# Chapter 3

# The Ray-tracing Algorithm

In this chapter the FORTRAN77 program used to calculate seismic first arrival times and ray-paths in media characterised by either smooth or sharp velocity contrast is presented. The algorithm approximates linearly varying slowness inside cells defined by a regular grid. In this way direct, refracted, diffracted as well as diving rays can be modelled. The algorithm employs approximate ray propagation equations that allow for a very fast computation. The effects of the approximations used in the algorithm have been assessed by comparison with a widely used algorithm and have been found to be negligible. This algorithm has been chosen for the seismic refraction experiments presented in Chapter 5 because its speed as well as its accuracy are well suited for its use as forward routine in the global inversion performed by Genetic Algorithms.

## 3.1 Introduction

Ray-tracing algorithms are widely used to model the observed travel times of seismic waves. These algorithms must be able to accurately trace rays through complex velocity distributions and for most applications they must also be efficient both in terms of speed and memory requirements. This is particularly true in the case of seismic tomography where datasets are large and there is widespread use of inverse modelling. The tomographic inversion of travel-time data usually involves the use of search algorithms that span the solution space by iterative calls to a ray-tracing routine. Global inversion techniques such as Genetic Algorithms and

Simulated Annealing require a large number of function evaluations in order to effectively search the solution space. In seismic applications this means that a large number of ray-tracings is required. Accordingly there is growing need for very fast ray-tracing routines.

Vidale [8] described a ray-tracing algorithm in which ray propagation was modelled by calculating the travel times between a network of nodes within the seismic slowness field. This idea of approximating ray-tracing with concepts inherited from graph theory has been adopted by other authors, and a number of variations to that algorithm followed [6, 7, 4, 5, 1, 2]. In the algorithms proposed by Vidale [8] and Moser [6] the traveltime calculation is performed at regularly spaced nodes and the slowness is allowed to vary linearly from node to node. In the other papers a scheme is adopted in which the slowness field is discretised by constant slowness cells and the calculation is performed only at the cell borders, resulting in a very fast computation. However, limitations imposed by the use of constant velocity cells may invalidate the improvement in speed obtained by performing calculations only at the cell borders, since complex media may be reconstructed only by the use of a large number of cells, which in turns slows down the computation. In this chapter a modification to the algorithm proposed by Asakawa and Kawanaka [1] is presented in which this limitation is addressed and a method to approximate linear variation of slowness within the velocity field is proposed.

## 3.2 Asakawa and Kawanaka's algorithm

In Asakawa and Kawanaka's method [1] the slowness field is discretised by the use of rectangular constant slowness cells. The ray propagation calculations are performed only at discretised points located at the borders of the cells. These points are here called calculation points.

The seismic ray propagation from one cell to the other is governed by the 'linear traveltime interpolation scheme' according to which the traveltime at any location along the border of a cell is the linear interpolation between the travel times at adjacent calculation points. The procedure is illustrated in Figure 3.1. Suppose the traveltime at points $A$ and $B$ has already been calculated and the traveltime

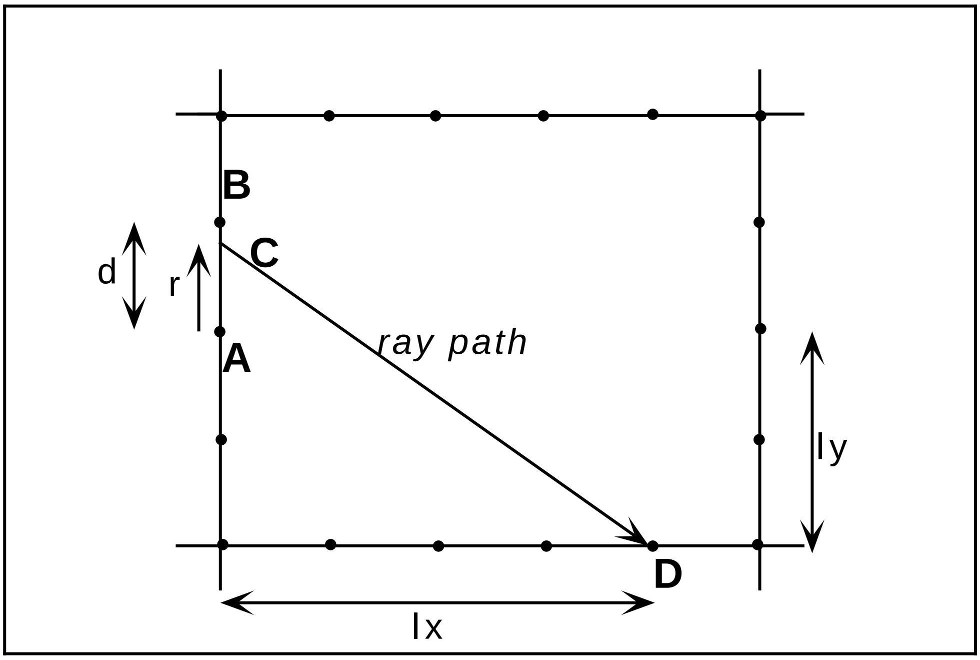


Figure 3.1: Visual representation of the procedure described by equations 3.4 and 3.5. The point C is allowed to vary continuously between point A and B. The actual location of point C is chosen is such a way that the arrival time at point D is minimum.

to point $D$ is required. Following Fermat's principle we determine the point $C$ in the segment $A-B$ in which the ray crosses the cell border, that, once connected to the point $D$, gives the fastest traveltime to the point $D$.

Due to the 'linear traveltime interpolation scheme', the traveltime at point $C$ is given by:

$$T_C = T_A \frac{d-r}{d} + T_D \frac{r}{d} \quad (3.1)$$

where $r$ is the distance between point $C$ and $A$, $T_A$ is the traveltime at point $A$, $T_B$ is the traveltime at point $B$ and $d$ is the distance between point $A$ and $B$.

Thus, the traveltime to point $D$ is:

$$T_D = T_C + S\sqrt{l_x^2 + (l_y + r)^2} \quad (3.2)$$

where $l_x$ is the horizontal distance between $D$ and $A$, $l_y$ is the vertical distance between the same points and $S$ is the slowness inside the cell.

This equation can be conbined with equation 3.1 to give:

$$T_D = T_A \frac{d-r}{d} + T_D \frac{r}{d} + S\sqrt{l_x^2 + (l_y + r)^2} \quad (3.3)$$

According to Fermat's principle, the correct traveltime to point $D$ is chosen by determining the minumum possible value for $T_D$, i.e., by differentiating equation 3.3 with respect to $r$.

This gives:

$$T_D = T_A + \Delta T \frac{l_y}{d} + \frac{l_x}{d}\sqrt{S^2 d^2 - \Delta T^2} \quad (3.4)$$

with:

$$r = \frac{\Delta T l_x}{\sqrt{S^2 d^2 - \Delta T^2}} - l_y \quad (3.5)$$

where $\Delta T$ is the difference $T_B - T_A$. The reader is referred to Asakawa and Kawanaka original paper for a detailed description of the method and the derivation of the above equations.

The traveltimes for all possible paths connecting the seismic source to a calculation point in the slowness field are calculated by the iterative use of these equations. How these process is carried out is described in section 3.5. Then Fermat's principle is used to select the correct traveltime, i.e., the one corresponding to the fastest arrival. Once this is done for all the calculation points a discretised 'traveltime field' across the entire domain is obtained.

It is important to realise that the ray-paths are not required to determine the arrival time at the receivers. During the calculation of traveltimes the ray paths are not stored, since the process of storing the minimum traveltime paths at this stage would be massive from both a computational and memory point of view. If ray-paths are required these are calculated once the traveltime field across the entire domain has been obtained. At this stage ray paths may be easily traced by running backward from the receiver positions to the source. This is achieved by connecting a receiver to the calculation point with the lowest traveltime value in the cell in which the receiver lies and so on until the source location is reached.

If ray-paths are traced, an alternative way to obtain the traveltimes is to add the results of multipling the length of individual segments of the ray-path by the corresponding slownesses. A discussion about the accuracy of the two alternatives in given below in the section 'Accuracy and efficiency'.

We have seen that the calculation points are located only at the borders of the cells. No calculation points within the individual cells are used. This is the key to the algorithm's efficiency.

The fact that the ray-paths are allowed to cross the cells borders at any location (i.e. not only at the calculation points) through the use of equations 3.4 and 3.5, is important to mantain the accuracy of the calculation. Details the accuracy of the algorithms are given in the original paper by Asakawa and Kawanaka.

This algorithm was used in the physical model experiment presented in Chapter 5.

## 3.3 The linear-slowness variation approximation

Modelling diving rays is very important for refraction experiments. Equations 3.4 and 3.5 allow only for the modelling of direct, refracted and diffracted rays in velocity fields characterised by both smooth and sharp velocity variations. For example, the constant slowness assumption of Asakawa and Kawanaka does not allow for diving rays inside horizontal layers. In this case the rays are always refracted along the upper border of the cells. This happens because if the rays dived into the constant slowness layers, they would take a path longer than the one due to the refraction along the upper border. With the slowness constant, such a path would result in a longer traveltime, that can not be chosen because of Fermat's principle.

With this configuration the only way to reduce the errors in the travel time estimation would be to subdivide horizontal layers into a large number of thin layers, i.e., to use a large number of constant-slowness cells. However, this would result in a significant increase in the computation effort.

The model and equation 3.4 can be modified to approximate linear variation of slowness inside the cells as follows. While in the original algorithm a constant slowness value is assigned to a cell, in this modified algorithm the slowness field is defined at the nodes of a regular grid, i.e. at the vertices of the cells. The slowness value at the calculation points at the borders of the cell is then assumed to be the linear interpolation of the slowness values at the two vertices of the cell border. The seismic ray is then assumed to propagate inside the cell with a slowness equal to the average between the slowness values at the points where the ray crosses the cell borders. A similar approach applied to a regularly spaced grid nodes scheme has been already proposed in the literature and can be found in [8, 6]. Consequently equation 3.4 changes so that $S$ becomes the average between the slowness at points $C$ and $D$. This simple modification allows for rays to dive inside flat layers with vertical velocity gradients.

The accuracy of this bi-linear approximation is discussed in the next section. Notice that a proper linear slowness variation in the domain could be modelled only by the use of triangular cells.

## 3.4 Accuracy and efficiency

The accuracy and speed of the algorithm is a function of the number of cells in the domain and of the numbers of calculation points along the cell borders.

In section 3.2 it has been shown that traveltimes can be evaluated either with or without ray-tracing. The possibility of obtaining traveltimes without actually tracing the rays is not mentioned in the original paper by Asakawa and Kawanaka. The travel times calculated with or without ray-tracing differ because both the original and the modified equations presented above involve some kind of approximation. The effect of the number of calculation points along the cell borders and the calculation of travetimes with or without ray-tracing is now discussed.

The accuracy of the algorithm including the linearly varying slowness approximation and the influence of the number of points along the cells borders have been tested numerically against the SEIS83 ray tracing algorithm [3], an algorithm often used for seismic refraction applications. A comparison has been made using synthetic first-arrival travel time data generated using a model defined by a 9x5 grid whose spacing is 1000 m in the horizontal direction and 100 m in the vertical direction. A linear slowness vertical gradient is assumed between the grid nodes (the same model has been used to generate the synthetic seismic refraction data

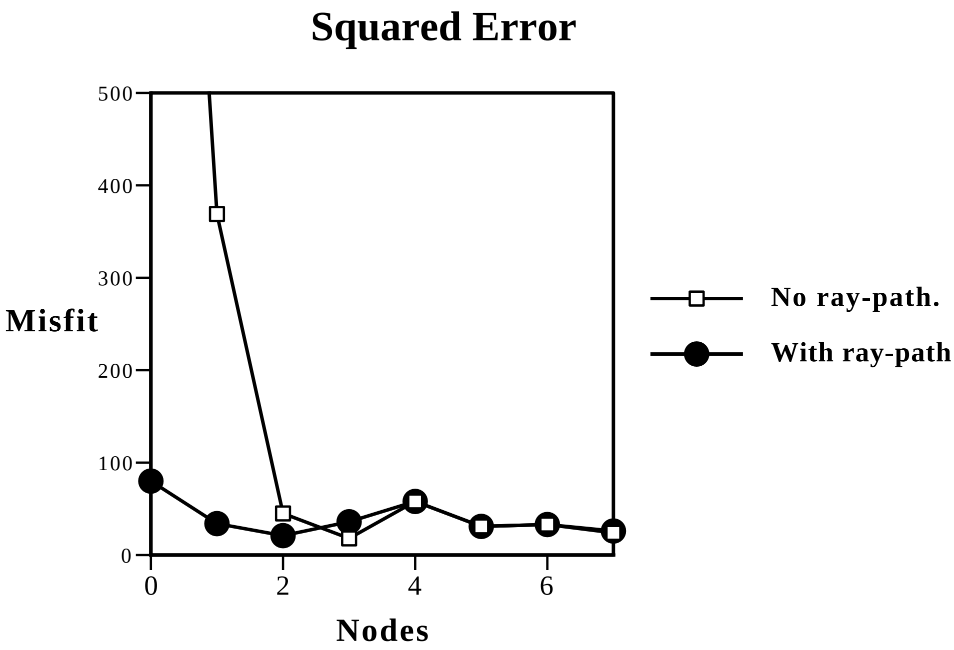


Figure 3.2: Squared misfit between the traveltimes calculated by SEIS83 and the algorithm here presented both with (filled circles) and without raytracing (squares) as a function of the number of nodes along the cells border.

presented and discussed in more detail in Chapter 5. The arrival times from a source positioned at one end of the model to 9 regularly spaced receivers have been calculated using both algorithms. The misfit (calculated in milliseconds) between the two data sets has been squared and shown in Figure 3.2 as a function of the number of nodes along the cells borders.

The algorithm has been run both with and without ray-tracing. Clearly, for the number of calculation points along the cells borders equal to or larger than 2, the discrepancy between the two sets of travel-times tends to become roughly constant and negligible. In order to better evaluate the misfit between the two algorithms the actual numeric results expressed in milliseconds for the implementation with 3 calculation points along the cell borders and no ray-path computation is presented in Table 3.3 and compared with the results from SEIS83. The disagreement never exceeds 1% and decreases at increasing source-receiver offsets.

Figure 3.4 shows the CPU time, expressed in seconds in a 486DX-PC with a 33 MHz processor for increasing numbers of calculation points along the cell borders in the synthetic test previously described, for travel times calculated both with

| Receivers | SEIS83 | Ray-Tr. code | Difference % |
|---|---|---|---|
| 1 | 198.1 | 200.0 | 0.95 |
| 2 | 295.0 | 296.2 | 0.40 |
| 3 | 390.1 | 392.1 | 0.51 |
| 4 | 484.9 | 487.7 | 0.57 |
| 5 | 579.2 | 578.5 | 0.12 |
| 6 | 659.2 | 658.4 | 0.12 |
| 7 | 738.3 | 738.2 | 0.01 |
| 8 | 817.3 | 817.9 | 0.07 |
| 9 | 897.7 | 897.3 | 0.04 |

Figure 3.3: Numeric comparison between SEIS83 and the code here presented implemented with 3 calculation points along the cells borders and no ray-path computation. The results in column 2 ans 3 are presented in milliseconds while the difference in column 4 is given in percentual.

and without ray-tracing. The advantage of reducing the number of nodes and avoiding raytracing is evident. From Figures 3.2 and 3.4 it is clear that by using only 2 or 3 nodes along the cell borders and no ray-tracing a good compromise between speed and accuracy can be obtained.

Finally, Figure 3.5 shows the ray-diagram for the slowness model presented above, in which 15 seismic sources and 54 seismic receivers has been modelled. This is intended to simulate a seismic refraction survey. Since the travel-time/ray path calculation is performed only at the cell borders, there is no bending of rays inside the cells but the presence of diving rays within the model is clearly seen. Such a result would have not been possible with constant velocity cells.

## 3.5 Algorithm implementation

The algorithm's implementation in FORTRAN77 is presented in Appendix A. The implementation of equations 3.5 and the modified equation 3.4 is straightforward and is carried out in the subroutine LINSHOT.

In order to show how the traveltimes are evaluated at each calculation point consider the example in Figure 3.6 where the seismic first arrival from the seismic source to point $D$ is sought (notice that for the sake of clarity only one calculation

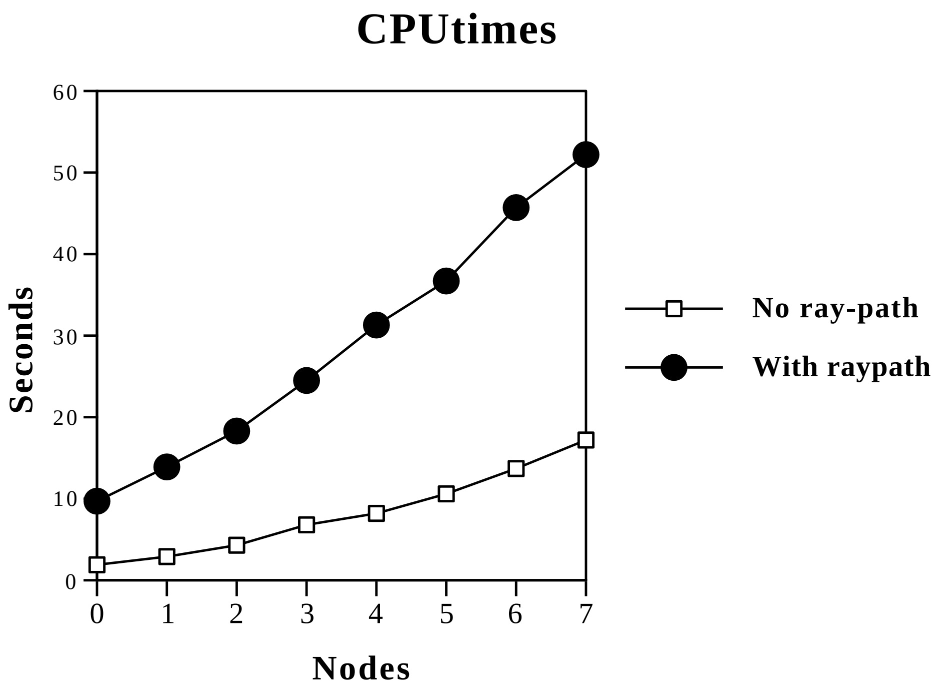


Figure 3.4: CPU time expressed in seconds in a 486DX-PC with a 33 MHz processor for increasing number of points along the cells borders for traveltime calculation both with (filled circles) and without (white squares) raytracing.

point along the cell borders has been modelled here). First the seismic ray is shot from the seismic source to all calculation points at the borders of the cell in which the source is located (see dashed segments in the left-hand side cell in Figure 3.6). Then the traveltime to points located in the cells adjacent to the source cell are calculated. For example, the arrival time at point $D$ is then evaluated by checking all the possible paths the seismic ray could take from the source to $D$. In the example in Figure 3.6 a ray can reach $D$ by crossing the cell anywhere in segment $AB$ or $BC$. Thus, equations 3.5 and 3.4 are applied to segments $AB$ and $BC$. This assures that the traveltimes along all the available paths from the source to point $D$ are calculated and the fastest arrival time is selected, according to Fermat's principle.

The procedure mustt be performed for all calculation points in the domain. The calculation of the ray propagation must proceed outwards from the cell containing the source. This operation is performed by the subroutine RAYMAIN and it is sketched in Figure 3.7. First the calculation is performed for all the cells in the

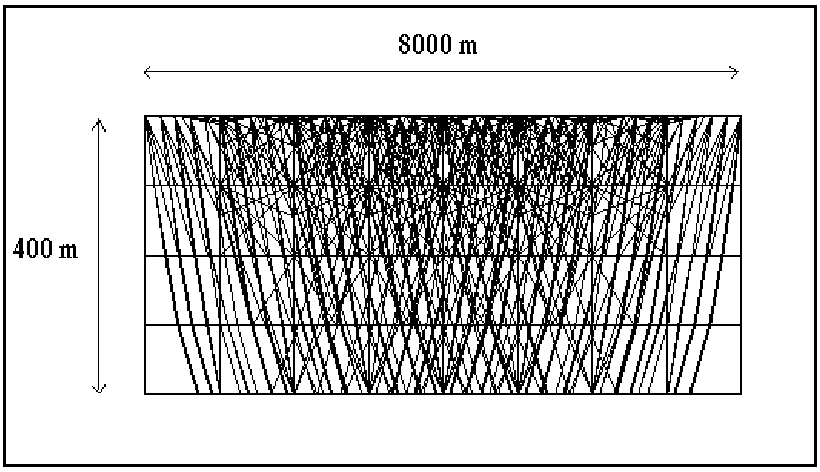


Figure 3.5: Ray-diagram for a vertical linearly varying slowness layered model.

column where the source is located downward (arrow 1 in the picture). Then from the right-hand side of this column rays are shot from each point in the right-hand direction, and similar procedure is accomplished on the left-hand side until all the domain is covered (arrows 2 in the picture). Eventually, when all the calculation points have been reached the travel time field is completely defined. At this stage the arrival time at the location of any receiver is similarly calculated by first connecting the receiver to any point in the cell within which the receiver lies. This creates a number of possible arrival times among which the minimum is then chosen. In the program the ray propagation is performed by the routines TRASMITX, TRASMITY, DIFFRACT that control the propagation from the calculation points located on the horizontal axis, the vertical axis and the cell vertices, respectively.

Care must be taken to ensure all the possible ray paths are checked and that no redundant calculation is performed, to keep the code as fast as possible. Particular attention should also be paid to ensure that refraction does not occur at the borders of the cells in more complex velocity distributions. Once the calculation for each cell is completed, possible refraction along the horizontal axis must be checked before shooting at adjacent cells in order to avoid possible errors to be propagated

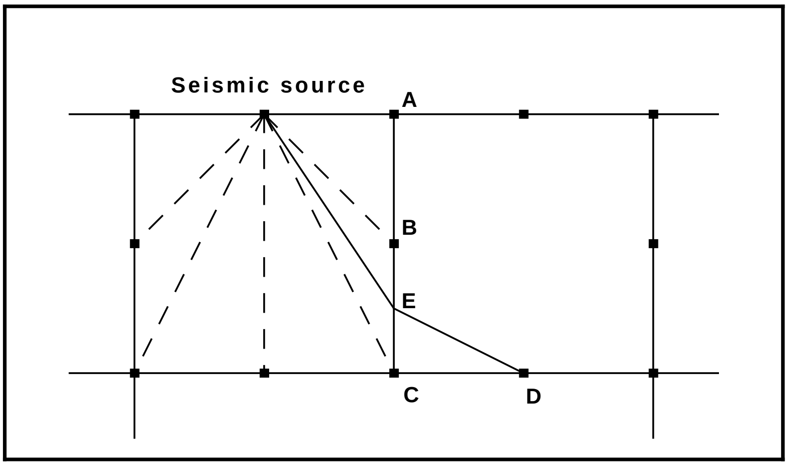


Figure 3.6: Schematic description of the ray propagation form the seismic source to any calculation point in the domain.

in the entire domain. Similarly, once the traveltimes have been determined at the vertical borders of a column of cells, refraction along the vertical axis should also be checked. This task is accomplished by the subroutines VREFR1 and VREFR2.

If the ray-paths are required a similar procedure must be performed backwards. From any receiver location the point at which the ray-path crosses the border of the cell where the receiver lies is calculated using equation 3.5 and then stored. The calculation is then performed iteratively from such points backwards until the source is reached and the path completed. Such a task is performed by the subroutines BACKDIFF, BACKX and BACKY.

## 3.6 Discussion

The aim of this research is to test the applicability of Genetic Algorithms to geophysical problems, including seismic refraction tomography. Genetic Algorithms, as a global optimisation technique, require a large number of forward calculations. On small workstations commonly used in research this can be best achieved by the use of very fast routines, similar to the one presented here.

As has been stated above Asakawa and Kawanaka's linear traveltime interpolation scheme is simply an approximation of the more complicated equations governing the propagation of seismic energy in complex media. The same applies to the modified equation 3.4. Also, the linearly varying slowness is not taken into

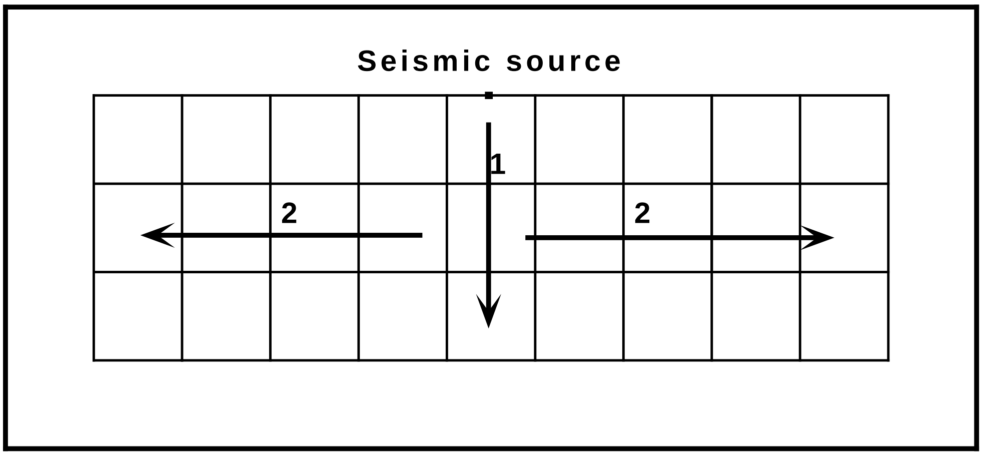


Figure 3.7: Schematic description of how the ray propagation should be performed. First the calculation is performed for all the cells in the column where the source is located downward (arrow 1) and then outwards (arrows 2)

account in equation 3.5. An accurate solution for equation 3.5 with the assumption of linear slowness variation has been calculated but resulted in a complex calculation that strongly affects the efficiency of the algorithm.

It is worthwhile drawing attention to a number of observations in the use of the algorithm presented in this chapter:

- Asakawa and Kawanaka's linear traveltime interpolation scheme has been tested in the original paper and a further test of its validity has been performed on the physical model experiment described in Chapter 5. In this experiment the original equation 3.4 and not the modified one, was used because, as it will be shown in Chapter 5, the medium under analysis was characterised by constant velocity layers;

- the linearly varying slowness approximation has been tested against the widely used SEIS83 algorithm that employs more accurate modelling of the ray propagation equations, and the results were shown to be particularly satisfactory;

- the statements in the comments above refer to tests on models characterised by relatively simple geometry. Far more complicated media has been modelled in the Nevoria experiment presented in Chapter 5 where the results obtained with Genetic Algorithms showed a good agreement with results from previous studies on the same area. In Chapter 7 an extensive study of the effects on the inversion process due to inaccuracies in the forward calculation is presented where it is shown that no accurate results may be achieved if large errors are present in the forward calculation. Accordingly, the good results obtained with Genetic Algorithms in Chapter 5 should be considered as an indirect demonstration of the satisfactory accuracy of the ray-tracing routine here presented. A more detailed discussion of this point is presented in Chapter 7;

- an acceptable balance between accuracy and speed in the forward calculation has been achieved with the use of this ray-tracing routine. If Genetic Algorithms are to be routinely applied to more general seismic refraction problems the inclusion of more accurate modelling of ray propagation in complex media need to be carefully considered. This last point is further analysed in the conclusions presented in Chapter 8 where directions for possible improvements are given.

# Bibliography


[1] Asakawa and Kawanaka. Seismic ray tracing using linear travel-time interpolation. *Geophysical Prospecting*, 41:99–111, 1993.

[2] S. Cao and S. Greenhalgh. Calculation of the seismic first-break time field and its ray path distribution using minimum traveltime tree algorithm. *Geophysics*, 114:593–600, 1993.

[3] V. Cerveny and I. Psencik. Seis83-numerical modeling of seismic waves fields in 2-d laterally varying layred structures by the ray method. In *Documentation of Earthquake Algorithms*, pages 36–40. World Data Center for Solid Earth Geophysics, 1984.

[4] R. L. Coultrip. High-accuracy wavefront tracing traveltime calculation. *Geophysics*, 58:284–292, 1993.

[5] R. Fisher and J. M. Lees. Shortest path ray tracing with sparse graphs. *Geophysics*, 58:987–996, 1993.

[6] T. J. Moser. Shortest path calculation of seismic rays. *Geophysics*, 51:59–67, 1991.

[7] W. A. Schneider, K. A. Ranzinger, A. H. Balch, and C. Kruse. A dynamic programming approach to first arrival traveltime computation in media with arbitrarily distributed velocities. *Geophysics*, 57:39–50, 1992.

[8] J. Vidale. Finite-difference calculation of traveltime. *Bull. of Seism. Soc of Am.*, 78:2062–2076, 1988.