This is an excerpt from the Ph.D thesis of F. Haslinger (1998). The full thesis is available either as PDF or PostScript file from the author.

4.4 Forward Solution and Model Parametrization

The importance of the forward solution calculation for seismic tomography has already been emphasized at the beginning of this chapter (p. 70ff). Unfortunately it is up to now not possible to use full 3D wave-theory in seismic tomography. Computation of theoretical travel times is therefore restricted to ray theory, the high-frequency approximation of the elastodynamic equation of motion. One should be aware that in inhomogeneous media the high-frequency approximation is only valid if the velocity gradient is small comapred to the frequency of the seismic wave. Also, the use of infinitesimally thin rays too precisely constrains the location in 3D space from where the information (the ΔT) comes, whereas in reality (wave theory) the information is always influenced by a finite volume, the Fresnel volume of the ray.

The amplitude of velocity heterogeneities and their spatial extent which can be resolved with 3D tomography therefore depends on the accuracy of the ray theoretical forward solution (ray tracing) in estimating the correct travel time and path in a given velocity model. The absolute accuracy of a ray tracing scheme, however, is very difficult to assess. One way to determine the influence of ray tracing on tomographic images is to use different forward solving schemes and to compare the results.

The combination of approximate raytracing and pseudo-bending in the current version of the SIMULPS-code (further referred to as ART_PB ray tracing) is proven to work well in most LET applications, but is suspected to yield inaccurate results for ray paths exceeding ~40 km length (e.g. Eberhart-Phillips, 1986). Also the ART_PB is still an approximate raytracer, which only allows limited deformation of the initially circular ray paths. The dimension of the Ionia95 network is about 150 km $\times$ 150 km, and based on the results of Section 4.2, velocity variations up to $\pm$15% are to be expected. To assess the effects of possible inaccuracy of the ART_PB ray tracing for such a problem, a highly accurate 3D shooting ray tracer, based on Virieux (1991), is implemented in the SIMULPS code.

Implementation of an Accurate 3D Raytracer in SIMULPS

A Short Review of Ray Tracing

As stated above, all ray tracing schemes employ the high-frequency approach to solve the elastodynamic wave equation. An exhaustive discussion can for example be found in Aki & Richards (1980), here only some basic equations shall be given.

The scalar equation of wave propagation in an isotropic heterogeneous medium can be written as

$$\nabla^2\Phi = \frac{1}{v^2}\frac{\partial^2\Phi}{\partial t^2} , \qquad (4\text{-}1)$$

where $\Phi(\boldsymbol{x},t)$ is a scalar wave field and $v(\boldsymbol{x})$ the three-dimensional velocity field. For high frequencies a harmonic solution of (4-1) can be given as

$$\Phi(\boldsymbol{x}, t) = A(\boldsymbol{x})\exp[i\omega(t-T(\boldsymbol{x}))] , \qquad (4\text{-}2)$$

where $A(\boldsymbol{x})$ is the amplitude and $T(\boldsymbol{x})$ the travel time at a point $\boldsymbol{x}$. This is a high frequency solution because only then the amplitude and travel time function will be frequency independent. Introducing (4-2) in (4-1) the travel time is described by the Eikonal equation

$$\nabla^2 T(\boldsymbol{x}) = u^2 = \frac{1}{v^2} \qquad (4\text{-}3)$$

where u is the slowness (reciprocal velocity). $T(\boldsymbol{x}) = const$ then describes the wave fronts, and the normals to the wave front, which define the direction of wave propagation, constitute the seismic rays. By introducing a parametrization $\boldsymbol{x} = \boldsymbol{x}(s)$, where s is the arc length along the ray, one can derive the ray equation

$$\frac{d}{ds}\left(\frac{1}{v}\frac{d\boldsymbol{x}}{ds}\right) = \nabla\left(\frac{1}{v}\right) . \qquad (4\text{-}4)$$

The ray equation can numerically either be solved as initial value problem, where the ray direction at a starting point is given (shooting), or as boundary value, where the ray starting point and end point are given (bending). In both cases the solution of (4-4) will yield a ray path, and integrating over this path will give the travel time for this ray.

Description of Ray Tracing by Hamiltonian Perturbation

The raytracing scheme implemented here is a shooting method, where the ray which connects station and receiver in the given velocity distribution is found by varying the initial azimuth φ and take-off angle θ at the source. For the variation of the initial angles first order perturbation therory is used. The general theory is described in Virieux et al. (1988), Virieux (1991) and Virieux & Farra (1991). In the following, a brief summary of the concept and the fundamental equations is given.

Introducing the slowness vector $\boldsymbol{p} = \nabla T$ the eikonal equation (4-3) can be cast into a Hamiltonian formalism as proposed by Burridge (1976):

$$H(\boldsymbol{x}, \boldsymbol{p}, \tau) = \frac{1}{2}[\boldsymbol{p}^2 - u^2(\boldsymbol{x})] \qquad (4\text{-}5)$$

$\boldsymbol{x}(\tau)$ is the position along the ray and τ is a sampling parameter along the ray, defined by $dT = \boldsymbol{p}d\boldsymbol{x} = u^2 d\tau$. The eikonal equation implies that $H=0$ along a ray, and the ray tracing equations are then given by Hamiltons canonical equations

$$\dot{\boldsymbol{x}} = \nabla_{\boldsymbol{p}} H = \boldsymbol{p}$$
$$\dot{\boldsymbol{p}} = -\nabla_{\boldsymbol{x}} H = \frac{1}{2}\nabla_{\boldsymbol{x}} u^2 \qquad (4\text{-}6)$$

where $\nabla_{\boldsymbol{x}}$ and $\nabla_{\boldsymbol{p}}$ denote the gradients with respect to vectors $\boldsymbol{x}$ and $\boldsymbol{p}$, respectively. System (4-6) has then to be solved for $\boldsymbol{x}(\tau)$, $\boldsymbol{p}(\tau)$ with the given initial values (shooting angles) to find the raypath and, by integrating over $\boldsymbol{p}(\tau)$, the traveltime. Shooting normally implies that the initial values (first guess) have to be adjusted so that the ray surfacing point reaches the station with a required accuracy. For these adjustments the concept of paraxial rays proves to be very useful. An already traced ray, described by $\boldsymbol{x}_c(\tau)$ and $\boldsymbol{p}_c(\tau)$, will be called the central ray. Position and slowness of the paraxial ray are then given by

$$\boldsymbol{x}(\tau) = \boldsymbol{x}_c(\tau) + \delta\boldsymbol{x}(\tau) \qquad \boldsymbol{p}(\tau) = \boldsymbol{p}_c(\tau) + \delta\boldsymbol{p}(\tau) \qquad (4\text{-}7)$$

where $\delta\boldsymbol{x}$ and $\delta\boldsymbol{p}$ are the perturbations of position and slowness of the central ray. These perturbations have to satisfy the paraxial ray tracing equations, deduced by first order linear perturbation of (4-6):

$$\begin{bmatrix} \delta\dot{\boldsymbol{x}} \\ \delta\dot{\boldsymbol{p}} \end{bmatrix} = \begin{bmatrix} \nabla_{\boldsymbol{p}}\nabla_{\boldsymbol{x}} H & \nabla_{\boldsymbol{p}}^2 H \\ -\nabla_{\boldsymbol{x}}^2 H & \nabla_{\boldsymbol{x}}\nabla_{\boldsymbol{p}} H \end{bmatrix} \begin{bmatrix} \delta\boldsymbol{x} \\ \delta\boldsymbol{p} \end{bmatrix} \qquad (4\text{-}8)$$

where H and its derivatives are computed on the central ray. In order that a solution of (4-8) represents a paraxial ray, the additional constraint of $\delta H = 0$ must be fulfilled. Two paraxial rays, both with $\delta\boldsymbol{x}(0) = 0$, one with $\delta\boldsymbol{p}(0)$ associated with a change in φ and one with $\delta\boldsymbol{p}(0)$ associated with a change in θ, are necessary to update the shooting angles. From the conditions of these paraxials at the surface and the distance of the surfacing point of the central ray to the station, variations in θ and φ can be estimated; with a few iterations of this process convergence on the station is normally reached.

The solution of the ray tracing equations (4-6) and (4-8) require the integration of a set of differential equations. A fourth order Runge-Kutta solver is used in the numerical integration. This requires the first order derivatives of the sqared slowness distribution to be continuous and a certain smoothness of the second order derivatives to ensure numerical stability, which is obtained by representing the squared slowness distribution with cardinal B-splines of order four. The B-spline representation also allows very efficient programming of the calculation of the derivatives, which drastically speeds up computation time. In the following this ray tracing scheme will be abbreviated RKP-ray tracing (**R**unge-**K**utta + **P**erturbation).

Considerations on Model Parametrization

An important prerequisite to allow the comparison of different ray tracers is to ensure that the physical model (the given 3D velocity structure) is similar within the significance level for each

of the ray tracers. This may sound rather trivial, but if one realizes that more or less every ray tracer requires a different way of parametrization of the 3D velocity field it becomes evident that this matter has to be treated carefully.

ART_PB ray tracing uses a model where velocities are defined on nodes (intersections of grid lines) on a grid which may be irregularly spaced in each direction. For any arbitrary point $\boldsymbol{x}$ the velocity $v(\boldsymbol{x})$ is then obtained by 3D linear interpolation on the eight nearest neighbors (the cube of grid-nodes enclosing $\boldsymbol{x}$). For the RKP-raytracing the model has to be parametrized as squared slowness u^2, and cubic B-spline interpolation (see Box) is used to obtain $u^2(\boldsymbol{x})$ for any point $\boldsymbol{x}$. For computational efficiency, the u^2 -grid has to be equidistantly spaced in each direction. Also, cubic B-spline interpolation uses the four next neighbors in each direction for the computation of an interpolated value. As can be seen from Figure 4.1b, the B-spline interpolated curve is much smoother than the lineraly interpolated curve. If the two curves are taken to represent velocity-depth functions it is clear that one cannot expect that rays with the same start- and endpoint would have similar paths and traveltimes in the different velocity representations.

In order to use the RKP ray tracing with the same velocity model as the ART_PB ray tracing two problems have to be solved: 1) Values for squared slowness on an evenly spaced grid must be obtained from an unevenly spaced velocity grid. 2) B-spline interpolation must be modified in a way that remaining differences between linearly interpolated velocities and B-spline interpolated squared slownesses do not significantly affect the resulting rays. Note that this need only arises because the main aim of this study is a comparison of the effect of the two ray tracers. In principle there is no geophysical reason to prefer a linearly interpolated model to a B-spline interpolated one.

Box: *Cubic B-spline Interpolation*

The principles of cubic B-spline interpolation can be explained on the 1D case, the extensions to 3D is then straightforward. A thorough discussion can be found in de Boor (1987).

Given a discrete set of points (x_i, $v(x_i)$), with the coordinate x and the value $v(x)$, sorted in ascending order, the *linear* interpolation to obtain $v(x_k)$ at any point x_k with $x_{i-1} < x_k \leq x_i$ can be written as

$$v(x_k) = \sum_{i-1,\,i} l_j(x_k) \cdot v(x_j) \qquad l_j(x_k) = 1 - \frac{|x_k - x_j|}{x_{i-1} - x_i} \; .$$

For every point i the l_i then is a triangular function which is 1 at x_i and 0 at x_{i-1}, x_{i+1}. The l_i are called basis functions and as the interpolated value $v(x_k)$ depends only on the two next neighbors, this interpolation is also called linear B-spline of order 2. A linearly interpolated curve normally has discontinuity in its first derivatives at each point x_i. To obtain interpolated curves which are continuous to the second derivative, cubic B-spline interpolation of order 4 is used. The interpolated value is now controlled by the four next neighbors x_{i-2}, x_{i-1}, x_i, x_{i+1}, and the basis functions are cubic polinomials. If the four control values are numbered from -2 to 1 (see Fig. 4.1) the interpolation function then is

$$v(x_k) = \sum_{j=-2}^{1} l_j(x_k) \cdot v(x_j) \; .$$

For equidistant spacing dx of the x_i the basis functions take the form

$$l_{-2} = \frac{1}{6}(1-u)^3$$

$$l_{-1} = \frac{1}{6}[4 + 3u^2(-2+u)]$$

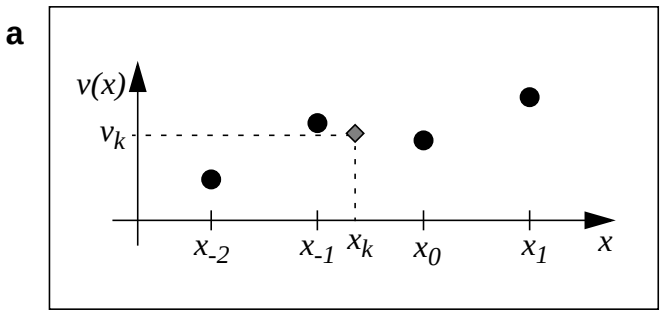
$$l_0 = \frac{1}{6}[1 + 3u(1+u-u^2)]$$

$$l_1 = \frac{1}{6}u^3$$

$$u = \frac{|x_k - x_{-1}|}{dx} \; .$$

Figure 4.1 shows the principle of cubic B-spline interpolation and a set of points with the interpolated curve.

The extension to three dimensions is sketched in Figure 4.2.



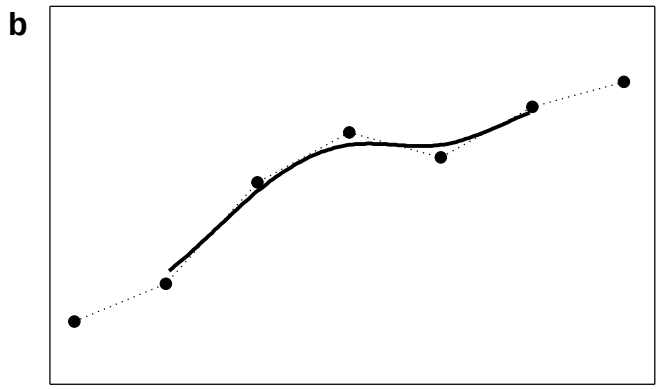


Figure 4.1 Principle of cubic B-spline interpolation. **a)** From four points $(x_{[-2,1]}, v(x_{[-2,1]}))$ the value $v(x_k)$ at x_k is interpolated. **b)** A B-spline curve (solid black line) inter-polated from a set of points (solid circles). The dotted line depicts the linear interpolation.

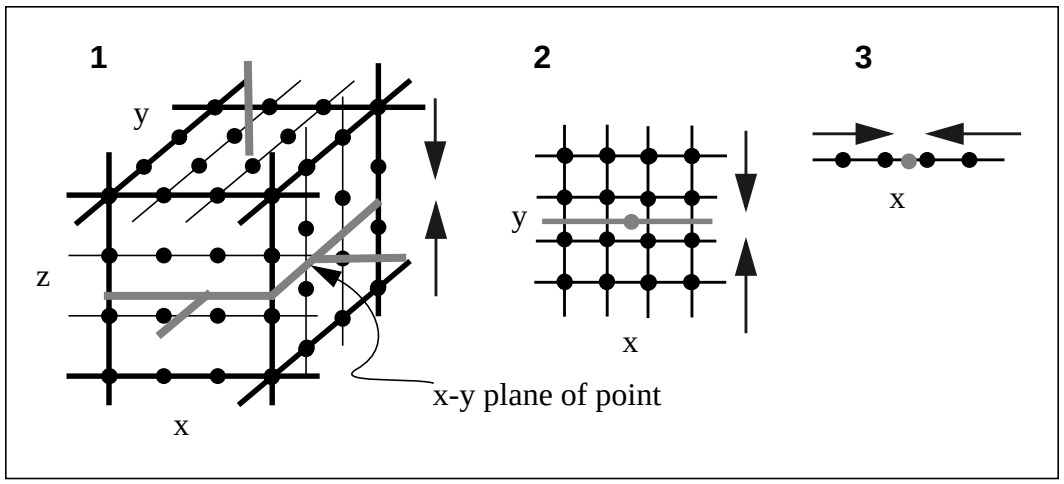


Figure 4.2 Scheme for 3D cubic B-spline interpolation. The solid black circles are the control values (grid nodes) and the grey circle is the target point. **1)** Interpolation along z onto x-y plane of point. **2)** interpolation along y onto x-coordinate of point. **3)** Interpolation along x onto point.

Implementation

Basis of the implementation of the RKP-ray tracer in SIMULPS is a code by J. Virieux (pers. comm., 1998), which has been successfully used in two LET studies, in the Gulf of Corinth, Greece (Le Meur et al., 1997) and in the northern Tien Shan, central Asia (Ghose et al., 1998). In this code the initial velocity model is transformed into squared slowness u^2, which is also the model parameter used for the inversion. Some additional preparatory work and a couple of changes to the RKP-code are necessary to use the RKP ray tracing in SIMULPS:

- The possibility of an unevenly spaced inversion grid is retained. Therefore the velocity field for ray tracing has to be resampled on an even grid which will be called ray tracing grid (RT-grid).
- The RT-grid will be made up of velocities. For every point along a ray the velocity $v(x,y,z)$ will be obtained by cubic B-spline interpolation. From this v the squared slowness values u^2 and their first and second order derivatives are then given by

$$u^2 = \frac{1}{v^2} \qquad \frac{\partial}{\partial x_i}u^2 = -\frac{2}{v^3}\cdot\frac{\partial v}{\partial x_i}$$
$$\frac{\partial^2}{\partial x_i \partial x_j}u^2 = \left(\frac{6}{v^4}\cdot\frac{\partial v}{\partial x_i}\cdot\frac{\partial v}{\partial x_j}\right) - \left(\frac{2}{v^3}\cdot\frac{\partial^2 v}{\partial x_i \partial x_j}\right) \qquad (4\text{-}9)$$

  This requires additional 24 multiplications, 3 divisons and 6 subtractions to compute the full system of u^2 and its first and second order derivatives for each raypoint, which is about 1/10 of the number of operations needed for the B-spline interpolation.
- For the evenly spaced RT-grid a much smaller grid-spacing is used than in the original velocity grid and linear 3D interpolation is used to derive the RT-grid from the inversion grid. This will provide nearly identical velocity fields for the two ray tracing approaches (Fig. 4.3). At the same time the velocity field for the RKP ray tracer can be rougher than in the original implementation.
- The initial shooting angles will be computed with the ART_PB raytracing. Initial values close to the 'true' values are crucial for the success of the Hamiltonian perturbation. The regional search algorithm of the ART and the refinement of the pseudo-bending have shown to yield much better results than the previously used simple 1D-shooting. Initial angles have to be computed on the first iteration and every time when (due to the change in velocities or hypocentral coordinates) the Hamiltonian perturbation starting from previously stored angles fails to reach the station.

Testing on Synthetic Models

RKP ray tracing has already been thoroughly tested by Virieux and co-workers (Virieux, 1991, Le Meur et al., 1997, Ghose et al., 1998), but some things have substantially changed in this implementation, e.g. the representation of the velocity field and the initial search for take off angles. Some tests are therefore conducted to assess the performance of the ray tracer in the new

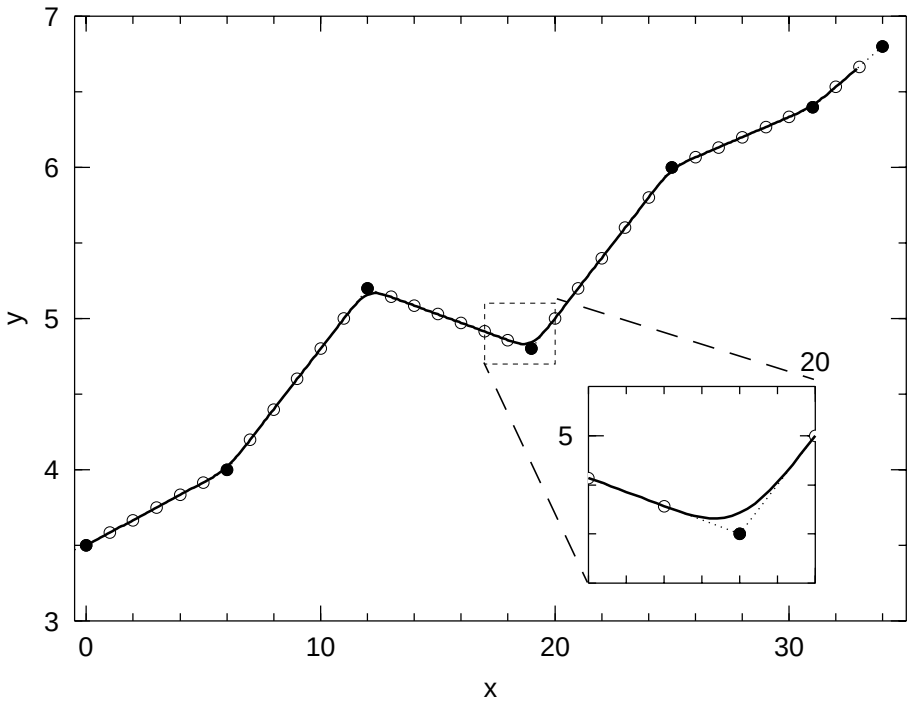


Figure 4.3 Refined B-spline interpolation. Solid black circles are the original unevenly spaced input points, empty circles (and the solid circles) are the evenly distributed points resulting from linear interpolation. The solid black line is the B-spline interpolated curve from the empty circles, the dotted line is the linearly interpolated curve from the solid circles. The zoom-in shows the slight difference at a location where the gradient is changing. In terms of model grids for tomography x may correspond to depth [km] and y to velocity [km/s]. The dotted line then resembles the velocities used in ART_PB ray tracing and the solid line those used in RKP ray tracing.

implementation.

The numerical stability of the computation can be tested by comparison of forward and reverse ray tracing, where source and receiver positions are exchanged. In Appendix C.1 these tests are described in detail. It is shown that the numerical accuracy of the ray tracing in this implementation is around 10 ms. This is of the same order of magnitude or better than the assumed onset time accuracy for high quality local earthquake data (see Chapter 2).

In a second test, ART_PB ray tracing and RKP ray tracing are compared. For a homogeneous half-space no differences in travel times or ray path geometries are visible. For a velocity gradient model with low- and high-velocity anomalies, significant differences in ray path and travel time occur for rays exceeding 60 km length (Appendix C.2). From comparing forward and reverse ray tracing it can be inferred that the difference is caused mainly by inaccuracies of the ART_PB ray tracing. This confirms previous findings (e.g. Thurber, 1983, Eberhart-Phillips, 1986) that ART_PB ray tracing should be regarded with caution for rays longer than ~60 km.

In Figure 4.4 the results of the tests are summarized. For ray lengths up to ~50 km the travel time differences between ART_PB and RKP ray tracing are within the numerical accuracy of about 10 ms. For longer ray paths ART_PB yields systematically slower rays with large differences between forward and reverse ray tracing. This might probably be due to the hard-wired

maximum number of ray-segments in ART_PB ray tracing which effects longer ray paths more than shorter ones. But no further testing of the ART_PB ray tracing has been undertaken in this work. RKP rays also show an increased travel-time uncertainty above 60 km ray length but much less than ART_PB rays.

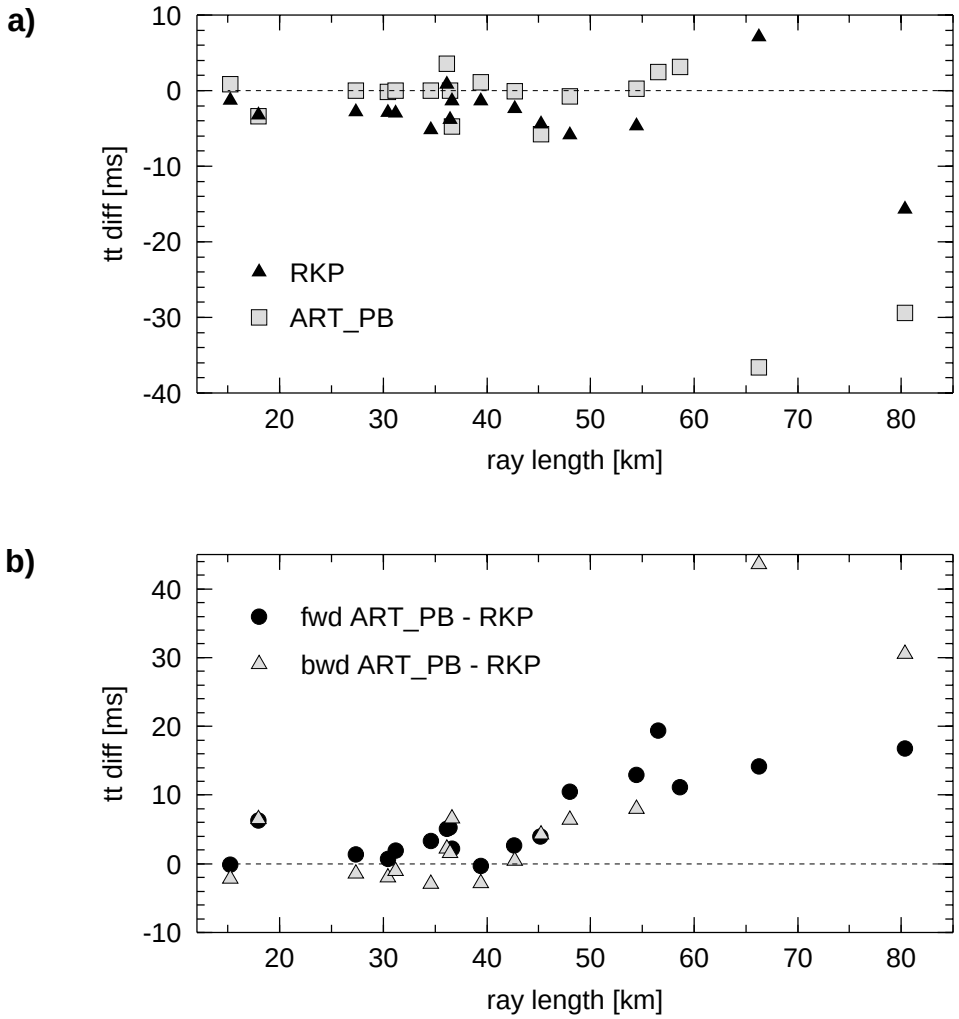


Figure 4.4 **a)** Travel time differences between forward (fwd) and reverse (bwd) ray tracing for RKP fwd-bwd (triangles) and ART_PB fwd-bwd (squares). For both ray tracers an accuracy of ~10 ms is inferred for ray lengths up to 60 km. For longer rays the RKP shows better performance than ART_PB. **b)** Differences between ART_PB and RKP forward (circles) and reverse (triangles) ray tracing. Up to ~50 km ray length the differences are insignificant (within the ray tracing accuracy). For longer rays RKP ray tracing gives faster travel times.

Computational Cost

With the advent of more and more powerful computers, the importance of the computational cost of solving the forward problem is decreasing. Complementary to that the growth of data sets used for tomographic studies again takes computational power to its limits. As a comparative measure, RKP ray tracing for one ray takes about four times as long as ART_PB ray tracing if

no initial shooting angles are provided (i.e. at the first time a ray has to be traced) and about 2 times as long when initial angles are available from previous successful ray tracings. Main reasons for the increase in computation time is the B-spline interpolation on the velocity field for every ray point, and the increased number of ray points for RKP rays (the maximum number of ray points is fixed for ART_PB ray tracing). As the code was written to allow maximum control on the performance of the ray tracer there may be some room for speeding up calculations.

Appendix C

RKP Ray Tracer Testing

C.1 Numerical Stability of RKP Ray Tracing

Travel times for shooting or bending forward raytracers are computed by summing up the travel times for all ray segments (a ray segment is defined as the path *s* between 2 consecutive raypoints *r, r_1*). For RKP ray tracing the travel time for one ray segment *S* is computed by $T_{S(r,\,r_1)} = ampr \cdot u^2$, where *ampr* is the parameter that controls the step length for ray tracing (d**x** = *ampr* · **p**). *ampr* is a value around 1, and u^2 has significant digits to 10^{-4}, so summing up T_S for a large number of ray segments may lead to numerical instabilities. To test this a series of tests are performed using 3 different methods of computing T_S .

time1 is computed using the slowness value at the midpoint of the current ray segment :

$$time1 = \sum_S ampr \cdot u^2(midpoint_s)$$

time2 is computed using Simpson's extended rule :

$$time2 = \sum_R T_{S(Simpson)}(r, r_1)$$

time3 is computed using the trapezoidal rule:

$$time3 = \sum_R ampr \cdot (u^2{}_r + u^2{}_{r_1})/2$$

Simpson's extended rule integration (Press et al., 1986: Numerical Recipies, pp 105ff) uses the parameter *jmax* to increase the number of points between *r* and r_1 and so to get a more accurate estimate for T_S than with trapezoidal integration, especially when there are strong velocity gradients along a path. For *jmax*=1, *time2* and *time3* are equal. With decreasing *ampr* and increasing *jmax* the accuracy of the raypath and the traveltime should increase, but at the same time the computational cost increases. The effects on the raypath can be neglected (raypaths are similar within a few tens of meters) with *ampr* smaller than 2. Thus we look for the largest *ampr* and smalles *jmax* which yield stable travel time estimations and compare this time to the results with *time1* and *time3*; as these are computationally more efficient they are preferable if the obtained travel time is accurate enough.

The tests consisted of shooting between three station-event pairs in both directions, *forward* from the event to the station and *reverse* from the station to the event. As the effect of *ampr* is

much greater than of *jmax*, the results are presented in three tables for *ampr* values of 0.5, 1.0 and 1.5. The velocity field used for the tests has a vertical gradient of 0.2 km/s from the surface to 17 km, 0.1 km/s from 17 to 27 km and then again 0.2 km/s. It contains one low- and one high-velocity block with ±15% anomaly respectively, which only influence the S3/E3 rays.

The raypaths for forward and reverse shooting are similar to within a few tens of meters for all three station-event pairs, so the differences in travel times between the two rays are due to numerical effects of the travel time estimation, and can be regarded as the limit of accuracy which can be achieved.

From the results (Tables C.1 - C.3) it can be seen that for the computation of *time2*, a value of *jmax*=5 is sufficient to ensure stable travel times in all cases. Also, as expected, with decreasing values of *ampr* the differences between *time1*, *time2* and *time3* decrease. The differences in travel times for reciprocal shots decrease significantly for S2/E2 and S3/E3 between *ampr*=1.5 and *ampr*=1.0, they increase slightly for S1/E1, but they are always greater than the differences between the individual *time1, time2, time3* values. From the point of stability of computed traveltimes this results justify that a value *ampr* = 1.0 is sufficient for stable travel time calculation in LET problems. This leads to an average ray segment length of ~200m, which makes sense for velocity parametrizations on grids with 1km or larger grid spacing. For very small and detailed models *ampr* should probably be reduced. The computation of traveltimes using simple trapezoidal rule integration (*time3*) is computational the most efficient one and does not produce any significant different results to the other methods for *ampr* = 1.0 or 0.5. *time3* is therefore used in the RKP implementation in SIMULPS.

The travel times shown in the tables are those computed for the actual raypath obtained from shooting. As shooting lacks a boundary condition at the endpoint of the ray (i.e. the ray will normally not hit the receiver exactly), the endpoints of the reciprocal rays are not equal. The maximum endpoint misfit in the tests is ~30 m, and the average is ~20 m. With the paraxial theory a travel-time correction can be computed to account for these misfits. The correction is within a few milliseconds and the values for the uncorrected and corrected times for *ampr*=1.0 (times computed as *time3*) are given in Table C.4. The correction values are within the error levels concluded previously, so one could argue whether a correction really is necessary. But the differences between the reciprocal travel times generally decrease when the correction is applied, and this indicates that the correction does makes sense and should be taken into account.

From these tests it can be concluded that the absolute accuracy of travel times computed for RKP ray tracing is around 10 ms. This is of the same order of magnitude as the arrival time picking accuracy assumed for high frequency local earthquake data with good S/N ratio. But, considering the various error sources in arrival time estimation and the resulting cumulative uncertainty, which normally is between 20 ms and 100 ms, travel times from ray tracing with 10 ms uncertainty are sufficient for almost all LET applications.

endpoint - starting point	ray length [km]	time1	time2				time3
			jmax= 1	jmax= 5	jmax= 10	jmax= 20	
S1-E1	15.202	3.9868	3.9887	3.9875	3.9875	3.9875	3.9887
E1-S1	15.202	3.9893	3.9914	3.9900	3.9900	3.9900	3.9914
S2-E2	43.097	8.7071	8.7090	8.7077	8.7077	8.7077	8.7090
E2-S2	43.080	8.6938	8.6956	8.6944	8.6944	8.6944	8.6956
S3-E3	79.835	14.7235	14.7254	14.7241	14.7241	14.7241	14.7254
E3-S3	79.859	14.7051	14.7069	14.7057	14.7057	14.7057	14.7069

Table C.1: ampr = 1.5

endpoint - starting point	ray length [km]	time1	time2				time3
			jmax= 1	jmax= 5	jmax= 10	jmax= 20	
S1-E1	15.202	3.9823	3.9831	3.9826	3.9826	3.9826	3.9831
E1-S1	15.202	3.9882	3.9891	3.9885	3.9885	3.9885	3.9891
S2-E2	43.097	8.6985	8.6993	8.6987	8.6987	8.6987	8.6993
E2-S2	43.080	8.6983	8.6991	8.6986	8.6986	8.6986	8.6991
S3-E3	79.835	14.7023	14.7031	14.7025	14.7025	14.7025	14.7031
E3-S3	79.859	14.7121	14.7129	14.7123	14.7123	14.7123	14.7129

Table C.2: ampr = 1.0

endpoint - starting point	ray length [km]	time1	time2				time3
			jmax= 1	jmax= 5	jmax= 10	jmax= 20	
S1-E1	15.202	3.9876	3.9878	3.9877	3.9877	3.9877	3.9878
E1-S1	15.202	3.9912	3.9915	3.9913	3.9913	3.9913	3.9915
S2-E2	43.097	8.6984	8.6986	8.6985	8.6985	8.6985	8.6986
E2-S2	43.080	8.7016	8.7018	8.7016	8.7016	8.7016	8.7018
S3-E3	79.835	14.7097	14.7099	14.7097	14.7097	14.7097	14.7099

endpoint - starting point	ray length [km]	time1	time2				time3
			jmax= 1	jmax= 5	jmax= 10	jmax= 20	
E3-S3	79.859	14.7174	14.7177	14.7175	14.7175	14.7175	14.7177

Table C.3: ampr = 0.5

endpoint - starting point	time3 [s] corrected	time3 [s] uncorrected	difference between shooting directions	
			corrected [ms]	uncorrected [ms]
S1-E1	3.9839	3.9831	-4.4	-6.0
E1-S1	3.9883	3.9891		
S2-E2	8.6962	8.6993	-0.6	0.2
E2-S2	8.6968	8.6991		
S3-E3	14.7076	14.7031	-6.1	-9.8
E3-S3	14.7137	14.7129		

Table C.4: Travel times with and without end-point correction

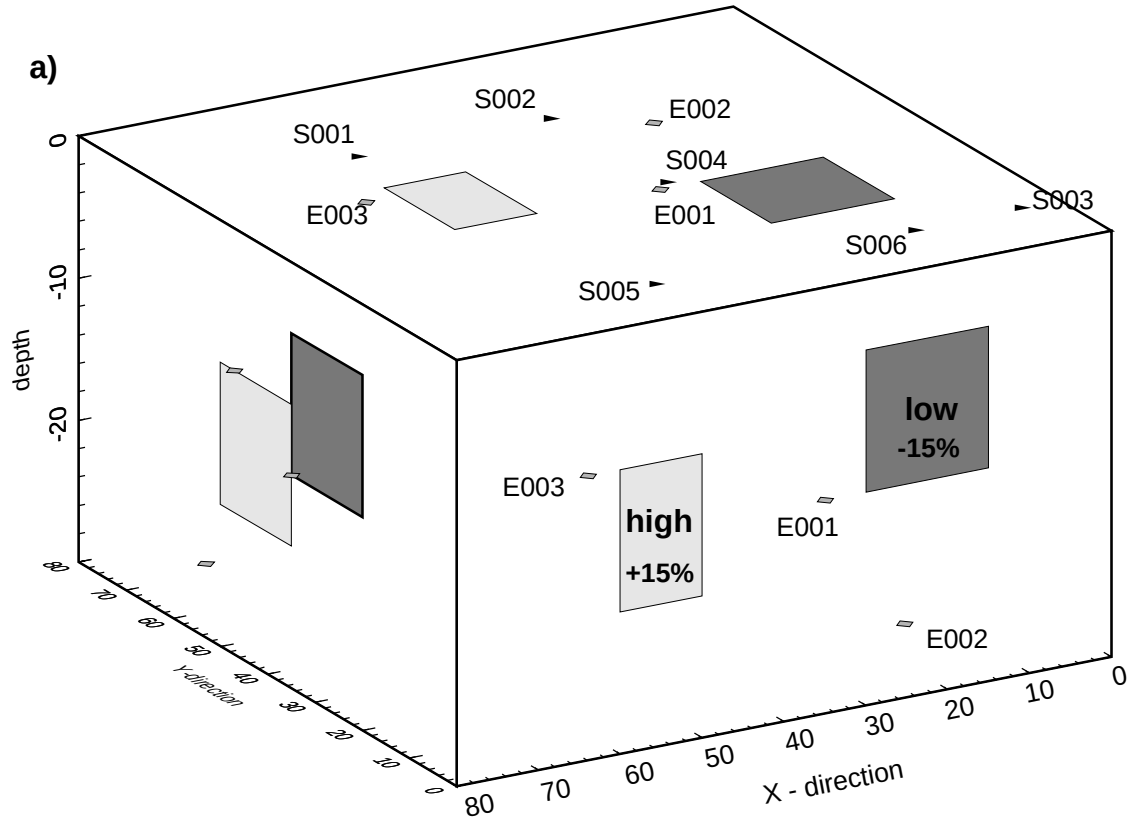
C.2 Comparing ART_PB and RKP Ray Tracing

After testing the numerical stability of the RKP ray tracing a second test is undertaken to assess the differences between ART_PB and RKP ray tracing. The synthetic velocity model and shot - receiver configuration (slightly different from that used in C.1) for this test are shown in Figure C.1. For each shot - receiver pair again forward and reverse ray tracing is done using ART_PB and RKP. Thus the numerical accuracy of one ray tracing scheme as well as the differences between the two can be investigated. The ray tracing results are summarized in Table C.5. For ray length less than ~60 km the travel times from both ray tracers, comparing forward and reverse ray tracing, can be assigned uncertainties below 10 ms (see also Fig. 4.6a). The differences between the two ray tracing schemes become larger than 10 ms for ray lengths >50 km (Fig. 4.6b). From the ray-path plots (Figs. C.2, C.3) it can be seen that for these rays also discernible differences in the ray paths exist. Apparently the RKP ray tracing can better adapt especially to the presence of low velocity zones. Still the differences are rather small and especially for tomographic inversions with grid node spacing larger than about 5 km no significant effects due to different ray paths should emerge. One drawback of the RKP ray tracing also becomes visible: as it is a shooting method it can happen that no valid ray (i.e. a ray which reaches its target within a certain maximal distance) can be found within the allowed number of iterations. This happens

here for the rays (E002 - S003) and (E002 - S005). Both these rays (calculated from the receiver to the source) touch the border of the low velocity anomaly shortly before reaching the shot location, and the adjustment process of the initial angles is not able to accomodate for this.

rec-shot pair	ray length [km]	ART-PB times [s]			RKP times [s]			diff ART-PB - RKP [ms]	
		forward	reverse	diff [ms]	forward	reverse	diff [ms]	forwd	revrs
S001-E001	39.42	7.6786	7.6775	1.1	7.6789	7.6803	- 1.4	- 0.3	- 2.8
S002-E001	30.41	6.4165	6.4167	- 0.2	6.4158	6.4187	- 2.9	0.7	- 2.0
S003-E001	48.02	9.1432	9.1440	- 0.8	9.1327	9.1386	- 5.9	10.5	6.4
S004-E001	15.22	3.5410	3.5402	0.8	3.5411	3.5424	- 1.3	- 0.1	- 2.2
S005-E001	34.57	7.1111	7.1111	0.0	7.1088	7.1140	- 5.2	3.3	- 2.9
S006-E001	36.14	7.7072	7.7037	3.5	7.7023	7.7015	0.8	5.1	2.2
S001-E002	42.65	7.9590	7.9591	-0.1	7.9563	7.9587	- 2.4	2.7	0.4
S002-E002	27.35	5.4303	5.4303	0.0	5.4289	5.4317	- 2.8	1.4	-1.4
S003-E002	56.52	9.9349	9.9325	2.4	9.9155	--	--	19.4	--
S004-E002	31.15	6.1127	6.1127	0.0	6.1108	6.1138	- 3.0	1.9	-1.1
S005-E002	58.64	10.3209	10.3178	3.1	10.3098	--	--	11.2	--
S006-E002	54.43	9.6479	9.6477	0.2	9.6350	9.6397	- 4.7	12.9	8.0
S001-E003	17.91	4.2874	4.2908	- 3.4	4.2811	4.2843	- 3.2	6.3	6.5
S002-E003	36.63	7.5820	7.5868	- 4.8	7.5798	7.5812	- 1.4	2.2	6.6
S003-E003	80.37	13.7696	13.7990	- 29.4	13.7528	13.7685	- 15.7	16.8	30.5
S004-E003	36.40	7.4953	7.4953	0.0	7.4900	7.4938	- 3.8	5.3	1.5
S005-E003	45.19	8.8864	8.8912	- 5.8	8.8826	8.8870	- 4.4	4.0	4.2
S006-E003	66.26	12.0506	12.0872	- 36.6	12.0365	12.0436	- 7.1	14.1	43.6

Table C.5: Results for forward and reverse ray tracing with ART_PB and RKP for the model shown in Figure C.1.



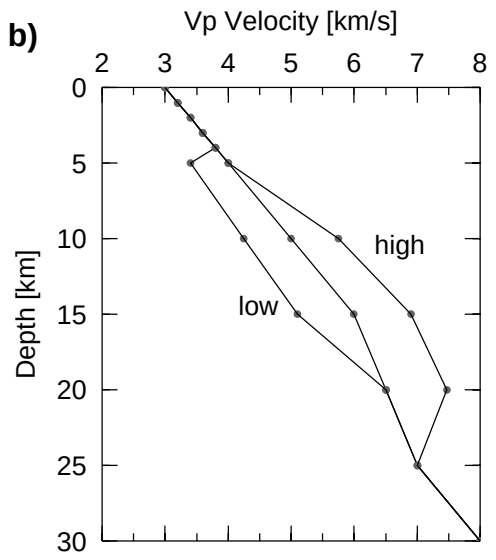


Figure C.1 **a)** Synthetic velocity model and shot - receiver distribution. Shot locations (grey diamonds) are marked E00x, receiver locations (black triangles) S00x. **b)** v(z) profiles through the undisturbed model and the center of the high and low velocity anomalies. Note that the velocity anomalies decay linearly to the undisturbed model within 5 km from the shown borders (resp. within 1 km at the upper border of the low velocity model, as also apparent from the v(z) profiles).

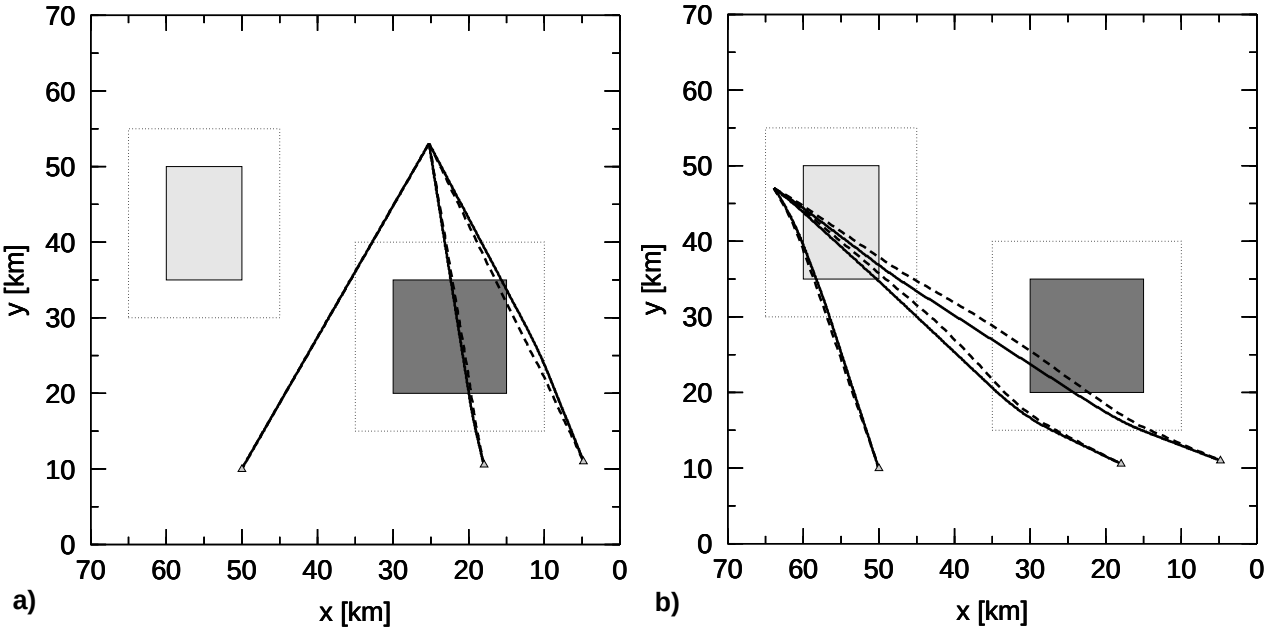


Figure C.2 Horizontal projections of rays resulting from ART_PB (dashed lines) and RKP (solid lines) ray tracing to stations S003, S006, S005 (from right to left, cf Fig. C.1). **a)** Source E002 **b)** Source E003. Dotted lines around the velocity anomalies denote region where anomaly decays to the background model. Note the significant differences for rays crossing both velocity anomalies.

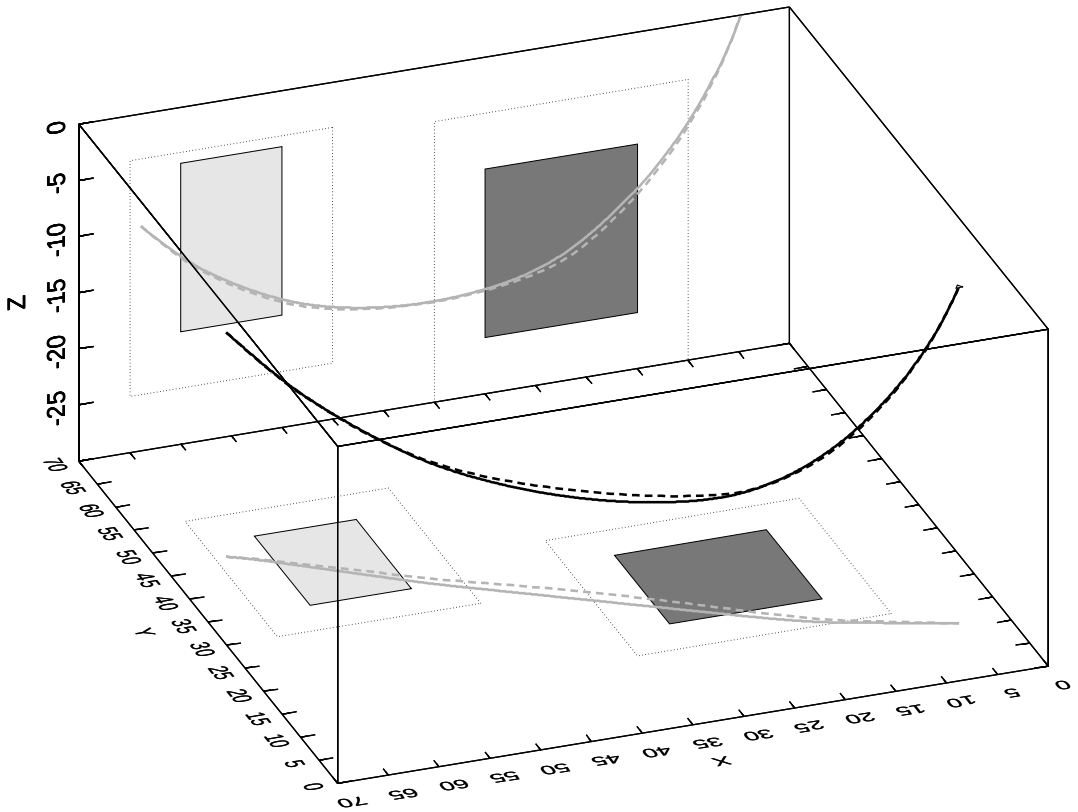


Figure C.3 3D perspective plot for the RKP (solid line) and ART_PB (dashed line) rays from source E003 to receiver S003. Projections of the rays on the bottom xy-plane and the backward xz-plane are shown with grey lines. The RKP ray is 17 ms faster than the ART_PB ray.