

Massively Parallel Computing of Shortest Raypath and Traveltime in 2-D and 3-D Models

Jason C. Kao¹ and How-Wei Chen²

(Manuscript received 6 March 1996, in final form 28 July 1996)

ABSTRACT

Based on Huygens' principle, the authors present an accurate and computationally efficient method to compute the shortest raypath and traveltime in a two- and three-dimensional (2-D and 3-D) space of a discrete block model. The efficiency of the method is achieved through approximation, while the accuracy of the calculated traveltime solely depends on machine precision. The accuracy of the raypath is realized by the small increment in the orientation of the ray incidence. Whether the computational efficiency and accuracy can be justified depends on the model's complexity and requirements in its own application. In addition, the feasibility of implementing the algorithm on the Cray T3D Massively Parallel Processors (MPP) is proposed.

The velocity distribution in a 2-D space is discretized into homogeneous polygonal cells. The search for the shortest traveltime and path between two given points can be reduced to a discrete graph searching. In the general 3D case, the velocity model is characterized by discrete convex blocks bounded by polyhedral surfaces. Although the 3-D algorithm is a straightforward extension of the 2-D case, the computing operations in 3-D are much more CPU intensive.

The method is demonstrated with examples showing raypaths and wavefronts in 2D and 3D block models. On the basis of these examples, the proposed algorithm is capable of solving the optimal raypaths from different source points in parallel on the MPP system.

(Key words: Massive parallel computing, Raypath, Traveltime, 2-D, 3-D, MPP)

1. INTRODUCTION

The accurate computation of traveltime and raypath for wavefields passing through a complex earth model from a specified source is essential for a variety of geophysical applica-

¹ Cray Research, Inc., Earth Sciences Application Division, Eagan, Minnesota, U.S.A.

² Institute of Seismology and Institute of Applied Geophysics, National Chung Cheng University, Chiayi, Taiwan, R.O.C.

tions, e. g., seismic migration, tomographic inversion and earthquake hypocenter location. Current interest in 3-D prestack Kirchhoff or reverse-time depth migration has added an impetus to the studies of an efficient and accurate method to generate first arrival traveltimes on complex velocity models.

Conventional methods of traveltimes computations are based on shooting and bending methods (Julian and Gubbins, 1977; Um and Thurber, 1987), in which the minimum traveltime raypath between two prescribed locations is calculated. Such methods are generally quite accurate for those media with moderate varying velocities. However, both methods have serious drawbacks. The schemes occasionally suffer from such problems as the convergence to a local minimum, the inability to find diffracted raypaths and the disappearance of rays within the shadow zones and waveguides. Also, the stability criterion may fail under certain situations. The problems become even more severe in 3-D media.

Furthermore, such methods in principle can only be used to calculate one raypath between two specific points in a sequential manner. In a practical 3-D prestack migration process, the objective is to fill every grid point in a 3-D volume with the first arrival traveltime from an arbitrary source location. Computing traveltime at all grid points becomes an extremely CPU-intensive task. The conventional scheme, thus, becomes impractically slow for actual data processing.

A scheme based on the expanding wavefront theory was recently developed and presented as an alternative to conventional raypath computing. Reshef and Kosloff (1986) proposed the earlier version of this idea based on the eikonal equation solution. Vidale (1988, 1990) presented a finite-difference scheme to calculate the traveltime at all corners of cubic cells. Van Trier and Symes (1991) proposed the improved version of solving the eikonal equation via an upwind finite-difference scheme. Dellinger (1991) later applied it to anisotropic media. Qin *et al.*, (1992) further improved the algorithm by calculating the traveltime along expanding wavefronts instead of along the expanding squares. Raypaths from the source point to all grid points can be obtained by tracing the secondary source points in succession through grid points.

Based on Huygens' principle and Dijkstra's algorithm (1959), Nakanishi and Yamaguchi (1986), Moser (1990) and Saito (1989, 1990) developed an efficient method of finding the shortest raypath tracing on a model consisting of uniform-velocity cells. In their studies, the earth model was parameterized into a network of nodes on the boundaries of block cells. Calculating the traveltime from a source is thereby reduced into a problem of sparse graphic searching. Fischer and Lees (1993) and Klimes and Kvasnicka (1994) developed a similar technique for improving the efficiency of the shortest path ray tracing with error analysis.

The above procedures explicitly calculate minimum traveltimes on the network nodes along an expanding wavefront. The shortest raypath approach posed many advantages over previous schemes. For instance, this method considers the incidence of waves in all possible takeoff angles from the source. It is applicable to any type of source-receiver geometry and complex structure model with strongly varying velocities. Moreover, it always finds the global minimum on the expanding wavefront; additionally, head waves and diffraction waves are calculated where they are the first arriving energies, etc.

The scheme's computational efficiency is another major concern about the shortest raypath

approach. Raypaths from a source to all grid points in the model can be constructed in one pass. In fact, the cost of generating a family of raypaths is no more than that of calculating one path between any two specified points. This algorithm is particularly appropriate for implementation on a distributed memory MPP system, as an even higher level of parallelism can be realized by parallelly computing several families of shortest paths from different sources.

Although there are no restrictions with the classical ray theory, some drawbacks are inherent just from the sheer nature of the sparse graphs approach. The accuracy of the method in approximating the raypath strongly depends on the compactness of a network's discrete nodes. If the nodes are close together, the raypaths from the source can be changed with a very small increment in a takeoff angle. An excellent approximation of ray tracing can be produced. However, including too many nodes in the network would greatly add to an undesirable burden on computer resources which would subsequently reduce the computational efficiency.

It is intuitively clear that a network with a smaller number of nodes and larger block sizes theoretically generates a better raypath approximation of the actual ray connecting two specified points than the one with more cells. Closely examining Figure 1 indicates that the system with smaller rectangular cells actually generates a poorer approximation. Therefore, as long as the desired geological features are incorporated into the model, selecting a grid as coarse as possible is preferable so as to achieve computational efficiency and accuracy.

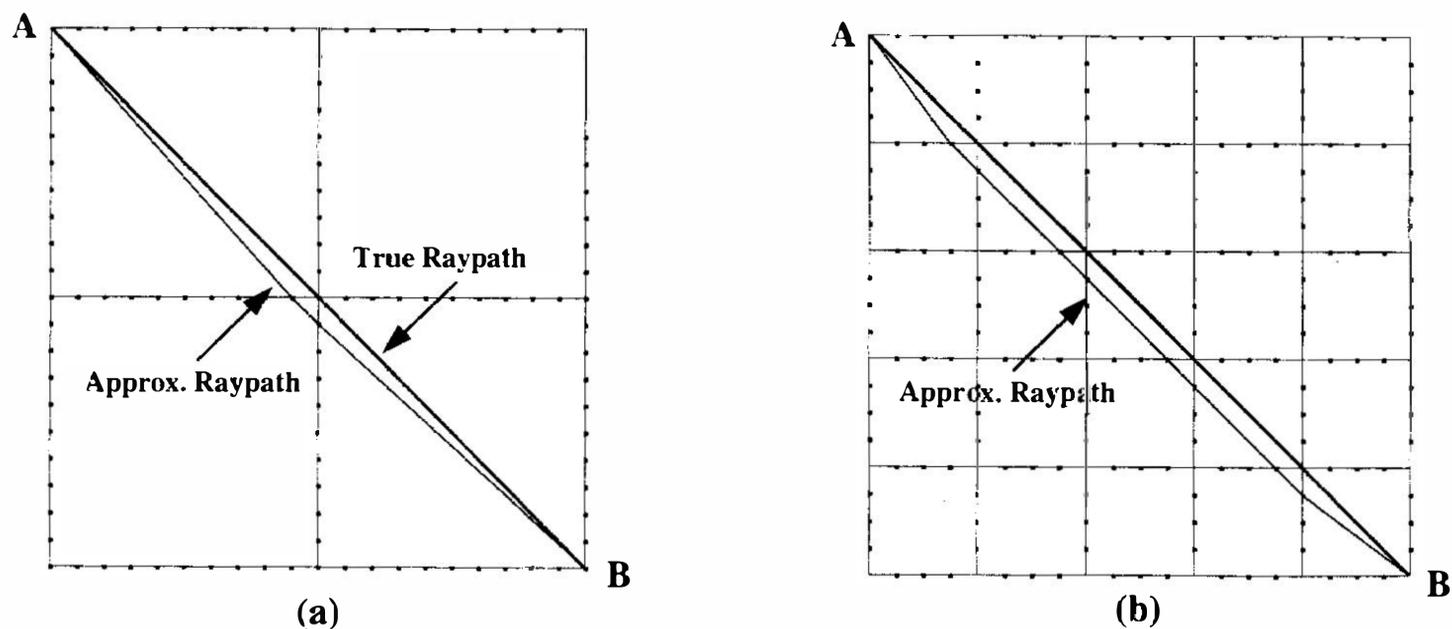


Fig. 1. With regard to a uniform velocity 2-D model, this figure shows the problem of inaccuracy of computing the shortest raypath between two points A and B due to the sparseness of nodes on the networks. With a given node interval distance, network (b) with a smaller cell size clearly generates a poorer raypath approximation to the true ray than the raypath in network (a) composed of larger cells and a smaller number of nodes.

Motivated by those results, in this study, a more flexible network is designed by allowing the earth model to be composed of homogeneous blocks of irregular size. In the 2-D case, each block is a convex polygon characterized by polygon vertices and associated velocities. A network of nodes on the boundaries of the block is generated by the algorithm. For general 3-D

media, the model consists of blocks of convex polyhedron bounded by polygonal facets as the boundaries. The new networks normally contain fewer nodes on the network and, thus, release the burden on computational time and memory. Moreover, this allows the cell boundaries to fit better on a complex structure, such as irregularly shaped fault planes, dipping beds and salt domes, without resorting to stair-step approximations inherent with square cells. Figure 2 compares the network systems for 2-D models.

This paper first illustrates the algorithm of the construction of the node and edge network in a 2-D block model and the computation of the traveltimes and raypaths from a given source point to all nodes. A series of examples of increasing complexity are employed to illustrate the salient characteristics of travel-time and raypath computations. A straightforward extending of the procedure into the 3-D problem is then addressed. To achieve maximum performance, the key concepts and features regarding machine dependent architecture and the computing environment are necessary. In the following section, the concept of the massively parallel processing (MPP) computation for a highly scalable heterogeneous supercomputing system is first introduced.

2. MACROARCHITECTURE OF CRAY T3D MPP

Cray T3D MPP is a scalable heterogeneous computing system with multiple-instruction multiple-data (MIMD) character in architecture. It is scalable from 16 to 1024 nodes with two processing elements (PEs) in each (Figure 3a). The DEC Alpha microprocessor, with the capability of 150 MFLOPS peak performance and a direct-mapped data cache of 256 lines (32 bytes per line), is the heart of every PE. The memory of the T3D is physically distributed; however, it is globally addressable by each PE. Each PE on a node owns a single bank of local Dynamic Random-Access Memory (DRAM) external to the Alpha processor. The memory interface between the PE and local memory is the Cray custom interface. This circuitry extends the local virtual address to point into a global address space. Each PE can directly read from and write to other PE memories. Although each PE can directly address the memory of every other PE, the PE accesses its own DRAM much faster.

The interconnected network is a 3-D torus designed to minimize the network distances and provides the highest known bisection bandwidths (Figure 3b). High-performance (150 MHz) switch nodes are used in the network, which can operate bidirectionally in each dimension and handle interprocessor communications without interrupting the processor. The CRAY T3D allows latency hiding, capable of reducing the memory latency of accessing remote memory through either preloading or overlapping with other computations.

Improving the procedure's efficiency entails minimizing communication overhead and maximizing sustained computational rates by considering all macroarchitecture characteristics of the T3D MPP system in the design of the algorithm. The flexible synchronization mechanism is important for parallel computing on MPP machines.

3. SEARCHING THE SHORTEST RAYPATH

The 2-D earth model, regardless of the velocity complexity, is considered to be composed of a finite number of uniform-velocity polygonal cells. The algorithm request that all cells be

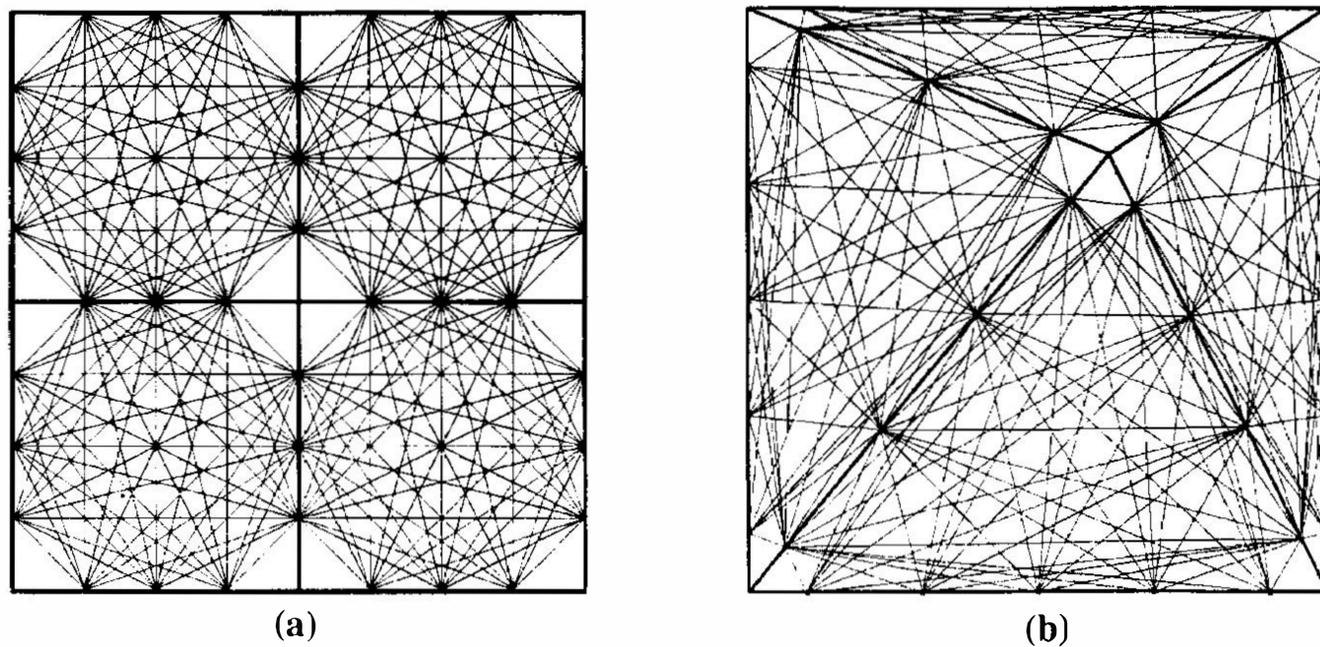


Fig. 2. Comparison of the organizations of nodes and edges for a 2-D block model. (a) Moser's network composed of square cells (b) the network considered in the present algorithm composed of irregular polygonal cells.

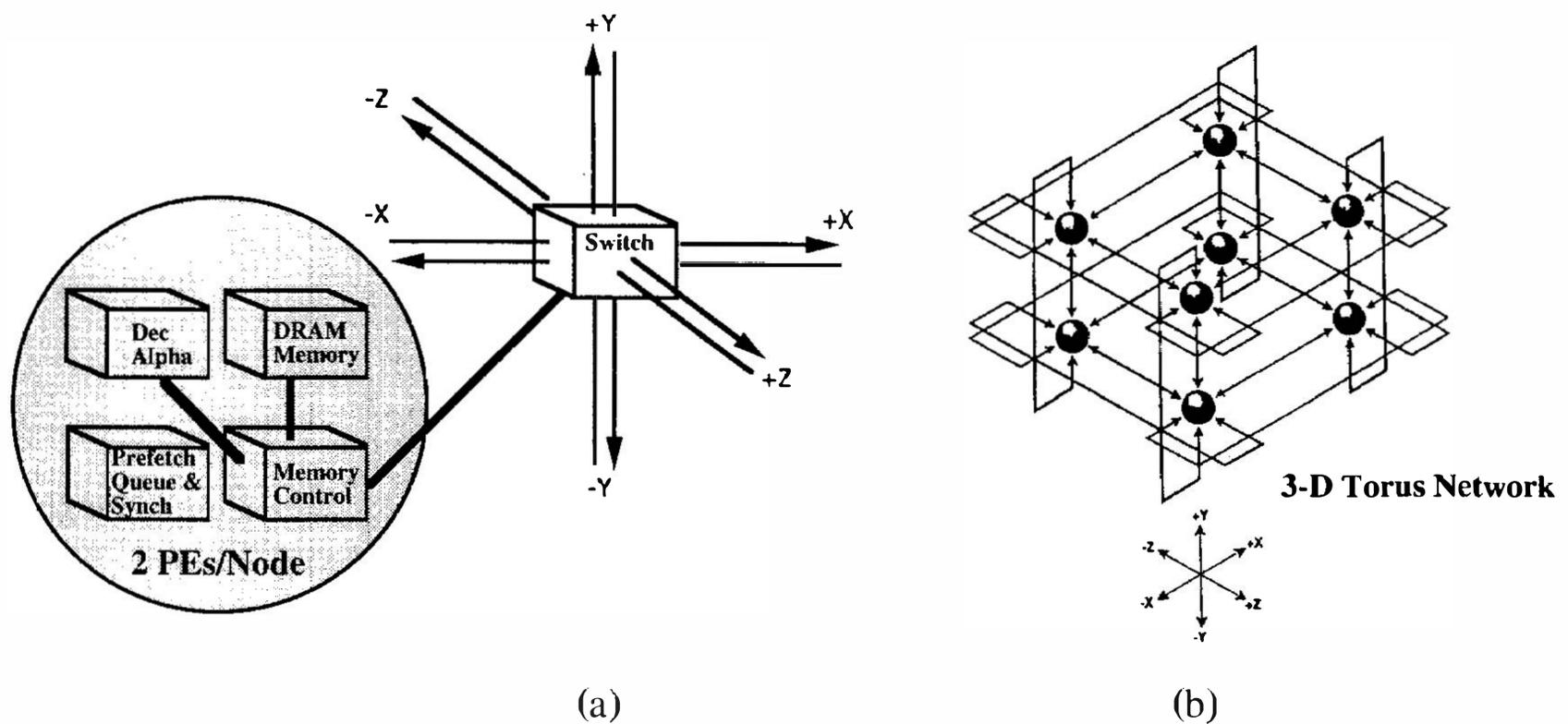


Fig. 3. The macroarchitecture of a scalable MPP system used for parallel computing. (a) The system node consists of two PEs sharing the switch and network support logic for high-speed synchronization. (b) The PEs are connected by a 3-D torus system interconnected network. This design allows for an extremely fast remote memory access for efficient MPP system usage.

convex to ensure that all segments of the raypaths connecting any two nodes on the boundaries of the cell must lie entirely within it.

In accordance with Moser's paper, the present authors do not place nodes at the cell vertices in such a way that all network nodes are shared by two blocks, except those on the outer boundaries. The input parameters required to construct the network are vertex coordinates, the fixed distance between two neighboring nodes and the velocities associated with each discrete block. Coordinates of all network nodes are automatically calculated and distributed over the local memories of all available PEs. Meanwhile, for each node, information about the block numbers, which are shared with this node, is also saved.

The node coordinates and block number information are related to the network system's geometry and should not be changed at any time throughout the computation. Therefore, no data communication on these arrays is required. The computation efficiency is thus significantly improved. An array of the same length needs to be allocated to store the traveltime values which are updated throughout the processing. Fetching data from a remote PE is considerably slower than data moving within the local memory (Numrich, 1994; Koeninger, 1992). Accordingly, the storage of nodal coordinates is arranged so that all nodes on the boundaries of one cell are saved into the local memory of a specific PE to minimize data communication.

Dijkstra's algorithm is applied concurrently at all PEs to pick out the node with the minimum traveltime which can function as the new secondary source for the next iteration of radiation. Calculating the traveltime between any two nodes on the cell boundaries is relatively easy and rapid. The major cost of the method arises from the checking of the wavefront position and finding the secondary source on the cell boundaries. However, the sorting and searching process cannot be vectorized. This is the most time-consuming part of the scheme. As the traveltime information is distributed over all PEs, the process actually executes in parallel to pick out local minimums. A global minimum is then selected from the sequence of local minimums so that the first arrival traveltime at this node can be calculated. Although interprocessor communications cannot be completely avoided, the algorithm is designed to reduce the data movements among PEs as much as possible.

4. ALGORITHM AND IMPLEMENTATION

The present authors basically follow Moser's procedure in applying Dijkstra's algorithm to find a new source along the expanding wavefront. However, due to the distinctness in the graphic network put forth and the characteristics of the hardware architecture, the algorithm used here is not the same as that addressed in Moser's work.

The procedure initially determines the coordinates of all network nodes from the given model geometries and the node interval. The calculated node positions are evenly distributed over the local memory of the available PEs. A one-dimensional array with the same dimension of node number must be allocated to store the traveltime at each node. During iterative computational procedure, the traveltime information at each node is also distributed over the PEs. During the computation, all nodes are divided into two disjoint sets, open and closed, denoted by P and C. Closed set C consists of nodes having had wavefronts from the source propagated through them, whereas all nodes in the open set P have not been hit by the wavefront.

Initially, all nodes reside in set P, and the traveltimes at all nodes are set to be infinite. In

the beginning of the process, straight raypaths radiate from the source point into all nodes on the boundaries of the cell containing the source point. The traveltimes of all nodes on the cell boundaries are then updated. In set P, a node with minimum traveltime is picked as the secondary source point for the next iteration of wavefront radiating. It is assumed that the secondary source point is considered to be hit by the wavefront and must be moved into set C from set P.

Because each node is shared by two cells, the secondary source becomes a new starting point for emitting rays into all boundary nodes of those cells. The traveltimes on each node of ray incidence are then updated. A node in set P with the minimum arrival time is again selected as the new secondary source. The same process is repeated until the wavefront goes through all the network nodes thereby, moving all nodes from set P into set C.

The entire procedures described above can be presented in terms of a pseudo code and further translated into either Rational Fortran (Ratfor for short) or standard Fortran language. The algorithm can be formulated as follows:

INITIALIZATION:

1. Calculate the nodal coordinates of the network and distribute over the local memory of PEs.
2. Put all nodes in set P and initiate C to be empty.
3. Define the traveltime at all nodes as infinite.
4. Calculate the traveltime from the source point to all nodes on the boundaries of the cell containing the source.

ITERATIVE TRAVELTIME UPDATING:

While (set P is not empty) {

1. Find the node p in P with the earliest arrival time.
2. Use p as the secondary source and update the traveltime of all nodes in set P which are also on the boundaries of cells containing node p.
3. Move node p into set C. }

It should be noted that since node coordinates are determined by the geometry of the block model, the nodal coordinate system is independent of the source position and should not be changed during the processing. This procedure allows for the computation of raypaths from several separate sources in parallel. Designing a program for massively parallel computing is based on the Cray MPP Fortran (CRAFT) (Pase *et al.*, 1992), and inter-processor communications are facilitated via shared memory access library routines (Barriuso, 1994).

5. SYNTHETIC EXAMPLES

In the following synthetic applications, two- and three-dimensional models are used to demonstrate the behavior of the algorithm. A series of examples of increasing complexity is employed to illustrate the salient characteristics of traveltime computations. A relatively simple and graben-like model with a moderate velocity contrast is presented. A circular body with the characteristics of high velocity contrast and steep dip is also used to mimic salt and magma

chamber structures. Finally, a three-dimensional block model is presented to demonstrate the effectiveness of the 3-D computation.

5.1 Two-dimensional Models

5.1.1 Simple and fault-and-tilted block models

Figures 4, 5 and 6 demonstrate the scheme by means of a relatively simple 2-D example. Figure 4 shows a higher velocity block embedded in a lower velocity background. To create a system to fit the algorithm, the medium is divided into five discrete convex polygons in which 287 nodes are generated onto the boundaries. Figure 5 describes the procedures of computing raypath and traveltime for a source located inside a block. The raypath from a source point to all boundary nodes of the cell are first calculated based on Dijkstra's algorithm. Traveltimes at any position inside the polygonal cells can be estimated from those on the cell boundaries. The minimum arrival time is picked along the cell boundaries as the secondary source. The secondary source, denoted as C, is used again for the next iteration to radiate rays to all nodes that share this source. The calculated raypath and corresponding traveltime from C are then updated again for the next iteration. The procedure is executed in parallel to determine the local minimums. Finally, a global minimum of traveltime is selected from the sequence of local minimums. The traveltime within each block can be obtained by first determining the traveltime along each raypath. A bi-directional linear interpolation is then performed to allocate the actual traveltime desired for display. The computation can be performed over local machines, such as a workstation or PC. Figure 6 shows the final shortest raypaths and first arrival traveltime isochron contours. Some blank areas in Figure 6 that do not contain raypaths are the result of searching for the final global minimum of traveltime. Figure 7 depicts a somewhat realistic 2-D graben-like structure model with the geological features of steep normal fault planes and tilted fault blocks. The velocity change across any adjacent blocks is moderate enough to represent most of the geological features generally encountered. Figure 8 displays the calculated raypaths and isochronous contours.

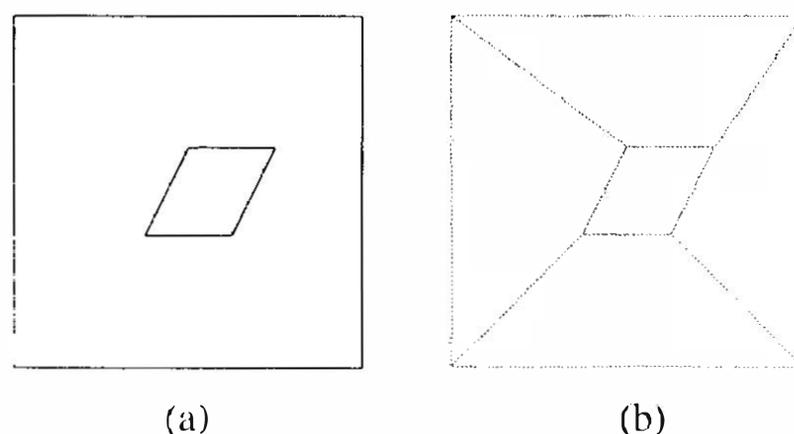


Fig. 4. Construction and parameterization of a simple 2-D block model used for traveltime and path calculation. (a) A polygonal block with velocity 1.5 times higher than the homogeneous background is embedded in the 2-D body. (b) The algorithm creates a network of 287 nodes evenly spread over twelve boundary edges of five convex polygons.

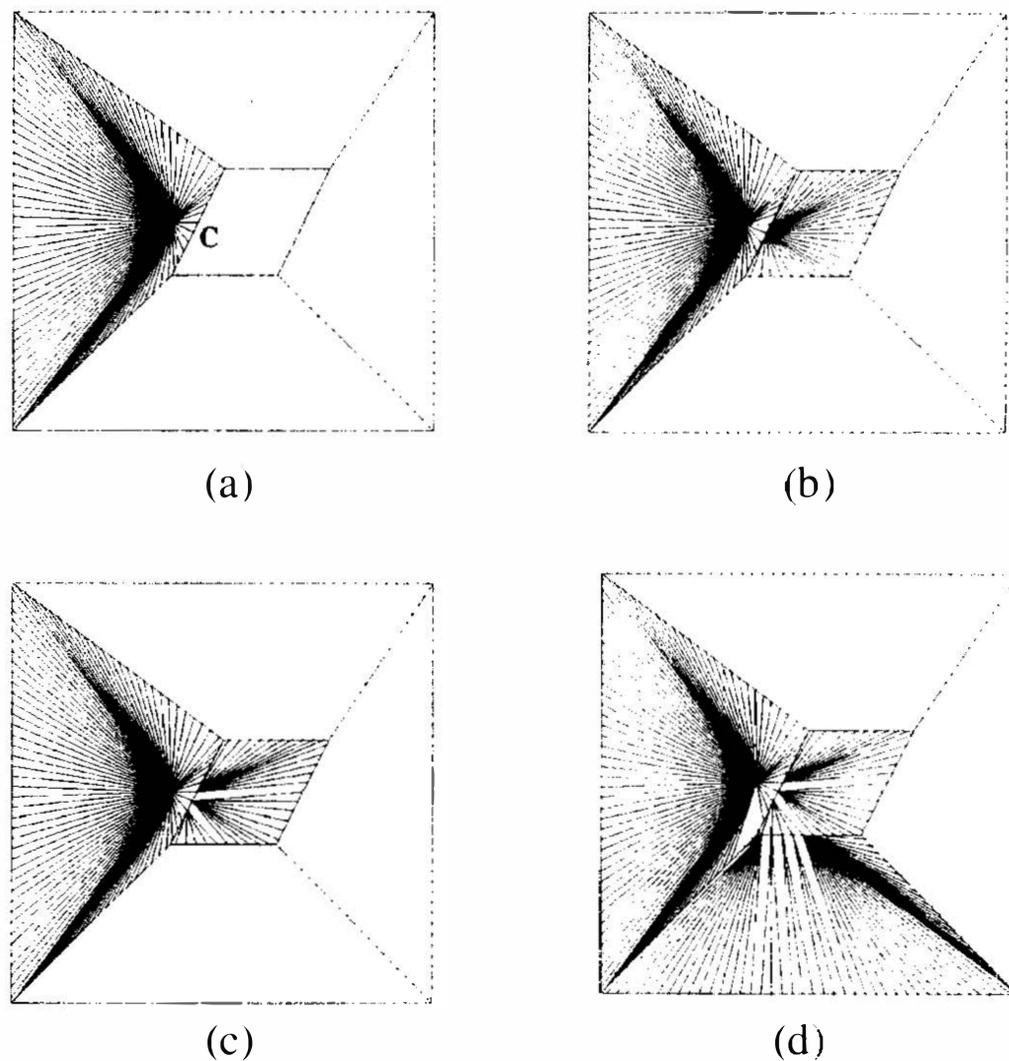


Fig. 5. The intermediate procedures of computing raypath and traveltime from a source located inside the block. (a) The raypath from a source point to all boundary nodes of the cell containing the source are calculated on the basis of Dijkstra's algorithm. The minimum arrival time is picked at node C. (b) C is the secondary source that emits rays to all boundary nodes of the cells sharing this source. (c) Localized raypaths and traveltimes are updated for all of the possible secondary sources. (d) The procedure is repeated throughout the entire blocks.

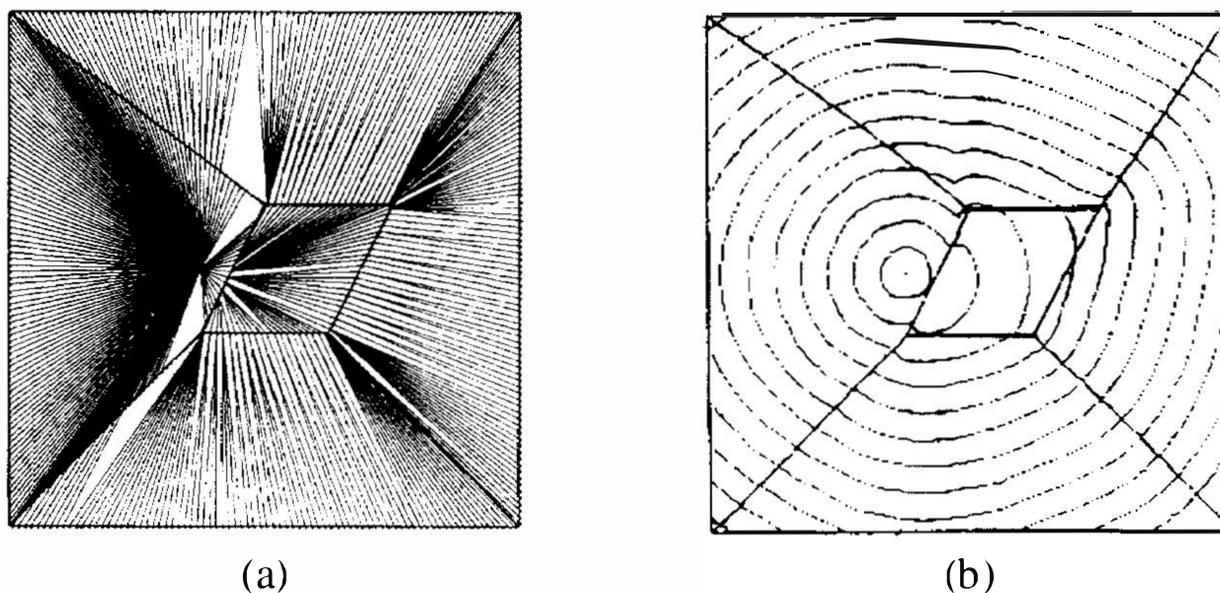


Fig. 6. (a) The final shortest raypath calculated from a source point to all nodes of the system. (b) The contour of isochrons which show the expanding of a wavefront from the source.

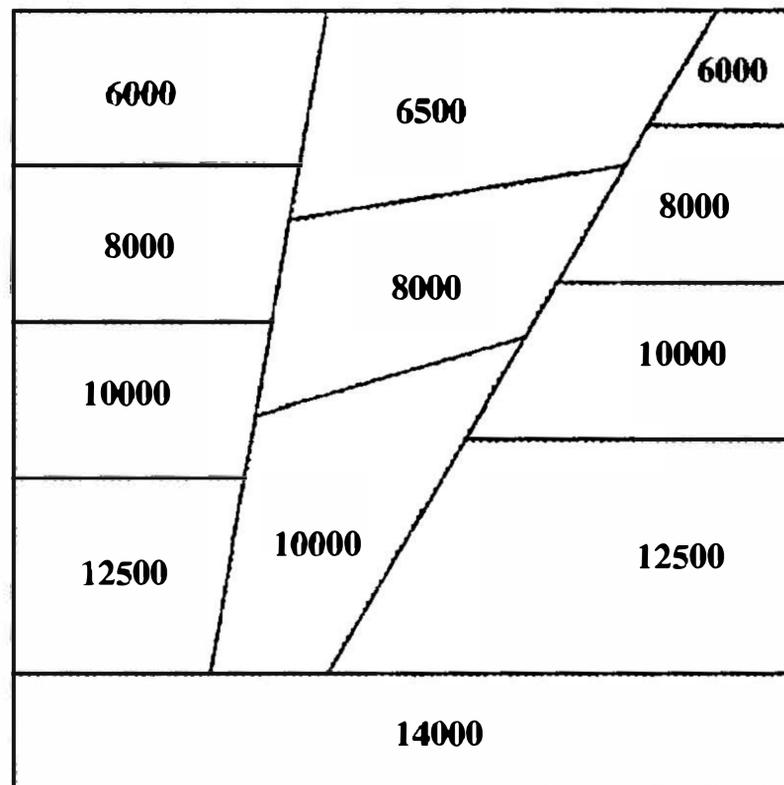


Fig. 7. A piecewise smooth medium representing velocity distribution over a 2-D model with graben-like geologic features. The numbers inside each block indicate its velocity in ft/sec. The network generated from the algorithm is composed of 946 nodes over thirty seven edges of twelve convex polygons.

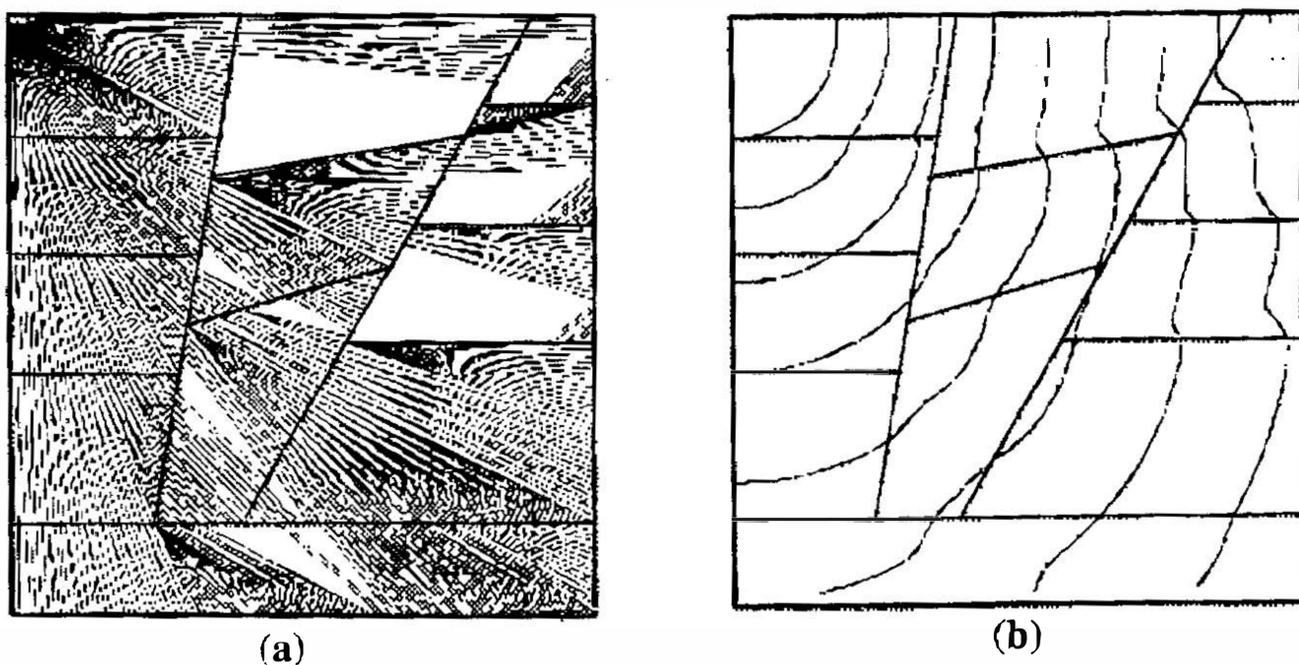


Fig. 8. (a) The final shortest raypath calculated from a source point to all nodes of the system. (b) The contour of isochrons which show the propagation of a wavefront from the source.

5.1.2 Circular model

An important but difficult application in prestack seismic processing is the mapping of the bottom of the salt diapir structure and the features beneath salt. Determining the image (both the lateral and vertical extents) of the magma chamber is another important application. The difficulty arises because the irregular shape and steep flanks of the salt structure produce highly distorted observations. Waves transmitted through the top of the salt are strongly refracted, while most of those reflected from the bottom of the salt arrive at the earth's surface out of the normal recording apertures and are not available for imaging.

The capability of obtaining accurate results along the structure boundaries with a sharp first order velocity contrast and a steeply dipping interface is necessary when traveltimes and raypath information are used for prestack depth migration. For instance, traveltimes are used as a necessary excitation time imaging condition in prestack reverse-time migration (Chang and McMechan, 1986; Chen and McMechan, 1992) for reconstructing the wavefield and for subsurface structure imaging. Moreover, both raypath and traveltimes also function as inputs for prestack Kirchhoff migration.

Figure 9 illustrates the advantages of applying the algorithm to model with a large velocity contrast and severe structure variation. Figure 9 shows a low velocity circular body embedded in a higher velocity background. The model is composed of twenty three discrete convex polygons, and the circular body is approximated with sixteen segments. The traveltimes (thus, wavefront) inside this body are obtained through bi-directional linear interpolation based on the calculated traveltimes data along the structure boundary. Other types of interpolation scheme, such as the bi-cubic spline or minimum curvature techniques can be used. Outside the body, the effects of this high velocity anomaly are clearly demonstrated by the appearance of

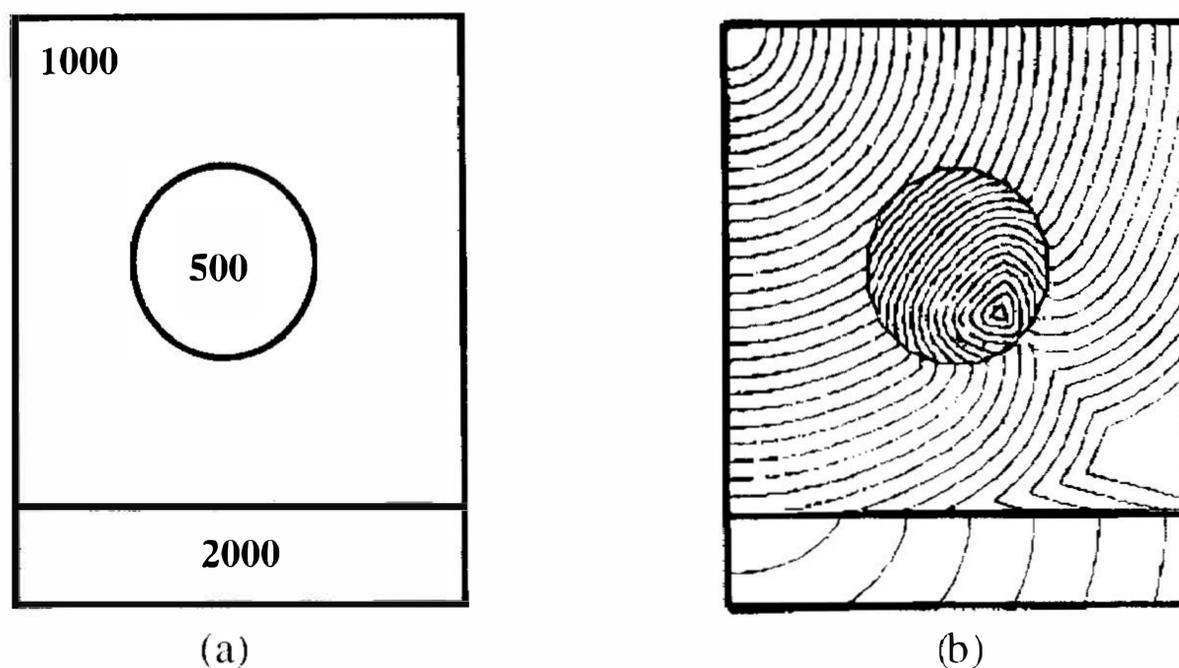


Fig. 9. A low velocity circular body embedded in a higher velocity background and a flat layer at the bottom. (a). The circle is approximated by a 16-side polygon with source at the upper-left corner. The number within each subblock is the velocity in ft/sec. Traveltime contours (b) indicate the focusing effect within the circular body and the generation of head waves due to the high velocity contrast along a flat interface.

diffracted wavefronts in the shadow zone. Inside, the velocity contrast is such that it provokes a closure of wavefronts (focusing) due to the refracted rays. The focusing effect, along with refracted, diffracted and head waves are particularly distinguishable in Figure 9. Although the model is geologically irrelevant, the proposed method does not raise any computational problems even for extremely severe first order velocity contrasts.

The model is particularly important to test the algorithm's flexibility, stability and accuracy. Because the image of the subsurface structure is highly influenced by the accuracy of the image condition obtained from different traveltime generators. A distorted or even vanished image can be obtained when a low velocity body exists between the source and the major target structure to be imaged. This can occur in the presence of an irregularly shaped low velocity body that often creates a shadow zone. The main advantage of using this approach is that traveltime information can be obtained not only along the structure boundary and within a low velocity circular body but also within the shadow zone. Also, the large velocity contrast between the salt and the surrounding sediments frequently produces anomalously large (post-critical) reflections and correspondingly reduced amplitudes for reflections from deeper structures. A highly distorted low-wavelength image of very large amplitude that contaminates the actual image obtained can be observed in the migrated section when post-critically refracted waves are included.

5.2 Three-dimensional Model

The concept of computing in a 2-D medium can naturally be extended into the 3-D case. The 3-D model is still considered here as a piecewise smooth medium consisting of discrete blocks. Each block is a constant velocity convex polyhedron with polygonal facets as boundaries. All network nodes are evenly spread over the boundary facets. It should be noted that all nodes are placed inside the boundary facet such that each one is shared by two and only two blocks. Defining the input parameters for a complex 3-D model is rather cumbersome. However, once the network system is created, the computational scheme in the 3-D case is the same as that in the 2-D model, except that the memory size is greatly increased and the floating point operations are much more intensive.

For simplicity, the procedure was experimented over a 3-D body consisting of a higher velocity block embedded into a uniform velocity background as shown in Figure 10(a). The 3-D body is divided into seven discrete subregions, each of which is a convex polyhedron. The nodes at each boundary facet are generated from the scheme. Figure 10(b) depicts the polyhedral regions and nodes on one facet. Given a source point at the center of the top surface, Figure 11(a) shows the shortest rays originating from the source to specified nodes on the outer planes. Figure 11(b) depicts the wavefronts at several time steps. Figure 12 shows the successive expanding of the first arrival surfaces in the 3-D medium.

6. DISCUSSION AND CONCLUSIONS

In this study, the authors have developed and tested a parallel algorithm of the sparse graphic searching method which employs Dijkstras' algorithms and Hugen's principle of finding the shortest raypath and their corresponding travel-time for 2-D and 3-D models. Raypaths

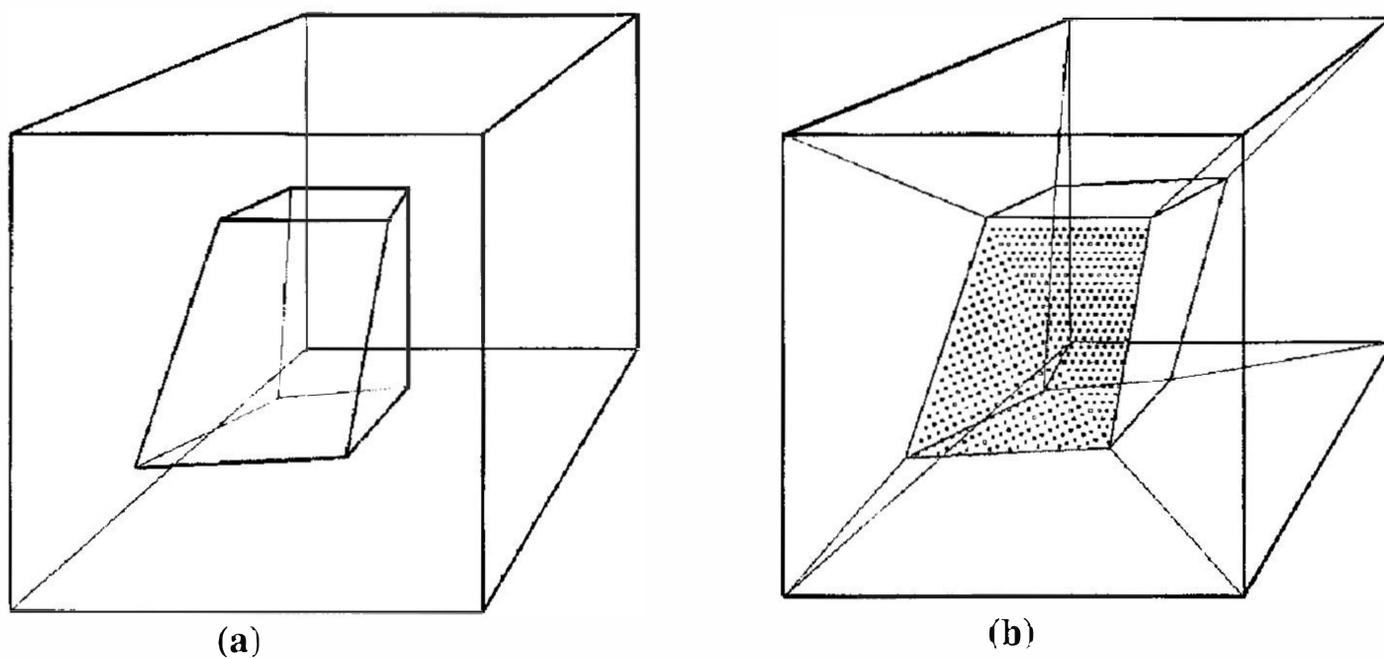


Fig. 10. (a) A homogeneous block with velocity 1.5 times higher than the background medium is embedded into the 3-D body. (b) The 3-D volume is divided into seven subregions of convex polyhedrons. 14082 nodes are generated over twenty four facets from the algorithm. For simplification, this figure only shows nodes on one specific boundary..

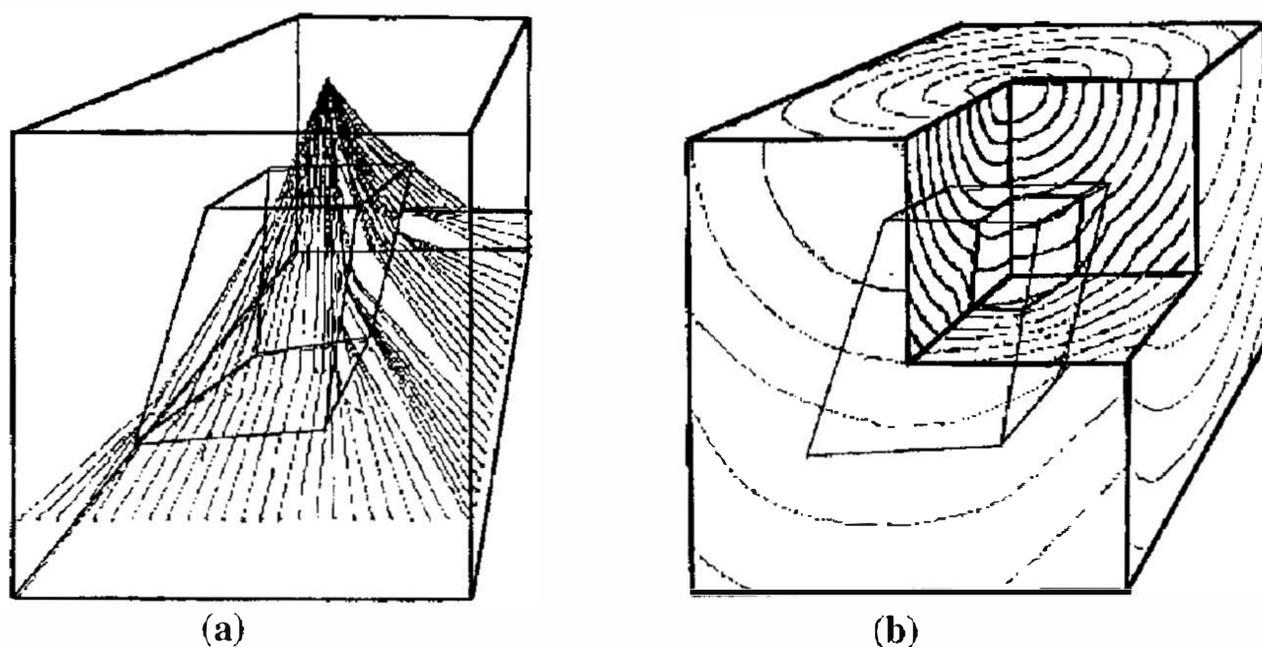


Fig. 11. The corresponding shortest raypaths and traveltimes obtained from the model. (a) The rays radiated from the source located on the center of the top surface to some receivers on outer boundaries. The refracted rays are well depicted. (b) The 3-D view of wavefronts passing through the 3-D medium.

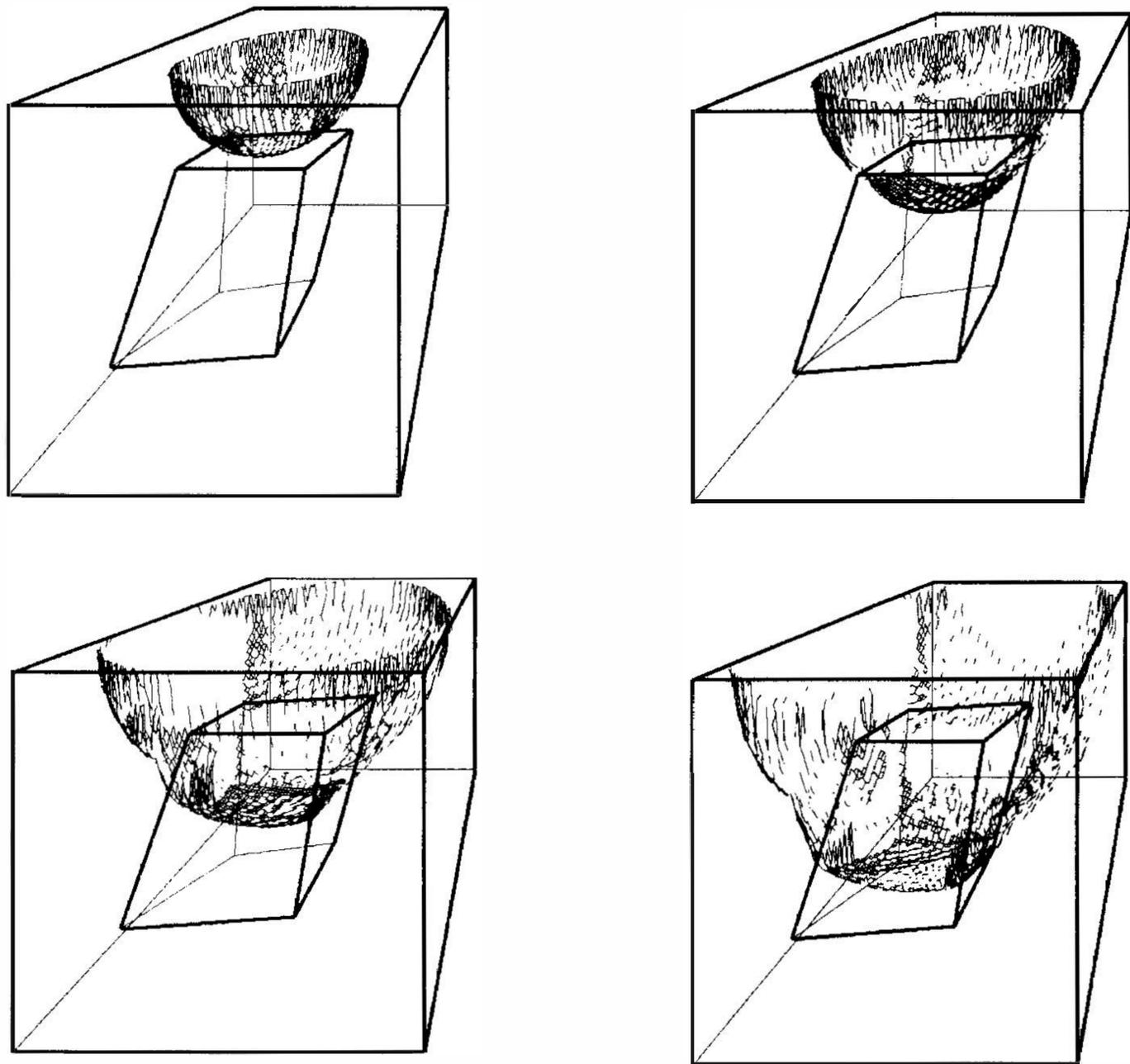


Fig. 12. Successive expanding of first arrival traveltime surfaces (wavefronts) in the 3-D medium.

from a source to all grid points in the model can be constructed in one pass. An even higher level of parallelism can be achieved by the parallel computing of several families of shortest paths from different sources. The approach in this paper is relatively new with respect to the parallel nature of the algorithm. Additionally, the improvement of Moser's method provides an intuitive approach to calculate traveltime and raypath. The computational efficiency and accuracy of the scheme are the major concerns of the current shortest raypath approach. The scheme is more flexible than the conventional approach for it allows for the earth model to be composed of a homogeneous convex polygon of irregular size. The proposed method also allows for coarse grid along the edges for traveltime and raypath calculation. The implementation of the parallel version of the algorithm on a distributed memory MPP system ensures the entire scheme is rapid and efficient.

Since the approach here is developed for a homogenous cell, the accuracy of the computed traveltime is only related to machine run-off error (i.e., machine epsilon). Comparing the accuracy between the calculated and analytical solutions is unnecessary, which is in con-

trast to the finite-difference approach of traveltime computing that inherits an anisotropy property of traveltime values at each grid point. The computational efficiency and accuracy can be justified on the basis of the complexity of the model and the requirements in their own application. In fact, for a 2-D case, such as the model in Figure 4, the execution time on an MPP machine takes much less than a second. The scalar version of the program which runs on a sun Sparc/20, including input and output to a disk, takes only 0.3 second of wall clock time.

The shortest raypath method is a rapid yet reliable approach for computing first arrival traveltime at every grid of the 2-D and 3-D block model. The originality of the algorithm is the searching of the shortest raypaths from a source point to all nodes of the model network. The traveltime at any location of the model can be calculated from values on the cell boundaries. Due to the parallel nature of the scheme, the algorithm is particularly appropriate for implementation on a MPP system.

The algorithm is, in principle, applicable to the model of an isotropic medium of any kind. So far, the authors have only considered the raypaths in cells of a constant velocity. For more practical velocity distributions, a heterogeneous media with continuously varying velocity blocks should be considered. In such a velocity model, rather than a straight segment, raypaths may bend smoothly within the cells. As an alternative approach to accommodate variable velocity in a heterogeneous medium, the model can be constructed by using small blocks (the constant-velocity blocks should be sufficiently small) compared to the minimum resolution required. However, the primary purpose in the current algorithm here aims to maintain the advantages of using larger block size for better approximation. Therefore, this approach has been ruled out. The problem can be resolved by computing the bending raypath within the cell. That will be the future research of the authors.

The current approach is developed with the computation of first arrival traveltime over every grid of any model in mind. Head waves, refracted and diffracted rays are automatically computed. The approximation of reflected rays can be obtained by simultaneously computing all the rays from all source locations along the reflector boundary.

Potential applications of the algorithm include real-time earthquake hypocenter determination for earthquake seismology studies and the computation of excitation time imaging condition for prestack depth migration. Furthermore, real-time parallel computing of traveltime from any earthquake to any specified location may become one of the very important factors in designing an early warning system and the earthquake mitigation program which are currently being investigated in Taiwan.

Nevertheless, the experiments of the method on the 2-D and 3-D block models show the efficiency and accuracy of the algorithm even in the presence of extremely severe, arbitrarily shaped velocity contrasts. The scheme's parallel nature makes the proposed algorithm particularly appropriate for implementation on a Massively Parallel Processor type system that permits the displaying of results interactively on a graphic terminal.

Acknowledgments The author would like to thank the National Science Council of the Republic of China (ROC) for its financial support of this research under grant number NSC 84-2621-P-194-003B. The computations were performed on a 256 PEs T3D MPP system installed at Cray Research headquarters in Eagan, Minnesota. Cray computation time was also

provided by Cray Research Inc. The authors thank two anonymous reviewers for their constructive comments and suggestions.

REFERENCES

- Barriuso R., and A. Knies, 1994: SHMEM Technical Note for C. Cray Research, Inc. Report, Eagan, MN.
- Chang, W. F., and G. A. McMechan, 1986: Reverse-time migration of offset VSP data using the excitation time imaging condition. *Geophysics*, **51**, 67-84.
- Chen, H. W., and G. A. McMechan, 1992: 3-D Pre-stack depth migration for salt and subsalt structures using reverse-VSP data. *J. Seism. Expl.*, **1**, 281-291.
- Dellinger, J., 1991: Anisotropic finite-difference traveltimes, Expanded Abstracts, 61st Annual Int. SEG Meeting, 1530-1533.
- Dijkstra E., 1959: A note on two problems in connection with graphs. *Numer. Math.*, **1**, 269-271.
- Fischer R., and J. Lees, 1993: Shortest path ray tracing with sparse graphs. *Geophysics*, **58**, 987-996.
- Julian, B., and D. Gubbins, 1977: Three-dimensional seismic ray tracing. *J. Geophys.*, **43**, 95-113.
- Koeninger, K., 1992: Massive Parallel Performance at Cray Research. Cray Channels, Eagan MN.
- Klimes L., and M. Kvasnicka, 1994: 3-D network ray tracing. *Geophys. J. Int.*, **116**, 726-738.
- Moser, T., 1991: Shortest path calculation of seismic rays. *Geophysics*, **56**, 59-67.
- Nakanishi, I., and K. Yamaguchi, 1986: A numerical experiment on nonlinear image reconstruction from first-arrival times for two-dimensional island arc structure. *J. Phys. Earth*, **34**, 195-201.
- Numrich, R., 1994: The Cray T3D Address Space and How to Use It, Cray Research, Inc. Report, Eagan, MN.
- Pase, D., P. MacDonald, and A. Meltzer, 1992: Fortran Programming Model, Cray Research, Inc. Report, Eagan, MN.
- Qin, F., Y. Luo, K. Olsen, W. Cai, and G. Schuster, 1992: Finite-difference solution of the Eikonal equation along expanding wavefronts. *Geophysics*, **57**, 478-487.
- Reshef, M., and D. Kosloff, 1986: Migration of common-shot gathers. *Geophysics*, **51**, 324-331.
- Saito, H., 1989: Traveltimes and raypaths of first arrival seismic waves: computation method based on Huygens' principle. Expanded Abstracts, 59th Annual Int. SEG Meeting, 244-247.
- Saito, H., 1990: 3-D ray-tracing method based on Huygens' principle. Expanded Abstracts, 60th Annual Int. SEG Meeting, 1024-1027.
- Um, J., and C. Thurber, 1987: A fast algorithm for two-point seismic ray tracing. *Bull. Seism. Soc. Am.*, **77**, 972-986.
- Van Trier, J., and W. W. Symes, 1991: Upwind finite-difference calculation of traveltimes. *Geophysics*, **56**, 812-821.

- Vidale, J., 1988: Finite-difference calculation of traveltimes. *Bull. Seism. Soc. Am.*, **78**, 2062-2076.
- Vidale, J., 1990: Finite-difference calculation of traveltimes in three dimensions. *Geophysics*, **55**, 521-526.