4) is based on advancing the computational front for the functions $u(x, z)$ and $v(x, z)$. The traveltime field $t(x, z)$ is found subsequently by integrating the function $v(r, \theta)$ with respect to r. In cylindrical coordinates the scheme is based on using equation (4) to compute the values of $u(r, \theta)$ on a new computational front of constant radius, by using the values of $u(r, \theta)$ and $v(r, \theta)$ from the previous computational front. Equation (4) becomes

$$u(r + \Delta r, \theta) = u(r, \theta) + \frac{\Delta r}{\Delta \theta} \Delta v(r, \theta). \tag{5}$$

Starting with a constant velocity condition in the immediate vicinity of the source location ($u(r,\theta) = 0, v(r,\theta) = s(r,\theta)$), we can advance the computational front using the finite-difference equation (5). Once the values of $u(r,\theta)$ are known, the values of $v(r,\theta)$ can be computed using the eikonal equation:

$$v(r,\theta) = \sqrt{s^2(r,\theta) - \frac{u^2(r,\theta)}{r^2}}. \tag{6}$$

An insight into the Engquist-Osher scheme applied to Van Trier's algorithm

The practical implementation of the above formulation is centered around the problem of advancing the computational front. The Engquist-Osher scheme provides a way to compute Δv by imposing a time minimization condition along three points of the finite-difference stencil. The scheme calculates $\frac{\Delta v}{\Delta \theta}$ to approximate the partial derivative (with respect to θ), of the function $v(r,\theta)$, by using the values of $v(r,\theta)$ in points of minimum traveltime; $v(r,\theta)$ varies only in θ along the constant radius computational front. Given three consecutive points on the computational front with constant radius, the values of the functions $u(r,\theta)$ and $v(r,\theta)$ vary only in the variable θ. Each function will have the values u_{j-1}, u_j, u_{j+1} and v_{j-1}, v_j and v_{j+1} respectively, in the three points of the stencil. From equation (6) we can write $v(r,\theta)$ as a function of $u(r,\theta)$:

$$F(u) = v(r,\theta) = \sqrt{s^2(r,\theta) - \frac{u^2(r,\theta)}{r^2}}.$$

The Engquist-Osher scheme computes Δv as

$$\Delta v = F(min(u_{j+1},\bar{u})) - F(min(u_j,\bar{u})) + F(max(u_j,\bar{u})) - F(max(u_{j-1},\bar{u})), \tag{7}$$

where $\bar{u} = 0$, at the point where $\frac{\partial t}{\partial \theta} = 0$. For this case the value of $F(\bar{u}) = v = slowness$ from equation (6). Because the values of u_{j-1}, u_j and u_{j+1} are compared against zero ($\bar{u} = 0$), the scheme needs only the sign of the function u in the three points of the stencil. Equation (7) allows for eight cases as a function of the positive or negative values of u_{j-1}, u_j and u_{j+1} in the three points of the stencil. The eight cases are shown in Figure 1.

The calculation of Δv is done for locations where the value of the traveltime function $t(r,\theta)$, is minimum. From Figure 1 one can see that only cases 1 and 5 will actually give a first order correct value. In both cases the values for the function v are chosen from the three points of the stencil where the value of the time is minimum. While for cases 1 and 5 the accuracy of the scheme is unquestionable, for the rest of the cases some approximations are introduced. The other six cases also calculate the values of Δv using the points where the value of the time is minimum. However, the value of Δv is divided by a constant $\Delta\theta$, even though the function $v(r,\theta)$ is estimated over a different interval $\Delta\theta$. A potentially more accurate algorithm would calculate the exact value of $\Delta\theta$ for each intermediate case. The algorithm can be designed to

Figure 1: Eight possible cases for the Engquist-Osher scheme. The thick vertical bars represent the sign of $u = \frac{\partial t}{\partial \theta}$, the continuous line represents the values of the traveltime over the three points of the stencil, the dashed vertical lines represent the location where $\bar{u} = \frac{\partial t}{\partial \theta} = 0$; the dots on the time line show the coordinates for the function $v(r, \theta)$ which contribute to the difference Δv. On the right side are the values for the computed Δv for each case.

calculate the locations of the minimum travel time in the three point stencil interval and the span over the θ axis, necessary to divide the value Δv.

3-D ALGORITHM

In 3-D the eikonal equation is

$$u^2 + v^2 + w^2 = s^2 \tag{8}$$

where

$$\begin{cases} u = \frac{\partial t(x,y,z)}{\partial x} \\ v = \frac{\partial t(x,y,z)}{\partial y} \\ w = \frac{\partial t(x,y,z)}{\partial z} \end{cases}$$

For a spherical-coordinates system

$$w^2 + \frac{v^2}{r^2} + \frac{u^2}{r^2 \sin\phi^2} = s^2 \tag{9}$$

where

$$\begin{cases} w = \frac{\partial t(r,\theta,\phi)}{\partial r} \\ v = \frac{\partial t(r,\theta,\phi)}{\partial \phi} \\ u = \frac{\partial t(r,\theta,\phi)}{\partial \theta} \end{cases}$$

The cross derivative equation (4) is transformed into the spherical coordinates system

$$\begin{cases} \frac{\partial u}{\partial r} = \frac{\partial w}{\partial \theta} \\ \frac{\partial v}{\partial r} = \frac{\partial w}{\partial \phi} \end{cases} \tag{10}$$

The finite-difference equivalent of equation (5) is the system

$$\begin{cases} u(r + \Delta r, \theta, \phi) = u(r, \theta, \phi) + \frac{\Delta r}{\Delta \theta} \Delta w(r, \theta, \phi) \\ v(r + \Delta r, \theta, \phi) = v(r, \theta, \phi) + \frac{\Delta r}{\Delta \phi} \Delta w(r, \theta, \phi) \end{cases} \tag{11}$$

which is used to advance the stencil for a new radial increment. Once the values of the functions $u(r, \theta, \phi)$ and $v(r, \theta, \phi)$ are known on the spherical front with constant radius $(r + \Delta r)$, the third function $w(r, \theta, \phi)$ can be calculated using the eikonal equation

$$w(r + \Delta r, \theta, \phi) = \sqrt{s^2(r + \Delta r, \theta, \phi) - \frac{v^2(r + \Delta r, \theta, \phi)}{(r + \Delta r)^2} - \frac{u^2(r + \Delta r, \theta, \phi)}{(r + \Delta r)^2 \sin^2 \phi}}. \tag{12}$$

The value of the traveltime is found by integration:

$$t(r+\Delta r,\theta,\phi) = t(r,\theta,\phi) + \frac{\Delta r}{2}[w(r+\Delta r,\theta,\phi) + w(r,\theta,\phi)].$$

In equation (11), the Engquist-Osher scheme is applied twice, once for calculating the values of Δw across three points of consecutive values of θ, and second for calculating the values of Δw across three consecutive values of ϕ. The computational front advances in spherical shells, and on each shell the computations advance a circle at a time. The angle θ is the horizontal angle while the angle ϕ is the vertical angle. The Engquist-Osher scheme is applied along each three consecutive points on the circle with constant vertical angle ϕ to determine Δu from the equation $\Delta u = \frac{\Delta r}{\Delta \theta}\Delta w$. For each circle of constant vertical angle ϕ the Engquist-Osher scheme is applied for three points $(\phi - \Delta\phi)$, ϕ and $(\phi + \Delta\phi)$, which are perpendicular on the circle in the (r,θ,ϕ) coordinates. The scheme is completely vectorizable.

3-D TRAVELTIME MAPS

The first example shown in Figure 2 represents sections through a 3-D traveltime field for a constant velocity of 2 km/s. The $100 \times 100 \times 100$ grid is evenly sampled laterally and in depth, with a sample interval of 10 m. The source is situated at the surface, in the middle of the plane. Figure 2a and 2b represent vertical orthogonal planes passing through the source. The vertical plane in Figure 2d is offset with a distance (X=300 m) from the source. Figure 2c shows the values of the traveltime field on a horizontal surface passing through the middle of the cube. Figure 3 displays the difference between the analytical solution and the computed traveltime field for each panel shown in Figure 2. The distribution of the errors is shown through lines of constant value. The errors accumulate with distance from the source. The values of the absolute error on each surface (of order 10^{-6}) are shown in Figure 4.

The constant velocity model is a poor example to examine a finite-difference algorithm based on the Engquist-Osher scheme, as the functions $u(r,\theta,\phi)$ and $v(r,\theta,\phi)$ are zero on spherical shells of constant radius r. Therefore the errors in the constant velocity case are merely due to numerical integration and interpolation from cartesian to spherical coordinates.

Figure 5 shows results for a traveltime field in a depth variable velocity medium. the computations are done on the same $100 \times 100 \times 100$ grid with a spacing of 10 m. The three panels shown in Figures 5a, 5b and 5c represent vertical planes situated at increasing distance from the source. Figure 5d displays the depth variable velocity model. The 3-D algorithm is an order of magnitude more unstable than the 2-D algorithm for comparable grid parameters in spherical, and cylindrical coordinates respectively. The instability problems disappear for smaller grid intervals in spherical coordinates, but one is limited by the amount of computer memory which can be allocated for an increasing number of grid points. The absolute error is not negligible at the sides of the panel, as can be seen from Figure 6c and 6d. Maximum values for

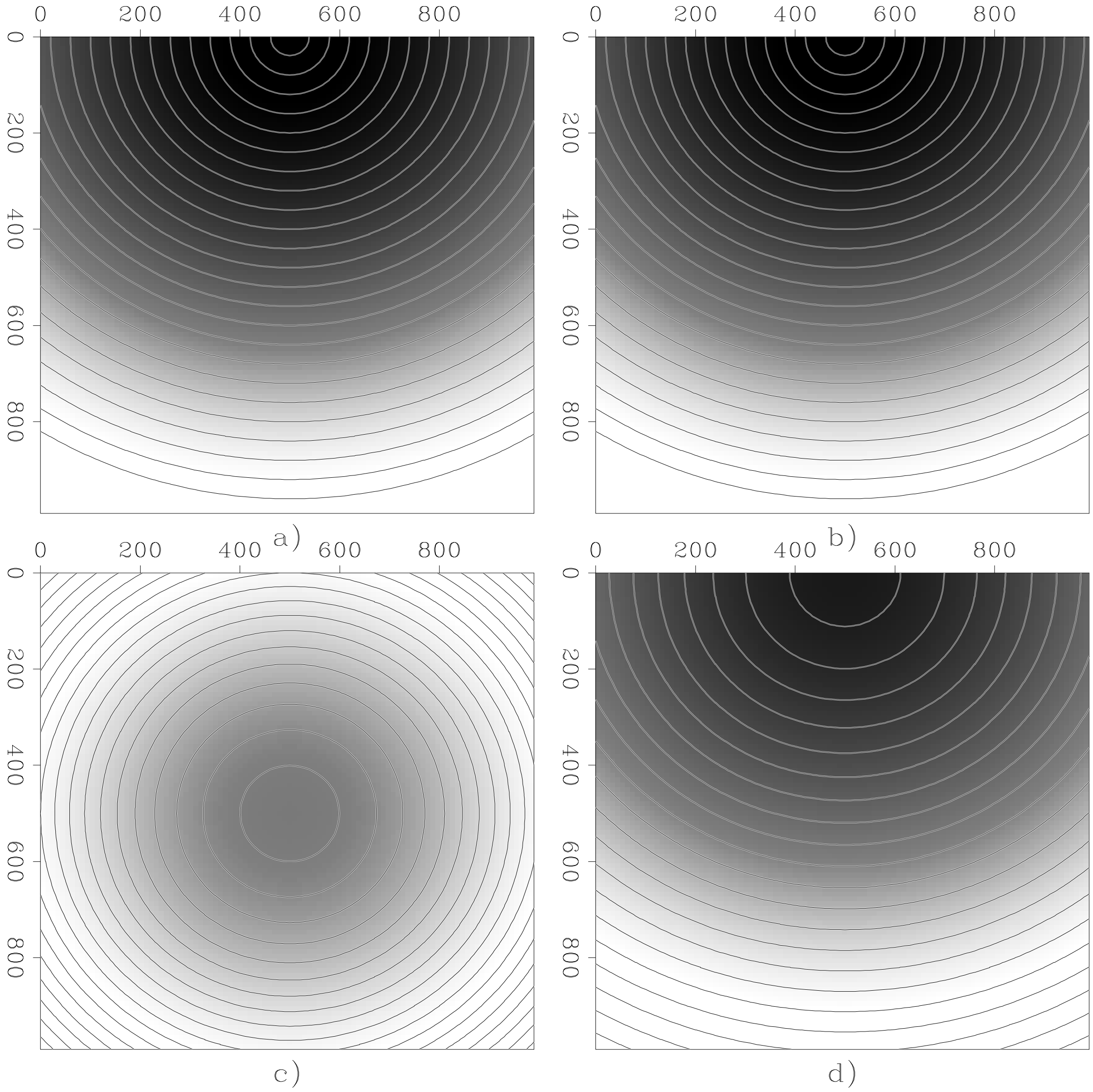

Figure 2: Traveltime map for a constant 3-D velocity model (1000 m each side) with the source at the surface. a) Vertical section through the middle of the cube for constant X. b) Vertical section through the middle of the cube for constant Y. c) Horizontal section through the middle of the cube for constant Z. d) Vertical section off the middle of the cube for constant X (X=800 m). mihai2-Ray3Dmc [NR]

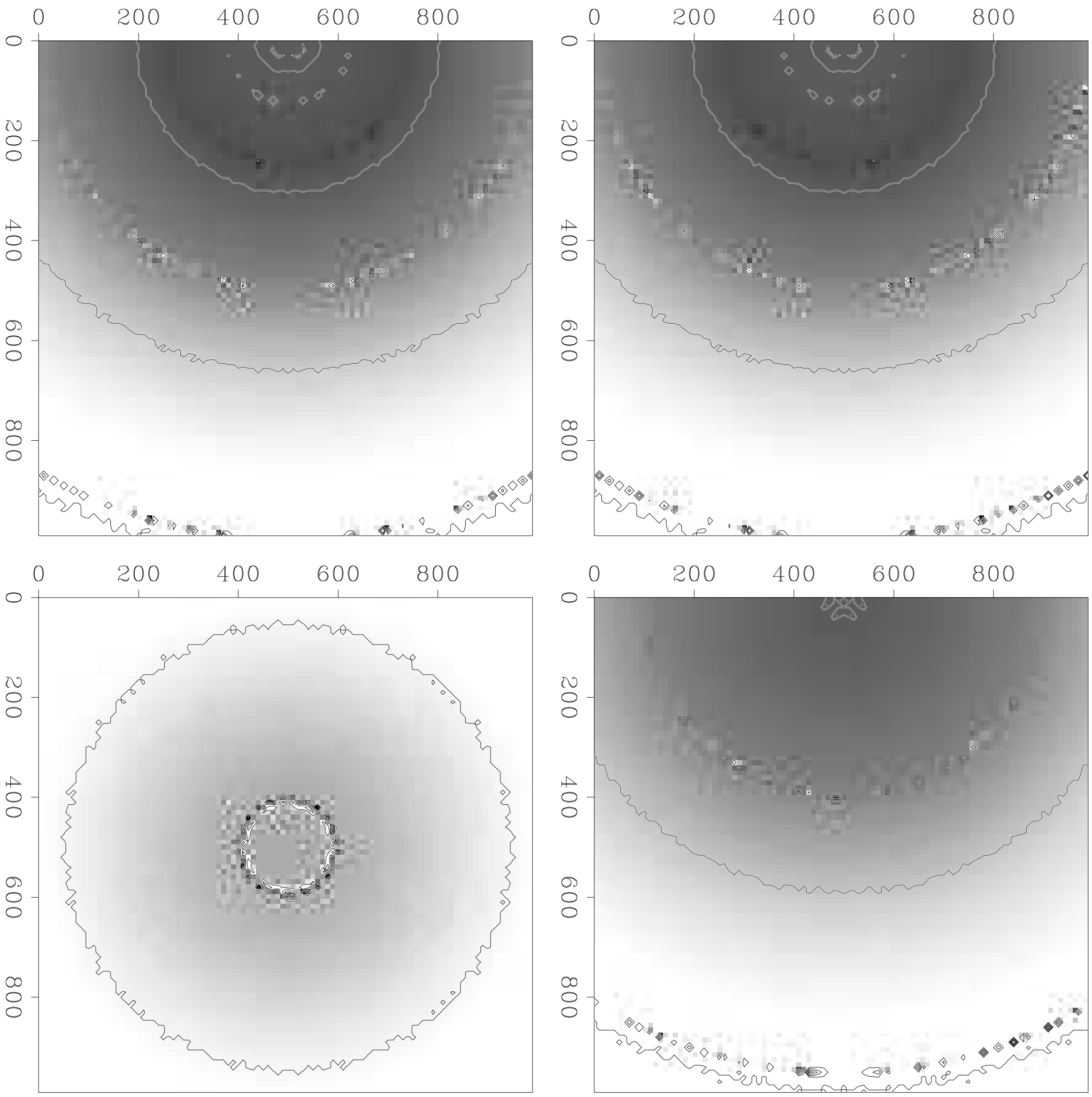

Figure 3: Difference between the computed traveltime map and the analytical solution for a constant velocity model. Each figure represents the difference between the analytical solution and the corresponding panel in Figure 2. The error accumulates with distance from the source. mihai2-Diff3Dmc [NR]

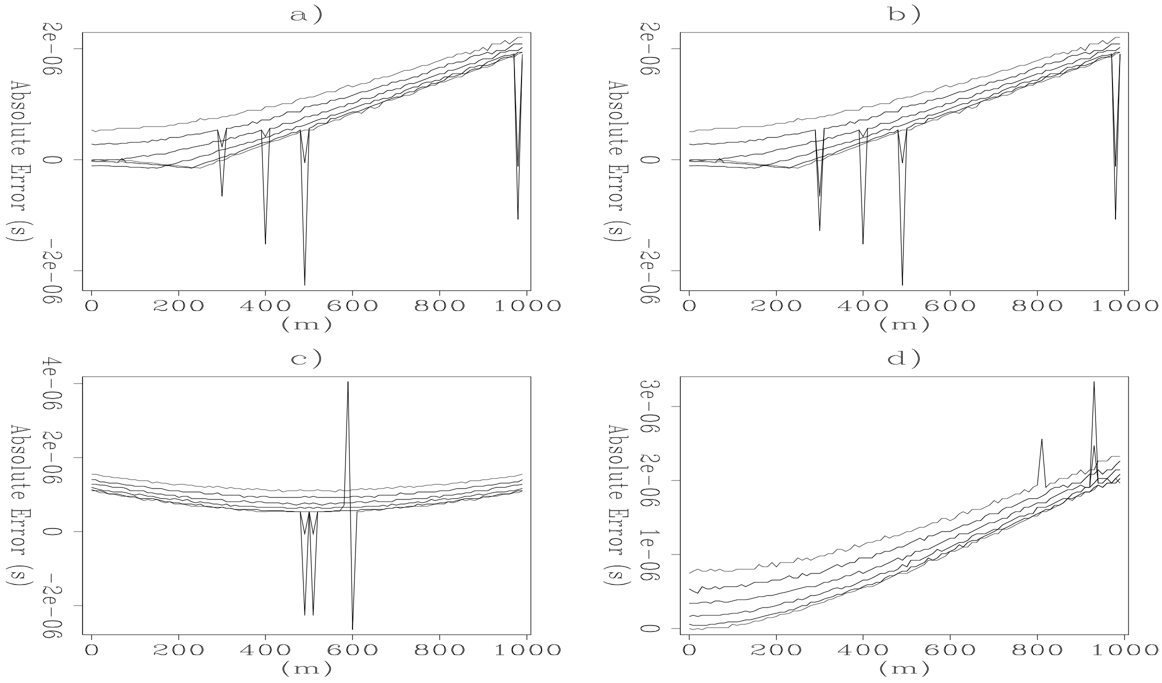

Figure 4: Errors in the finite-difference calculations. Each panel shows several curves which represent the absolute error across the corresponding sections in Figure 3. mihai2-GraphDiffmc [NR]

the error appear on the edges of the model studied, at higher values of the traveltime field. The traveltimes in the 3-D model tend to be higher, as the difference between the 3-D section and the 2-D section is always positive.

Numerical instability problems

The present drawback to the Van Trier and Symes algorithm is that for a given grid size the algorithm is numerically unstable. The problem is that presently one cannot determine a priori a stable grid spacing. The program will execute without any evidence of numerical instability. Only by examining the results one can determine if numerical instability has occurred. By reducing the grid spacing one can resolve this problem. Further work is necessary to detect automatically numerical instability or determine the stability conditions as a function of the velocity model and grid spacing. Detecting numerical instability and correcting it at run-time will be an important improvement. In 3-D the instability problem is more serious compared to 2-D. The velocity models I tested need to have smooth velocity variations in order to run in the presently available computer memory without presenting numerical instability.

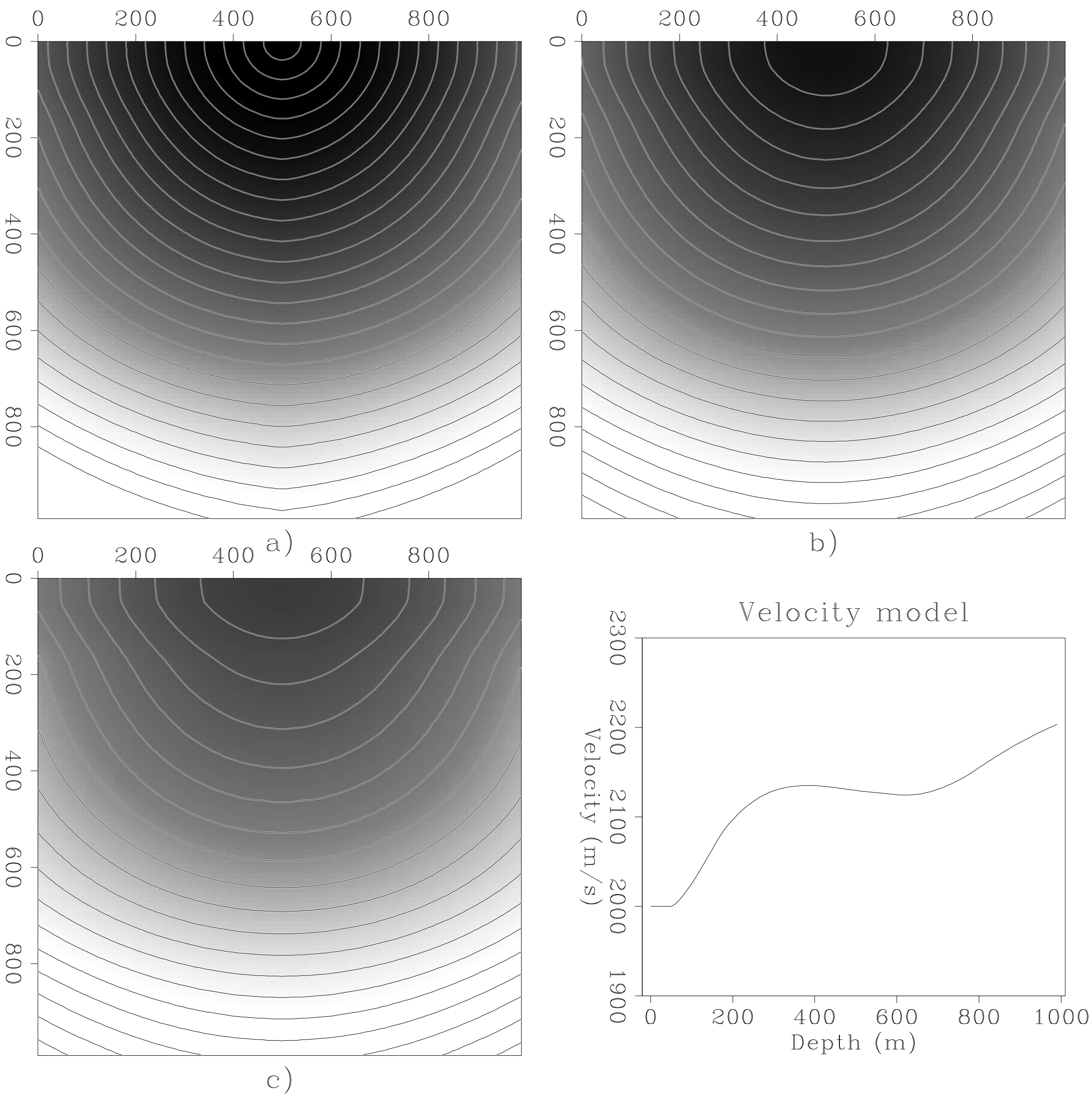

Figure 5: Traveltime map for a 3-D depth variable velocity model (1000 m each side) with the source in the middle of the top surface. a) Vertical section through the source. b) Vertical section with offset (X=250 m) from the source. c) Vertical section with offset (X=450 m) from the source. d) Velocity model. mihai2-Ray3Dm1 [NR]

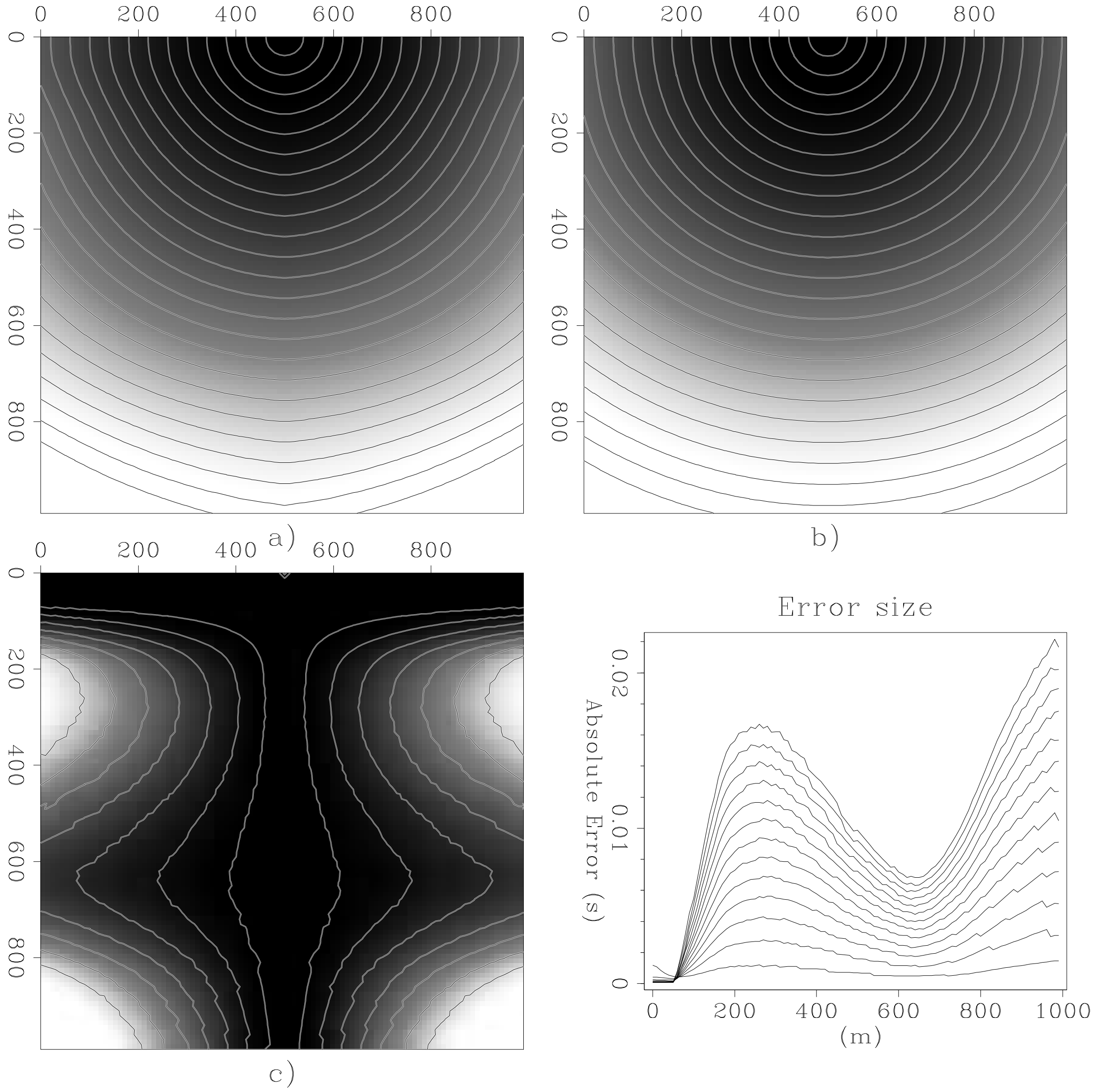

Figure 6: Comparison between 3-D and a 2-D finite-difference traveltime map for the depth variable velocity model shown in Figure 5d. a) Vertical section through the source in the 3-D model. b) 2-D traveltime field for the same depth variable velocity model. c) Difference between the 3-D traveltime field and the 2-D traveltime field. d) Absolute errors for the 3-D vs. 2-D difference. mihai2-Ray3Ddiff [NR]

CONCLUSION

I have presented an insight in the Engquist-Osher minimization scheme used in Van Trier's finite-difference traveltime calculations. I explained the approximations made by the scheme and the probable source of numerical instability. I extended Van Trier's algorithm to the 3-D case and showed examples of finite-difference traveltime maps for 3-D models with constant velocity and relatively mild variations in velocity.

REFERENCES

Engquist, B., and Osher, S., 1980, Stable and entropy satisfying approximations for transonic flow calculations: Math. Comp., **34**, 45–75.

Van Trier, J., 1989, Finite difference calculation of travel times: SEP–**61**, 103–113.

Van Trier, J., and Symes, W., 1990, Upwind finite difference calculation of traveltimes: SEP–**65**, 41–58.

Vidale, J., 1988, Finite-difference calculation of travel times: Bulletin of the Seismological Society of America, **78**, 2062–2076.