GEOPHYSICS, VOL. 56, NO. 1 (JANUARY 1991); P. 59-67, 12 FIGS.

Shortest path calculation of seismic rays

T. J. Moser*

ABSTRACT

Like the traveling salesman who wants to find the shortest route from one city to another in order to minimize his time wasted on traveling, one can find seismic raypaths by calculating the shortest traveltime paths through a network that represents the earth. The network consists of points that are connected with neighboring points by connections as "long" as the traveltime of a seismic wave along it. The shortest traveltime path from one point to another is an approximation to the seismic ray between them, by Fermat's principle. The shortest path method is an efficient and flexible way to calculate the raypaths and traveltimes of first arrivals to all points in the earth simultaneously. There are no restrictions of classical ray theory: diffracted raypaths and paths to shadow zones are found correctly. There are also no restrictions to the complexity or the dimensionality of the velocity model. Furthermore, there are no problems with convergence of trial raypaths toward a specified receiver nor with raypaths with only a local minimal traveltime. Later arrivals on the seismogram, caused by reflections on interfaces or by multiples, can be calculated by posing constraints to the shortest paths. The computation time for shortest paths from one point to all other points of the networks is almost linearly dependent on the number of points. The accuracy of the results is quadratically dependent upon the number of points per coordinate direction and the number of connections per point.

INTRODUCTION

There are two traditional methods to compute seismic raypaths between two points in the earth: shooting and bending (Julian and Gubbins, 1977). Shooting tries to find raypaths leaving one source point by solving the differential equations that follow from ray theory for different initial conditions until the trial ray arrives at the preassigned point. Bending has Fermat's principle as a starting point; it tries to find a raypath between two points by searching the minimal traveltime path between them.

Both methods have serious drawbacks. By shooting a fan of rays leaving the source, one can obtain an impression of the wave field. However, convergence problems are known to occur frequently, especially in three dimensions. Also, shooting will not find diffracted raypaths or raypaths in shadow zones, where ray theory breaks down. With bending, one can find every raypath satisfying Fermat's principle, even a diffracted one, but only for one source-receiver pair at a time, and it is not certain whether the path has an absolute minimal traveltime or only a local minimal traveltime. These drawbacks result in low efficiency from the methods and incomplete results. The problems are even more severe in three-dimensional than in two-dimensional ray tracing.

In this paper, a method is presented that avoids the disadvantages of shooting and bending. It uses an idea of Nakanishi and Yamaguchi (1986) to approximate raypaths by the shortest paths in networks as a starting point. I then investigate methods to improve the efficiency of the search for the minimum time path and show how it is possible to treat reflected arrivals in the same way.

Network theory and shortest paths in networks are an abstract formulation of problems that appear in many different branches of science and technology. They are usually of discrete nature: there are a finite number of objects, and the exact solution of the problem can be found in a finite number of steps. One example is the road map, where the network consists of cities and connections between them; one can ask for the shortest path from one city to another, or the shortest closed circuit along all cities, and so on. A wide variety of techniques from network theory and especially from the theory on shortest paths is available; see Deo and Pang (1984) for a review of the literature.

One advantage of the shortest path method is that all shortest paths from one point are constructed simulta-

Presented at the 59th Annual International Meeting, Society of Exploration Geophysicists.
Manuscript received October 20, 1989; revised manuscript received July 9, 1990.
*Department of Theoretical Geophysics, University of Utrecht, Budapestlaan 4, Box 80.021, 3508 TA Utrecht, The Netherlands.

neously. This follows from the nature of shortest path algorithms: calculating one path costs just as little computation time as calculating all paths. It can be applied for instance in the simulation of common-shotpoint gathers.

Although a search that ignores the differential equations and even Snell's law does not seem to be efficient at first sight, seismic ray tracing can make direct, practical use of the increased efficiency of shortest path algorithms. Efficiency has been improved by several orders of magnitude in the last 30 years by introducing sophisticated data structures (Gallo and Pallottino, 1986). The most efficient one for ray tracing can be selected by a simple comparison of the available algorithms. The shortest path method is designed to find a good approximation to the globally minimal traveltime and traveltime path. It is therefore especially fit for applications in traveltime tomography. Later arrivals on a seismogram can only be computed if they can be formulated as constrained shortest paths. For instance, the shortest paths constrained to visit one point of a set of points that form a scatterer or interface approximate diffracted and reflected raypaths.

Finally, from the abstract structure of a network there is no notion of dimensionality of the space, so two-dimensional and three-dimensional ray tracing are possible with the same algorithms.

SHORTEST PATHS IN NETWORKS AND SEISMIC RAYPATHS

Before the more theoretical analysis, I illustrate the possibilities of networks to represent approximate raypaths. One important property of a seismic raypath is given by Fermat's principle: the raypath is a spatial curve along which the traveltime is stationary. The construction of a ray between a seismic source and a receiver can be based on this principle. One could enumerate all curves connecting the source and the receiver and look for the minimum traveltime curve. This is usually done by bending an initial guess of the raypath so that the traveltime along it is decreased, until stationarity is achieved. The analogy between a seismic raypath and a shortest path in a network provides an alternative use of Fermat's principle.

In this approach, the relevant part of the earth is represented by a large network consisting of points connected by arcs. Each point or node is connected with a restricted number of points in its neighborhood but not with points that lie farther away. It is therefore possible to travel from one node to another via the connections. The network indeed resembles a three-dimensional road map. As on a road map, the connections between nodes have lengths. This length is to be understood as a weight of the connection; for example, in the application to seismic ray tracing, it is the traveltime of a seismic wave between the two nodes. In seismic ray tracing, the connection will have equal length for both directions along it, by virtue of the reciprocity principle. In other applications of network theory, the weight can be the electrical resistance in an electrical network, the cost in an economical decision tree, or whatever quantity must be minimized in a discretized medium.

When the length of an arc is chosen equal to the traveltime of a seismic wave, one can hope that the shortest traveltime path between two nodes approximates the seismic raypath between them. This is likely when the nodes of the network are distributed such that almost any raypath can be approximated by paths through the network. To this end, regular distributions of the nodes and of the connections between them are introduced. Such a distribution is required not only to give reasonable approximations to seismic raypaths, but also results in a considerable saving of memory space. Two organizations of networks are used in this paper to illustrate the shortest path method. Both illustrations are given only in two dimensions, although the network theory does not impose any restriction on the dimensionality of the space. All quantities are made dimensionless; the horizontal distance x and the depth z range from 0 to 100.

In the first example, taken from Nakanishi and Yamaguchi (1986), the nodes are distributed regularly on the boundaries of rectangular cells in which the propagation velocity of seismic waves is constant (Figure 1). Two nodes are only connected when there is no cell boundary between them. The traveltime between two connected nodes is defined as their Euclidian distance multiplied with the slowness of the

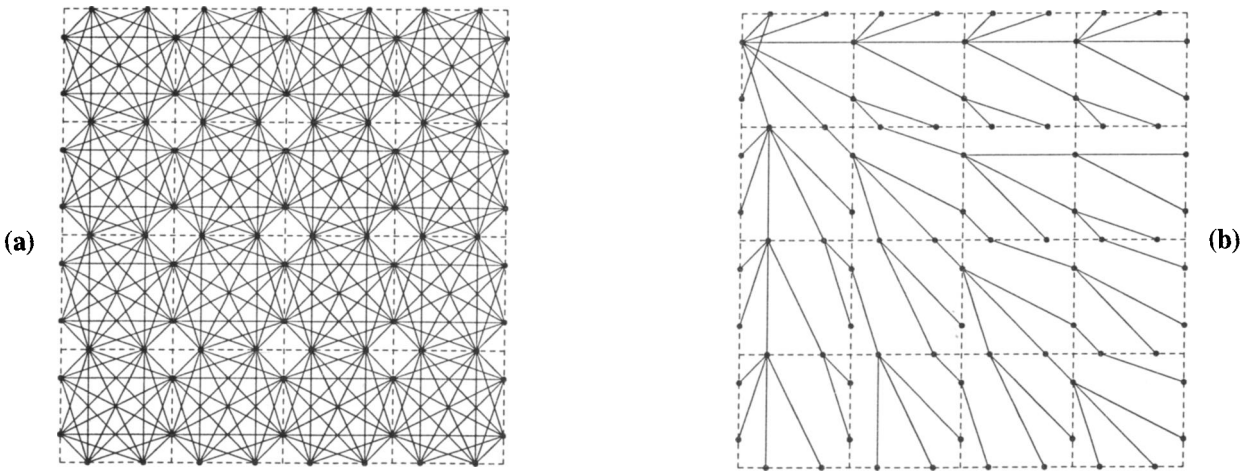

FIG. 1. Cell organization of a network. (a) Dashed lines: cell boundaries. Black circles: nodes. Solid lines: connections. (b) Shortest paths from one node to other nodes in a homogeneous model.

cell in between. Figures 2a and 2b show the shortest paths in two networks. The model of Figure 2a has a constant velocity of 1.0, so that the shortest paths approximate straight lines, and the model of Figure 2b has a (dimensionless) velocity distribution $c = 1.0 + 0.01z$, so that the shortest paths approximate circular raypaths. The cell organization of networks is particularly suitable for the application to seismic tomography, as in Nakanishi and Yamaguchi (1986).

The second example uses a rectangular grid of nodes, of which each node is connected with the nodes in a neighborhood of it. The seismic velocity or slowness is sampled at the node locations. The traveltime between two connected nodes is defined as their Euclidian distance multiplied with the average slowness of the two nodes. This organization facilitates the drawing of contours of equal traveltimes and velocities and, more important, the introduction of interfaces. Figure 3a shows a rectangular grid of 5×5 nodes, each one connected with at most eight neighbors. In Figure 3b the shortest paths from the upper left node are plotted for a homogeneous model.

The shortest paths in more general media follow expected patterns. In Figure 4a a medium with velocities ranging from 1.0 to 2.0 is represented by a network of 50×50 nodes and at most 48 connections per node. The shortest paths from one node at the left side of the model are plotted. It can be seen that they converge in high-velocity regions and try to avoid low-velocity zones.

The traveltimes along shortest paths to the 50 nodes at the right side of the model can be compared with traveltimes calculated with the shooting method for seismic ray tracing. The raypaths in Figure 4b have been calculated by a numerical solution of the ray equation for 50 fixed initial directions with a fourth-order Runge-Kutta scheme (Stoer and Bulirsch, 1980). No attempt has been made to reach a preassigned receiver point; with a standard bisection method this will usually take about five times more computation time. The raypaths are discretized into 25 points; the traveltimes along them are correct up to 1 percent compared to reference rays, consisting of 100 points. In this setting, the shortest path calculations and the shooting calculations take roughly the same computation time and the traveltimes are correct up to 1 percent for both methods. Although the accuracy and the efficiency may be of the same order, the results of the calculations are different. Intersections of raypaths and triplications in the traveltime curves are found correctly by the shooting method, but the shortest path method will give only the first arrivals. On the other hand,

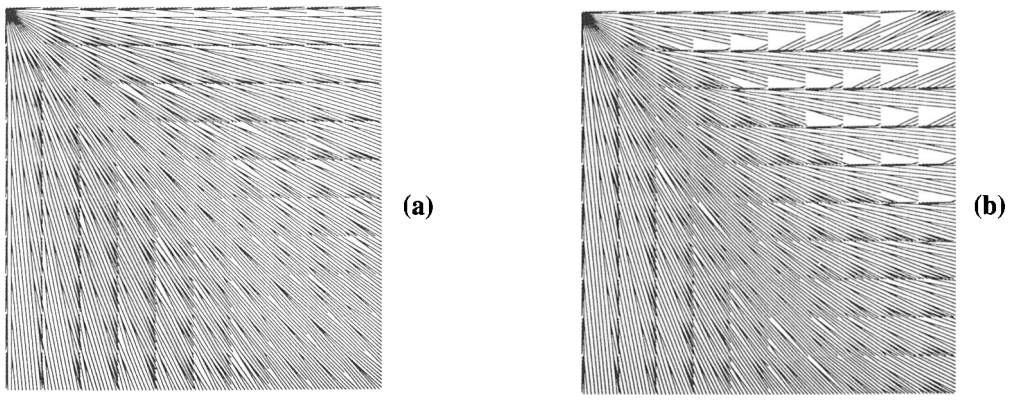

FIG. 2. Shortest paths in cell networks with 10×10 cells and 10 nodes per cell boundary. (a) Homogeneous model. (b) Linear velocity model $c = 1.0 + 0.01z$.

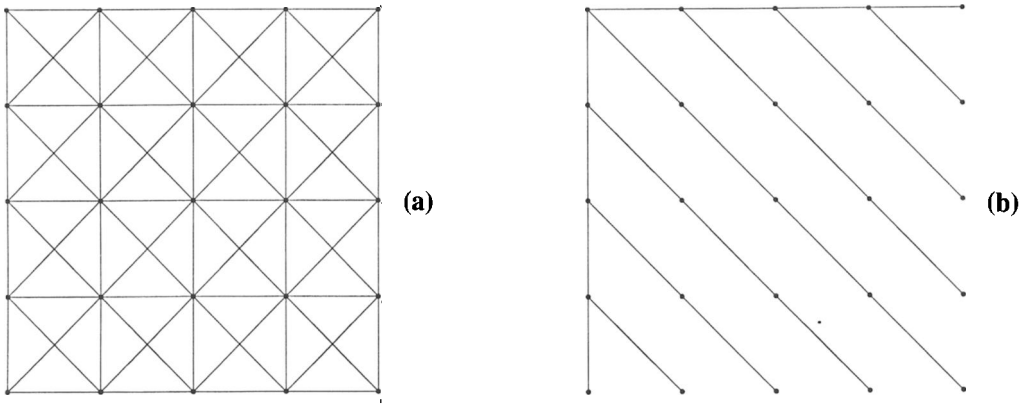

FIG. 3. Grid organization of a network. (a) 5×5 nodes, each one connected with at most eight neighbors. (b) Shortest paths in a homogeneous model in the network of Figure 3a.

the latter method has calculated the shortest paths to the 2450 other points in the same time. A careful consideration of purpose, accuracy, and efficiency of the ray-tracing problem is therefore necessary. The efficiency and accuracy of the shortest path method are given below.

The robustness of the shortest path method comes out in the presence of discontinuities in the velocity field. In Figure 5 a salt dome structure is simulated with two homogeneous layers with velocities 1.0 and 4.0 velocity is m/s, separated by a curved interface. Refractions can be seen at the top of the salt dome and diffracted paths at the flanks.

Finally, it is clear that the cell and the grid organization of the network may be not the best choices to represent a particular velocity model. Especially when the velocity varies rapidly, grid refinement techniques can be applied to make optimal use of the nodes and of the available memory space.

SHORTEST PATH ALGORITHMS AND THEIR EFFICIENCY

The following notation and definitions describe the shortest path algorithms. $G(N, A)$ is a graph, that is, a set N containing n nodes together with an arc set $A \subset N \times N$. A network (G, D) is a graph with a weight function $D{:}N \times N \rightarrow \mathrm{IR}$ that assigns a real number to each arc. D can be represented by a matrix (d_{ij}) (Figure 6). For ray-tracing purposes, it can be assumed that D is symmetric by virtue of reciprocity, $d_{ij} = d_{ji}$, and nonnegative, $d_{ii} = 0$ for $i \in N$ and $d_{ij} \geq 0$ for $i, j \in N$. By convention, $d_{ij} = d_{ji} = \infty$ denotes the case when the nodes i and j are not connected. The forward star of node i, $FS(i)$, is the set of nodes connected with i. All networks representing a velocity model used in this paper are sparse: it can be assumed that each node is connected only with the nodes in a small neighborhood of it. This means that the number of elements in the forward stars is bounded by a small number m, with $m \ll n$ and $m = O(1)$ $(n \rightarrow \infty)$.

A path is a sequence of nodes and connections succeeding each other. The traveltime along a path from one node to another is defined as the sum of the weights of the connections of the path. A shortest path is a path with the smallest possible traveltime. It may not be uniquely determined, as observed from Figure 1. The shortest paths from the source node s to all other nodes calculated with the available algorithms form a so-called shortest path tree, with its root at s and its branches connecting the other nodes. There is one and only one way to reach a node from the source node through such a tree and there are no loops. One consequence of the tree structure is that the shortest paths are described completely by an array of pointers: prec(i) is the preceding node of i on the shortest path from s to i with, by definition, the source node s equal to its own predecessor. A shortest path can be extracted from this array by repeating $\{j := \mathrm{prec}(i),\ i := j\}$ until $i = s$. The traveltime along the shortest path from s to i is denoted with $tt(i)$. The shortest traveltimes from the point source s obey Bellman's (1958) equations:

$$tt(i) = \min_{j \neq i} [tt(j) + d_{ij}] \qquad i, j \in N \tag{1a}$$

subject to the initial condition

$$tt(s) = 0. \tag{1b}$$

The traveltime to a node i is the minimum of the traveltimes to neighboring nodes j plus the weight of the connection between both. These equations follow easily from the observation that if $tt(i)$ were not equal to

$$\min_{j \neq i} [tt(j) + d_{ij}],$$

there would exist a path with a shorter traveltime than $tt(i)$. The node j that minimizes $tt(j) + d_{ij}$ is exactly the preceding node of i on the shortest path from s to i: $j = \mathrm{prec}(i)$. Bellman's equations suggest a scheme of constructing shortest traveltimes. All initial traveltimes are infinite except the

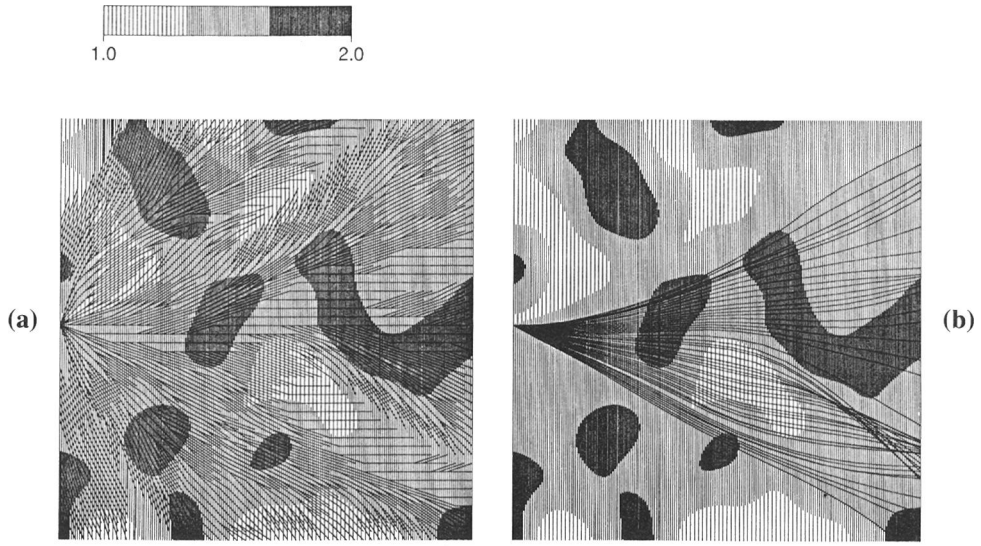

FIG. 4. Shortest paths in a smooth medium. (a) Shortest paths in a smoothly varying medium covered by a grid of 50 × 50 nodes, each one connected with at most 48 neighbors. (b) Raypaths in the same medium as in Figure 4a from the same source location, calculated with a Runge-Kutta shooting algorithm.

source node $tt(s) = 0$. It is now possible to repeat the nonlinear recursion

$$tt(i) := \min_{j \in N} [tt(j) + d_{ij}] \tag{2}$$

for all $i \in N$ until no traveltime can be updated any more. This will certainly happen after many iterations. The shortest paths from one node to all other nodes are then calculated. In fact, all efficient algorithms calculate all shortest paths from one node simultaneously, although not all that information may be useful.

Dijkstra's (1959) algorithm arranges an order of nodes to be updated so that after exactly n iterations the shortest paths are found. The nodes are divided in a set P of nodes with known traveltimes and a set Q of nodes with not yet known traveltimes along shortest paths from s. Initially, P is empty and $Q = N$. The minimum traveltime node of Q is s. It has a known traveltime $[tt(s) = 0]$, so it can be transferred to P.

The traveltimes of all nodes connected with s, all $j \in FS(s)$, are then updated in agreement with equation (2). The node in Q with the smallest tentative traveltime will not be updated any more. It can therefore be transferred to P, and the nodes from Q connected with it are again updated. This process of finding the minimum tentative traveltime node, transferring it to P, and updating its forward star is repeated exactly n times. The complete shortest path tree is then constructed. Dijkstra's algorithm can thus be formulated as follows:

(1) Initialization

$Q := N \quad tt(i) := \infty$ for all $i \in N$
$P := \emptyset \quad tt(s) := 0$

(2) Selection

Find $i \in Q$ with minimal traveltime $tt(i)$

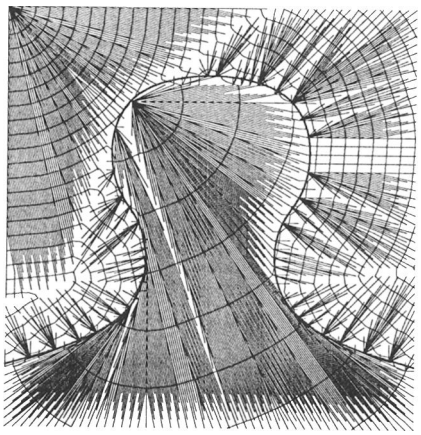

FIG. 5. Shortest paths and isochrons in a piecewise smooth medium consisting of two homogeneous layers with velocities 1.0 (upper layer) and 4.0 (lower layer), separated by a curved interface (heavy line) and covered by a grid of 50 × 50 nodes, each one connected with at most 120 neighbors.

(3) Updating

$tt(j) := \min \{tt(j), tt(i) + d_{ij}\}$ for all $j \in FS(i) \cap Q$
transfer i from Q to P

(4) Iteration check

If $P = N$ stop
else go to 2.

Dijkstra's algorithm is the classical algorithm for the computation of shortest paths, but alternatives have been developed that are several orders of magnitude more efficient. To consider the computational efficiency of Dijkstra's algorithm, the number of operations can be counted. The initialization step requires n operations to initiate the traveltimes. The selection step requires n comparisons the first time, but after each iteration one comparison less, because the number of elements of Q decreases each iteration with one. The updating costs at most as many operations as there are elements in $FS(i)$, namely, m. Therefore the total number of operations is

$$n + (n-1) + (n-2) + \cdots + 1 + m \times n = O(n^2)(n \to \infty),$$

so the computation time is essentially quadratrically dependent upon the number of nodes.

The selection of the minimum traveltime node [step (2)] turns out to be the most time-consuming step of the algorithm, because at each of the n iterations the entire set Q of tentative traveltime nodes must be scanned. The scanning could be omitted if the traveltimes in Q were ordered completely in a waiting list. The minimum traveltime node could be found immediately, since it would be the first of the waiting list. However, each updating requires the updated node to be shifted to its right position in the waiting list. This costs again $O(n)$ comparisons per iteration. Consequently, a complete ordering does not improve the quadratic dependence of the computation time on the number of nodes.

An alternative was introduced by Johnson (1977) and described by Gallo and Pallottino (1986). First, all nodes in Q with tentative traveltimes ∞ can be removed from the waiting list, since they will never be the smallest. The rest of the nodes in Q are then partially ordered in a so-called "heap." A heap is an array of elements $a(i)$, $i = 1, n$, such that

$$a(i) \le a(2i) \tag{3a}$$

and

$$a(i) \le a(2i+1) \tag{3b}$$

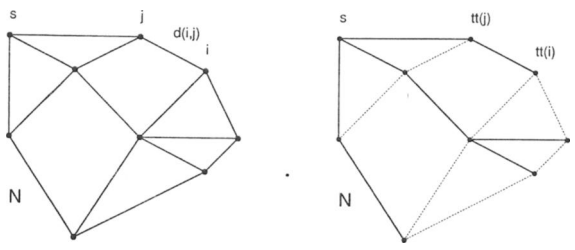

FIG. 6. The definition of a network (left) and a shortest path tree (right).

for $i = 1, \ldots, n/2$. Consider, for instance, the traveltimes

77, 80, ∞, 75, 89, 97, 90, 93, ∞, 70, 101, 99, 87, ∞, 74, 91.

After the removal of the infinite traveltimes, these can be ordered in a heap as

70, 75, 74, 80, 87, 77, 93, 89, 101, 99, 97, 90, 91.

The advantages of the heap structure come out when these numbers are represented as a tree (Figure 7). Conditions (3) are visualized in that each number is smaller than or equal to the two numbers just below it. It can be seen that the number of elements in one "generation" or horizontal layer grows exponentially with the height of the heap. Therefore, there are only log n generations for a heap of n nodes. The minimum traveltime can be found immediately: it is the uppermost number. When some finite traveltime is updated, it may violate conditions (3), so it must ascend or descend to restore the heap structure. Removing the minimum traveltime node from the heap and adding an infinite traveltime node whose traveltime has become finite can also be formulated as descents and ascents. These cost at most log n operations. The total number of operations for the calculation of a shortest path tree is now

$$\log n + \log (n-1) + \cdots + \log 1 + m \times n = O(n \log n)$$

$$(n \to \infty),$$

which is much faster than the quadratical computation time of the original Dijkstra algorithm.

The computational times are plotted for four different shortest path algorithms on a series of networks in Figure 8. The calculations were done on a GOULD PN9000 with one CPU, 8 MB memory space, and a Unix operating system. Only the shortest path calculation times are measured, not the input and output of data. The networks are cell networks with 10 nodes per cell boundary and the number of cells in both x and z directions increasing one by one from 1 to 100. For each of the hundred networks, one shortest path calculation is done for the four different algorithms: the original Dijkstra algorithm (DIJKSTRA), the Dijkstra algorithm with Q organized as a heap (HEAP), and an alternative of Dijkstra algorithm with Q organized as a queue (LQUEUE), respectively, as a so-called double-ended queue (LDEQUE). See Gallo and Pallottino (1986) for details on LQUEUE and LDEQUE. The original Dijkstra algorithm indeed appears to have a quadratical computational complexity. In the application to seismic ray tracing, LQUEUE and LDEQUE are not as fast as promised in the literature on shortest paths. HEAP is the most efficient algorithm; it has a computation time that is almost linearly dependent upon the number of nodes. This is caused by the sparseness of the networks used in ray tracing: the heap size is much smaller than the set of all nodes during the whole process.

CONSTRAINED SHORTEST PATHS AND REFLECTION SEISMOLOGY

A restriction to seismic ray tracing with the shortest path method is that only the absolutely shortest paths are found. Later arrivals on the seismogram, like reflections and multiples, caused by discontinuities in the spatial velocity distribution, do not travel along the shortest path between the source and the receiver and will not be found by a simple shortest path algorithm. Yet they are of scientific and economic importance because they contain additional information about the earth's structure. It is therefore necessary to impose a constraint on the shortest paths, which can be formulated as the demand to visit a specified set of nodes that lie on the interface.

The solution to the constrained shortest path problem is as follows. First, all shortest paths from the source node s to all other nodes are calculated with Dijkstra's algorithm. The traveltimes of the nodes on the interface are then selected, remembered, and ordered in a heap. All other traveltimes are again set to infinity. The set Q is reinitialized to N and P is reemptied. Dijkstra's algorithm is then restarted at step (2), the selection step. The resulting traveltimes are the traveltimes of shortest paths that are constrained to visit the interface node set.

To see how this works, consider Figure 9. A model is generated that consists of three homogeneous layers with velocities 1.0, 1.2, and 1.4, separated by curved interfaces

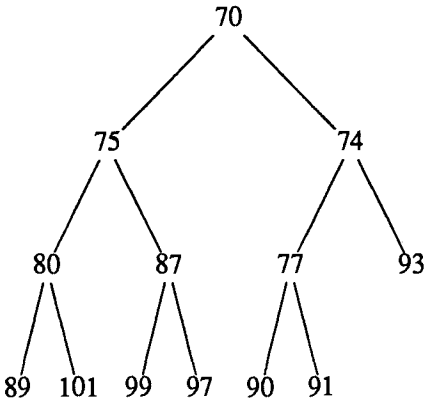

FIG. 7. The tree structure of a heap.

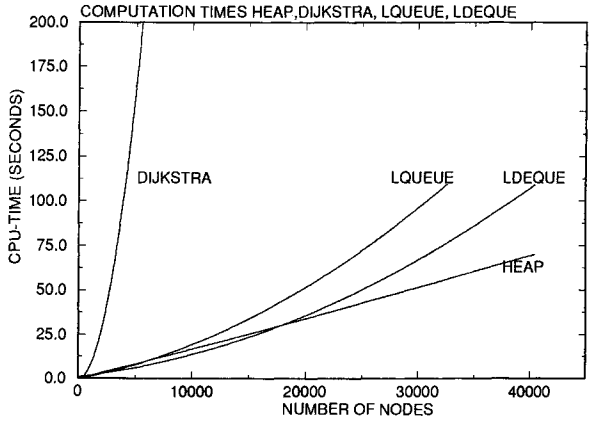

FIG. 8. Computation times of various shortest path algorithms as a function of the number of nodes.

(heavy lines). The shortest paths from one source point at the earth's surface are shown in Figure 9a. It is known a priori that points with a nonzero scattering coefficient are real physical scatterers. Nodes on interfaces or, more exactly, nodes with neighbors with a much different velocity are real scatterers, so they should "reflect" shortest paths. Therefore, they are selected and gathered in a new source node set, before restarting the algorithm. The result is shown in Figure 9b. The shortest paths in the uppermost layer are reflections on the first interface. They satisfy approximately the law of incidence and reflection. The paths in the second layer are exactly the same paths as the unconstrained shortest paths in Figure 9a. This is no surprise, because these paths automatically satisfy the constraint to visit one of the interface nodes. It can be seen that Snell's refraction law holds, as far as permitted by the network structure. The procedure of collecting scattering nodes in a new source node set and restarting the algorithm can be repeated ad infinitum, so one can calculate as many multiple reflections as desired. Figure 10 shows the continuation of the constrained shortest paths of Figure 9b, now under the constraint to visit the second interface. The paths are the shortest ones that visit the first interface first and then the second interface. In the third layer, both constraints are again satisfied automatically. The paths in the first and second layers are reflections on the second interface.

The physical significance of the results follows from a combination of Huygens' principle and Fermat's principle. The paths are shortest in traveltime between the source and the interfaces and between the interfaces and the receiver, and the points on the interfaces connecting the shortest path segments act as secondary sources, provided they are real scatterers.

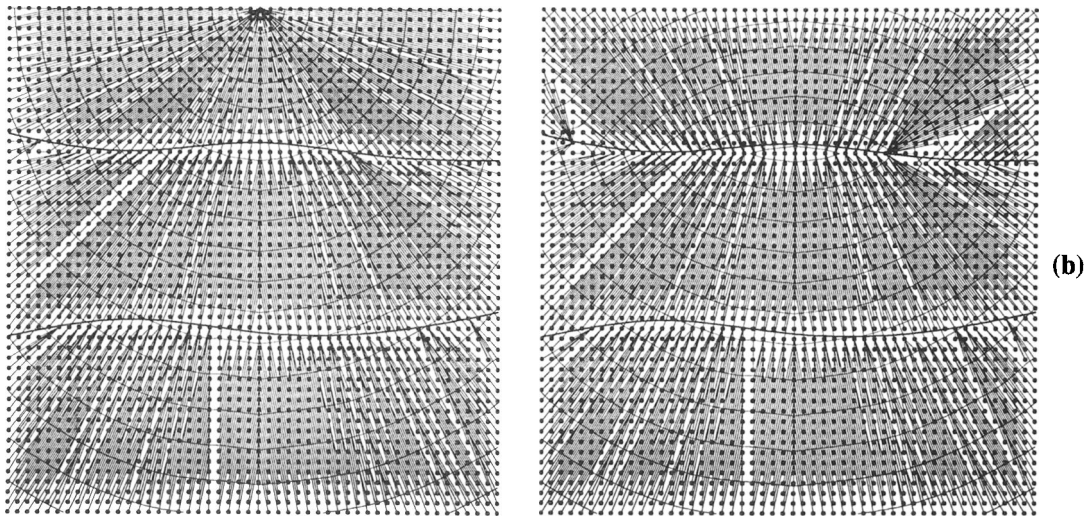

FIG. 9. Constrained shortest paths. (a) The primary field in a medium with constant velocities 1.0 (upper layer), 1.2 (middle layer), and 1.4 (lower layer) and a grid network of 50 × 50 nodes, each one connected with 100 neighbors. (b) The secondary field, consisting of the second part of paths that are forced to visit the first interface.

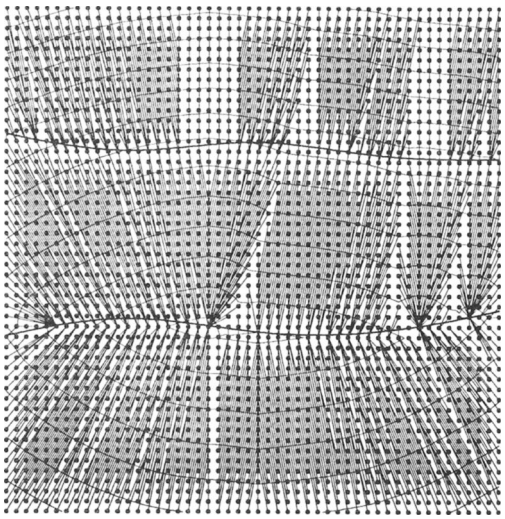

FIG. 10. Reflections on the second interface in the model of Figure 9.

ACCURACY OF THE SHORTEST PATH METHOD

By forcing seismic raypaths to follow the connections of a network, one introduces errors in the ray geometry and in the traveltime along the ray. These errors are mainly caused by two factors: the space discretization and the angle discretization.

Clearly, one error source is the sampling of the velocity field by a finite number of nodes. Rapid variations in the velocity field may be missed by a coarse grid, so that the shortest paths consist of line segments that are too long to give a reasonable approximation to the curved raypaths.

A simple example shows that the space discretization is not the only error source. Consider, for instance, a homogeneous velocity model covered by a grid network, with each node connected only with its north, east, south, and west neighbors. Such a network could imitate the streets of New York, with its streets and avenues perpendicular to each other. The shortest paths in two such networks are shown in Figure 11, together with the traveltime contours. The network of Figure 11a has 10×10 nodes; the network of Figure 11b 50×50 nodes. It is obvious that increasing the number of nodes does nothing to the accuracy of the results: the traveltime contours are straight lines instead of circular wavefronts in both networks. One could hope that the shortest paths would approximate the straight raypaths in the homogeneous model, but they are far from unique, so a great number of paths have the same (shortest possible) traveltime.

Therefore, the error in the traveltime and the ray geometry depends upon the number of nodes and the number of connections per node. When the average Euclidian distance between two connected nodes is denoted with δx and the average smallest angle between two connections leaving one node with $\delta\phi$, the traveltime error $E = T(\text{exact}) - T(\text{approximate})$ obeys an asymptotical relation of the type

$$E = \alpha_{00} + \alpha_{10}\delta x + \alpha_{01}\delta\phi + \alpha_{20}\delta x^2 + \alpha_{11}\delta x\delta\phi + \alpha_{02}\delta\phi^2 + \sum_{i+j>2} \alpha_{ij}\delta x^i\delta\phi^j, \quad (\delta x, \delta\phi \to 0), \tag{4}$$

where α_{ij} are numbers that depend upon the variation of the velocity field and the network. It can be shown that

$$\alpha_{00} = \alpha_{10} = \alpha_{01} = 0.$$

so that the traveltimes are correct up to the second order in the space and angle discretization.

This result will not be proven here, but it can be illustrated in a linear velocity model. Such a model can be thought to be more or less representative for sufficiently smooth models, since they can be approximated by linear velocity regions for small δx. Effects of the discretization of nonsmooth models are visible at the interfaces in Figures 9 and 10.

The calculations of Figure 2b are done for different space and angle discretizations. The traveltimes in the velocity field $c = 1.0 + 0.01z$ are computed in a series of cell networks, with n_x, the number of cells in the x and z direction, increasing from 2 to 50 with step 1, and n_r, the number of nodes per cell boundary, for each n_x, increasing from 2 to 15. n_r is a measure for $1/\delta\phi$ and n_x, for $1/\delta x$. The traveltimes are compared with the analytical formula for traveltimes in constant gradient media (Červený, 1987). The absolute difference is averaged over all nodes farther away from the source than 10 and divided by the computed traveltime. This average relative traveltime error is plotted logarithmically in Figure 12. It can be seen that the error is smaller than 0.1 percent for moderately large networks. This error can be compared with the computational complexity by counting the total number of nodes n as a function of n_x and n_r:

$$n = 2n_r(n_x + n_x^2).$$

For a cell network of 30×30 cells and 10 nodes per cell boundary, the average relative traveltime error is 0.0939 percent. The total number of nodes is 18 600 and the CPU time for this calculation is 31.7 s for the LDEQUE algorithm and 32.2 s for the HEAP algorithm (see Figure 8). The logarithmic plot of the average relative traveltime error (Figure 7) shows that the error curves tend to a $1/n_r^2$ relation

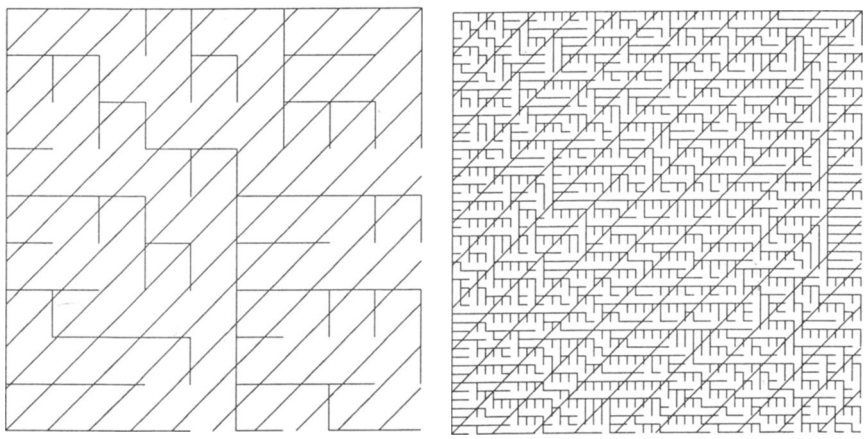

FIG. 11. Shortest paths in the streets of New York. (a) 10×10 nodes. (b) 50×50 nodes.

for $n_x \to \infty$. This illustrates the fact that $\alpha_{00} = \alpha_{01} = 0$. $\alpha_{10} = 0$ can be illustrated in the same way.

CONCLUSIONS AND DISCUSSION

The shortest path method is a flexible means of calculating seismic raypaths and shortest traveltimes. It can easily be coded in FORTRAN, because of the general abstract formulation of network theory. There is no need for extra software for complicated structures, nor for three dimensions. The method constructs a global ray field to all points in space, so there are no problems with the convergence of trial rays toward a receiver. The method finds the absolute minimal traveltime path instead of getting stuck in a local minimal traveltime path. Later arrivals on the seismograms can be found by an extra run of the shortest path algorithm. The shortest path method has a computation time that is almost linearly dependent upon the number of nodes. Its accuracy is quadratically dependent upon the number of points per coordinate direction and the number of connections per point. Even when this may not be enough for a few rays, the shortest paths are good initial guesses for additional bending. The multiple raypaths that can be seen in Figure 4b cannot be modeled with the shortest path method, at least not using the suggestions of the section on "constrained shortest paths and reflection seismology."

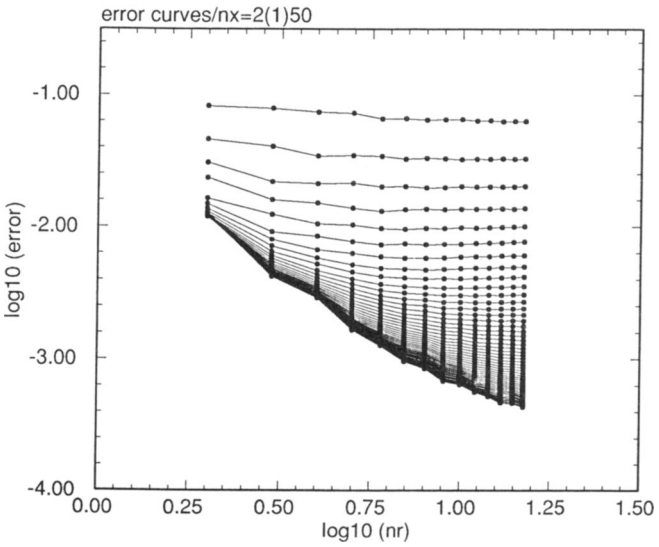

FIG. 12. The traveltime error as a function of the number of nodes and the number of connections per node in a linear velocity model covered by cell networks.

ACKNOWLEDGMENTS

This research is financially supported by the Netherlands Technology Foundation (STW). I would like to thank Prof. Dr. G. Nolet and Dr. R. Snieder from the Department of Theoretical Geophysics, Prof. Dr. K. Helbig from the Department of Exploration Geophysics, and Prof. Dr. A. van der Sluis from the Mathematical Institute in Utrecht; Prof. Dr. J. K. Lenstra from the Centrum voor Wiskunde en Informatica (CWI) in Amsterdam; and Prof. Dr. Ph.L. Toint from the Department of Mathematics of the Faculté Universitaires ND de la Paix in Namur, Belgium, for their scientific support.

REFERENCES

Bellman, R., 1958, On a routing problem: Quart. Appl. Math., **16**, 88–90.
Červený, V., 1987, Ray tracing algorithms in three-dimensional laterally varying layered structures, *in* Nolet, G., Ed., Seismic tomography: D. Reidel Publ. Co., 99–133.
Deo, N., and Pang, C., 1984, Shortest-path algorithms: taxonomy and annotation: Networks, **14**, 275–323.
Dijkstra, E. W., 1959, A note on two problems in connection with graphs: Numer. Math., **1**, 269–271.
Gallo, G., and Pallottino, S., 1986, Shortest path methods: A unifying approach: Mathematical Programming Study, **26**, 38–64.
Johnson, D. B., 1977, Efficient algorithms for shortest paths in sparse networks: Journal of the ACM, **24**, 1–13.
Julian, B. R., and Gubbins, D., 1977, Three-dimensional seismic ray tracing: J. Geophys., **43**, 95–114.
Nakanishi, I., and Yamaguchi, K., 1986, A numerical experiment on nonlinear image reconstruction from first-arrival times for two-dimensional island arc structure: J. Phys. Earth, **34**, 195–201.
Stoer, J., and Bulirsch, R., 1980, Introduction to numerical analysis: Springer-Verlag.