Interpolation of ray-theory travel times within ray cells

Petr Bulant & Luděk Klimeš
Department of Geophysics, Charles University, Ke Karlovu 3, 121 16 Praha 2, Czech Republic. Fax: +420-2-2191 1214, Tel: +420-2-2191 1213,
E-mails: bulant@seis.karlov.mff.cuni.cz, klimes@seis.karlov.mff.cuni.cz

1 Introduction

3-D ray tracing, especially the computation of ray-theory travel times at nodes of dense grids of points has become quite important in recent years. The possibility to compute the travel times is of principal importance in imaging of depth structures, seismic tomography or hypocenter determination techniques. Several methods based on decomposition of the model volume into ray cells, and on further interpolation within individual ray cells, were introduced.

In the *wavefront tracing* method (Vinje *et al.* 1996a, 1996b) the entire wavefront is represented by a triangular mesh with a ray at each node. The wavefront is time-step by time-step propagated through the model. The space between two consecutive wavefronts is decomposed into ray cells, each of them being bounded by a triangle at the older wavefront, by the corresponding triangle at the newer wavefront, and by segments of the three rays connecting the triangles. The receivers located within the ray cells are assigned values of interpolated travel times and other quantities. If there are some difficulties with the interpolation, new rays may be computed, and the ray cell may be split into smaller cells.

In the *recursive seismic ray modelling* method by Moser and Pajchel (1997), the computation is organized in "depth-first search" rather than "breadth-first search", as it is in the wavefront tracing method. The computation of a selected ray cell is followed by the computation of the cell in the next time step. At each interface, ray tubes corresponding to all the elementary waves under consideration are recursively generated. This procedure is continued until the whole ray tube, composed of individual ray cells, is computed, including reflections, transmissions and diffractions. From the point of view of interpolation, the main difference as compared to the wavefront tracing method is that the neighbouring ray cells are not available when interpolating in a ray cell.

In the *controlled initial-value ray tracing* method (Bulant 1997a,b) just ray tubes corresponding to a prescribed elementary wave are considered, without splitting at structural interfaces. The model volume is decomposed into ray tubes in the first step, and the interpolation within the ray cells is carried out in the second step. This is because the method was proposed as a post-processing method to a two-point ray tracing algorithm based on the triangularization of the domain of ray take-off parameters (for details, see Bulant 1997a,b). Thus the system of ray tubes and ray cells is given before the interpolation process takes place. Each ray cell corresponds to the space in the ray tube, limited by two planes which approximate wavefronts or structural interfaces. The values of the interpolated quantities at the vertices of a ray cell are available for interpolation.

In: Seismic Waves in Complex 3-D Structures, Report 7 (Department of Geophysics, Charles University, Prague, 1998), pp. 17–32

Whereas the above-mentioned methods differ in the manner of the decomposition of the model volume into ray cells, they all contain some procedure of interpolation of the computed quantities to the receivers (usually nodes of a receiver grid) located within a single ray cell. We propose bilinear and bicubic interpolation schemes. The bilinear interpolation is easy and robust. It incorporates both the decision, whether a receiver lies in the ray cell, and the interpolation of quantities computed along the rays to the receiver. The scheme uses only the values of quantities at the vertices of the ray cell. Thus the scheme should be applicable in any modelling method based on interpolation within ray cells. If the partial derivatives are known in addition to the functional values at the vertices of the ray cell, the bicubic interpolation scheme may be used to increase the accuracy of the interpolation. The bicubic interpolation is designed for travel times and the bilinear interpolation for other quantities.

2 Decomposition of a ray tube into ray cells

As mentioned above, first of all we trace rays and decompose the model volume into ray tubes. Each *ray tube* is bounded by three rays, which form a triangle in a 2-D domain of ray take-off parameters. We triangularize the ray parameter domain by so-called homogeneous triangles. Each *homogeneous triangle* is formed by rays of the same *ray history*, i.e. by rays which pass the same model blocks, interact with the same interfaces, display the same behaviour at caustics, and terminate in the same area for the same reason. Thus the width of the ray tubes may be kept below a prescribed limit, and the possibility of interpolation within the ray tubes is guaranteed. See figure 1.

For details on triangularization of the ray–parameter domain by homogeneous triangles (i.e. for details on creating the ray tubes) refer to Bulant (1996). Note that the demarcation belts between homogeneous triangles of different ray histories produce gaps between the corresponding ray tubes. These gaps may then cause gaps in the interpolated quantities in the receiver grid, where travel times and other quantities cannot be interpolated. To reduce the gaps as much as possible, we must carefully decide, what information we wish to include in the ray histories.

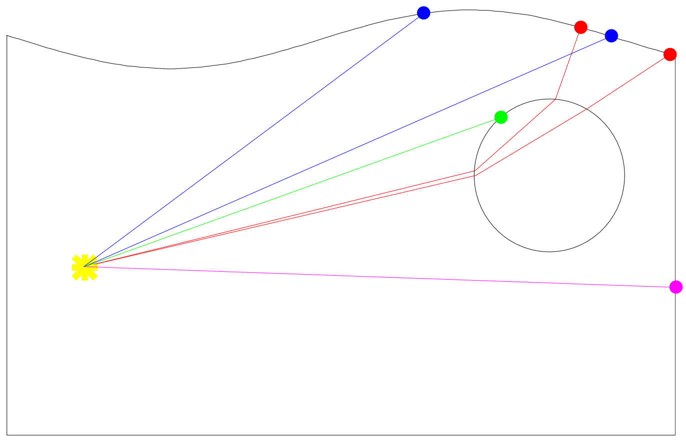

Figure 1: Rays with different ray histories in a seismic model. Only rays with the same ray histories, displayed here in the same colours, are suitable for creating ray tubes to be used to interpolate between the rays.

After being computed, each ray is stored on a disk as a set of points of two types. First, there are the points on the ray stored with travel-time step Δt; we shall refer to them as *"time points"*. Second, there are the points of interaction of the ray with interfaces and other surfaces (reflection, transmission or termination); we shall refer to them as *"interaction points"*. A ray tube is thus represented as a set of the above points on the three rays, which form the ray tube. See figure 2. For interpolation to be feasible, the ray tube must be decomposed into smaller parts.

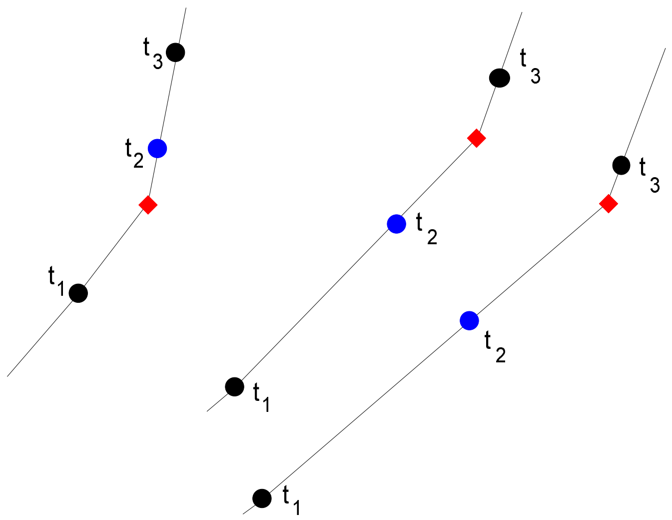

Figure 2: Points on three rays forming a ray tube in the place where the rays cross an interface. Black and blue dots represent the points at travel-time levels t_1, t_2 and t_3 ("time points"). Red diamonds are the points of intersection of the rays with the structural interface ("interaction points").

We decompose the ray tube into *ray cells*. *Regular ray cells* are defined by six points on the rays. Both the bottom and the top of the regular ray cell are formed by triangles defined by three points on the rays. Mostly these are the "time points" of the same time level t. *Degenerate ray cells* are formed by five or four points. They occur mainly at a point source, where the bottom of a ray cell is formed by a single point, before and after structural interfaces, where one or two points of the top of a ray cell coincide with the corresponding point(s) of the bottom, and before the end surface where the ray tube terminates.

When decomposing ray tubes into ray cells, we must assure, that the sides of all the ray cells are the same for neighbouring ray tubes. Otherwise some gridpoints of a receiver grid might be considered to be located within none, or within both of the neighbouring ray cells.

We start the decomposition at a source, taking the first points on the rays as the bottom of the first ray cell. Thus the bottom is formed by a single point for point sources, by a line segment for straight-line sources, and by a triangle for curved-line sources and surface sources. Then we proceed along the rays to the second points on the rays, taking the second points simultaneously as the top of the first ray cell and as the bottom of the second ray cell. We continue this procedure along the rays and create regular ray cells, until we reach the first "interaction point" on any of the rays. Here we postpone proceeding along this ray, and we proceed only along the remaining two rays (or one ray), until we reach the "interaction points" on all the three rays. We may thus create one or more degenerate ray cells.

Now the three "interaction points" form the top of the last created ray cell. If they lie at the end surface, the decomposition of the ray tube into ray cells is completed. If they are located at a structural interface, we continue the process of creating ray cells beyond this interface. We may again have to postpone proceeding along some ray(s) and we may create some degenerate ray cell(s), until we get "time points" with the same travel time at the top of the recently created ray cell. We can then continue with creating regular ray cells with equal travel times at their bottoms and tops. See figure 3.

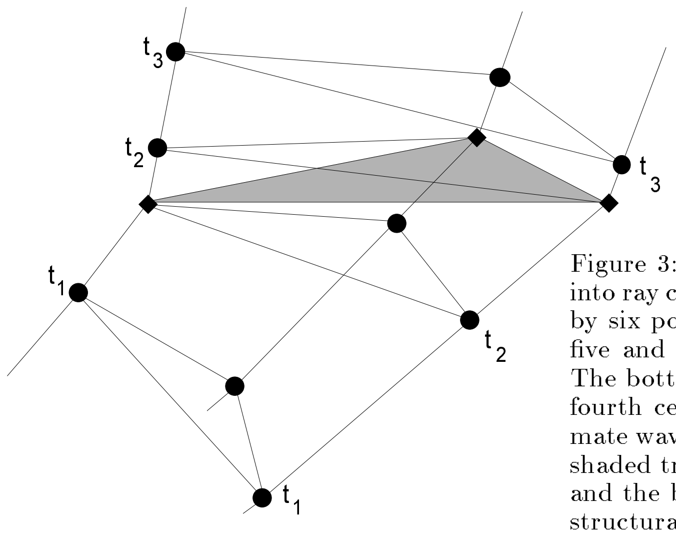

Figure 3: The ray tube from figure 2 decomposed into ray cells. We can see one regular ray cell formed by six points, then two degenerate cells formed by five and four points, and again a regular ray cell. The bottom of the first ray cell and the top of the fourth cell are formed by triangles which approximate wavefronts at travel-time levels t_1 and t_3. The shaded triangle, which is the top of the second cell and the bottom of the third cell, approximates the structural interface.

Note that once we have reached an "interaction point" on any one of the three rays, we are sure that we will reach "interaction points" of the same kind also on the remaining two rays, because all the rays of the ray tube have the same ray history. It should be also pointed out that the described method of decomposition of a ray tube into ray cells prevents us from interpolating across structural interfaces.

3 Determination of a ray cell

By determination of a ray cell we understand the decision, whether an arbitrary point lies within the ray cell, or not. Assume a ray cell defined by six points $\mathbf{B}_1, \mathbf{B}_2, \mathbf{B}_3, \mathbf{C}_1, \mathbf{C}_2, \mathbf{C}_3$ on three rays, where points $\mathbf{B}_i = ({B_i}^1, {B_i}^2, {B_i}^3)^T$ are at the bottom, and points $\mathbf{C}_i$ at the top of the ray cell, see figure 4. We introduce a local coordinate system parametrized by parameters w_1, w_2, w_B. The coordinates of arbitrary point $\mathbf{X}$ are then given as

$$\mathbf{X} = \mathbf{A}_3 + w_1(\mathbf{A}_1 - \mathbf{A}_3) + w_2(\mathbf{A}_2 - \mathbf{A}_3) \tag{1}$$

where

$$\mathbf{A}_i = \mathbf{C}_i + w_B(\mathbf{B}_i - \mathbf{C}_i), \ i = 1, 2, 3. \tag{2}$$

Equations (1) and (2) may also be expressed as

$$\mathbf{X} = w_B w_i \mathbf{B}_i + w_C w_i \mathbf{C}_i \tag{3}$$

with $w_C = 1 - w_B, \quad w_3 = 1 - w_1 - w_2$. Equation (1) is the parametric description of planar triangle $\mathbf{A}_1\mathbf{A}_2\mathbf{A}_3$. Equations (2) are parametric descriptions of straight-line segments $\mathbf{B}_1\mathbf{C}_1$, $\mathbf{B}_2\mathbf{C}_2$ and $\mathbf{B}_3\mathbf{C}_3$, approximating the ray segments. Note that only in the homogeneous parts of the model the straight segments lie on the exact rays.

If point $\mathbf{X}$ lies within the ray cell, all values w_1, w_2, w_3, w_B are within range $< 0, 1 >$. Thus, for each receiver, which might be located within the selected ray cell, we compute its local coordinates w_1, w_2, w_B and quantities w_3, w_C. If all of them display values from range $< 0, 1 >$, point $\mathbf{X}$ is within the ray cell, and the values of all required quantities will be determined by interpolation within the ray cell.

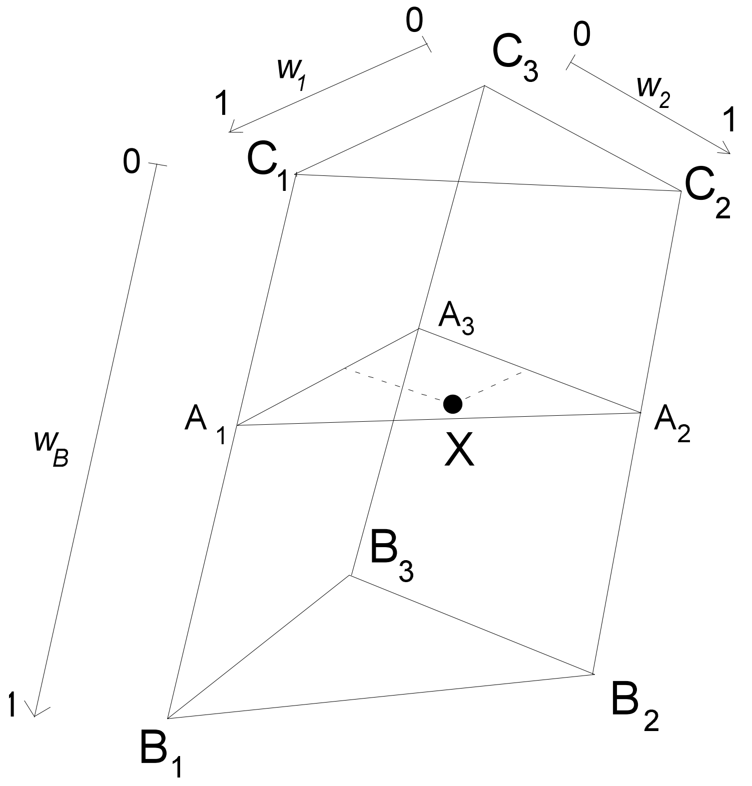

Figure 4: A ray cell is defined by six points $\mathbf{B}_1, \mathbf{B}_2, \mathbf{B}_3, \mathbf{C}_1, \mathbf{C}_2, \mathbf{C}_3$, situated on three rays. Point $\mathbf{X}$ is examined, whether it is located within the ray cell. Its position may be expressed using the local coordinate system parametrized by parameters w_1, w_2, w_B. If all of the quantities w_1,w_2,w_B,$1 - w_1 - w_2$ are in range $< 0, 1 >$, point $\mathbf{X}$ lies within the ray cell.

If the ray cell touches a caustic, two or three (in 3-D) different sets of values w_1, w_2, w_3, w_B may be found within range $< 0, 1 >$ for the same point $\mathbf{X}$. Different coordinates w_1, w_2, w_B of the same point $\mathbf{X}$ correspond to different ray coordinates $\gamma_1, \gamma_2, \gamma_3$, and the interpolation should be performed for each set of coordinates w_1, w_2, w_B.

3.1 Computation of w_B

First we compute the value of local coordinate w_B. We know that the volume V of the parallelepiped defined by three vectors $\mathbf{M}, \mathbf{N}, \mathbf{O}$ is given by the determinant of the 3×3 matrix $\widehat{\mathbf{M}}$ composed of the three vectors,

$$V = \det \widehat{\mathbf{M}} = \begin{vmatrix} M^1 & N^1 & O^1 \\ M^2 & N^2 & O^2 \\ M^3 & N^3 & O^3 \end{vmatrix} = \sum_{i,j,k=1}^{3} \epsilon_{ijk} M^i N^j O^k \tag{4}$$

where ϵ_{ijk} is a Levi-Civita symbol. Point $\mathbf{X}$ must lie in the plane determined by points $\mathbf{A}_1, \mathbf{A}_2, \mathbf{A}_3$. Thus the volume of the parallelepiped defined by vectors $\mathbf{A}_1 - \mathbf{X}, \mathbf{A}_2 - \mathbf{X}, \mathbf{A}_3 - \mathbf{X}$ should be equal to zero. According to equation (2)

$$\mathbf{A}_i - \mathbf{X} = \mathbf{C}_i - \mathbf{X} + w_B(\mathbf{B}_i - \mathbf{C}_i), \;\; i = 1, 2, 3. \tag{5}$$

Considering equations (4) and (5), we arrive at the cubic equation for parameter w_B in the form of

$$\det(\widehat{\mathbf{Z}} + w_B \widehat{\mathbf{Y}}) = a{w_B}^3 + b{w_B}^2 + c w_B + d = 0, \tag{6}$$

where $\widehat{\mathbf{Z}}$ is the 3×3 matrix composed of vectors $\mathbf{C}_i - \mathbf{X}$, $i = 1, 2, 3$, $\widehat{\mathbf{Y}}$ is the 3×3

matrix composed of vectors $\mathbf{B}_i - \mathbf{C}_i,\ i = 1, 2, 3,$

$$a = \det \widehat{\mathbf{Y}},$$

$$b = \sum_{i,j,k=1}^{3} \epsilon_{ijk} Z_1^i Y_2^j Y_3^k + \sum_{i,j,k=1}^{3} \epsilon_{ijk} Y_1^i Z_2^j Y_3^k + \sum_{i,j,k=1}^{3} \epsilon_{ijk} Y_1^i Y_2^j Z_3^k,$$

$$c = \sum_{i,j,k=1}^{3} \epsilon_{ijk} Y_1^i Z_2^j Z_3^k + \sum_{i,j,k=1}^{3} \epsilon_{ijk} Z_1^i Y_2^j Z_3^k + \sum_{i,j,k=1}^{3} \epsilon_{ijk} Z_1^i Z_2^j Y_3^k,$$

and

$$d = \det \widehat{\mathbf{Z}}.$$

Cubic term a equals zero if one of the vectors $\mathbf{B}_i - \mathbf{C}_i$ forming the ray tube is linearly dependent on the other two vectors (including the case in which the vector equals zero). Quadratic term b equals zero if two of the vectors $\mathbf{B}_i - \mathbf{C}_i$ are linearly dependent on the third. Thus for degenerate ray cells, when one or two of points $\mathbf{B}_i$ coincide with the corresponding points $\mathbf{C}_i$, cubic equation (6) reduces to a quadratic or linear equation, (*) but the interpolation scheme remains valid. This usually happens when a ray tube interacts with a structural interface, see figure 3.

We look for the roots of the cubic equation (6) only in range $< 0, 1 >$. Otherwise, the point $\mathbf{X}$ is out of the ray cell. We tried to solve the equation using analytical formulae from Numerical Recipes in Fortran (1997), but the solution was not precise enough for our purposes. Thus we solve the equation numerically. First we compute all the stationary points of the cubic function (6). Then we calculate the values of the cubic function at points 0, 1, and at all the stationary points from range $< 0, 1 >$, if any. We can thus split interval $< 0, 1 >$ into subintervals. Each subinterval is limited by points 0, 1, or by the stationary points, see figure 5.

We get zero to three subintervals of $< 0, 1 >$, such that the values of the cubic function at the endpoints of each subinterval have opposite signs, and each of the subintervals thus contains just one root of cubic equation (6). We can then easily find the roots using, e.g., the Newton method, because the function is monotonic in all the subintervals. The reader is reminded that all values of w_B from interval $< 0, 1 >$ are significant.

(*) We could thus decompose ray tubes into degenerate ray cells formed by only four points on the rays and gain from this important simplification of the cubic equation to a linear equation. This possibility will be studied in the future.

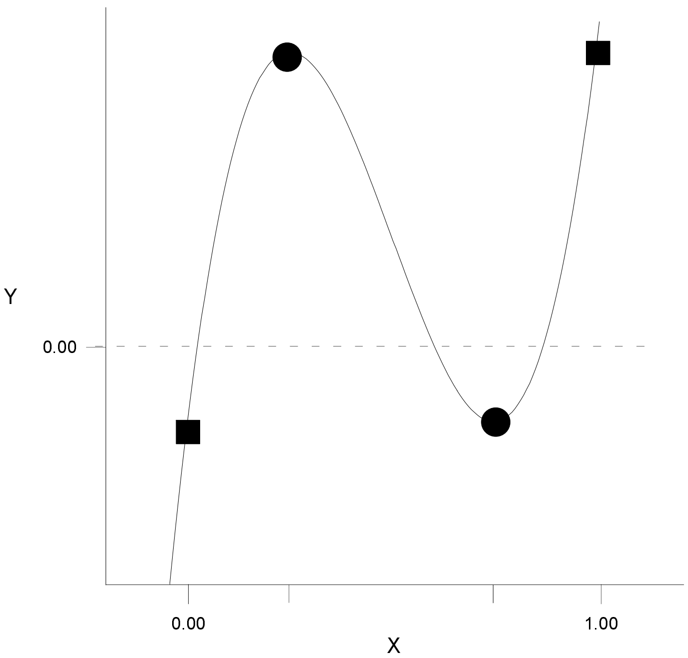

Figure 5: A cubic function with three roots in range $< 0, 1 >$. Once we have computed the two stationary points (marked by circles), the interval $< 0, 1 >$ may be split into three subintervals. Each subinterval contains just one root. The cubic function is monotonic in all the subintervals.

3.2 Computation of w_1, w_2

Once we have computed w_B from interval $< 0, 1 >$, we proceed to the computation of w_1, w_2, w_3. We calculate the coordinates of points $\mathbf{A}_i$ according to equation (2). Then, considering equation (1), we immediately get a system of linear equations for w_1, w_2 and w_3,

$$\widehat{\mathbf{A}}\mathbf{W} = \mathbf{X} \tag{7}$$

where $\widehat{\mathbf{A}}$ is a 3×3 matrix composed of vectors $\mathbf{A}_i$, $i = 1, 2, 3$, and $\mathbf{W} = (w_1, w_2, w_3)^T$.

However, points $\mathbf{A}_i$ may be very close to one another, vectors $\mathbf{A}_i$ may thus be almost parallel, and the determinant of matrix $\widehat{\mathbf{A}}$ may be very small. This might cause numerical errors in solving equation (7). It is thus better to construct two perpendicular vectors

$$\mathbf{E}_1 = (\mathbf{A}_1 - \mathbf{A}_3)\,|\mathbf{A}_2 - \mathbf{A}_3| + (\mathbf{A}_2 - \mathbf{A}_3)\,|\mathbf{A}_1 - \mathbf{A}_3|$$

and

$$\mathbf{E}_2 = (\mathbf{A}_1 - \mathbf{A}_3)\,|\mathbf{A}_2 - \mathbf{A}_3| - (\mathbf{A}_2 - \mathbf{A}_3)\,|\mathbf{A}_1 - \mathbf{A}_3|$$

in the plane defined by points $\mathbf{A}_i$, then to construct projections

$$\begin{aligned}
\mathbf{U}_1 =& ((\mathbf{A}_1 - \mathbf{A}_3)^T\mathbf{E}_1, (\mathbf{A}_1 - \mathbf{A}_3)^T\mathbf{E}_2)^T \quad ,\\
\mathbf{U}_2 =& ((\mathbf{A}_2 - \mathbf{A}_3)^T\mathbf{E}_1, (\mathbf{A}_2 - \mathbf{A}_3)^T\mathbf{E}_2)^T \quad ,\\
\mathbf{V} =& ((\mathbf{X} - \mathbf{A}_3)^T\mathbf{E}_1, (\mathbf{X} - \mathbf{A}_3)^T\mathbf{E}_2)^T
\end{aligned}$$

of vectors $\mathbf{A}_1 - \mathbf{A}_3$, $\mathbf{A}_2 - \mathbf{A}_3$, $\mathbf{X} - \mathbf{A}_3$ onto the plane defined by points $\mathbf{A}_i$, solve equations

$$\begin{pmatrix} U_1^1 & U_2^1 \\ U_1^2 & U_2^2 \end{pmatrix} \begin{pmatrix} w_1 \\ w_2 \end{pmatrix} = \begin{pmatrix} V^1 \\ V^2 \end{pmatrix}, \tag{8}$$

and calculate $w_3 = 1 - w_1 - w_2$.

The use of equation (8) in computing local coordinates w_1, w_2, w_3 cannot prevent us from the inaccuracy caused by rounding errors. Let us consider two neighbouring ray cells and point $\mathbf{X}$ situated at the side which is common to both the ray cells. Note that the side is usually not planar, because it is defined by four points on rays. For such point $\mathbf{X}$ one of coordinates w_1, w_2, w_3 is roughly zero; let it be w_i in the first cell and w_j in the second one. Theoretically either both w_i and w_j are equal to zero, or w_i times w_j is less than zero. Due to the rounding errors this is not true for point $\mathbf{X}$ located very close to the side of the ray cell, because we have used coordinates of all the points forming the corresponding ray cell to compute local coordinates w_i and w_j, and not only the coordinates of the four points common to both the neighbouring cells.

Thus point $\mathbf{X}$ may randomly be considered to be located either in both the cells, in one of the cells, or in no cell. See figure 6. On a PC, the relative width of the volume along the side of the ray cell, where the rounding errors described cause inaccuracy, is about 10^{-6} in order of magnitude as compared to the coordinates of the vertices of the side.

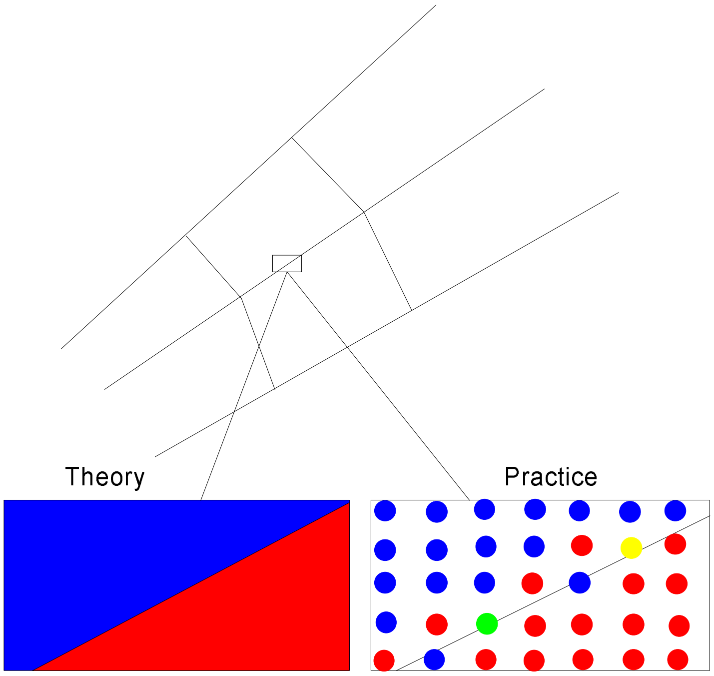

Figure 6: Theory and practice of determination, in which one of the two ray cells a given point is located. Theoretically, we should be able to decide correctly for any point, whether it is located inside the 'red' or 'blue' ray cells. In practice, the rounding errors cause that, for points located within a thin strip along the side separating the two ray cells, the decision based on equations (6) and (8) may be incorrect. An additional check of these errors is necessary.

Assume that point $\mathbf{X}$ is very close to side $\mathbf{B}_1, \mathbf{B}_2, \mathbf{C}_1, \mathbf{C}_2$. We can get rid of the described numerical problems by constructing virtual ray cell $\mathbf{B}_1, \mathbf{B}_2, \mathbf{B}'_3, \mathbf{C}_1, \mathbf{C}_2, \mathbf{C}'_3$. First we construct points $\mathbf{S} = (\mathbf{B_1} + \mathbf{B_2})/\mathbf{2}$ and $\mathbf{T} = (\mathbf{C_1}+\mathbf{C_2})/\mathbf{2}$. We then get the coordinates of points $\mathbf{B}'_3 = \mathbf{S} + (\mathbf{B_2} - \mathbf{B_1}) \times (\mathbf{T} - \mathbf{S})$ and $\mathbf{C}'_3 = \mathbf{T} + (\mathbf{C_2} - \mathbf{C_1}) \times (\mathbf{T} - \mathbf{S})$, where $\times$ stands for the vector product. We compute the local coordinates of point $\mathbf{X}$ in the virtual ray cell. According to their value we can unambiguously decide about the position of point $\mathbf{X}$ with respect to the side $\mathbf{B}_1, \mathbf{B}_2, \mathbf{C}_1, \mathbf{C}_2$. Because the values of the local coordinates in the virtual ray cell are influenced only by the coordinates of the four points located at the given side, the rounding errors should be exactly the same in both the neighbouring real ray cells. We must only order points $\mathbf{B}_1, \mathbf{B}_2, \mathbf{C}_1, \mathbf{C}_2$ equally for both the cells when constructing the virtual cells.

4 Interpolation within a ray cell

4.1 Bilinear interpolation

The simplest interpolation to fit the functional values at six points $\mathbf{B}_1$, $\mathbf{B}_2$, $\mathbf{B}_3$, $\mathbf{C}_1$, $\mathbf{C}_2$, $\mathbf{C}_3$ is the bilinear interpolation with respect to w_B and (w_1, w_2). Linear interpolation

$$\tau^{\mathbf{A}_k} = w_B \tau^{\mathbf{B}_k} + w_C \tau^{\mathbf{C}_k} \tag{9}$$

with respect to w_B is used to obtain the values $\tau^{\mathbf{A}_k}$ of an interpolated quantity τ at points $\mathbf{A}_k$ from known values $\tau^{\mathbf{B}_k}$ and $\tau^{\mathbf{C}_k}$ at the vertices of the ray cell. Linear interpolation

$$\tau^{\mathbf{X}} = w_1 \tau^{\mathbf{A}_1} + w_2 \tau^{\mathbf{A}_2} + w_3 \tau^{\mathbf{A}_3} \tag{10}$$

with respect to (w_1, w_2) is then used to obtain the values at point $\mathbf{X}$. This interpolation fits the functional values at the six vertices of the ray cell. Since the values at each cell face are dependent on the values at the vertices of the face only, the interpolated values are continuous across the cell faces. This interpolation seems convenient in interpolating all the quantities resulting from ray tracing, except for the travel time which should be interpolated more accurately, see figures 7a,b,c,d.

4.2 Bicubic interpolation

Assume that partial derivatives

$$p^i = \frac{\partial \tau}{\partial x^i} \tag{11}$$

are known in addition to the functional values of τ at six vertices $\mathbf{B}_1$, $\mathbf{B}_2$, $\mathbf{B}_3$, $\mathbf{C}_1$, $\mathbf{C}_2$, $\mathbf{C}_3$ of the ray cell. We may then wish to use the bicubic interpolation with respect to w_B and (w_1, w_2) to fit both the functional values and the first partial derivatives at the six vertices. To fit the functional values, we define cubic function

$$a(w) = w^2(3 - 2w) \tag{12}$$

with functional values $a(0) = 0$, $a(1) = 1$ and first derivatives $a'(0) = 0$, $a'(1) = 0$ at the endpoints of interval $< 0, 1 >$. To fit the derivatives, we define cubic functions

$$b(w_B) = (w_B)^2 p^i_{\mathbf{B}} [A^i(w_B) - B^i] \tag{13}$$

with functional values $b(0) = 0$, $b(1) = 0$ and derivatives $b'(0) = 0$, $b'(1) = p^i_{\mathbf{B}}[B^i - C^i]$ at the endpoints of interval $< 0, 1 >$. Here

$$\mathbf{A}(w_B) = w_B \mathbf{B} + (1 - w_B)\mathbf{C} \quad ,$$

see (3). Analogously for functions

$$b(w_C) = (w_C)^2 p^i_{\mathbf{C}} [A^i(w_C) - C^i] \quad . \tag{14}$$

Cubic interpolations

$$\boxed{\tau^{\mathbf{A}_k} = a(w_B)\tau^{\mathbf{B}_k} + a(w_C)\tau^{\mathbf{C}_k} + (w_B)^2 p^i_{\mathbf{B}_k} [\mathbf{A}^i_k - \mathbf{B}^i_k] + (w_C)^2 p^i_{\mathbf{C}_k} [\mathbf{A}^i_k - \mathbf{C}^i_k] \quad (15)}$$

and

$$\boxed{p^i_{\mathbf{A}_k} = a(w_B) p^i_{\mathbf{B}_k} + a(w_C) p^i_{\mathbf{C}_k} \quad (16)}$$

along the edges of the ray cell, parametrized by w_B, are used to obtain functional values $\tau^{\mathbf{A}_k}$ at points $\mathbf{A}_k$ together with their first derivatives $p^i_{\mathbf{A}_k}$. The derivatives will be used for the interpolation within triangle $\mathbf{A}_1\mathbf{A}_2\mathbf{A}_3$.

Cubic interpolation within triangle $\mathbf{A}_1\mathbf{A}_2\mathbf{A}_3$

$$\boxed{\tau^{\mathbf{X}} = \sum_{k=1}^{3} a(w_k)\tau^{\mathbf{A}_k} + \sum_{k=1}^{3}(w_k)^2 p^i_{\mathbf{A}_k}[X^i - \mathbf{A}^i_k] + w_1 w_2 w_3 \xi} \tag{17}$$

with respect to (w_1, w_2) is then used to obtain the values at point $\mathbf{X}$. The three functional values and six first partial derivatives at the vertices of planar triangle $\mathbf{A}_1\mathbf{A}_2\mathbf{A}_3$ are fitted by the cubic function of w_1 and w_2. Since a cubic function of two variables has 10 coefficients, there is a free coefficient ξ in the last equation. Coefficient ξ corresponds to the basis function $w_1w_2w_3$ which has both the functional values and the first derivatives at the vertices $\mathbf{A}_k$ equal to zero.

Let us now check the interpolation of simple linear, quadratic and cubic functions. First, assume a simple linear function

$$\tau(w_1, w_2) = w_3 \quad , \tag{18}$$

where $w_3 = 1 - w_1 - w_2$.
Then, as derived in the Appendix,

$$\tau^{\mathbf{X}} = w_3 + (\xi - 2)w_1w_2w_3 \quad . \tag{19}$$

Analogously for two other independent linear functions

$$\tau(w_1, w_2) = w_1 \quad \text{and} \quad \tau(w_1, w_2) = w_2 \quad . \tag{20}$$

Now assume a simple quadratic function

$$\tau(w_1, w_2) = w_3(1 - w_3) \quad . \tag{21}$$

The interpolation then yields

$$\tau^{\mathbf{X}} = w_3(1 - w_3) + (\xi - 2)w_1w_2w_3 \quad , \tag{22}$$

see the Appendix.
Analogously for both other independent quadratic functions

$$\tau(w_1, w_2) = w_1(1 - w_1) \quad \text{and} \quad \tau(w_1, w_2) = w_2(1 - w_2) \quad . \tag{23}$$

Coefficient ξ must depend linearly on the given interpolated quantities. The linear combination must be symmetric with respect to the permutation of points $\mathbf{A}_1, \mathbf{A}_2, \mathbf{A}_3$. There are only two linearly independent symmetric linear combinations of the given quantities,

$$\sum_{k=1}^{3} \tau^{\mathbf{A}_k} \quad \text{and} \quad \sum_{k=1}^{3}\sum_{l=1}^{3} p^i_{\mathbf{A}_k}[\mathbf{A}^i_l - \mathbf{A}^i_k] \quad . \tag{24}$$

There is no other linear combination of the given interpolated quantities, symmetric with respect to points $\mathbf{A}_1$, $\mathbf{A}_2$ and $\mathbf{A}_3$.

To interpolate linear and quadratic functions exactly, we thus have to take

$$\boxed{\xi = 2\sum_{k=1}^{3} \tau^{\mathbf{A}_k} + \frac{1}{2}\sum_{k=1}^{3}\sum_{l=1}^{3} p^i_{\mathbf{A}_k}[\mathbf{A}^i_l - \mathbf{A}^i_k] \quad .} \tag{25}$$

Figure 7a: Travel times in a homogeneous model interpolated bilinearly within wide ray tubes using equation (10). One colour circle corresponds to 0.2 s.

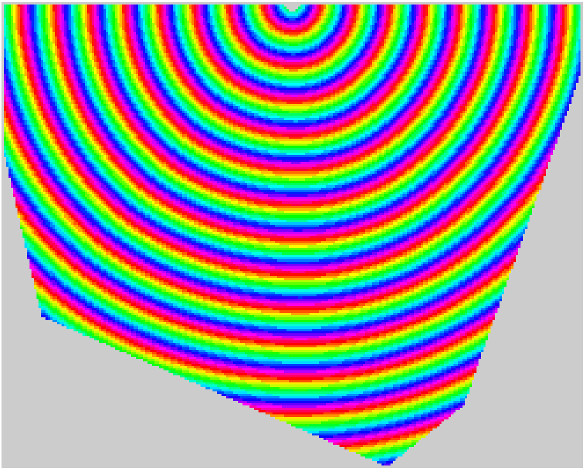

Figure 7b: Travel times in a homogeneous model interpolated within the same ray tubes using bicubic interpolation (17). The maximum travel time is about 2.7 s.

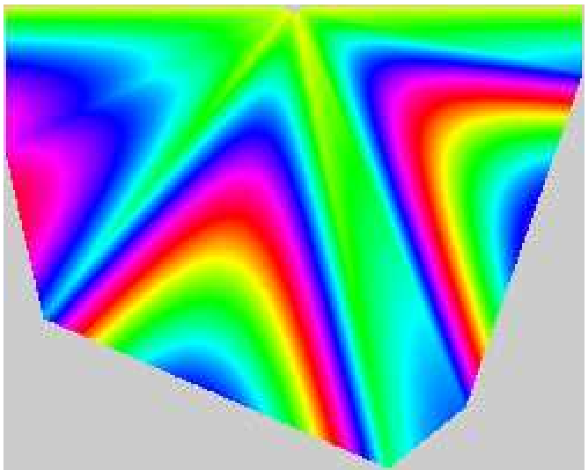

Figure 7c: Differences of the bilinearly interpolated travel times from the exact solution. The maximum difference is about 0.308 s.

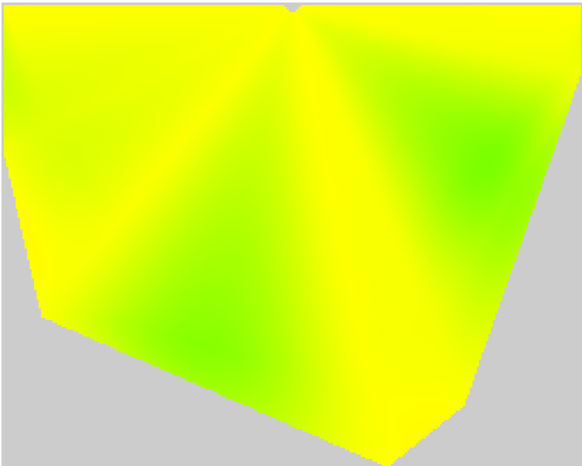

Figure 7d: Differences of the bicubically interpolated travel times from the exact solution. The maximum difference is about 0.017 s.

Here the first term is necessary to interpolate linear functions exactly, and the second term is added to interpolate also quadratic functions exactly.

Now assume a simple cubic function

$$\tau(w_1, w_2) = (w_2 - w_1)[(w_2 - w_1)^2 - 1] \quad . \tag{26}$$

The interpolation then yields

$$\tau^{\mathbf{X}} = (w_2 - w_1)[(w_2 - w_1)^2 - 1] \quad , \tag{27}$$

see the Appendix, and analogously for the two other independent cubic functions

$$\tau(w_1, w_2) = (w_3 - w_2)[(w_3 - w_2)^2 - 1] \text{ and } \tau(w_1, w_2) = (w_1 - w_3)[(w_1 - w_3)^2 - 1] \ . \tag{28}$$

The last independent cubic function

$$\tau(w_1, w_2) = w_1 w_2 w_3 \tag{29}$$

has zero functional values and first derivatives at all three vertices and thus yields

$$\tau^{\mathbf{X}} = 0 \quad . \tag{30}$$

There is no way of removing this inaccuracy in the interpolation of cubic functions without additional information.

The interpolation fits the values and the first spatial partial derivatives at the six vertices of the ray cell. The functional values at each cell face depend on the functional values and derivatives at the vertices of the face only. The interpolated values are thus continuous across the cell faces. The gradient of the interpolated function is continuous at the triangular cell faces and along the cell edges, but may be discontinuous at the tetragonal cell faces separating ray tubes. The interpolation fits linear and quadratic functions exactly.

5 Numerical example

A computation in model 98 is presented here as an example of interpolation of ray-theory travel times on a dense receiver grid. The sample synthetic model named "98" was created during the technical session of the consortium "Seismic Waves in Complex 3-D Structures" meeting in June 1998. The aim of the work done was not to create a realistic model, but to demonstrate the capabilities of program packages `MODEL` and `CRT`. The data and history files for the computation may be found in subdirectory `crt/98/` of the SW3D-CD-2 disk.

Model 98 is composed of three layers separated by two interfaces. The upper interface is curved, the lower one is formed by a dipping plane. The inclusion in the middle layer is approximately spherical. The first layer represents water with constant P-wave velocity and zero S-wave velocity. In other model blocks, the P-wave velocity has a vertical gradient and smaller lateral gradients, the S-wave velocity is a linear function of the P-wave velocity. See figures 8a,b. For more detail refer to Bulant & Klimeš (1998).

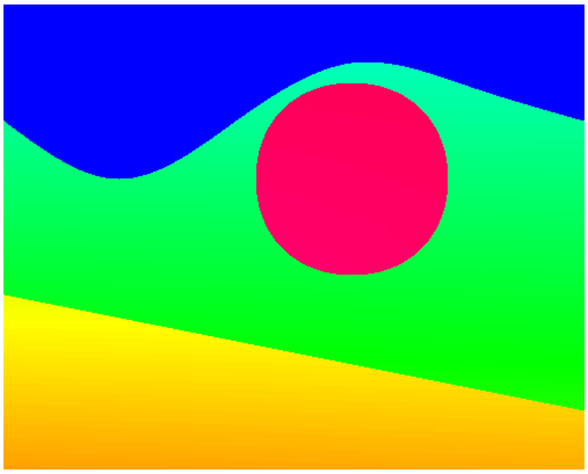

Figure 8a: P-wave velocity cross-section through the middle of model 98. The P-wave velocity is lowest at the top water layer and highest within the spherical body.

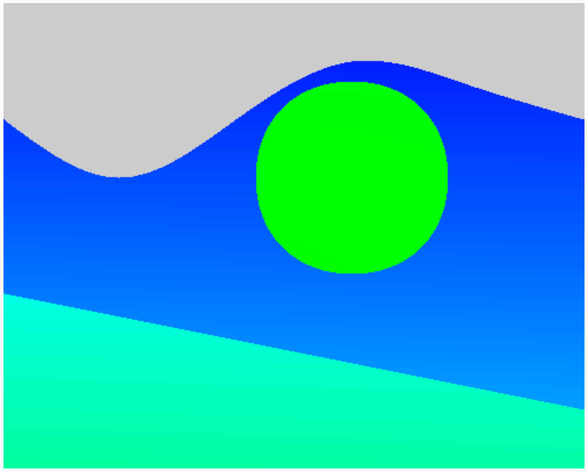

Figure 8b: S-wave velocity cross-section through the middle of model 98. The S-wave velocity equals zero in the first layer and it is a linear function of the P-wave velocity in the other layers.

The ray-theory travel times are computed for a vertical 2-D grid of 251*201 points embedded in the 3-D model. The point source is located at the same plane as the receivers at a depth of 0.001 km. See figure 9. The elementary wave refracted at each interface is considered. Only structural interfaces affect ray histories.

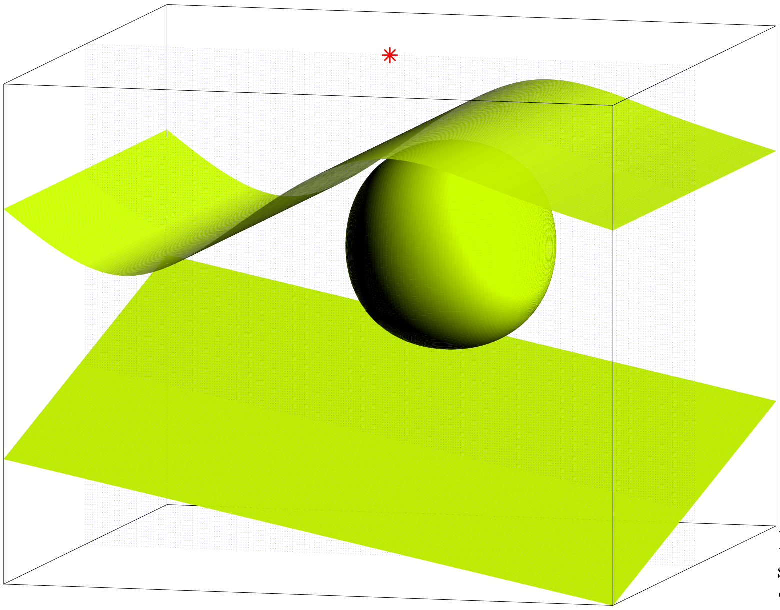

Figure 9: Structural interfaces, point source and 2D receiver grid in model 98.

The computed multi-valued travel-time maps may be sorted according to the arrival time (i.e. first arrival at the gridpoints, second arrival at the gridpoints, ...), see figure 10.

First arrival. Second arrival. Third arrival. Fourth arrival.

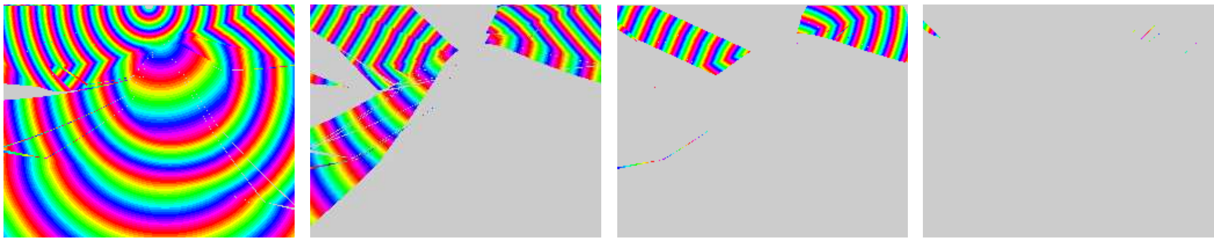

Figure 10: The single-valued travel-time maps sorted according to the arrival time.

Another possibility is to convert the computed multi-valued travel-time maps into the single-valued maps sorted according to the ray histories. See figure 11.

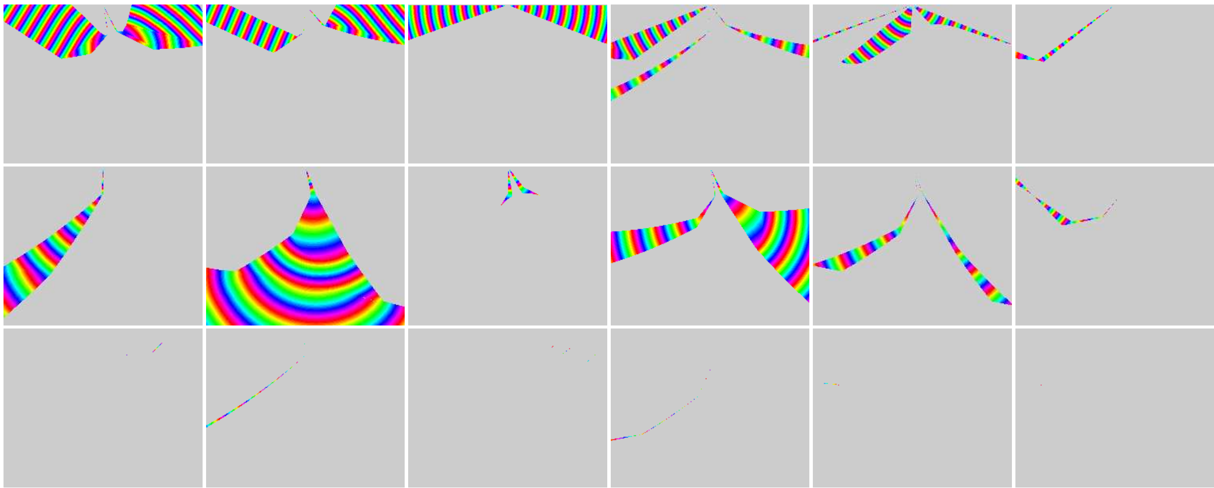

Figure 11: The single-valued travel-time maps sorted according to the ray histories.

The single-valued travel-time maps sorted according to the ray histories may be merged together manually. One result, which may be obtained, is shown in figure 12. Note that some travel times were omitted during the merging process, and so the resulting pictures are only three.

Travel times of rays refracted through the spherical body. | Travel times of rays not penetrating into the spherical body. | Travel times of rays weakly refracted through the middle layer back to the first layer.

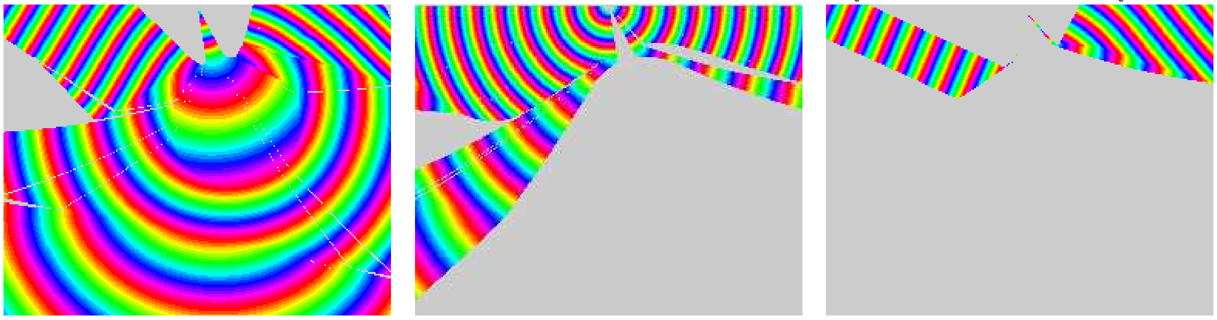

Figure 12: The single-valued travel-time maps sorted according to the ray histories from figure 11, merged into the three sections. Some computed travel times were omitted.

6 Conclusions

We have proposed a method for interpolation of ray-theory travel times and other quantities to the gridpoints of dense receiver grids. Ray tubes are decomposed into individual ray cells in the way, which assures that the sides of ray cells in the neighbouring ray tubes coincide, and no receiver points are omitted along these sides. The decomposition also prevents interpolation across structural interfaces.

Using the scheme, all the quantities computed along rays may be interpolated. The scheme requires the values of the interpolated quantities only at the vertices of the corresponding ray cell. It incorporates both the decision, whether a receiver lies in the ray cell and the interpolation to the receiver. The scheme is applicable to all ray cells formed by six, five or four points. Further investigation should concentrate on numerical tests of the scheme and on comparisons with other schemes.

The numerical example presented shows the applicability of the method to a 3-D model with interfaces.

Appendix: Derivation of equations (19), (22) and (27)

For a given function $\tau(w_1, w_2)$, we introduce complementary parametrizations

$$\begin{aligned}\tau_{12}(w_1,w_2) =& \tau(w_1,w_2)\\ \tau_{13}(w_1,w_3) =& \tau(w_1,1-w_1-w_3)\\ \tau_{23}(w_2,w_3) =& \tau(1-w_2-w_3,w_2) \quad .\end{aligned} \tag{A1}$$

Then

$$\begin{aligned}p^i_{\mathbf{A}_1}[X^i - A^i_1] =& p^i_{\mathbf{A}_1}\sum_j w_j[A^i_j - A^i_1]\\ =& p^i_{\mathbf{A}_1} w_2[A^i_2 - A^i_1] + p^i_{\mathbf{A}_1} w_3[A^i_3 - A^i_1]\\ =& w_2\frac{\partial\tau_{23}}{\partial w_2}(1,0) + w_3\frac{\partial\tau_{23}}{\partial w_3}(0,1)\end{aligned} \tag{A2}$$

and vice versa.

Equation (19)

Linear function

$$\tau(w_1, w_2) = w_3 \tag{18}$$

then has

$$\begin{aligned}&\tau^{\mathbf{A}_1} = 0,\ \tau^{\mathbf{A}_2} = 0,\ \tau^{\mathbf{A}_3} = 1,\\ &p^i_{\mathbf{A}_1}[\mathbf{X}^i - \mathbf{A}^i_1] = w_3,\ p^i_{\mathbf{A}_2}[\mathbf{X}^i - \mathbf{A}^i_2] = w_3,\ p^i_{\mathbf{A}_3}[\mathbf{X}^i - \mathbf{A}^i_3] = w_3 - 1\end{aligned} \tag{A3}$$

and interpolation (17) yields

$$\begin{aligned}\tau^{\mathbf{X}} =&(w_3)^2(3-2w_3) + (w_1)^2 w_3 + (w_2)^2 w_3 + (w_3)^2(w_3 - 1) + w_1 w_2 w_3 \xi\\ =&w_3[w_3(2-w_3) + (w_1)^2 + (w_2)^2] + w_1 w_2 w_3 \xi\\ =&w_3[(1-w_1-w_2)(1+w_1+w_2) + (w_1)^2 + (w_2)^2] + w_1 w_2 w_3 \xi\\ =&w_3 + (\xi - 2) w_1 w_2 w_3 \quad .\end{aligned} \tag{A4}$$

Equation (22)

Quadratic function

$$\tau(w_1, w_2) = w_3(1-w_3) \tag{21}$$

then has

$$\begin{aligned}&\tau^{\mathbf{A}_1} = 0,\ \tau^{\mathbf{A}_2} = 0,\ \tau^{\mathbf{A}_3} = 0,\\ &p^i_{\mathbf{A}_1}[\mathbf{X}^i - \mathbf{A}^i_1] = w_3,\ p^i_{\mathbf{A}_2}[\mathbf{X}^i - \mathbf{A}^i_2] = w_3,\ p^i_{\mathbf{A}_3}[\mathbf{X}^i - \mathbf{A}^i_3] = 1 - w_3\end{aligned} \tag{A5}$$

and interpolation (17) yields

$$\begin{aligned}\tau^{\mathbf{X}} =&(w_1)^2 w_3 + (w_2)^2 w_3 + (w_3)^2(1 - w_3) + w_1 w_2 w_3 \xi\\ =&w_3[w_3(1-w_3) + (w_1)^2 + (w_2)^2] + w_1 w_2 w_3 \xi\\ =&w_3[(1-w_1-w_2)(w_1+w_2) + (w_1)^2 + (w_2)^2] + w_1 w_2 w_3 \xi\\ =&w_3(1-w_3) + (\xi - 2) w_1 w_2 w_3 \quad .\end{aligned} \tag{A6}$$

Equation (27)

Cubic function

$$\tau(w_1, w_2) = (w_2 - w_1)[(w_2 - w_1)^2 - 1] \tag{26}$$

then has

$$\begin{aligned}&\tau^{\mathbf{A}_1} = 0,\ \tau^{\mathbf{A}_2} = 0,\ \tau^{\mathbf{A}_3} = 0,\ p^i_{\mathbf{A}_1}[\mathbf{X}^i - \mathbf{A}^i_1] = 2(w_2 - w_1 + 1),\\ &p^i_{\mathbf{A}_2}[\mathbf{X}^i - \mathbf{A}^i_2] = 2(w_2 - w_1 - 1),\ p^i_{\mathbf{A}_3}[\mathbf{X}^i - \mathbf{A}^i_3] = -(w_2 - w_1),\ \xi = 0\end{aligned} \tag{A7}$$

and interpolation (17) yields

$$\begin{aligned}\tau^{\mathbf{X}} =&2(w_1)^2(w_2 - w_1 + 1) + 2(w_2)^2(w_2 - w_1 - 1) - (w_3)^2(w_2 - w_1)\\ =&(w_2 - w_1)[2(w_1)^2 + 2(w_2)^2 - 2w_1 - 2w_2 - (1 - w_1 - w_2)^2]\\ =&(w_2 - w_1)[(w_1)^2 + (w_2)^2 - 2w_1 w_2 - 1]\\ =&(w_2 - w_1)[(w_2 - w_1)^2 - 1] \quad .\end{aligned} \tag{A8}$$

References

Bulant, P. (1996): Two–point ray tracing in 3-D. Pure and Applied Geophysics, **148**, 421–446.

Bulant, P. (1997a): Controlled initial–value ray tracing and two–point ray tracing in 3D. Extended Abstracts, P025, EAGE 59th Conference and Technical Exhibition, Geneva.

Bulant, P. (1997b): Two–point ray tracing and controlled initial–value ray tracing in 3-D heterogeneous block structures. Expanded Abstracts, pp. 1727–1730, SEG 67th International Exposition and Annual Meeting, Dallas.

Bulant, P. & Klimeš, L. (1998): Computations in the model composed during the 1998 consortium meeting. In: Seismic Waves in Complex 3-D Structures, Report 7, pp. 33–56, Department of Geophysics, Charles University, Prague.

Moser, T.J. and Pajchel, J. (1997): Recursive seismic ray modelling: applications in inversion and VSP. Geophysical Prospecting, **45**, 885–908.

Vinje, V., Iversen, E., Gjøystdal, H. & Åstebøl, K. (1996a): Estimation of multivalued arrivals in 3D models using wavefront construction – Part I. Geophysical Prospecting, **44**, 819–842.

Vinje, V., Iversen, E., Gjøystdal, H. & Åstebøl, K. (1996b): Part II: Tracing and interpolation. Geophysical Prospecting, **44**, 843–858.

Numerical Recipes Software (1996): Numerical Recipes in Fortran. The Art of Scientific Computing. Second Edition, pp. 178–180 (Chapter 5.6), http://www.nr.com/.